Synchronization Techniques for Burst-Mode Continuous Phase Modulation

By

Seyed Mohammad Ehsan Hosseini

Submitted to the graduate degree program in the Department of Electrical Engineering & Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dissertation Committee:

Erik S. Perrins, Chairperson

Shannon D. Blunt

Lingjia Liu

David W. Petr

Tyrone S. Duncan

Date Defended

The Dissertation Committee for Seyed Mohammad Ehsan Hosseini
certifies that this is the approved version of the following dissertation:

**Synchronization Techniques for
Burst-Mode Continuous Phase Modulation**

Erik S. Perrins, Chairperson

Date approved:

Abstract

Synchronization is a critical operation in digital communication systems, which establishes and maintains an operational link between transmitter and the receiver. As the advancement of digital modulation and coding schemes continues, the synchronization task becomes more and more challenging since the new standards require high-throughput functionality at low signal-to-noise ratios (SNRs). In this work, we address feedforward synchronization of continuous phase modulations (CPMs) using data-aided (DA) methods, which are best suited for burst-mode communications. In our transmission model, a known training sequence is appended to the beginning of each burst, which is then affected by additive white Gaussian noise (AWGN), and unknown frequency, phase, and timing offsets.

Based on our transmission model, we derive the Cramér-Rao bound (CRB) for DA joint estimation of synchronization parameters. Using the CRB expressions, the optimum training sequence for CPM signals is proposed. It is shown that the proposed sequence minimizes the CRB for all three synchronization parameters asymptotically, and can be applied to the entire CPM family.

We take advantage of the simple structure of the optimized training sequence in order to design a practical synchronization algorithm based on the maximum likelihood (ML) principles. The proposed DA algorithm jointly estimates frequency offset, carrier phase and symbol timing in a feedforward manner. The frequency offset estimate is first found by means of maximizing a one dimensional function. It is then followed by symbol timing and carrier phase estimation, which are carried out using simple closed-form expressions. We show that the proposed algorithm attains the theoretical CRBs for all synchronization parameters for moderate training sequence lengths and all SNR regions. Moreover, a frame synchronization algorithm is developed, which detects the training sequence boundaries in burst-mode CPM signals.

The proposed training sequence and synchronization algorithm are extended to shaped-offset quadrature phase-shift keying (SOQPSK) modulation, which is considered for next generation aeronautical telemetry systems. Here, it is shown that the optimized training sequence outperforms the one that is defined in the draft telemetry standard as long as estimation error variances are considered. The overall bit error rate (BER) plots suggest that the optimized preamble with a shorter length can be utilized such that the performance loss is less than 0.5 dB of an ideal synchronization scenario.

Acknowledgments

First and foremost, I would like to thank my adviser Dr. Erik Perrins for invaluable guidance he has given me over the years. This work would not have been accomplished without his excellent support, patience, and motivating attitude toward his students.

Needless to say, my family and friends have been very important to me. I would like to express my deepest appreciation to my parents for their encouragement and support. I am grateful to my brother, Keyvan, who have been helping my parents throughout the years that I have been abroad. Finally and especially, I would like to say thank you to my friends, colleagues, professors, and many others at the University of Kansas for providing an inspiring atmosphere.

Page left intentionally blank.

Table of Contents

Page left intentionally blank.

List of Figures

Page left intentionally blank.

Chapter 1

Introduction

1.1 Background

As the title of this dissertation suggests, three areas of digital communications are covered in this effort, which are: burst-mode transmissions, CPM schemes, and synchronization algorithms. Thus, we opt to provide a brief introduction to each of these topics in order to establish the basic concepts, notation and terminology. Interested readers may refer to textbooks such as [1, 2, 3] for an in-depth discussion of the aforementioned topics.

1.1.1 Burst-Mode Communications

Transmission of digital data can be performed in two different fashions: *continuous* transmission and *burst-mode* transmission. In the former case a data stream is established between the receiver and the transmitter for a long period of time, while in the latter case, the transmission consists of abrupt disjoint short packets, also known as bursts. Burst-mode transmission is in close association with time-division multiple-access (TDMA) networks, which break the transmission into multiple time slots, each one is dedicated to a different user.

A simplified example of burst-mode transmission is depicted in Figure 1.1. Each burst consists of two main parts: Training sequence and payload. The training sequence (also known as the preamble, pilot symbols or sync. word) is used to estimate and correct defects that

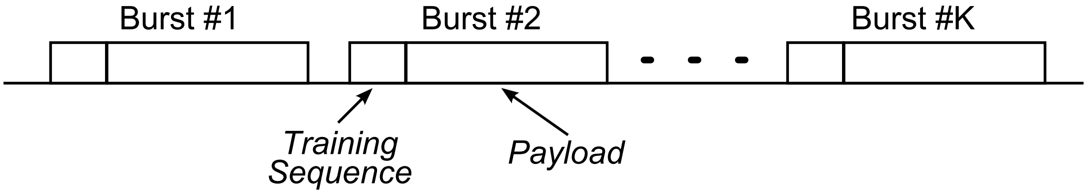


Figure 1.1. Structure of a burst-mode transmission

occur during the transmission. The payload is simply the information bits that construct the transmitted message. A unique word (UW) may also be placed in the burst for burst identification, i.e. if a burst is actually received. Finally, one can place a header part prior to the payload in order to distinguish the intended user. In this work, we assume the training sequence is located at the beginning of each burst, and hence, training sequence and preamble are used interchangeably.

1.1.2 Continuous Phase Modulation

Continuous phase modulation (CPM) [4] is a constant envelope modulation that exhibits power and bandwidth efficiency. Due to its constant envelope feature, CPM modulators do not require linear amplifiers, reducing the cost of the transmitter. However, CPM is a non-linear modulation with memory, which results in high complexity at the receiver compared to linear modulations such as phase-shift keying (PSK) and quadrature amplitude modulation (QAM).

The complex baseband CPM signal is

$$s(t) = \sqrt{\frac{E_s}{T_s}} \exp\{j\phi(t;\boldsymbol{\alpha})\} \tag{1.1}$$

where E_s is the energy per transmitted symbol and T_s is the symbol duration. The phase of the signal $\phi(t;\boldsymbol{\alpha})$ during the transmission is represented as

$$\phi(t;\boldsymbol{\alpha}) = 2\pi h \sum_{i=-\infty}^{\infty} \alpha_i q(t - iT_s) \tag{1.2}$$

where $\{\alpha_i\}$ is the sequence of M-ary data symbols selected from the set of $\{\pm1, \pm3, \ldots, \pm(M -$

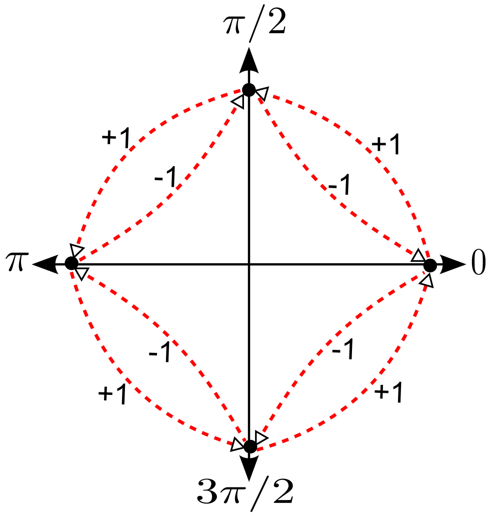
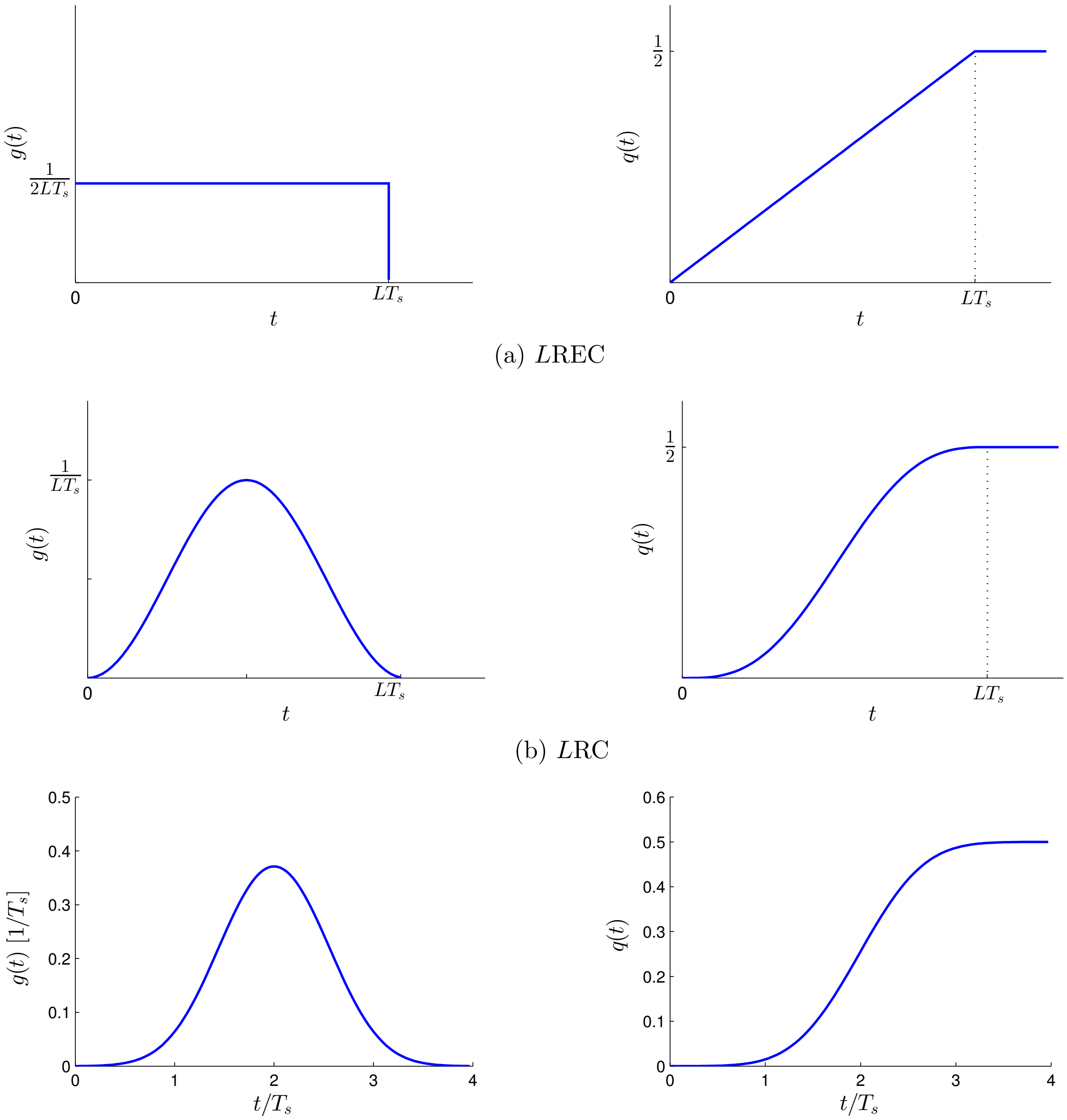


Figure 1.2. State diagram for binary $h = 1/2$ full-response CPM. The dashed lines show the phase variations based on the incoming data symbols.

$1)\}$. The variable h is called the modulation index. It can be a constant in *single-h* CPM or a variable in *multi-h* CPM waveforms. The waveform $q(t)$ is called the *phase response* and in general is represented as the integral of the *frequency pulse* $g(t)$, whose duration is LT_s. If $L = 1$ the signal is called *full-response* CPM, and for $L > 1$, it is called *partial-response* CPM. Based on CPM conventions, $q(t) = 0$ for $t < 0$, and $q(t) = 1/2$ for $t \geq LT_s$.

Three well-known frequency pulse shapes [3] and their phase responses are illustrated in Figure 1.3. LREC refers to a rectangular pulse with a duration of LT_s. LRC corresponds to a raised-cosine pulse of duration LT_s. Finally, the Gaussian pulse is used in Gaussian minimum shift-keying (GMSK), and has a bandwidth parameter of BT_s. We have illustrated the Gaussian pulse with $BT_s = 0.3$ that is used in the European cellular system called GSM. For this particular bandwidth parameter we have $L \approx 4$. Additionally, GMSK is a binary CPM with $h = 1/2$, which is closely related to the full response minimum shift-keying (MSK) modulation. In fact, binary CPMs with $h = 1/2$ are classified as MSK-type modulations, regardless of their frequency pulses.

Generally speaking, a CPM signal cannot be represented on the signal space using discrete points since the phase of the signal is constantly varying. However, a QPSK-like constellation



(a) LREC

(b) LRC

(c) Gaussian ($BT_s = 0.3$)

Figure 1.3. The pulse shapes for rectangular (a), raised-cosine (b), and Gaussian CPMs.

diagram is presented in Figure 1.2 for a better understanding of full-response binary $h = 1/2$ CPM and other CPMs in general. It can be seen that at exact time instances equal to multiples of T_s the signal's phase is one of $[0, \pi/2, \pi, 3\pi/2]$ values. Upon arrival of a new data symbol, the phase starts traveling to another adjacent constellation point continuously during T_s seconds. Additionally, the phase trajectory of the CPM signal can be expressed on a trellis which is used for optimal detection of CPM signals using the Viterbi [5] algorithm.

1.1.3 Synchronization Techniques

In digital transmissions, one is always interested in correct estimation of the data symbols regardless of what happens during the transmission. However, this cannot be achieved unless a reliable knowledge of the parameters that describe the transmission exists. A major class of such parameters are *synchronization parameters*, which can be viewed as reference points in the transmission.

Two of the main synchronization tasks are *timing* synchronization and *carrier* synchronization. Timing synchronization has to be performed in order to determine the correct sampling time of the received analog signal. The timing ambiguity is caused by the unknown delay between the transmitter and the receiver, which is also referred to as the symbol timing. Carrier synchronization consists of the estimation of both carrier frequency offset and its phase. The frequency offset occurs in passband transmissions due to the mismatch between oscillators at the receiver and transmitter and also because of the Doppler effect, i.e. frequency shifts due to the movement of the transmitter and/or receiver. It should be mentioned that carrier phase estimation is not always needed and one can use *noncoherent* techniques. However, noncoherent detection results in a performance degradation. The transmitted signal $s(t)$ affected by the aforementioned impairments can be formalized at the receiver as

$$r(t) = e^{j(2\pi f_d + \theta)} s(t - \tau) \tag{1.3}$$

where f_d is the frequency offset, θ is the carrier phase and τ is the symbol timing. Note that the channel noise is ignored in the above notation. Therefore, synchronization consists of estimating $[f_d, \theta, \tau]$ from the received signal $r(t)$. In the above notation, we have assumed τ is a fraction of the symbol duration, and hence, it is referred to as the symbol timing. In burst-mode communications, it is necessary to estimate the boundaries of each frame in order to satisfy that assumption. We refer to the latter task as the *frame synchronization*.

In the following sections, we encounter several terms regarding the classification of a given synchronization algorithm. Here, a short description of these terms is provided.

- **Feedback vs. Feedforward;** Feedback synchronizer refers to a closed-loop estimator in which the estimation is carried out based on partially synchronized $r(t)$ in a closed-loop fashion. The synchronization parameters are updated with the arrival of every new symbol. In a feedforward structure, a one-time estimate of the synchronization parameters is generated using the original received signal.

- **Data-Aided (DA) vs. Non-data-aided (NDA);** DA synchronizers take advantage of the prior knowledge of transmitted data symbols, whereas NDA synchronizers work on the statistical properties of the transmitted signal.

- **Decision-directed (DD)**. This class of synchronizers is designed based on DA estimation algorithms. However, they use estimated data symbols (decisions) rather than prior known symbols. In fact, synchronization and detection are combined in this class of synchronizers.

As mentioned earlier, synchronization mainly involves the estimation of reference parameters which is a problem in estimation theory [2]. Here, we touch on a couple of concepts in estimation theory that will be used frequently in our discussions.

A classical and widely-used approach in estimation theory and particularly in synchronization algorithms is Maximum Likelihood (ML) estimation. ML estimation is asymptotically, i.e. for large data records, optimal and unbiased [2, Theorem 7.1], which makes it attractive for researchers. The ML estimate of an unknown parameter λ from the observed received signal $\mathbf{r}$

is the value for which the likelihood function is maximized. This can be expressed as

$$\hat{\lambda}(\mathbf{r}) = \underset{\lambda}{\operatorname{argmax}}\{\Lambda(\mathbf{r};\lambda)\} \tag{1.4}$$

where $\hat{\lambda}$ is the estimated value of λ and $\Lambda(\mathbf{r};\lambda)$ is the likelihood function, i.e. the probability density function (pdf) of $\mathbf{r}$, which is also a function of λ. According to (1.4), ML estimation requires first the computation of the likelihood function and then its maximization. The maximization process can be done either analytically or numerically, e.g. via a grid search.

Another important concept in estimation theory, which is related to ML estimation, is the Cramér-Rao bound (CRB) [2], which is a lower bound on the estimation performance on *any* unbiased estimator in terms of mean-square error (MSE). The CRB is often used in synchronization studies as a benchmark to demonstrate how well a synchronizer performs, i.e. how close its actual MSE is to the CRB. Another application of the CRB in synchronization is the design of optimum training sequences for DA algorithms, because the CRB becomes a function of the transmitted data symbols in synchronization algorithms. Finally, it should be mentioned that the CRB is attained asymptotically, i.e. high signal-to-noise ratio (SNR) and large number of samples, in the case of ML algorithms.

1.2 Related Work

In this section, we provide an overview of the existing work regarding synchronization of burst-mode communications with an emphasis on CPM. Due to the scale and complexity of this problem, researchers have generally addressed it in three categories: training sequence design, carrier and symbol timing synchronization, and frame synchronization. Here, we present the related work using the same categories.

1.2.1 Training Sequence Design

The problem of deriving the optimum training sequence for DA estimation has been extensively studied [6, 7, 8, 9, 10, 11, 12, 13, 14], which can be categorized in terms of target application, optimization criteria and optimization method.

The CRB has been widely used for design and study of training sequences. Jiang *et al.* [7, 8] have derived the CRB for DA estimation of symbol timing and carrier phase for linearly modulated signals in the additive white Gaussian noise (AWGN) channels. Their method presents the CRB as a function of the autocorrelation of the data symbols by which they have compared the estimation performance of several sequences, including a continuous wave (CW), alternating-ones-and-zeros sequence, and pseudo-random sequences. However, they have not provided the optimization of the training sequence. In a more recent work [15], Tavares *et al.* have developed a method for training sequence design for DA estimation of synchronization parameters based on the CRB criteria. They have assumed linearly modulated signals in an AWGN channel and considered two main estimation scenarios: joint estimation of symbol timing and carrier phase with a known frequency offset, and, joint estimation of carrier phase and frequency offset assuming the time delay is known. Accordingly, the CRB expressions are derived and shown to have a similar form, i.e. the inverse of quadratic forms. The authors have introduced a transmit power constraint on the data symbols and performed the optimization such that the data symbols are treated as real numbers rather than M-ary symbols. The optimum data vectors are then quantized according to the modulation. It is shown that the optimum training sequence for symbol timing has an alternating structure while it turns out to be as a rather constant vector for carrier phase and frequency offset estimation. An important conclusion is that the optimum training sequence differs for each of the synchronization parameters in the case of linear modulations. In a similar effort for linear modulations [13], the authors have computed the training sequence that minimizes the CRB for joint estimation of symbol timing, carrier phase and frequency offset. In [13], the symbol timing CRB is minimized analytically while the training sequences which minimize the frequency offset and carrier phase CRBs are derived via numerical optimization techniques. Moreover, Rice and Perrins [12] have numerically identified the best training sequence for synchronization of offset QPSK (OQPSK) modulation and using the CRB method. It is reported that the best sequence for all three synchronization parameters is basically repeating the same OQPSK constellation point. Dabora *et al.* [6] have derived a high SNR expression for the symbol timing CRB of CPM signals in fast fading channels, which they used to obtain the best sequence for MSK signals. It should be mentioned that MSK is a specific and the simplest form of CPM signals, and hence, the results may not be generalized to the whole CPM family. Finally, there has also been works [16, 17] regarding designing training sequences for channel estimation based on the CRB.

Besides the CRB approach, a few *ad hoc* methods have been published for the design of training sequences. For example, [10] presents the optimum training sequence for symbol timing estimation of MSK modulation by first presenting the timing estimator and then investigating the effect of the sequence on its performance. In another work for synchronization of MSK [11], the authors have proposed repeating "1100" as the best preamble based on the spectral properties of this particular sequence. Finally, the authors in [9] have proposed an algorithmic approach for training sequence design for PSK modulations. They have defined an error function based on the autocorrelation of the symbols, which is minimized iteratively using a gradient descent algorithm. This method has two main drawbacks. First, its performance depends on the definition of the error function, i.e. there is no guarantee that the chosen function is the best candidate. Additionally, the optimization suffers from a large number of local minima, and hence, one has to run the algorithm multiple times with different initial seeds in order to possibly find the optimum sequence.

1.2.2 Symbol Timing and Carrier Recovery

Various synchronization algorithms for CPM have been presented in [1] with regard to each of the synchronization parameters, i.e. symbol timing, carrier phase and frequency offset. In this section, we provide an in-depth survey of more relevant works which are DA estimators that utilize a training sequence and/or feedforward algorithms that are suitable for burst-mode communications. We are also interested in joint estimation algorithms since we have assumed all three synchronization parameters are unknown.

One of the earliest yet important contributions on the DA synchronization of CPM signals was presented by Huber and Liu in [18]. They have proposed transformation of the CPM signal into the frequency domain using non-orthogonal basis functions. As a result, the timing delay is transformed into a phase offset that is estimated along with the carrier phase using a ML algorithm. However, the proposed algorithm works only for timing delays which are much smaller than the symbol duration, i.e. $\tau \ll T_s$, and hence, can only be used in a phased-locked loop (PLL) structure, which may also encounter false locks as mentioned by the authors. In a similar approach, Tang and Shwedyk [19] have used the *Walsh* transform for DA ML estimation of symbol timing and carrier phase for CPM signals. However, it still assumes small time delays, and hence, cannot be implemented in an open-loop structure. More recently, Zhao and Stüber [20] have presented a robust timing and phase estimation for CPM signals using minimum MSE (MMSE) criteria. Despite its robustness, the resulted MSE of the estimation is about an order of magnitude higher than the CRB, which means one has to utilize quite large training sequences in order to achieve a good estimation performance at low SNRs. Another interesting DA design for CPM is reported in [21], which addresses ML estimation of the symbol timing irrespective of the carrier phase in Rayleigh fading channels. In a more recent and novel approach, Maoilo [22] *et al.* have studied estimation of synchronization parameters in CPM signals based on the trellis representation and BCJR algorithm [23]. Although their discussion involves all three synchronization parameters, they were only able to derive a phase estimation algorithm due to the high complexity of the equations. Finally, it should be mentioned that all the aforementioned works assume perfect frequency offset estimation.

A significant amount of research has been dedicated to synchronization algorithms for a specific class of CPM known as MSK-type signals. This popularity is due to their approximation as OQPSK signals, which enables one to employ well-known detection and synchronization techniques for linear modulations. Nezami [24] has proposed a simple synchronization method for MSK modulation based on the discrete Fourier transform (DFT) of the received preamble. This method uses an alternating training sequence and estimates all three synchronization parameters. Despite its simplicity, its reported performance is quite poor in the low SNR region. An NDA feedforward *ad hoc* estimator for MSK has been presented in [25] that is capable of frequency offset estimation. Another NDA feedforward frequency estimator for MSK-type signals is presented in [26] and is based on a quadratic non-linear transformation of the signal. Morelli and Vitetta [27] have presented an NDA joint ML estimation algorithm for symbol timing and carrier phase that can be employed in feedforward schemes. This algorithm outperforms [25] at low SNR regions in terms of the MSE, however, it does not provide carrier frequency estimation. A feedforward DA frequency estimator for GSM standard with GMSK modulation is presented in [28]. Despite its relatively simple implementation, this algorithm is only applicable to small frequency offsets compared to the symbol rate.

Although the scope of this work is CPM signals, it is helpful to study some of the major contributions for synchronization of linear modulations in burst-mode transmissions. Morelli and D'Amico [29] have proposed a joint ML estimator for symbol timing, carrier phase and frequency offset estimation in AWGN channels. The main idea behind their work is employing alternating BPSK symbols as the training sequence which simplifies the likelihood function significantly leading to a rather simple estimation algorithm. Therefore, we can conclude that the training sequence not only affects the estimation performance but it may also affect the complexity of the estimation algorithm. Gunther and Moon [30] have presented a complete synchronization algorithm for burst-mode QPSK signals, which includes frame synchronization in addition to three conventional parameters. This algorithm is based on a statistical measure named *kurtosis*, which is related to fourth and second order moments of a random variable. This work highlights the importance of accurate frequency estimation in low SNR regions in terms of bit error rate (BER). It is shown through simulations that the errors in frequency offset estimation leads to errors in symbol timing and carrier phase estimation resulting in elevated BERs.

1.2.3 Frame Synchronization

An important task for the synchronization in burst-mode communications is determining the boundaries of a packet, i.e. the beginning of bursts in the time domain. This task is referred to as frame synchronization in the published work. In fact, almost all of the work reviewed in the previous section assumes that frame synchronization has been already accomplished prior to carrier and symbol timing recovery. The only exceptions are [11] and [30], which employ matched filter (MF) based frame synchronization structures for linear and MSK modulations, respectively. In this section, we review the published works regarding frame synchronization and their relation to the synchronization algorithms that were studied in the previous section.

Similar to other estimation and detection problems, ML estimation is a widely-used technique in frame synchronization methods. Perhaps the earliest analytical work on frame synchronization is Massey's paper [31], which derives a ML rule for locating a *sync word* embedded in the beginning of each frame for PSK modulations. The rule is basically a correlation operation with a correction term that accounts for the random data adjacent to the sync word. Despite its simplicity and optimality, it assumes perfect knowledge of frequency and phase offsets. Additionally, the symbols must be sampled at the correct time. More recently, Lee [32] has proposed a ML algorithm for frame synchronization of PSK signals in the presence of frequency offsets. This work treats the frequency offset and carrier phase as unwanted uniformly distributed parameters in the derivation of the likelihood function. Another ML algorithm [33] has been proposed for PSK modulations which jointly estimates the carrier phase, frequency offset and the start of frame. It should be mentioned that all of these works require a known training sequence, and hence, are well-suited with DA techniques reviewed in the previous section in order to estimate the fractional timing delay.

Another approach for frame synchronization is the likelihood ratio test (LRT), which is a fundamental concept in detection theory. Chiani and Martini [34] have derived LRT and generalized LRT (GLRT) metrics for detection of the sync. word in AWGN channels where they have shown these tests perform significantly better than correlations. However, they assume both carrier and symbol timing are recovered prior to frame synchronization. In a more recent work by Chiani [35], carrier phase is included in the analysis as a random variable, however, frequency offset and symbol timing are neglected. It should be noted that the LRT methods in [34, 35] can only be applied to linear modulations. There are also a few works [32, 36] that address frame synchronization of PSK signals in frequency uncertainty. However, they consider continuous transmissions where the training sequence is surrounded by data symbols.

In recent years, code-aided frame synchronization techniques have gained attention in the literature [37, 38, 39]. These techniques take advantage of embedded coding structure in the signal for improving the synchronization at low SNRs. For example, [39] reviews several code-aided techniques when linear modulations are combined with iterative coding schemes such as turbo codes. The main drawback of these techniques is that they require perfect sampling time and carrier knowledge. Herzet *et al.* [37] have proposed a ML estimation technique which works on the factor graphs representing error correction codes such as low-density parity check (LDPC) and convolutional codes. Despite its good performance at low SNRs, it is still limited to linear modulations and any frequency offset has to be resolved in advance. Finally, a code-aided frame synchronization algorithm which can be applied to CPM signals was proposed by Huh and Krogmeier [38] which exploits the inherent trellis of CPM signals. Unfortunately, this algorithm needs perfect carrier and symbol timing recovery prior to frame synchronization.

From an implementation standpoint, current standards have adopted algorithms with minimum complexity. For example, the second generation digital video broadcasting (DVB-S2) standard [40] employs a correlator and a peak detector for frame synchronization. Similarly, the authors in [11] design a *start-of-message* (SOM) word for the UHF MILSATCOM standard such that it has minimum correlation side-lobes, suggesting correlators for the frame synchronization task.

Despite the novel works about frame synchronization in recent years, there are still some major issues which need to be addressed. We can summarize them as follows.

1. Many of the techniques such as [31, 32, 39] require carrier and/or symbol timing recov-

ery. On the other hand, DA synchronization algorithms require a coarse estimate of the training sequence location.

2. Almost all the published work address the frame synchronization problem in the context of linear modulations. This leaves the frame synchronization of the CPM signals unanswered despite the demand. Only exceptions are [38, 41] which model CPM signals as a Markov chain and orthogonal frequency division multiplexing (OFDM) respectively.

3. The computational complexity of a frame synchronization algorithm should allow real-time operation with a reasonable amount of utilized resources.

1.3 Problem Statement

As the rapid growth of digital communication systems continues, more stringent requirements are imposed during the development phase. One of the main challenges in current systems design is the tight constraint on the available RF spectrum. Due to the exponential growth of commercial systems, many other applications have seen spectrum loss as more bandwidth is allocated to commercial wireless systems. Therefore, bandwidth efficient schemes such as OFDM and CPM have become attractive to developers in recent years despite their complexities. Another challenge is the ever-increasing data rates as the number of users increases while they demand access to large data sources such as video streams. These factors have led to implementation of more complex modulation and coding schemes that are capable of reliable operation at low SNRs. However, these modern techniques are computationally complex and the implementation cost is high. In fact, a state-of-the-art receiver is complex in its all three major blocks: synchronization, detection and error correction.

As the title of this work suggests, this research effort is positioned at the intersection of three concepts in digital communications, which are CPM, synchronization and burst-mode communications. Below, we briefly discuss the motivation behind the selection of each of these areas in this effort.

- **CPM**

  This modulation scheme has been selected for this work due to its attractive features which are spectrum and power efficiency. Additionally, less research has been dedicated to CPM compared to linear modulations due its complexity. Thus, it is an interesting research topic where one can tackle these complexities and make CPM an attractive choice for current and future applications.

- **Synchronization**

  Among a receiver's primary functions, synchronization can be seen as the front-end of the receiver, which needs to function reliably. Otherwise, degradations in synchronization will have a ripple effect on the performance of the subsequent blocks and in turn results in poor BER. Therefore, in this work, we focus our attention on synchronization methods. As we saw in the previous chapter, synchronization of CPM signals has not been studied adequately compared to other modulations, which is another reason for choosing it in this research.

- **Burst-mode Communication**

  Burst-mode transmission enables one transmitter to communicate with multiple users on demand. However, as its name suggests, this type of communication brings uncertainty to the receiver since the packets arrive abruptly and the *warm-up time* has to be minimized in order to save bandwidth. Consequently, a burst-mode receiver must be able to perform the synchronization task reliably in a short period of time. Hence, the synchronization of burst-mode communications is more demanding compared to continuous transmissions.

We are now in a position to express our research problem formally. Assume transmission of L_{pay} data symbols, which are selected from an M-ary alphabet set, i.e. $\{\pm 1, \pm 3, \cdots, \pm(M-1)\}$ using CPM signaling in a burst fashion. Additionally, each burst is assumed to be preceded with L_0 training symbols in order to facilitate the synchronization task. Therefore, the baseband transmitted burst using CPM signaling can be written as

$$s(t) = \sqrt{\frac{E_s}{T_s}} \exp\left\{ j2\pi h \sum_{i=0}^{L_0+L_{\text{pay}}-1} \alpha_i q(t - iT_s) \right\} \tag{1.5}$$

where E_s, T_s, h and $q(t)$ are the symbol energy, symbol duration, modulation index, and the CPM phase response respectively. Moreover, $[\alpha_0, \alpha_1, \cdots, \alpha_{L_0-1}]$ is the set of training sequence symbols and $[\alpha_{L_0}, \alpha_{L_0+1}, \cdots, \alpha_{L_0+L_{\text{pay}}-1}]$ is the information symbols in the burst. It is assumed that $s(t)$ is transmitted over an AWGN channel and the receiver has no knowledge regarding the carrier phase, frequency offset and propagation delay. Therefore, the received baseband signal is expressed as

$$r(t) = e^{j(\theta + 2\pi f_d t)} s(t - \tau) + w(t) \tag{1.6}$$

where θ is the unknown carrier phase, f_d is the frequency offset, τ is the timing offset (delay), and $w(t)$ is complex baseband AWGN with zero mean and power spectral density of N_0.

According to (1.5) and (1.6), two major questions arise with regard to the synchronization task. These questions are:

1. What is the *best* choice for the training sequence $[\alpha_0, \alpha_1, \cdots, \alpha_{L_0-1}]$?

2. How can one estimate f_d, θ and τ reliably according to the chosen training sequence?

The above questions form the essence of this dissertation. In the following, we elaborate on the above questions with regard to the existing work reviewed in the previous section.

1.3.1 Training Sequence Design

The training sequence design is an optimization problem in which the optimum $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_{L_0-1}]$ has to be selected from M^{L_0} possible sequences. Similar to any other optimization problem, the first yet most important task is to define the optimization's objective. Since we are investigating the synchronization problem, we can define the optimum training

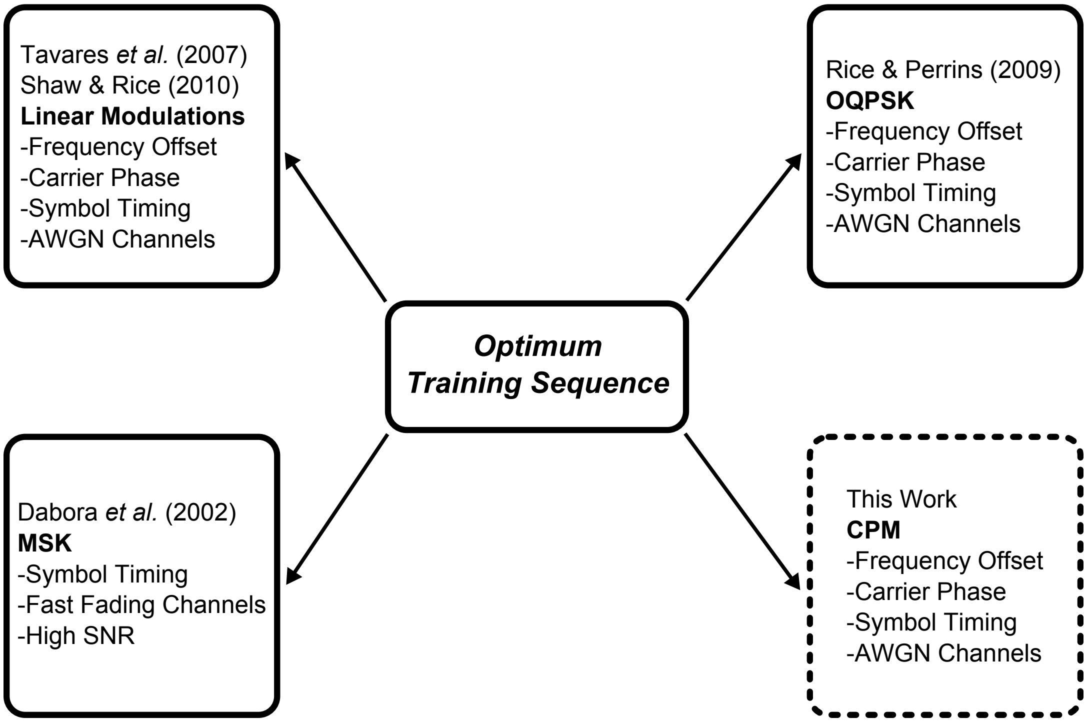


Figure 1.4. Development of the training sequence in different scenarios.

sequence according to

$$\boldsymbol{\alpha}^* = \operatorname*{argmin}_{\boldsymbol{\alpha}} \left\{ E_{\mathbf{r}|\boldsymbol{\alpha}}[(\hat{\lambda}(\mathbf{r}) - \lambda)^2] \right\} \tag{1.7}$$

where λ represents any of the three synchronization parameters, and $\hat{\lambda}(\mathbf{r})$ is its estimated value based on the channel observation $\mathbf{r}$. The expectation is performed with respect to the received signal conditioned on the underlying training sequence. In other words, the optimum training sequence is the sequence for which the MSE of an estimator is minimized. Since λ corresponds to three estimation parameters, i.e. $\lambda \in \{\theta, f_d, \tau\}$, (1.7) in fact represents three separate optimization problems, which can have similar or different solutions. Another challenge is that (1.7) does not convey any information about the estimation algorithm, and hence, its solution is algorithm dependent in general. Thus, the optimum training sequence should be derived with respect to a given estimation algorithm.

Figure 1.4 illustrates the proposed training sequence design problem with respect to the most related works in the literature. It is seen that our scenario is similar to [13] and [14] (top-left corner) where all three synchronization parameters are to be estimated in an AWGN channel. However, both of those works address linear modulations such as PSK while this work focuses on the more complex non-linear CPM modulation. The two other works by Debora [6] and Rice [12] investigate two instances of CPM in different types of channels. We also note that [6] considers only symbol timing estimation. In this work, we do not impose any conditions on the CPM signal and consider the estimation of frequency offset, carrier phase and symbol timing. Clearly, this attempt is distinguished from what has been performed by the research community, which addresses the broad area of synchronization for CPM signals.

1.3.2 Synchronization of Burst-Mode CPM

Once the optimum training sequence is designed, one has to design practical synchronization algorithms that are able to ideally approach the theoretical limits in terms of estimation MSE and/or BER performance. Based on our discussion so far, the synchronization problem can be stated as the DA estimation of $[\theta, f_d, \tau]$ from the received signal $r(t)$ given L_0 data symbols corresponding to the optimum training sequence. In the design of such algorithms, two key goals have to be considered. The first goal is synchronization capability at low SNRs. This is a result of existing powerful error correction codes such as LDPC and turbo codes, which have guaranteed reliable performance at very low SNRs, if the synchronization is carried out perfectly. Similar to any other communications algorithm, the computational complexity has a direct impact on the implementation cost and power consumption of the target hardware. Therefore, another goal is to design algorithms with minimum complexity.

Although several major contributions have been made toward the synchronization of CPM signals such as [18, 19, 42], they only address continuous-mode transmissions and assume there is no frequency offset. Thus, the joint estimation of all synchronization parameters in burst-mode transmissions is still an open question and requires further research. Our work is compared

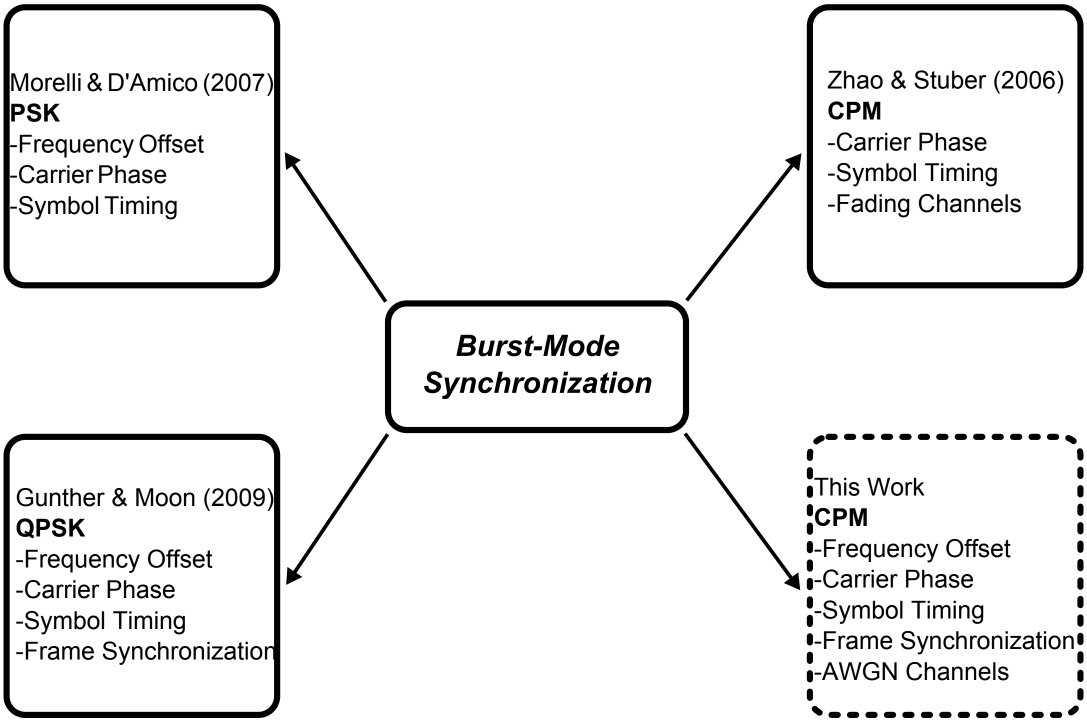


Figure 1.5. Development of synchronization algorithms for burst-mode communications.

to some of the most related and recent published works in the field of synchronization for burst-mode communications in Figure 1.5. It is again seen that the most comprehensive efforts [29, 30] are devoted to linear modulations (PSK here) due to their simplicity. From the CPM standpoint, Zhao and Stuber [20] have proposed a robust timing and phase synchronization algorithm which can be applied to all CPM schemes in burst-mode communications. Despite its good performance in fading channels, it is computationally complex and the frequency ambiguity has to be resolved prior to the synchronization. Furthermore, we have included frame synchronization problem in our work, which is quite necessary in burst-mode communications.

As a final note, the majority of published work treats each element of the burst-mode synchronization problem ,i.e. training sequence design and estimation algorithm, individually and with certain assumptions. However, we have combined each of these two elements such that the estimation algorithm is derived based on the optimized training sequence. To the best of our knowledge, this work is the first of its kind, which proposes a *complete* synchronization scheme for *general* CPM signals in burst-mode transmissions over AWGN channels.

Chapter 2

The Cramér-Rao Bound for Training Sequence Design for Burst-Mode CPM

2.1 Key Points of the Chapter

In this chapter, we study the CRB for CPM signals where frequency offset, carrier phase, and symbol timing are jointly estimated when transmitted over an AWGN channel. We consider a DA estimation scenario in which the estimator takes advantage of a known training sequence at the start of each burst. Thus, we first derive the joint CRBs as functions of a known training sequence and CPM parameters. By analyzing the CRB expressions, we propose the training sequence for which the CRB is minimized. We show that the proposed training sequence is asymptotically optimum for all three estimation parameters. Additionally, we compare the performance of the optimum training sequence with a random one by providing a closed-form expression for the unconditional CRB (UCRB) for symbol timing estimation of CPM signals. Comparing the UCRB and the CRB for the optimum training sequence reveals that a DA estimator with the optimum training sequence leads to significant gains in terms of the MSE of the estimation parameter when the underlying CPM scheme is non-binary and/or partial response.

2.2 Introduction

In burst-mode transmissions, one usually embeds a fixed data sequence to each burst for the purpose of fast and accurate receiver synchronization. This data sequence is known *a priori* to the receiver and is referred to as a training sequence or preamble. It is called a training sequence because DA synchronization algorithms take advantage of it in order to estimate synchronization parameters such as frequency offset, carrier phase, and symbol timing [1]. The performance of an arbitrary DA estimator may depend on the training sequence being used. Hence, a design challenge in dealing with DA estimators is choosing the best training sequence for a given length. Another question that arises in the design of the preamble is the performance gain that is achieved in comparison with a preamble-less design in which NDA estimators are employed. The latter technique does not have *a priori* knowledge of the transmitted symbols and is also known as *blind* estimation.

In this chapter, we use the CRB [2] as a tool in designing a training sequence that is asymptotically *optimum* for DA synchronization of burst-mode CPM signals. The CRB provides a performance limit (bound) on any *unbiased* estimator that is a function of the training sequence, and hence, we identify the optimum training sequence as the one that minimizes the CRB. Additionally, we consider the joint estimation of three synchronization parameters—frequency offset, carrier phase, and symbol timing. Therefore, the first step of our approach is computing the CRB for joint estimation of these three parameters for a known data sequence. It should be noted that the optimum training sequence can be algorithm-dependent and the CRB is only a lower bound on the MSE of an unbiased estimator. Nevertheless, it is known [2, Theorem 7.1] that ML estimators asymptotically achieve the CRB, making it a suitable criterion for designing the training sequence.

As mentioned earlier, the focus of this work is CPM due to the lack of analytical work on preamble design for this class of modulations. To the best of our knowledge, there is no analytical work on the design of training sequences for CPM in general. For instance, [11] introduces the “1100” repeating data pattern for UHF MILSATCOM standards solely based on its frequency spectrum. Mehlan and Meyr [10] have studied preamble design for symbol timing and frequency offset estimation in burst-mode MSK modulation, which is a very specific case of CPM. However, there is a demand for finding the best training sequence to be employed in current and future standards —especially in burst-mode communications. As an example, the integrated network enhanced telemetry (iNET) standard [43], which is currently being developed for telemetry applications, utilizes CPM schemes for transmitting burst packets, which indeed requires fast and precise synchronization based on a training sequence.

Unlike CPM, a few publications have addressed derivation of CRBs and training sequence design for linear modulations [8, 16, 14, 13, 12]. Jiang *et al.* in [8] have studied the CRB for joint DA estimation of carrier phase and symbol timing based on samples of the matched filter output. Although they present closed-form expressions for the CRB as a function of the data symbols, they do not find the best training sequence; they only compare the performance bounds for the alternating data sequence and a random sequence. The authors in [16] have addressed training sequence design for frequency-selective channels. They average the CRB for frequency estimation over the channel response and find the sequence that minimizes the CRB via a computer search. In independent and recent works [14, 13], the authors design the optimum training sequence for linearly modulated signals using the CRB approach. They confirm that the widely-used alternating sequence minimizes the symbol timing CRB in a joint estimation scenario. In [12], the best training sequence for burst-mode OQPSK is found via CRB simulations.

Using the CRB criterion, we make the following contributions. We derive the closed-form expressions for symbol timing, carrier phase, and frequency offset CRBs for joint DA estimation from CPM signals when transmitted over AWGN channels. The optimum training sequence for the symbol timing CRB is found analytically using a two-step optimization approach. Since our optimization problem is defined over a discrete set, i.e., M-ary data symbols, we make certain assumptions in our analysis. In order to validate these assumptions and the proposed training sequence, we resort to a computer search method known as a genetic algorithm (GA) [44], which will be shown to be suitable in this type of discrete optimization problem as a substitute for lengthy exhaustive searches. The GA is used for the optimization of the training sequence in other applications such as OFDM [45] and spatial multiplexing transmissions [46]. We show that the GA search either finds our proposed sequence or finds one that is similar yet suboptimum to our proposed sequence; in no case is a sequence found that is superior to our proposed sequence, which confirms our optimization method. Furthermore, we analytically derive the optimum sequence for carrier phase and frequency offset and this conveniently turns out to be the same sequence as the one proposed for symbol timing. This is in contrast to linear modulations (i.e., [13]) in which different sequences prove to be optimal for different estimation parameters. Moreover, we compare the performance of DA estimators with the proposed optimized training sequence and a randomly chosen training sequence. The true CRB for a random (unknown) data sequence in linear modulations has been presented in the literature [47, 15]. However, the nonlinear nature of CPM signals makes it almost impossible to present similar results herein. We tackle this problem by averaging the CRB over the probability density function (pdf) of the data sequence, as suggested in [48]. We derive a closed-form expression for the CRB for symbol timing estimation, which we are able to use in evaluating estimators with a random data sequence. We observe the performance gain that the optimized training sequence delivers compared to a random data sequence as a function of sequence length, alphabet-size and CPM phase response length.

The rest of this chapter is organized as follows. Section 2.3 presents the derivation of the joint CRB for CPM signals for a known data sequence. In Section 2.4, we derive the optimum training sequence via analysis and computer search. Section 2.5 discusses the CRB when the data sequence is random. The CRB results for some CPM scenarios are outlined and compared in Section 2.6. Finally, conclusions are drawn in Section 2.7.

2.3 CRB for CPM

Let us recall the complex baseband CPM signal as

$$s(t) = \sqrt{\frac{E_s}{T_s}} \exp\{j\phi(t;\boldsymbol{\alpha})\}, \tag{2.1}$$

where E_s is the energy per transmitted symbol and T_s is the symbol duration. The phase of the signal $\phi(t;\boldsymbol{\alpha})$ during the transmission of the preamble in each burst is represented as

$$\phi(t;\boldsymbol{\alpha}) = 2\pi h \sum_{i=0}^{L_0-1} \alpha_i q(t - iT_s), \tag{2.2}$$

where $\{\alpha_i\}$ is the sequence of M-ary data symbols selected from the set $\{\pm 1, \pm 3, \ldots, \pm(M-1)\}$.

Assuming transmission over an AWGN channel, the complex baseband representation of the received signal is

$$r(t) = \sqrt{\frac{E_s}{T_s}} e^{j(2\pi f_d t + \theta)} e^{j\phi(t-\tau;\boldsymbol{\alpha})} + w(t), \tag{2.3}$$

where θ is the unknown carrier phase, f_d is the frequency offset, τ is the timing offset, and $w(t)$ is complex baseband AWGN with zero mean and power spectral density N_0. The transmitted data symbols are denoted by $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_{L_0-1}]$. In the following, we denote the first term in (2.3) as the signal component of $r(t)$, i.e., $s(t, \mathbf{u}, \boldsymbol{\alpha})$ where $\mathbf{u} = [f_d, \theta, \tau]^T$ is the vector of unknown but deterministic parameters which are to be *jointly* estimated at the receiver.

Let us refer to an *unbiased* estimate of the unknown parameters by $\hat{\mathbf{u}}$. Regardless of the estimation method, one is generally interested in a lower bound on the variance of the estimation error as a performance metric. The CRB is a lower bound on the error covariance matrix $\mathbf{C}_{\hat{\mathbf{u}}}$ for the joint estimation of the u_i's,

$$E\{(\hat{u}_i - u_i)^2\} = [\mathbf{C}_{\hat{\mathbf{u}}}]_{i,i} \geq \left[\mathbf{I}(\mathbf{u})^{-1}\right]_{i,i} = \mathrm{CRB}(u_i|\boldsymbol{\alpha}), \tag{2.4}$$

where $\mathbf{I}(\mathbf{u})$ is the Fisher information matrix (FIM) with elements defined as

$$\mathbf{I}_{i,j}(\mathbf{u}) = -E\left[\frac{\partial^2}{\partial u_i \partial u_j} \ln(\Lambda(\mathbf{r}; \mathbf{u}, \boldsymbol{\alpha}))\right]. \tag{2.5}$$

Our known training sequence $\boldsymbol{\alpha}$ is implicit in the definition of $\mathbf{I}(\mathbf{u})$. Additionally, the expectation is performed with respect to the received signal given $\boldsymbol{\alpha}$ and $\mathbf{u}$. In (2.4), the *conditional* CRB for estimation of u_i is denoted as $\mathrm{CRB}(u_i|\boldsymbol{\alpha})$ in order to emphasize the knowledge of the training sequence. We note that the FIM is a symmetric matrix and is a function of the log-likelihood function (LLF) $\ln(\Lambda(\mathbf{r}; \mathbf{u}, \boldsymbol{\alpha}))$. The above LLF is defined based on a sampled version of $r(t)$, namely $\mathbf{r}$; However, one can easily derive a continuous time version based on the series representation of $r(t)$ as discussed in [49, Section 4.2.3]. Accordingly, the FIM elements corresponding to a CPM signal embedded in AWGN can be written as

$$\mathbf{I}_{i,j}(\mathbf{u}) = -\frac{2}{N_0}\int_0^{T_0} \mathrm{Re}\left[s(t, \mathbf{u}, \boldsymbol{\alpha})\frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial u_i \partial u_j}\right] dt. \tag{2.6}$$

where T_0 is the observation length, $\mathrm{Re}[\cdot]$ is the real part operator and * represents the complex conjugate operation. Note that $T_0 = L_0 T_s$, which is equal to the preamble duration.

The nine partial derivatives required for the computation of the FIM are

$$\frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial f_d\, \partial f_d} = -4\pi^2 t^2 \sqrt{\frac{E_s}{T_s}} e^{-j(2\pi f_d t+\theta)} e^{-j\phi(t-\tau;\boldsymbol{\alpha})} \tag{2.7}$$

$$\frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial f_d\, \partial \theta} = \frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial \theta\, \partial f_d} = -2\pi t \sqrt{\frac{E_s}{T_s}} e^{-j(2\pi f_d t+\theta)} e^{-j\phi(t-\tau;\boldsymbol{\alpha})} \tag{2.8}$$

$$\frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial f_d\, \partial \tau} = \frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial \tau\, \partial f_d} = -2\pi t \frac{\partial \phi(t-\tau;\boldsymbol{\alpha})}{\partial \tau} \sqrt{\frac{E_s}{T_s}} e^{-j(2\pi f_d t+\theta)} e^{-j\phi(t-\tau;\boldsymbol{\alpha})} \tag{2.9}$$

$$\frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial \theta\, \partial \theta} = -\sqrt{\frac{E_s}{T_s}} e^{-j(2\pi f_d t+\theta)} e^{-j\phi(t-\tau;\boldsymbol{\alpha})} \tag{2.10}$$

$$\frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial \theta\, \partial \tau} = \frac{\partial^2 s^*(t, \mathbf{u}, \boldsymbol{\alpha})}{\partial \tau\, \partial \theta} = -\frac{\partial \phi(t-\tau;\boldsymbol{\alpha})}{\partial \tau} \sqrt{\frac{E_s}{T_s}} e^{-j(2\pi f_d t+\theta)} e^{-j\phi(t-\tau;\boldsymbol{\alpha})} \tag{2.11}$$

$$\frac{\partial^2 s^*(t,\mathbf{u},\boldsymbol{\alpha})}{\partial\tau\,\partial\tau} = \left\{-j\frac{\partial^2\phi(t-\tau;\boldsymbol{\alpha})}{\partial\tau^2} - \left[\frac{\partial\phi(t-\tau;\boldsymbol{\alpha})}{\partial\tau}\right]^2\right\}\sqrt{\frac{E_s}{T_s}}e^{-j(2\pi f_d t+\theta)}e^{-j\phi(t-\tau;\boldsymbol{\alpha})}. \tag{2.12}$$

Substituting the above equations and $s(t,\mathbf{u},\boldsymbol{\alpha})$ of (2.3) in (2.6) cancels out the exponential terms. After taking the integral, we obtain the simplified FIM elements for the CPM as

$$\mathbf{I}_{11}(\mathbf{u}) = \frac{8\pi^2 T_0^3}{3T_s}\left(\frac{E_s}{N_0}\right) \tag{2.13}$$

$$\mathbf{I}_{12}(\mathbf{u}) = \mathbf{I}_{21}(\mathbf{u}) = \frac{2\pi T_0^2}{T_s}\left(\frac{E_s}{N_0}\right) \tag{2.14}$$

$$\mathbf{I}_{22}(\mathbf{u}) = \frac{2T_0}{T_s}\left(\frac{E_s}{N_0}\right) \tag{2.15}$$

$$\mathbf{I}_{13}(\mathbf{u}) = \mathbf{I}_{31}(\mathbf{u}) = \frac{4\pi\left(\frac{E_s}{N_0}\right)}{T_s}\int_0^{T_0} t\frac{\partial\phi(t-\tau;\boldsymbol{\alpha})}{\partial\tau}\,dt \tag{2.16}$$

$$\mathbf{I}_{23}(\mathbf{u}) = \mathbf{I}_{32}(\mathbf{u}) = \frac{2\left(\frac{E_s}{N_0}\right)}{T_s}\int_0^{T_0} \frac{\partial\phi(t-\tau;\boldsymbol{\alpha})}{\partial\tau}\,dt \tag{2.17}$$

$$\mathbf{I}_{33}(\mathbf{u}) = \frac{2\left(\frac{E_s}{N_0}\right)}{T_s}\int_0^{T_0} \left[\frac{\partial\phi(t-\tau;\boldsymbol{\alpha})}{\partial\tau}\right]^2 dt. \tag{2.18}$$

It is observed that the FIM elements for the frequency offset and carrier phase estimation when the timing is assumed to be perfectly known, i.e., $\mathbf{I}_{11}, \mathbf{I}_{12}, \mathbf{I}_{21}, \mathbf{I}_{22}$, *do not* depend on either the data sequence or the CPM characteristics [namely, M, h, L, and $q(t)$]. In fact, they only depend on the SNR, observation length and symbol duration. On the other hand, $\mathbf{I}_{13}, \mathbf{I}_{31}, \mathbf{I}_{23}, \mathbf{I}_{32}$, and $\mathbf{I}_{33}$ *do* depend on various modulated signal properties such as $\boldsymbol{\alpha}$, M, h, L, and $q(t)$ due to the presence of $\phi(t;\boldsymbol{\alpha})$ in the corresponding equations.

The derivative of the CPM phase for the data sequence $\boldsymbol{\alpha}$ and symbol timing τ is

$$\frac{\partial\phi(t-\tau;\boldsymbol{\alpha})}{\partial\tau} = -2\pi h\sum_{i=0}^{L_0-1}\alpha_i g(t-iT_s-\tau), \tag{2.19}$$

where $g(t) \triangleq \partial q(t)/\partial t$ is the frequency pulse. After substituting (2.19) in (2.16) to (2.18), the five FIM elements which are related to the symbol timing become

$$\mathbf{I}_{13}(\mathbf{u}) = \mathbf{I}_{31}(\mathbf{u}) = -\frac{8\pi^2 h\left(\frac{E_s}{N_0}\right)}{T_s}\sum_{i=0}^{L_0-1}\alpha_i\int_0^{T_0} tg(t - iT_s - \tau)\,dt \tag{2.20}$$

$$\mathbf{I}_{23}(\mathbf{u}) = \mathbf{I}_{32}(\mathbf{u}) = -\frac{4\pi h\left(\frac{E_s}{N_0}\right)}{T_s}\sum_{i=0}^{L_0-1}\alpha_i\int_0^{T_0} g(t - iT_s - \tau)\,dt \tag{2.21}$$

$$\mathbf{I}_{33}(\mathbf{u}) = \frac{8\pi^2 h^2\left(\frac{E_s}{N_0}\right)}{T_s}\sum_{i=0}^{L_0-1}\sum_{j=0}^{L_0-1}\alpha_i\alpha_j\int_0^{T_0} g(t - iT_s - \tau)g(t - jT_s - \tau)\,dt, \tag{2.22}$$

where we have changed the order of integration and summation in each of the above derivations. Finally, we can summarize the FIM as

$$\mathbf{I}(\mathbf{u}) = \frac{1}{T_s}\left(\frac{E_s}{N_0}\right)\begin{bmatrix} \frac{8\pi^2 T_0^3}{3} & 2\pi T_0^2 & -8\pi^2 hA \\ 2\pi T_0^2 & 2T_0 & -2\pi hB \\ -8\pi^2 hA & -2\pi hB & 8\pi^2 h^2 C \end{bmatrix}. \tag{2.23}$$

where the variables A, B and C are

$$A = \sum_{i=0}^{L_0-1}\alpha_i\int_0^{T_0} tg(t - iT_s - \tau)\,dt \tag{2.24}$$

$$B = \sum_{i=0}^{L_0-1}\alpha_i\int_0^{T_0} g(t - iT_s - \tau)\,dt \tag{2.25}$$

$$C = \sum_{i=0}^{L_0-1}\sum_{j=0}^{L_0-1}\alpha_i\alpha_j\int_0^{T_0} g(t - iT_s - \tau)g(t - jT_s - \tau)\,dt. \tag{2.26}$$

Note that each of these variables is a function of only τ and not f_d or θ.

Although the integrals required in the computation of A, B and C can be computed numerically, we can exploit the properties of the CPM phase response and simplify/approximate the computations. We begin the approximation of A by changing the integration variable to

$u = t - iT_s - \tau$

$$\begin{aligned} A &= \sum_{i=0}^{L_0-1} \alpha_i \int_{-iT_s-\tau}^{T_0-iT_s-\tau} (u + iT_s + \tau) g(u)\, du \\ &= \sum_{i=0}^{L_0-1} \alpha_i \left[\int_{-iT_s-\tau}^{T_0-iT_s-\tau} u g(u)\, du \;+\; (iT_s + \tau) \int_{-iT_s-\tau}^{T_0-iT_s-\tau} g(u)\, du \right] \\ &\approx \sum_{i=0}^{L_0-1} \alpha_i \left(\Gamma + \frac{\tau}{2} + i\frac{T_s}{2} \right), \end{aligned} \tag{2.27}$$

where the first integral is denoted as Γ and is approximated by $\Gamma = \int_{-\infty}^{\infty} t g(t) dt$. The value of Γ depends on the frequency pulse of the particular CPM signal. The second integral is basically the area under the frequency pulse which is equal to $1/2$ according to the CPM definition. In both of the aforementioned approximations, we have assumed that the integral limits cover the entire non-zero area under $g(t)$. However, this is not true for the last α_i's in the sequence when we are dealing with partial response CPM. In fact, the value of those integrals becomes smaller than Γ and $1/2$ for the last few symbols. Hence, we propose a better approximation by truncating the last $\lfloor L/2 \rfloor$ symbols, which makes the effective length of the sequence approximated by $L_1 = L_0 - \lfloor L/2 \rfloor$ where $\lfloor x \rfloor$ is the largest integer not greater than x. By using the same method, B is approximated according to

$$B \approx \frac{1}{2} \sum_{i=0}^{L_1-1} \alpha_i. \tag{2.28}$$

Let us introduce the correlation function of $g(t)$ as

$$R_g(t) = \int_{-\infty}^{\infty} g(u) g(u + t)\, du, \tag{2.29}$$

where $R_g(t)$ is an even function of t. Therefore, we approximate C as

$$C \approx \sum_{i=0}^{L_0-1} \sum_{j=0}^{L_0-1} \alpha_i \alpha_j R_g((i - j)T_s). \tag{2.30}$$

Table 2.1. The correlation function $R_g(t)$ for some commonly used CPM pulse shapes.

Pulse Shape	$R_g(0)$	$R_g(\pm T_s)$	$R_g(\pm 2T_s)$
1REC	0.25	0	0
2REC	0.125	0.0625	0
3REC	0.0833	0.0556	0.0278
1RC	0.375	0	0
2RC	0.1875	0.0312	0
3RC	0.125	0.0589	≈ 0
GMSK ($BT_s = 0.3$)	0.1597	0.0443	≈ 0

The correlation function $R_g(t)$ is tabulated in Table 2.1 at $t = nT_s$ for some commonly used CPM pulse shapes.

The dependency of the CRBs on τ is examined via simulation. In this set of simulations, we randomly generated one million 32-bit data sequences and computed their CRBs for LRC and LREC schemes where $h = 1/2$ and $M = 2$. For each data sequence, the symbol timing CRB is computed for different values of τ in $[-T_s/2, T_s/2]$ and the normalized deviation is computed via

$$\delta = \frac{\max\{|\mathrm{CRB}(\tau) - \mathrm{CRB}(\tau = 0)|\}}{\mathrm{CRB}(\tau = 0)}. \tag{2.31}$$

Accordingly, the empirical cumulative distribution function (CDF) of δ is presented in Figure 2.1. We can see for more than 90% of the sequences the deviation is less than 6%. Hence, we may conclude that the CRB is weakly dependent on τ, and thus, we assume $\tau = 0$ for the computation of CRBs in the following section. Therefore, the FIM becomes independent of all of the estimation parameters and is simply denoted by $\mathbf{I}$ in the rest of our discussion.

2.4 Best Sequence Design

Since the FIM is a function of the data sequence $\boldsymbol{\alpha}$, the performance bound of a DA estimator can be optimized by choosing the sequence that minimizes the CRB. In general, the best training sequence might be different for each of the synchronization parameters. We first address the optimum sequence for symbol timing. The optimum sequence design for frequency offset and

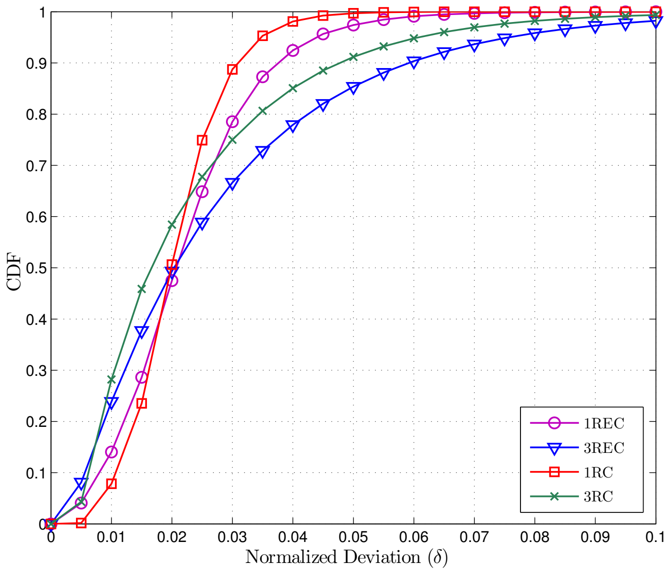


Figure 2.1. The CDF showing the effect of τ variations on symbol timing CRB for several CPM schemes.

carrier phase are discussed later.

2.4.1 Symbol Timing

In this section, we derive the symbol sequence $\boldsymbol{\alpha}$, which minimizes the entry in $\mathbf{I}(\mathbf{u})^{-1}$ corresponding to the error variance of symbol timing estimation. Here, we fix the SNR value to $E_s/N_0 = 1$ and symbol duration to $T_s = 1$ as they are scaling factors for the FIM and the resulting solution is independent of them. The FIM can be partitioned into four submatrices such that its inverse can be computed easily using the Lemma in [2, p. 572]. Taking the inverse of the FIM based on (2.23), the $(3,3)$-th element of $\mathbf{I^{-1}}$ becomes

$$\mathrm{CRB}(\tau|\boldsymbol{\alpha}) = \left[\mathbf{I}(\mathbf{u})^{-1}\right]_{33} = \left(8\pi^2h^2C - \left[-8\pi^2hA - 2\pi hB\right]\begin{bmatrix}\mathbf{I}_{11} & \mathbf{I}_{12}\\ \mathbf{I}_{21} & \mathbf{I}_{22}\end{bmatrix}^{-1}\begin{bmatrix}-8\pi^2hA\\ -2\pi hB\end{bmatrix}\right)^{-1}, \tag{2.32}$$

where the variables A, B, and C are replaced by their approximated values. Our goal is to minimize (2.32) or equivalently maximize its denominator. It is seen that the second term in

(2.32) is expressed in a quadratic form, i.e., $p = \mathbf{v}^T \mathbf{J} \mathbf{v}$, where $\mathbf{v}$ and $\mathbf{J}$ are

$$\mathbf{v} = \left[-8\pi^2 hA - 2\pi hB\right]^T, \tag{2.33}$$

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_{11} & \mathbf{I}_{12} \\ \mathbf{I}_{21} & \mathbf{I}_{22} \end{bmatrix}^{-1}. \tag{2.34}$$

It is trivial to show that the above quadratic term is non-negative. This is due to the fact that $\mathbf{J}^{-1}$ is in the form of the FIM which is positive-definite [2]. Thus, its inverse $\mathbf{J}$ exists and is positive-definite itself. Therefore, p is positive for all the values of $\mathbf{v} \neq 0$ according to the definition of positive-definite matrices. In order to maximize the denominator of (2.32), one has to maximize C and minimize p simultaneously. Based on the above discussion, $(A, B) = (0, 0)$ is the only solution which makes $p = 0$, and hence, it minimizes p. However, it might not be the same sequence for which C is maximized. On the other hand, the discrete values of $\boldsymbol{\alpha}$, i.e., $\alpha_i \in \{\pm 1, \pm 3, \ldots, \pm(M-1)\}$, prevent us from using well-known continuous optimization techniques. Thus, we initially assume C is independent of A and B. Therefore, the data sequence that minimizes the CRB for symbol timing is the solution to

$$\operatorname*{argmax}_{\boldsymbol{\alpha}} C \qquad \text{subject to} \quad A = B = 0. \tag{2.35}$$

We continue our analysis by assuming a binary CPM signal where $\alpha_i = \pm 1$. Higher order constellations will be discussed later on. This gives us a better understanding of the sequences that make $A = B = 0$. Moreover, we define the variables A' and B'

$$A' = \sum_{i=1}^{L_1} i\alpha_{i-1} \tag{2.36}$$

$$B' = \sum_{i=1}^{L_1} \alpha_{i-1}, \tag{2.37}$$

which are closely related to A and B respectively. It is easily seen that $A = B = 0$ if and only if $A' = B' = 0$. For a given L_1, there exist several binary sequences that satisfy both of these conditions from which the sequence that maximizes C has to be found via a search algorithm or analytically. Let us first examine C and see how it is affected by $\boldsymbol{\alpha}$ elements. According to (2.30), C can be expanded as

$$C = R_g(0) \sum_{i=0}^{L_0-1} \alpha_i^2 + 2R_g(T_s) \sum_{i=0}^{L_0-2} \alpha_i \alpha_{i+1} + \cdots . \tag{2.38}$$

We observe that the value of C does not depend on the data sequence in the case of full response CPM because $R_g(nT_s) = 0$ for $n > 0$ where n is an arbitrary integer number. Thus, we can select any sequence which satisfies conditions $A' = B' = 0$ as the optimum sequence. Partial response CPM requires more investigation. First, we notice that the first term in the above expansion is constant for all the sequences. Additionally, we assume that $R_g(nT_s) \geq 0$, which is true for the widely-used CPM schemes listed in Table 2.1. Therefore, we can state that C is maximized when all the symbol pairs, i.e., $\alpha_i \alpha_{i+n}$, have the same sign. In our discussion, we emphasize on the adjacent symbols, i.e., $\alpha_i \alpha_{i+1}$, due to the fact that the remaining factors, $R_g(nT_s)$ for $n > 1$, are very small according to Table 2.1.

Based on the above discussion, we propose the optimum training sequence as illustrated in Figure 2.2 that satisfies $A' = B' = 0$ while having a minimum number of sign transitions. The first $L_1/4$ symbols in the sequence are all -1 followed by $+1$ symbols having a number of $L_1/2$. The remainder of the sequence consists of all -1 symbols, which might be longer than $L_1/4$ because $L_1 < L_0$ for partial response CPM. It is trivial to show that our proposed sequence satisfies $A' = B' = 0$. Additionally, it only has two sign transitions which results in a near maximum value of C according to (2.38). The only competitors are the sequences with zero or one transition. However, they cannot satisfy our conditions on A' and B'.

Although our discussion has been limited to binary CPM, we can easily extend the results to M-ary CPM. Similar to the binary case, our goal is maximizing C while satisfying $A' = B' = 0$ because (2.32) is valid regardless of the alphabet size. If we re-examine the expanded form of C in (2.38), we notice that by selecting α_i's to be $\pm(M-1)$ instead of ± 1 in the proposed binary sequence, one is able to set C to its maximum possible value when $\alpha_i \in \{\pm 1, \pm 3, \cdots, \pm(M-1)\}$.

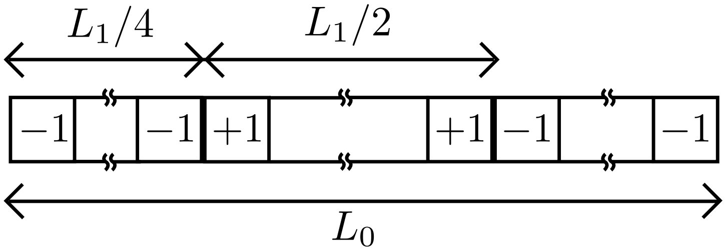


Figure 2.2. The proposed sequence for the joint estimation of frequency offset, carrier phase and symbol timing in CPM signals. This sequence and its negative are asymptotically optimum.

This procedure scales C by a factor of $(M-1)^2$. Moreover, scaling of the symbols in our proposed sequence by a constant factor does not affect the validity of $A' = B' = 0$ according to (2.36) and (2.37).

The proposed training sequence minimizes the symbol timing CRB which can be expressed as

$$\mathrm{CRB}_{\min}(\tau) \approx \frac{T_s}{\left(\frac{E_s}{N_0}\right) 8\pi^2 h^2 (M-1)^2 \left[L_0 R_g(0) + 2(L_0-5) R_g(T_s)\right]}. \tag{2.39}$$

The approximation in the above expression is due to the approximations in computation of A, B and C as well as the optimization process. The effect of these approximations will be discussed in Section 2.4.4.

2.4.2 Frequency Offset and Carrier Phase

Using a similar approach, we derive the inverse of the FIM elements $\mathbf{I}_{11}^{-1}$ and $\mathbf{I}_{22}^{-1}$ corresponding to frequency offset and carrier phase respectively. After some rearrangement, they can be written in the form of

$$\mathrm{CRB}(f_d|\boldsymbol{\alpha}) = \left[\mathbf{I}(\mathbf{u})^{-1}\right]_{11} = \frac{3}{2\pi^2 L_0^3} \times \underbrace{\frac{CL_0 - \frac{B^2}{4}}{CL_0 - \frac{B^2}{4} - \frac{3}{4L_0^2}(BL_0 - 4A)^2}}_{\gamma_1} \tag{2.40}$$

$$\mathrm{CRB}(\theta|\boldsymbol{\alpha}) = \left[\mathbf{I}(\mathbf{u})^{-1}\right]_{22} = \frac{2}{L_0} \times \underbrace{\frac{CL_0^3 - 3A^2}{CL_0^3 - 3A^2 - (BL_0 - 3A)^2}}_{\gamma_2}, \tag{2.41}$$

where we can prove that γ_1 and γ_2 are variables greater than or equal to one. This is due to the fact that both of them are in the form of $a/(a-b)$. In both cases, $a-b$ is basically the determinant of the FIM which is non-negative since the FIM is a positive definite matrix. Moreover, b is non-negative which makes $a \geq b \geq 0$. Consequently, γ_1 and γ_2 are minimized when the term which corresponds to b becomes equal to zero. This implies that the CRB for the frequency offset is minimized when $BL_0 = 4A$ and the CRB for the carrier phase is minimized when $BL_0 = 3A$. Theoretically, there exist several combinations of A and B which make either of these conditions satisfied. However, we opt for $A = B = 0$ such that it minimizes the CRBs for frequency offset and carrier phase simultaneously regardless of L_0. This result is very convenient because we have already considered it for designing the best sequence for symbol timing estimation. Hence, our proposed sequence minimizes the CRB for all of the three estimation parameters at the same time.

Recalling the original FIM in (2.23), the minimum value of the CRB for frequency offset estimation is

$$\mathrm{CRB}_{\min}(f_d) \approx \frac{3T_s}{2\pi^2 T_0^3 \left(\frac{E_s}{N_0}\right)} \tag{2.42}$$

and the minimum CRB for carrier phase estimation becomes

$$\mathrm{CRB}_{\min}(\theta) \approx \frac{2T_s}{T_0 \left(\frac{E_s}{N_0}\right)}. \tag{2.43}$$

Unlike $\mathrm{CRB}_{\min}(\tau)$, $\mathrm{CRB}_{\min}(f_d)$ and $\mathrm{CRB}_{\min}(\theta)$ do not depend on any of the CPM parameters while they are inversely proportional to the sequence length and its cubed version, respectively, for a given SNR.

As a final remark, we note that the optimum sequence forces $\mathbf{I}_{13}$, $\mathbf{I}_{23}$, $\mathbf{I}_{31}$ and $\mathbf{I}_{32}$ to become zero. Thus, one can say that the optimum training sequence *decouples* the estimation of symbol timing from carrier phase and frequency offset. Moreover, it is seen that the FIM 2×2 sub-matrix that corresponds to the joint estimation of f_d and θ is not a function of $\boldsymbol{\alpha}$ and leads to (2.42) and (2.43). From the symbol timing point of view, the optimum training sequence not only decouples τ but also maximizes $\mathbf{I}_{33}$ so that $\text{CRB}(\tau|\boldsymbol{\alpha})$ is minimized. This justifies our two-step optimization approach in which we first set A and B to zeros (decoupling) and then find the sequence that maximizes C.

2.4.3 Genetic Algorithm Search

So far, we have ignored the dependency of C and (A, B) in deriving the optimum sequence. In this section, we take advantage of computer search methods in order to confirm our proposed sequence as the best sequence. Clearly, an exhaustive search is quite time consuming even for moderate length sequences because of the vast search space. An alternative way is using systematic (guided) computer search algorithms which are frequently used in optimization problems. In this work, we resort to the GA search technique which is a type of *heuristic* search algorithm. The GA mimics the laws of natural selection (evolution) in the search process. GAs represent the potential solutions of a specific problem, i.e., our optimization problem, in *chromosome*-like data structures. The most commonly used data structure is binary vectors (bit-strings). This in fact motivates us to employ GAs in our optimization since our candidate solutions are already in the form of binary vectors and thus require no further encoding.

Due to the scope of this work, we refrain from explaining the GA operation. Interested readers may refer to [44] for further discussion. The GA parameters such as population size, crossover probability p_c and mutation probability p_m interact nonlinearly with each other. Therefore, the optimum settings may not be determined analytically and one should find the best combination for a specific problem experimentally. Here, we chose population size to be 50, $p_c = 0.8$, $p_m = 0.02$ and maximum number of generations equal to 200. Additionally, we ran each optimization scenario for 200 times with different initial populations to ensure that the GA results in a global minimum rather than a local one. Finally, we use the exact values of A, B and C in our GA implementation.

Using the GA as discussed above, we found the optimum training sequence for three well-known CPM schemes when $L_0 = 36$, which are depicted in Figure 2.3. These plots correspond

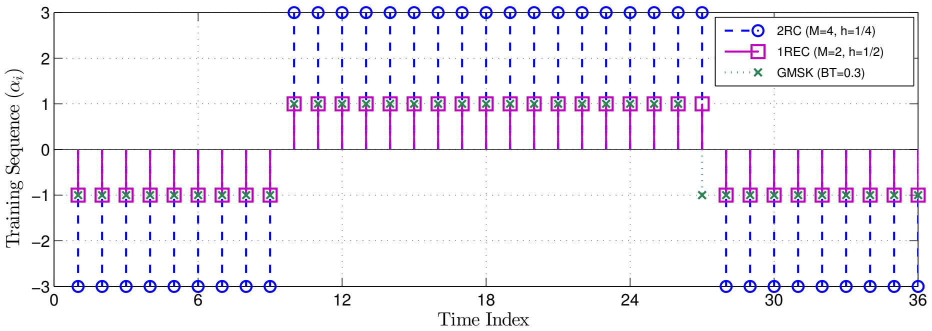


Figure 2.3. The optimum sequence for symbol timing estimation in CPM signals generated via genetic algorithm for $L_0 = 36$.

to binary h=1/2 1REC (MSK), GMSK with $L = 4$ and 4-ary $h = 1/4$ 2RC, which are examples of binary full response, binary partial response and M-ary partial response CPM respectively. Despite all these differences, the optimum sequences follow a pattern similar to the proposed sequence in Figure 2.2. It is observed that the 4-ary 2RC sequence is basically a scaled version of the optimum binary sequence. Also, the GMSK sequence has a different transition point compared to others since its effective length is smaller ($L_1 = 30 < L_0 = 32$). More importantly, these plots confirm our proposed sequence and the effectiveness of the GA in such a large search space (M^{36} possibilities). Finally, we ran the GA search for the same schemes with sequence lengths of 16 to 128. The ratio of the resulted minimum CRB to our proposed sequence's CRB is plotted in Figure 2.4. It can be seen that this ratio is greater than or equal to one, which means that the GA is not able to find a sequence better than the proposed one. In the case of MSK, the GA is successful in all cases and thus the ratio is always unity. In other CPM examples, this ratio becomes slightly larger than one as the sequence length increases because the search space becomes so large that the GA converges to a local minimum. This issue is more obvious for the 4-ary case since the size of the search space is squared relative to the binary case. Nevertheless, the GA results validate the accuracy of the assumptions we made in our analysis for deriving the optimum training sequence.

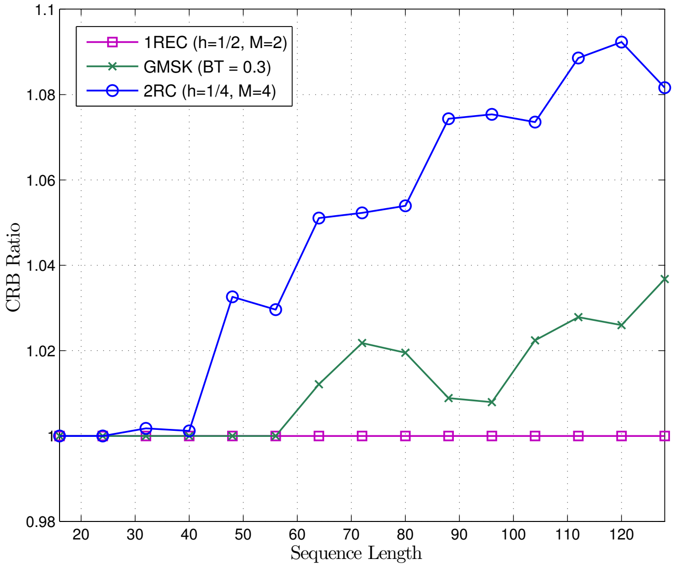


Figure 2.4. The ratio of symbol timing CRB obtained from GA search to the proposed sequence for different training sequence lengths and CPM schemes. The proposed sequence has an equal or lower CRB in all cases.

2.4.4 Remarks

The derivation of the optimum training sequence involved certain approximations and assumptions. The effect and importance of these approximations can be summarized as follows.

- Approximating A, B and C as in (2.27) and (2.30) introduces an *edge-effect* in the computations since we assumed the integrals cover the entire non-zero area of the integrands. It should be noted that this is only present in the integrals corresponding to the edges of the training sequence. Therefore, it becomes less significant as the length of the sequence is increased. Moreover, we introduced the effective length for partial response signals in order to reduce the edge-effect. The edge-effect does not exist in the case of full response signals assuming $\tau = 0$.

- It can be argued that there might be a sequence for which the symbol timing CRB, (2.32), is less than for the proposed one, which would violate our assumptions. Here, we consider a worst case scenario, in which a hypothetical sequence $\tilde{\boldsymbol{\alpha}}$ makes $(A, B) = 0$ while it maximizes C. The maximum possible value of C is $\tilde{C} \approx L_0 R_g(0) + 2(L_0 - 1)R_g(T_s)$ based on (2.38) and knowing that $R_g(nT_s) \approx 0$ for $n > 1$ in practical CPM schemes. The proposed sequence $\boldsymbol{\alpha}^*$ will result in $C^* = \tilde{C} - 8R_g(Ts)$ because it has two sign transitions. Thus, we can write the ratio of resulting CRBs using (2.32) as

$$\frac{\mathrm{CRB}(\tilde{\boldsymbol{\alpha}}|\tau)}{\mathrm{CRB}(\boldsymbol{\alpha}^*|\tau)} = \frac{\tilde{C} - 8R_g(T_s)}{\tilde{C}} = 1 - \frac{8R_g(Ts)}{L_0 R_g(0) + 2(L_0 - 1)R_g(T_s)}, \tag{2.44}$$

which approaches to 1 as L_0 increases and the fact that $R_g(0) \gg R_g(T_s)$. Therefore, the proposed training sequence is shown to be asymptotically optimum in terms of minimizing the symbol timing CRB. Note that this issue is not present in the carrier frequency and phase CRB optimization.

To be precise, there is no guarantee that the proposed sequence is optimum in general due to the above approximations, i.e. there might be sequences for which the CRB is smaller than the proposed one. However, we observed that these approximations become minimal as the sequence length is increased, and hence, the proposed training sequence is asymptotically optimum. Furthermore, the GA search results follows the proposed sequence for short and moderate sequence lengths. Thus, we refer to the proposed sequence as the optimum sequence for the sake of simplicity while we bear in mind that it might be near-optimum for shorter lengths.

2.5 CRB for Random Data Sequence

2.5.1 True CRB

After finding the optimum sequence, it is interesting to compare the estimation performance of the optimum sequence and a random (unknown) data sequence in terms of the CRB. This requires the computation of the true CRBs considering the data sequence $\boldsymbol{\alpha}$ as an unwanted or nuisance parameter. The computation of the CRB in this scenario is known to be quite challenging [1], especially for CPM signals due to their nonlinear nature. In this section, we briefly present the true CRB for a special case of CPM (binary and full-response).

We take advantage of the PAM representation of CPM signals [50] for computing the likelihood function. When the CPM signal is binary and full-response, it can be expressed as

$$s(t) = \sum_i a_i c_0(t - iT_s) \tag{2.45}$$

where a_i's are *pseudo symbols* and $c_0(t)$ is a pulse with duration of $2T_s$. When $h = 1/2$, the pseudo symbols become uncorrelated and can be detected according to:

$$\tilde{a}_n = \begin{cases} \displaystyle\int_{2T_s} \mathrm{Re}[r(t)c_0(t - nT_s - \tau)e^{-j(2\pi f_d t + \theta)}]dt & n - \text{even} \\ \displaystyle\int_{2T_s} \mathrm{Im}[r(t)c_0(t - nT_s - \tau)e^{-j(2\pi f_d t + \theta)}]dt & n - \text{odd} \end{cases}. \tag{2.46}$$

It should be noted that the imaginary unit j is ignored so that both even and odd time symbols are treated the same way. Therefore, the likelihood function can be written as

$$\Lambda[r(t); \mathbf{u}, \mathbf{a}] = \prod_{i=0}^{L_0-1} \exp\left(\frac{2E_s}{N_0}\tilde{a}_i a_i\right) \tag{2.47}$$

Because the transmitted symbols are assumed to be random, they can be taken out of the equation by averaging the likelihood function, i.e.

$$\Lambda[r(t); \mathbf{u}] = E_{\mathbf{a}}\{\Lambda[r(t); \mathbf{u}, \mathbf{a}]\} = \prod_{i=0}^{L_0-1} E_{\mathbf{a}}\left\{\exp\left(\frac{2E_s}{N_0}\tilde{a}_i a_i\right)\right\} \tag{2.48}$$

where $a_i = \pm 1$ with equal probability of 1/2 for binary modulation. After averaging over the pseudo symbols and mathematical simplifications, the partial derivatives of the LLF are found as

$$\frac{\partial \ln \Lambda[r(t); \mathbf{u}]}{\partial u_k} = \sum_{i=0}^{L_0-1} \left(\frac{2E_s}{N_0}\right) \frac{\partial \tilde{a}_i}{\partial u_k} \tanh\left(\frac{2E_s}{N_0}\tilde{a}_i\right). \tag{2.49}$$

Finally, the FIM elements are computed using

$$\mathbf{I}(\mathbf{u})_{k,l} = \left(\frac{2E_s}{N_0}\right)^2 \sum_{i=0}^{L_0-1} \sum_{j=0}^{L_0-1} E\left\{\frac{\partial \tilde{a}_i}{\partial u_k} \frac{\partial \tilde{a}_j}{\partial u_l} \tanh\left(\frac{2E_s}{N_0}\tilde{a}_i\right) \tanh\left(\frac{2E_s}{N_0}\tilde{a}_j\right)\right\} \tag{2.50}$$

where the expectation is performed over the received waveform.

Although it is possible to further simply (2.50) for a given pulse shape $c_0(t)$, we proceed using Monte Carlo simulations. In this fashion, we generate random binary data sequences which are modulated using the given CPM scheme and sent over an AWGN channel. The pseudo symbols are estimated by two matched filters (MFs) defined in (2.46). Additionally, the partial derivatives $\partial\tilde{a}_i/\partial u_k$ are found by another group of MFs, which are basically the partial derivatives of (2.46) with respect to appropriate parameter.

2.5.2 UCRB

The previous section showed us the challenges of computation of true CRB for CPM signals as it was only feasible for the simplest form of CPM. In this section, we resort to a different method such that the conditional CRB of Section 2.3 is averaged over $\boldsymbol{\alpha}$, and hence, it becomes independent of the data sequence.

As presented in [48], averaging both sides of (2.4) with respect to $\boldsymbol{\alpha}$ results in a lower bound on the *unconditional* estimation of τ (or any other parameter) from the received signal,

$$E_{\mathbf{r}}\left[(\hat{\tau} - \tau)^2\right] \geq E_{\boldsymbol{\alpha}}\left[\mathrm{CRB}(\tau|\boldsymbol{\alpha})\right]. \tag{2.51}$$

We refer to the right hand side of (2.51) as the UCRB, which should not be confused with the true CRB or the modified CRB (MCRB) [51]. The true CRB is different in the way that averaging takes place on the likelihood function of the received signal such that it is no longer a function of any nuisance parameters, i.e., $\boldsymbol{\alpha}$ in our problem. On the other hand, the MCRB is computed by averaging the FIM elements over the nuisance parameter.

In this section, we derive a closed-form expression for the symbol timing UCRB in the same scenario as previously, i.e., joint estimation of the synchronization parameters from CPM signals in AWGN. Let us first rearrange the symbol timing CRB according to (2.32)

$$\mathrm{CRB}(\tau|\boldsymbol{\alpha}) = \frac{T_s}{E_s/N_0} \frac{1}{\boldsymbol{\alpha}^T \mathbf{G} \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{D} \mathbf{J} \mathbf{D}^T \boldsymbol{\alpha}} = \frac{T_s}{E_s/N_0} \frac{1}{\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha}}, \tag{2.52}$$

where $\mathbf{Q}$ is an $L_0 \times L_0$ matrix defined as $\mathbf{Q} = \mathbf{G} - \mathbf{D}\mathbf{J}\mathbf{D}^T$. $\mathbf{G}$ is an $L_0 \times L_0$ correlation matrix defined as

$$\mathbf{G} = 8\pi^2 h^2 \begin{bmatrix} R_g(0) & R_g(T_s) & \dots & R_g((L_0-1)T_s) \\ R_g(-T_s) & R_g(0) & R_g(T_s) & \dots \\ \vdots & \vdots & \ddots & \dots \\ R_g((1-L_0)T_s) & \dots & R_g(-T_s) & R_g(0) \end{bmatrix}, \tag{2.53}$$

and $\mathbf{D}$ is an $L_0 \times 2$ matrix representing A and B with a form of

$$\mathbf{D} = \begin{bmatrix} -8\pi^2 h(\Gamma + \tau/2) & -\pi h \\ -8\pi^2 h(\Gamma + \tau/2 + T_s/2) & -\pi h \\ \vdots & \vdots \\ -8\pi^2 h(\Gamma + \tau/2 + (L_0-1)T_s/2) & -\pi h \end{bmatrix}. \tag{2.54}$$

Finally, $\mathbf{J}$ is defined in (2.34). We note that $\mathbf{Q}^T = \mathbf{G}^T - \mathbf{D}\mathbf{J}^T\mathbf{D}^T = \mathbf{Q}$ because $\mathbf{G}$ and $\mathbf{J}$ are symmetric matrices. Thus, $\mathbf{Q}$ is also a symmetric (Hermitian) matrix. Additionally, $\mathbf{Q}$ appears in a quadratic form in the denominator of (2.52). Because the CRB cannot be negative, the term $\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}$ is positive for all non-zero values of $\boldsymbol{\alpha}$, and hence, $\mathbf{Q}$ is a positive definite matrix. The above quadratic form can also be viewed as a random variable in which the random vector $\boldsymbol{\alpha}$ is transformed by $\mathbf{Q}$, i.e., $Z = \boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}$.

In order to derive the UCRB, one has to take the expectation of (2.52) with respect to $\boldsymbol{\alpha}$. This requires the computation of $E\{1/Z\}$ where Z is a discrete random variable with a non-trivial probability mass function (pmf). In order to tackle this problem, we can express

$1/Z$ in the vicinity of $\mu_Z = E\{Z\}$ using the Taylor series, i.e.,

$$\frac{1}{Z} = \frac{1}{\mu_z} - \frac{Z-\mu_z}{\mu_z^2} + \frac{(Z-\mu_z)^2}{\mu_z^3} + \cdots . \tag{2.55}$$

The above series expansion allows us to obtain $E\{1/Z\}$ using moments of Z itself rather than its reciprocal. Additionally, we approximate (2.55) by using the first three terms of its Taylor series. This requires us to compute only the first two moments. Based on the above discussion, we take the expectation of (2.55) and apply it to (2.52). Hence, the UCRB can be approximated as

$$\text{UCRB}(\tau) = E_{\boldsymbol{\alpha}}\{\text{CRB}(\tau|\boldsymbol{\alpha})\} \approx \frac{T_s}{E_s/N_0}\frac{E\{Z^2\}}{\mu_z^3}. \tag{2.56}$$

The only part left is the computation of μ_z and $E\{Z^2\}$. Due to the symmetry of $\mathbf{Q}$ one can decompose it such that [52, Theorem 6.2] $\mathbf{Q} = \mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V}$ where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{L_0}]^T$ is a unitary matrix consisting of eigenvectors of $\mathbf{Q}$. $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{L_0})$ is also a diagonal matrix with diagonal elements that correspond to eigenvalues of $\mathbf{Q}$. Using the eigen-decomposition, we can decompose Z such that

$$Z = \boldsymbol{\alpha}^T(\mathbf{V}^T\boldsymbol{\Lambda}\mathbf{V})\boldsymbol{\alpha} = \mathbf{x^T}\boldsymbol{\Lambda}\mathbf{x} = \sum_{i=1}^{L_0}\lambda_i X_i^2, \tag{2.57}$$

where $\mathbf{x} = \mathbf{V}\boldsymbol{\alpha} = [X_1, X_2, \cdots, X_{L_0}]^T$. We observe that $\mathbf{x}$ is a vector of random variables each of which is a transformation of $\boldsymbol{\alpha}$ via one of the eigenvectors of $\mathbf{Q}$. The importance of eigen-decomposition of Z is apparent in (2.57), where it becomes a linear combination of random variables X_i^2. Furthermore, the X_i's are uncorrelated because of the orthogonality of eigenvectors. These properties enable us to investigate statistical properties of Z. Following (2.57), the expected value of Z is computed by

$$E\{Z\} = \sum_{i=1}^{L_0}\lambda_i E\{X_i^2\} = \sum_{i=1}^{L_0}\lambda_i \mathbf{v}_i^T E\{\boldsymbol{\alpha}\boldsymbol{\alpha}^T\}\mathbf{v}_i = \frac{M^2-1}{3}\sum_{i=1}^{L_0}\lambda_i, \tag{2.58}$$

where the last equality holds, first because $E\{\boldsymbol{\alpha}\boldsymbol{\alpha}^T\}$ is a diagonal matrix with diagonal elements of $(M^2-1)/3$ when dealing with M-ary modulation, and second, $\|\mathbf{v}_i\| = 1$. Therefore, we are able to compute the mean of Z by summing the eigenvalues of $\mathbf{Q}$. The second moment of Z can be derived as

$$E\{Z^2\} = \boldsymbol{\lambda}^T E\{\mathbf{x_2 x_2^T}\}\boldsymbol{\lambda} = \boldsymbol{\lambda}^T \mathbf{C_{x_2}}\boldsymbol{\lambda}, \tag{2.59}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_{L_0}]^T$ and $\mathbf{x_2} = [X_1^2, X_2^2, \ldots, X_{L_0}^2]^T$. The computation of the covariance matrix of $\mathbf{x}_2$, i.e., $\mathbf{C_{x_2}} = E\{\mathbf{x_2 x_2^T}\}$, is not straightforward and requires computing $E\{X_i^4\}$ and $E\{X_i^2 X_j^2\}$. However, we can make an approximation by assuming the X_i's are Gaussian random variables with zero mean and variance of $(M^2-1)/3$. This is a good approximation due to the central limit theorem because X_i is a linear combination of independent random variables α_j's, i.e., $X_i = v_1^{(i)}\alpha_1 + v_2^{(i)}\alpha_1 + \cdots + v_{L_0}^{(i)}\alpha_{L_0}$ where $v_k^{(i)}$ is the k-th element of $\mathbf{v}_i$. The Gaussian approximation becomes more accurate as the sequence length grows and the eigenvector constructing X_i does not have large variations. The latter is because the central limit theorem requires summation of *identically* distributed random variables. Based on this assumption, $\mathbf{C_{x_2}}$ becomes

$$[\mathbf{C_{x_2}}]_{ij} \approx \begin{cases} \dfrac{(M^2-1)^2}{3} & i = j \\ \dfrac{(M^2-1)^2}{9} & i \neq j. \end{cases} \tag{2.60}$$

The second case is true because uncorrelated Gaussian random variables are independent as well, and hence, $E\{X_i^2 X_j^2\} = E\{X_i^2\}E\{X_j^2\}$. In Appendix A, we have computed the exact value of $\mathbf{C_{x_2}}$ where we show that the above approximation is valid even without Gaussian approximation.

Based on the above discussion, the first step in computing UCRB(τ) is deriving eigenvalues of $\mathbf{Q}$ and employing them in (2.58) and (2.59). Although eigen-decomposition can be performed using widely-available numerical analysis software, it is still a computationally complex task. Examining (2.58) and (2.59) reveals that we do not have to compute the individual eigenvalues. According to (2.58), one only needs to calculate sum of the eigenvalues of $\mathbf{Q}$ which is equal to the trace of that matrix, i.e., $\mu_z = \frac{M^2-1}{3}\text{trace}(\mathbf{Q})$. Furthermore, using (2.60) in (2.59) leads to

$$\begin{aligned} E\{Z^2\} &\approx \frac{(M^2-1)^2}{9}\left[\sum_i \lambda_i\right]^2 + \frac{2(M^2-1)^2}{9}\sum_i \lambda_i^2 \\ &= \frac{(M^2-1)^2}{9}\text{trace}(\mathbf{Q})^2 + \frac{2(M^2-1)^2}{9}\text{trace}(\mathbf{Q}^2). \end{aligned} \tag{2.61}$$

Thus, we conclude that the computation of UCRB(τ) based on (2.56) can be achieved by calculating the trace of the matrix $\mathbf{Q}$ and its square rather than its eigen-decomposition.

The computation of UCRB(f_d) and UCRB(θ) requires further investigation because each of them is expressed in the form of the expected value of the *ratio* of two quadratic forms [see (2.40) and (2.41)], i.e., UCRB $= E\{\boldsymbol{\alpha}^T\mathbf{P}\boldsymbol{\alpha}/\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}\}$. As a first degree approximation, they can be calculated by taking the ratio of two expected values, that is UCRB $\approx E\{\boldsymbol{\alpha}^T\mathbf{P}\boldsymbol{\alpha}\}/E\{\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}\}$. Nonetheless, more accurate values can be obtained numerically via Monte Carlo simulations.

2.6 Discussion and Results

The results for the symbol timing CRB using our optimum sequence for different CPM schemes and $L_0 = 32$ are shown in Figure 2.5. As we mentioned earlier, the effective length of a sequence is less than L_0 for partial-response schemes, which may violate our implicit assumption that L_1 is a multiple of 4. We approached this problem by rounding the values shown in Figure 2.2 to the nearest integer and building the optimum sequence upon those values. For instance, the effective length for GMSK case, assuming $L = 4$, is $L_1 = 32 - \lfloor 4/2 \rfloor = 30$, which is clearly not a multiple of 4. However, one can construct the optimum sequence for GMSK scheme such that it consists of eight -1's, followed by fifteen $+1$'s, followed by nine -1's according to Figure 2.2. The same discussion applies to the 4-ary 2RC example in which $L_1 = 31$ except for that we should use ± 3 symbols instead of ± 1 ones.

Based on (2.39), the symbol timing CRB for optimum training sequence is affected by modulation parameters for a given sequence length. These effects are depicted in Figure 2.5 for four CPM examples. The lowest CRB shown is attained using 4-ary symbols and a 2RC

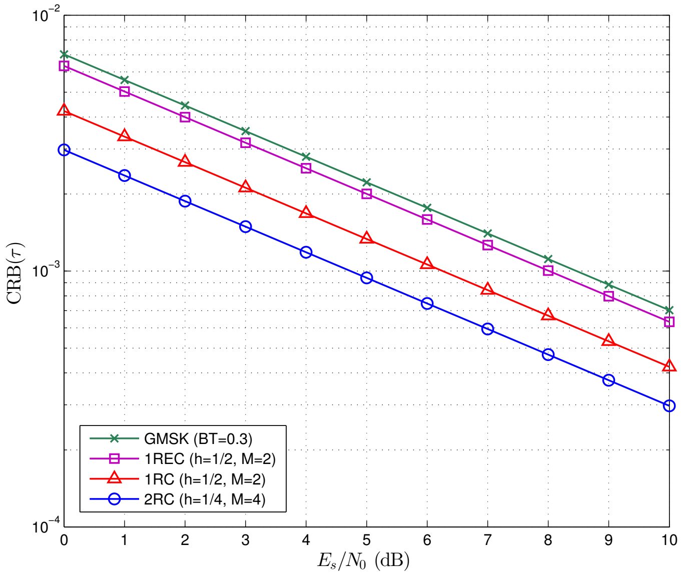


Figure 2.5. The CRB for symbol timing estimation of different CPM schemes for training sequence shown in Figure 2.2 and $L_0 = 32$.

frequency pulse, mainly because the symbol timing CRB is inversely proportional to $h^2(M-1)^2$ according to (2.39). This metric is 9/4 times larger for this scheme than the other three binary schemes. Among binary schemes, 1RC results in the lowest CRB. The reason is that 1RC has the largest value for $R_g(0)$ based on Table I, and hence, the smallest $\text{CRB}_{\text{min}}(\tau)$ value. The GMSK frequency pulse has the smallest $R_g(0)$, which results in the highest CRB value. It should be noted that these schemes have different bandwidths for a fixed symbol rate. Thus, an investigation should be carried out based on the CRB, signal bandwidth and symbol rate in order to make a fair decision on the scheme to be deployed in a particular application. This is beyond the scope of the current work and is a topic for future research. As mentioned earlier, the optimum frequency offset and carrier phase CRBs do not depend on any of the CPM parameters including the frequency pulse. Thus, we do not show plots similar to Figure 2.5 for frequency offset and carrier phase because they do not provide any useful comparison in terms of frequency pulse, phase response length and alphabet size.

The performance of the optimum training sequence versus a random sequence can be obtained in terms of the ratio of the UCRB to the optimum training sequence's CRB for any given

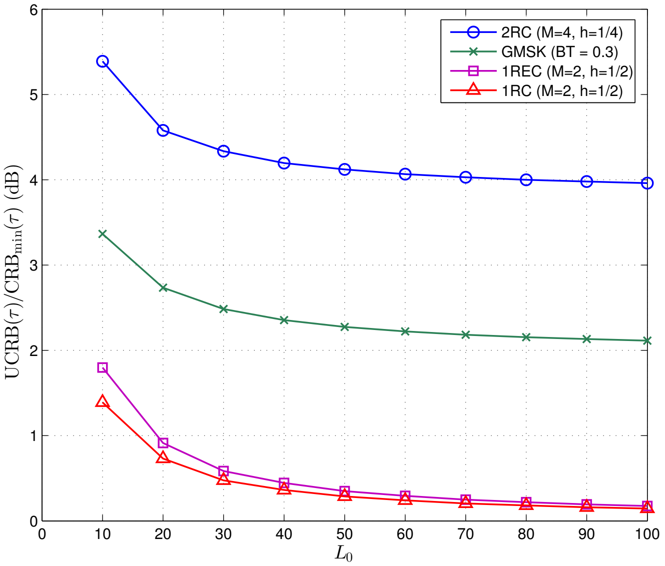
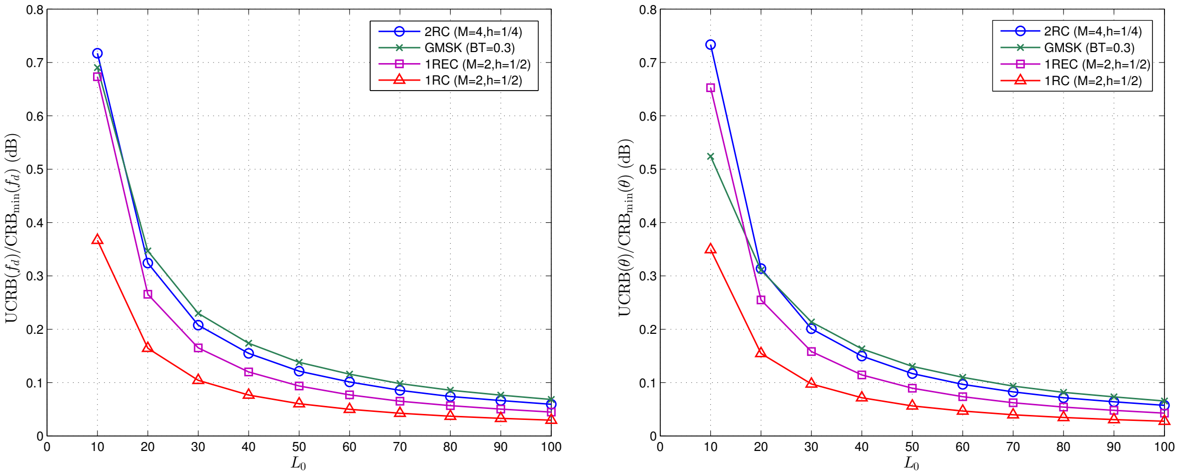


Figure 2.6. The ratio of the UCRB to the optimum training sequence's CRB for symbol timing estimation of different CPM schemes.

modulation. This ratio is plotted in Figure 2.6 as a function of sequence length for several CPM schemes. Additionally, the ratio is presented in terms of dB which can be interpreted as the performance gain that one can expect when a DA estimator with optimum training sequence is utilized rather than a random training sequence. It is seen that the CRB ratio decreases with a non-uniform rate as the sequence length increases, i.e., it has a high decline rate for shorter sequence lengths and converges to a limit at longer sequences. This is because as the sequence length increases, a random data sequence becomes less likely to be one of the *worst* sequences. In other words, a randomly selected data sequence would exhibit, to some extent, the statistical properties of data symbols for sufficiently large sequences. This causes the $\mathrm{CRB}(\tau|\boldsymbol{\alpha})$ to converge to $\mathrm{CRB}(\tau|\bar{\boldsymbol{\alpha}})$ in which $\bar{\boldsymbol{\alpha}}$ is a hypothetical sequence which possesses the average statistics of α_i. Despite the convergence for all the schemes, the limit to which the ratio is converging differs greatly. First, we note that for binary and full-response CPM signals such as 1REC and 1RC, the UCRB converges to the optimum CRB for longer sequences. This is due to the fact that $E\{\alpha_i^2\} = \alpha_i^2 = 1$ in binary signals, which makes C [defined in (2.30)] a constant for full-response signals. On the other hand, the CRBs for partial response and/or



(a) (b)

Figure 2.7. The ratio of the UCRB to the optimum training sequence's CRB for frequency (a) and phase (b) estimation of different CPM schemes.

non-binary schemes exhibit a performance gap even for long sequences. For example, the UCRB is almost 4 dB larger than the optimum CRB when a 4-ary partial-response CPM with the 2RC frequency pulse is used. This suggests the importance of employing a DA estimator based on the optimum training sequence when using partial-response or non-binary CPMs. Moreover, we computed the UCRB for frequency and phase estimations via simulations and it turned out to be weakly dependent on the underlying CPM–similar to the optimum training sequence CRB. For instance, $\mathrm{UCRB}(f_d)/\mathrm{CRB}_{\min}(f_d)$ and $\mathrm{UCRB}(\theta)/\mathrm{CRB}_{\min}(\theta)$ with respect to L_0 are plotted in Figures 2.7 (a) and 2.7 (b) respectively for the aforementioned CPMs. It can be observed that the frequency and phase estimations are less sensitive to the selection of the training sequence especially for longer training sequences. Finally, it should be mentioned that the CRB ratios in Figures 2.6 and 2.7 do not depend on the SNR as both $\mathrm{CRB}_{\min}$ and UCRB are inversely proportional to E_s/N_0.

It is interesting to investigate the relation between the UCRB, MCRB and the true CRB in our joint estimation problem since they all treat the training sequence as a random nuisance parameter. The MCRB for symbol timing estimation of CPM signals in additive noise is presented in [1, p. 68]. Based on our definitions, we can write

$$\mathrm{UCRB}(\tau) = E_{\boldsymbol{\alpha}}\{\mathbf{I}_{33}^{-1}\} \geq E_{\boldsymbol{\alpha}}\left\{\frac{1}{\mathbf{I}_{33}}\right\} \geq \frac{1}{E_{\boldsymbol{\alpha}}\{\mathbf{I}_{33}\}} = \mathrm{MCRB}(\tau), \tag{2.62}$$

where the first inequality holds because $\mathbf{I}_{33}^{-1} \geq 1/\mathbf{I}_{33}$ for a positive definite matrix [53, p. 81] and the second one is due to Jensen's inequality [52, p. 861]. Thus, the UCRB in this joint estimation case is expected to be higher than the MCRB. Unfortunately, there is no explicit relation between the UCRB and the true CRB in general. Additionally, the computation of the true CRB for general CPM signals has not been accomplished yet. Using the method of Section 2.5.1, we have computed the true CRB and compared it with the other two CRB bounds for binary CPM with 1REC and 1RC frequency pulses when the sequence length is 32. It can be seen in Figure 2.8 that the UCRB is higher than the MCRB regardless of the SNR as expected in (2.62). On the other hand, the true CRB is higher than both MCRB and UCRB at low SNRs while it converges to the MCRB at high SNRs. This convergence is mathematically shown in [48, Eq. 14] and is called the *asymptotic* CRB (ACRB), which is the true CRB for high E_s/N_0. The counter-intuitive behavior of the UCRB at high SNR, i.e., UCRB>CRB, is caused by different types of underlying estimators. The true CRB assumes an estimator which is unbiased on *average* with respect to $\boldsymbol{\alpha}$ while the UCRB applies to estimators that are unbiased for all $\boldsymbol{\alpha}$. Two examples of similar situations are presented in [54] in which the nuisance parameters are continuous random scalars. Furthermore, Figure 2.8 highlights the importance of DA estimation at low SNRs since UCRB can also be interpreted as an *averaged* lower bound for DA estimators.

As a final note, it is instructive to discuss the structure and spectral properties of the proposed sequence to provide some insights into its optimal performance. The main property of the optimum sequence is that the transmitted symbols are constant for a long duration which forces the CPM phase to increase (decrease) with a relatively constant rate in that interval. This results in two dominant frequency components at $\pm h(M-1)/2T_s$ Hz. In a sense, this is similar to the "1100" repeating pattern for the full-response MILSATCOM scheme

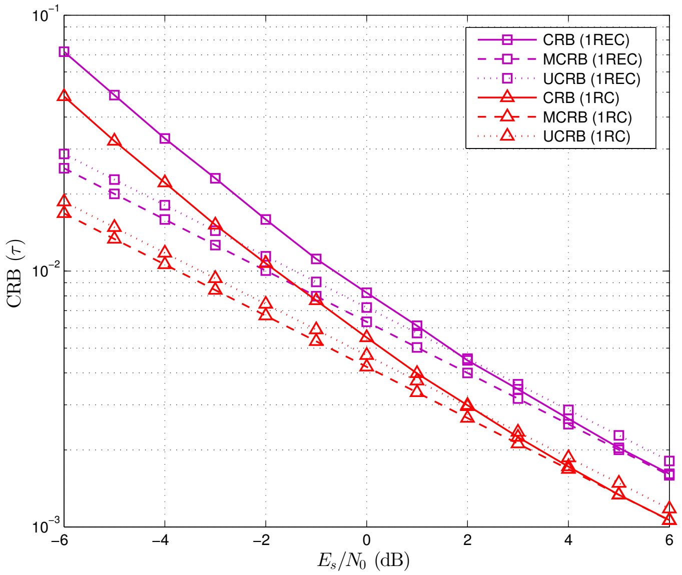


Figure 2.8. The comparison between different CRB bounds for symbol timing estimation of binary full-response CPM schemes. These bounds correspond to a random data sequence with a length of 32.

presented in [11] because it has a period of 4 resulting into $\pm 1/4T_s$ frequencies. However, the MILSATCOM sequence generates rather strong harmonic frequencies as well while the optimum training sequence does not produce such harmonics because it only has two transition points in terms of the direction that the signal phase is changing. Hence, the proposed sequence prevents leakage to the adjacent bands. As another example, the results in [12] suggest that the same constellation point should be repeated as a training sequence for OQPSK with a half-sine pulse shape. We note that this particular scheme is basically binary 1REC CPM (MSK) where their proposed sequence is the alternating sequence in terms of the CPM symbols. This sequence differs from our proposed sequence because the OQPSK representation causes the signal to not have a constant envelope at the very beginning and end. Our computations reveal that the alternating sequence is still a good choice and its CRB is only 1% higher than our sequence for this specific scheme. However, unlike our proposed sequence, the alternating sequence is not a good choice for partial response CPM. For instance, the alternating sequence results in a CRB seven times higher than our optimum sequence in the case of binary 3REC CPM signal.

2.7 Conclusions

We presented the derivation of the CRB for joint estimation of frequency offset, carrier phase, and symbol timing for a known data sequence in CPM signals. We explored the CRB expressions to find the data sequence for which the CRB is minimized. Our analysis in deriving the optimum training sequence involved certain assumptions and approximations, which were later validated via a computer search. Furthermore, we proved that the effect of these approximations are reduced for longer sequences, and hence, it is asymptotically optimum in a strict sense. We were able to show that the optimum training sequence is the same for all three estimation parameters for CPM signals. Additionally, the performance of the optimum training sequence was compared against a random sequence by providing a closed-form expression for the symbol timing UCRB. It was shown that the optimum training sequence brings about significant estimation gains when one is dealing with short training sequences, low SNR operation, partial response or non-binary CPM schemes.

Page left intentionally blank.

Chapter 3

Timing, Carrier, and Frame Synchronization of Burst-Mode CPM

3.1 Key Points of the Chapter

In this chapter, we propose a complete synchronization algorithm for CPM signals in burst-mode transmission over AWGN channels. The timing and carrier recovery are performed through a DA maximum likelihood algorithm, which jointly estimates symbol timing, carrier phase, and frequency offsets based on the optimized synchronization preamble found in Chapter 2. Our algorithm estimates the frequency offset via a one dimensional grid search, after which symbol timing and carrier phase are computed via simple closed-form expressions. The MSE of the algorithm's estimates reveals that it performs very close to the theoretical CRB for various CPMs at SNRs as low as 0 dB. Furthermore, we present a frame synchronization algorithm that detects the arrival of bursts and estimates the start-of-signal. We simulate the performance of the frame synchronization algorithm along with the timing and carrier recovery algorithm. The bit error rate results demonstrate near ideal synchronization performance for low SNRs and short preambles.

3.2 Introduction

A major source of receiver complexity is the synchronization task, especially in burst-mode transmissions where the warm-up or acquisition time must be kept as small as possible. This task has become even more challenging due to the introduction of powerful error correction codes such as LDPC codes, which require accurate synchronization at low SNRs in order to achieve the full coding gain. Feedforward synchronization is a common approach in this type of application since it requires a shorter acquisition time compared to closed-loop methods [1]. Moreover, a known synchronization preamble is usually appended to the beginning of each burst, which assists the synchronization via DA algorithms.

The majority of works on synchronization of CPM in burst-mode transmissions have addressed MSK-type modulations using NDA algorithms, e.g. [25, 26, 27]. In addition to their limited application, these methods do not perform as well as DA algorithms in low SNRs. Huang *et al.* [55] have proposed a feedforward DA joint symbol timing and frequency offset estimation algorithm for GMSK signals. The performance of this *ad hoc* method relies on the amount of frequency offset and symbol timing error. A few DA synchronization algorithms have been presented in the literature for general CPM signals in different environments [18, 19, 20, 56]. Huber and Liu [18] proposed an ML joint timing and phase synchronization algorithm for AWGN channels. In a related work [19], the Walsh transform is used in order to derive the synchronization algorithm. Both of these algorithms assume the timing offset is much smaller than the symbol duration in order to function properly. This limits their application in burst-mode feedforward receivers as the timing offset in practice can have any arbitrary value. Another DA joint phase and timing estimation algorithm is proposed in [20], which is based on the MMSE and Kalman filter criteria. Despite its robustness in time-variant channels and short preambles, this method is implemented in a closed-loop manner that requires multiple initialization steps. Moreover, its MSE is shown to be significantly larger than the CRB even at high SNRs. Another DA algorithm is proposed in [56] for space-time coded CPM over Rayleigh channels, which only tackles the symbol timing estimation. One important issue with all the aforementioned DA algorithms is that the carrier frequency offset has not been taken into account. Blind frequency estimators such as [57, 58] can be employed prior to symbol timing and phase estimations. However, the accuracy of these frequency estimators is away from the CRB [57] especially in low to moderate SNRs [58]. Residual frequency offsets result in poor timing and phase estimates as well as poor signal demodulation.

Another challenge in synchronization of burst-mode signals is estimation of the burst start point, i.e. start-of-signal (SoS). This task, which is referred to as frame synchronization, is crucial in DA algorithms where the boundaries of the known preamble have to be identified. Several sophisticated frame synchronization algorithms [32, 36, 59] have been proposed for PSK signals in AWGN where a frequency offset is present. The performance of the algorithm in [36] depends on the amount of frequency offset, which has to be much smaller than the symbol rate. Choi and Lee [32] have assumed continuous transmissions, where the preamble is surrounded by random data. Although burst-mode transmission is introduced in [59], the authors have assumed there is no guard interval between bursts and the preamble is preceded by random data (similar to the continuous mode). Moreover, it assumes the tentative location of the preamble is known within an uncertainty window. Such a knowledge might not be always available, particularly when the receiver is just powered on.

In this chapter, we present a feedforward DA ML algorithm for joint estimation of frequency offset, symbol timing, and carrier phase in burst-mode CPM signals. The proposed approach takes advantage of the optimized preamble of Chapter 2, which jointly minimizes the CRBs for all three synchronization parameters. We show that the proposed algorithm is capable of performing quite close to the CRB for various CPMs and SNRs. Although we consider an AWGN channel, the results can be applied to time-varying channels too since practical wireless channels can be assumed to be static during the preamble period. In such environments, the estimation results should be used in conjunction with tracking algorithms such as [42].

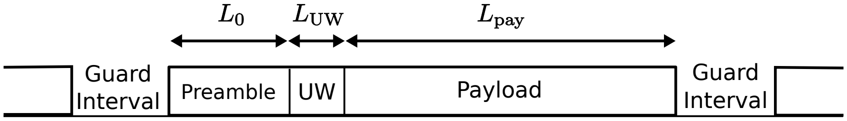


Figure 3.1. The Burst-Mode Transmission Model.

Additionally, we present a frame synchronization algorithm that detects the arrival of bursts and estimates the SoS within ideally one sample time. We discuss how our approach extends the frame synchronization algorithms in [32, 36] to our problem, i.e. CPM signals and burst-mode transmissions. We note that the order in which these two problems are addressed in this chapter is the reverse of their implementation in practice where frame synchronization must be applied prior to timing and carrier recovery.

The remainder of this chapter is organized as follows. Section 3.3 introduces our burst-mode transmission model. In Section 3.4, the joint ML timing and carrier estimation is proposed. Section 3.5 describes the frame synchronization algorithm. Simulation results of the synchronization algorithm are reported in Section 3.6, and Section 3.7 concludes the chapter.

3.3 Burst-Mode Transmission Model

In our model, we consider transmission of disjoint packets of data, i.e. bursts. The transmitter is assumed to be turned on at an unknown time in order to transmit a single burst after which it is turned off again. Each burst has a known duration and structure at the receiver, which is depicted in Figure 3.1 and consists of three parts. The first part is the synchronization preamble or training sequence. It consists of L_0 known and optimized data symbols, which are used to estimate synchronization parameters. Although the preamble can be used for channel estimation too, we only focus on the synchronization task. The next section in the burst is denoted as the unique word (UW), which is utilized to identify the bursts and determine the location of data symbols within a burst. It is assumed to be a pseudo-random sequence of L_{UW} symbols. The last part is the *payload*, which carries L_{pay} information symbols.

By applying the aforementioned model to our CPM notation, the phase of the signal during transmission of each burst is represented as

$$\phi(t;\boldsymbol{\alpha}) = 2\pi h \sum_{i=0}^{L_b-1} \alpha_i q(t - iT_s) \tag{3.1}$$

where α_i is the sequence of M-ary data symbols selected from the set of $\{\pm1, \pm3, \ldots, \pm(M-1)\}$. L_b is the total number of such symbols in a burst, that is $L_b = L_0 + L_{\text{uw}} + L_{\text{pay}}$.

Let us recall our synchronization problem. The complex baseband representation of the received signal is

$$r(t) = e^{j(2\pi f_d t+\theta)} s(t - \tau) + w(t) \tag{3.2}$$

where θ is the unknown carrier phase, f_d is the frequency offset, τ is the timing offset, and $w(t)$ is complex baseband AWGN with zero mean and power spectral density N_0. We denote the transmitted data symbols during the preamble by $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_{L_0-1}]$. Our goal is to determine the synchronization parameters, i.e. $\mathbf{u} = [f_d, \theta, \tau]^T$, by observing the preamble portion of the burst, which corresponds to $\boldsymbol{\alpha}$. Here, it is assumed that $\mathbf{u}$ is a vector of unknown but deterministic parameters which are to be *jointly* estimated at the receiver. Note that $\boldsymbol{\alpha}$ is implicit in the definition of $s(t)$.

Since data arrives in bursts at the receiver, τ can assume any value. However, a DA estimator requires the approximate knowledge of τ in order to perform the estimation algorithm on the received preamble. Therefore, we decompose τ into two parts based on

$$\tau = \mu T_s + \varepsilon T_s \tag{3.3}$$

where $\mu \geq 0$ is an integer that represents the *integer delay* and $-0.5 < \varepsilon < 0.5$ represents the *fractional delay.* In this work, we address these two components separately. First we assume μ is known and the goal is to estimate ε, f_d and θ. Later in Section 3.5, we consider estimation of μ, i.e. the SoS location, regardless of f_d and θ values.

The last item we need to specify is the synchronization preamble. In Chapter 2, we introduced the optimum training sequence for joint estimation of $\mathbf{u}$ based on the CRB criterion. This

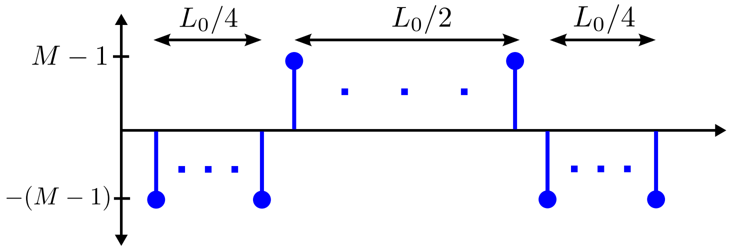


Figure 3.2. The optimum synchronization preamble (training sequence) for M-ary CPM signals containing L_0 symbols.

sequence, which is depicted in Figure 3.2, minimizes the CRBs for f_d, θ and ε simultaneously. It also has a similar pattern for the entire CPM family. We emphasize that the optimality of the above sequence is subject to certain approximations in general, which were discussed in Chapter 2. In spite of this, we refer to this particular sequence as the optimum sequence so that its main property is highlighted. We exploit the structure of the preamble in order to facilitate the algorithm design process and then to reduce its complexity. Note that there is a slight difference in transition points of the preamble compared to the sequence of Chapter 2, i.e. Figure 2.2. In this Chapter, we assume the same preamble regardless of the phase response length for the sake of clarity. Nevertheless, the results can easily be applied to partial-response CPMs. Moreover, we will consider partial-response examples in the rest of our discussion.

3.4 Maximum Likelihood Timing and Carrier Synchronization

3.4.1 Derivation of the Algorithm

Reliable detection of CPM signals depends on accurate timing and carrier synchronization, which requires knowledge of f_d, θ and τ. These parameters can be estimated via various techniques. In this work, we apply joint ML estimation in which $\boldsymbol{\alpha}$ is known to the receiver. The likelihood function for the estimation of a set of parameters from a waveform in AWGN is given in [3]. It can be easily shown that in our problem, i.e. when the signal is complex and constant envelope, the joint LLF for the synchronization parameters is expressed within a constant factor of

$$\Lambda[r(t);\tilde{f}_d,\tilde{\theta},\tilde{\varepsilon}] = \mathrm{Re}\left[\int_{\tilde{\varepsilon}T_s}^{T_0+\tilde{\varepsilon}T_s} e^{-j(2\pi\tilde{f}_d t+\tilde{\theta})} r(t)s^*(t-\tilde{\varepsilon}T_s)dt\right] \tag{3.4}$$

where $\tilde{f}_d$, $\tilde{\theta}$ and $\tilde{\varepsilon}$ are hypothetical values for f_d, θ and ε respectively, and $T_0 = L_0T_s$ is the preamble duration. Note that we disregard μ in this section for the sake of clarity. According to the ML criterion, we choose the trial values that maximize (3.4) as the best estimates for the unknown parameters $\mathbf{u}$. We denote the ML estimates as $\hat{\mathbf{u}} = [\hat{f}_d,\hat{\theta},\hat{\tau}]^T$.

In practice, $r(t)$ is sampled N times per symbol. This results in a discrete-time version of the LLF as

$$\Lambda(\mathbf{r};\tilde{\nu},\tilde{\theta},\tilde{\varepsilon}) \approx \mathrm{Re}\left[\sum_{n=0}^{NL_0-1} e^{-j(2\pi n\tilde{\nu}+\tilde{\theta})} r[n]s^*_{\tilde{\varepsilon}}[n]\right] \tag{3.5}$$

where $\nu = f_dN/T_s$, i.e. the normalized frequency offset with respect to the sampling frequency. $r[n]$ and $s_\varepsilon[n]$ are the sampled versions of $r(t)$ and $s(t-\varepsilon T_s)$ at $t = nT_s/N$ respectively. Note that $\tilde{\varepsilon}$ is assumed to be zero in the integral limits of (3.4) in order to derive (3.5). This is the main contributor to the approximation in the above given that the sampling frequency is large enough to avoid aliasing.

Based on (3.5), the maximization of the LLF requires at least a two dimensional grid search on $(\tilde{\nu},\tilde{\varepsilon})$ in general because both of these parameters are embedded inside the above summation. Therefore, we are interested in a method that decouples ε and ν. We note that the preamble of Figure 3.2, regardless of its underlying CPM, can be divided into three parts, each of which having the same data symbols. This distinct pattern causes the CPM phase to change with a uniform rate of approximately $\pi h(M-1)$ radians per symbol in the same direction for each part. We have illustrated this fact in Figure 3.3 by plotting the unwrapped phase response of three different CPMs when preamble of Figure 3.2 with $L_0 = 16$ is utilized. The first signal phase corresponds to the 1RC frequency pulse with binary data symbols and $h = 1/2$. Additionally, the partial-sresponse 4-ary 2RC CPM is provided in which $h = 1/4$. The GMSK scheme with $BT_s = 0.3$ is also included, which is binary, $L = 4$ and $h = 1/2$. We have compared each case

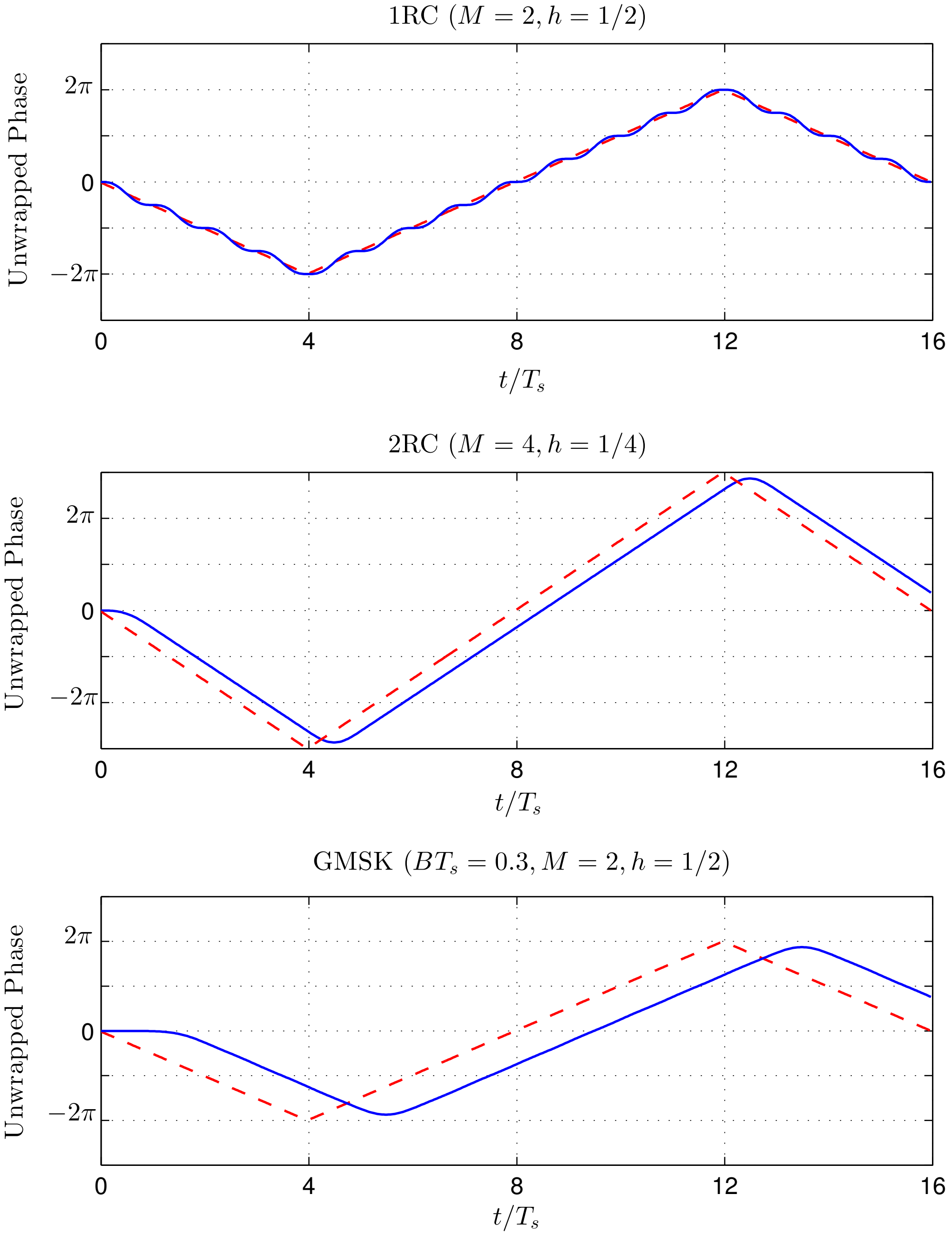


Figure 3.3. The phase response of different CPMs to the optimum training sequence (shown in solid lines). The dashed lines show the response of the same sequence to the 1REC CPM with the same h.

with the phase response of 1REC frequency pulse to the same $\boldsymbol{\alpha}$ and h. It is observed that despite the fundamental differences between their frequency pulses, the overall phase response of all CPM signals are approximately similar. More detailed observations can be made as the following:

1. GMSK and 2RC phase responses follow a straight line within each part similar to the 1REC pulse shape in spite of their bell-shaped pulses. This is due to the overlap of the frequency pulses when the subsequent data symbols are the same, which leads to uniform phase variations.

2. The overall phase response is delayed when partial-response CPMs such as 2RC and GMSK are employed. We denote this lag time by T_l which is equivalent to N_l samples.

3. 1RC CPM shows the largest deviations from the 1REC phase response because its frequency pulse is full-response (non-overlapping) and has the highest peak.

Based on the above discussion, we approximate the phase response of any given CPM signal to the optimum preamble $\boldsymbol{\alpha}^*$ with a delayed version of 1REC CPM to $\boldsymbol{\alpha}^*$ and the same h . In fact, the optimum preamble enables us to accurately apply a *piecewise* linear approximation to the phase of CPM. Therefore, the approximated phase response can be mathematically expressed as

$$\phi(t, \boldsymbol{\alpha}^*) \approx \begin{cases} -(M-1)\pi h \frac{t-T_l}{T_s} & T_l \leq t < \frac{T_0}{4} + T_l \\ (M-1)\pi h \frac{t-T_0/2-T_l}{T_s} & \frac{T_0}{4} + T_l \leq t < \frac{3T_0}{4} + T_l \\ -(M-1)\pi h \frac{t-T_0-T_l}{T_s} & \frac{3T_0}{4} + T_l \leq t < T_0 + T_l \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where T_l is fixed for a given CPM and is known to the receiver. In Appendix B, it is shown that $T_l = \frac{(L-1)}{2} T_s$ for symmetric $g(t)$, which is the case for rectangular, raised-cosine and Gaussian pulse shapes. In the rest of our discussion, we assume the channel observation starts from $t = T_l$, and hence, we ignore T_l. In practice, we can append $\lceil T_l/T_s \rceil$ "$-(M-1)$ symbols" to the end of the preamble for partial-response CPMs in order to avoid unwanted variations at the end of the observation interval, which is now shifted by T_l. Thus, we use (3.6) to express $s_\varepsilon[n]$ during the preamble transmission as

$$s_\varepsilon[n] \approx \begin{cases} \exp[-j(M-1)\pi h(\frac{n}{N}-\varepsilon)] & 0 \le n < \frac{NL_0}{4} \\ \exp[+j(M-1)\pi h(\frac{n}{N}-\frac{L_0}{2}-\varepsilon)] & \frac{NL_0}{4} \le n < \frac{3NL_0}{4} \\ \exp[-j(M-1)\pi h(\frac{n}{N}-L_0-\varepsilon)] & \frac{3NL_0}{4} \le n < NL_0. \end{cases} \tag{3.7}$$

We take advantage of the above approximation in order to simplify the LLF and its maximization algorithm. Using (3.7) in (3.5) results in

$$\begin{aligned} \Lambda^*(\mathbf{r};\tilde{\nu},\tilde{\theta},\tilde{\varepsilon}) \approx \mathrm{Re}\Big\{ e^{-j\tilde{\theta}} \Big[& \sum_{n=0}^{NL_0/4-1} e^{-j2\pi\nu n} r[n] e^{j(M-1)\pi h(n/N-\varepsilon)} \\ & + \sum_{n=NL_0/4}^{3NL_0/4-1} e^{-j2\pi\nu n} r[n] e^{-j(M-1)\pi h(n/N-L_0/2-\varepsilon)} \\ & + \sum_{n=3NL_0/4}^{NL_0-1} e^{-j2\pi\nu n} r[n] e^{j(M-1)\pi h(n/N-L_0-\varepsilon)} \Big] \Big\}, \end{aligned} \tag{3.8}$$

where $\Lambda^*(\cdot)$ represents the joint LLF given $\boldsymbol{\alpha}^*$. It is evident from (3.8) that the symbol timing is now decoupled from the frequency offset and can be moved outside the summations of the LLF. Hence, (3.8) can be simplified as

$$\Lambda^*(\mathbf{r};\tilde{\nu},\tilde{\theta},\tilde{\varepsilon}) \approx \mathrm{Re}\Big\{ e^{-j\tilde{\theta}} \Big[e^{-j(M-1)\pi h\tilde{\varepsilon}} \lambda_1(\tilde{\nu}) + e^{j(M-1)\pi h\tilde{\varepsilon}} \lambda_2(\tilde{\nu}) \Big] \Big\}, \tag{3.9}$$

where

$$\lambda_1(\tilde{\nu}) = \sum_{n=0}^{NL_0/4-1} e^{-j2\pi\tilde{\nu}n} r[n] e^{j(M-1)\pi hn/N} + e^{-j(M-1)\pi hL_0} \sum_{n=3NL_0/4}^{NL_0-1} e^{-j2\pi\tilde{\nu}n} r[n] e^{j(M-1)\pi hn/N}, \tag{3.10}$$

and

$$\lambda_2(\tilde{\nu}) = e^{j(M-1)\pi h L_0/2} \sum_{n=NL_0/4}^{3NL_0/4-1} e^{-j2\pi\tilde{\nu}n} r[n] e^{-j(M-1)\pi hn/N}. \tag{3.11}$$

As the estimation parameters are now decoupled, the maximization of the LLF becomes straightforward. Let us proceed by denoting the term in (3.9) which corresponds to symbol timing and frequency offset as

$$\Gamma(\tilde{\nu}, \tilde{\varepsilon}) = e^{-j(M-1)\pi h\tilde{\varepsilon}} \lambda_1(\tilde{\nu}) + e^{j(M-1)\pi h\tilde{\varepsilon}} \lambda_2(\tilde{\nu}). \tag{3.12}$$

It is easily seen that for any value of $(\tilde{\nu}, \tilde{\varepsilon})$, $\Lambda^*(\cdot)$ is maximized by choosing $\tilde{\theta}$ such that it rotates $\Gamma(\tilde{\nu}, \tilde{\varepsilon})$ towards the real axis, i.e.,

$$\tilde{\theta} = \arg\{\Gamma(\tilde{\nu}, \tilde{\varepsilon})\}. \tag{3.13}$$

which reduces the LLF to $|\Gamma(\tilde{\nu}, \tilde{\varepsilon})|$. Thus, the ML estimates of $\tilde{\nu}$ and $\tilde{\varepsilon}$ are found by maximizing

$$|\Gamma(\tilde{\nu}, \tilde{\varepsilon})|^2 = |\lambda_1(\tilde{\nu})|^2 + |\lambda_2(\tilde{\nu})|^2 + 2\mathrm{Re}[e^{-j2(M-1)\pi h\tilde{\varepsilon}} \lambda_1(\tilde{\nu}) \lambda_2^*(\tilde{\nu})] \tag{3.14}$$

with respect to $(\tilde{\nu}, \tilde{\varepsilon})$. The first two terms on the right-hand side of (3.14) do not depend on $\tilde{\varepsilon}$. Using a similar argument as $\tilde{\theta}$, the third term is maximized by selecting $\tilde{\varepsilon}$ according to

$$\tilde{\varepsilon} = \frac{\arg\{\lambda_1(\tilde{\nu})\lambda_2^*(\tilde{\nu})\}}{2(M-1)\pi h} \tag{3.15}$$

so that the term inside the real part operator in (3.14) becomes purely real and equal to $|\lambda_1(\tilde{\nu})\lambda_2^*(\tilde{\nu})|$. Therefore, the maximization of the LLF is now a one dimensional problem that results in the ML estimate of ν. This can be expressed mathematically in the form of

$$\hat{\nu} = \underset{\tilde{\nu}}{\mathrm{argmax}} \left\{ X(\tilde{\nu}) = |\lambda_1(\tilde{\nu})| + |\lambda_2(\tilde{\nu})| \right\}, \tag{3.16}$$

which in turn leads to the ML estimates of the normalized symbol timing and phase offset via

$$\hat{\varepsilon} = \frac{\arg\{\lambda_1(\hat{\nu})\lambda_2^*(\hat{\nu})\}}{2(M-1)\pi h}, \tag{3.17}$$

and

$$\hat{\theta} = \arg\left\{e^{-j(M-1)\pi h\hat{\varepsilon}}\lambda_1(\hat{\nu}) + e^{j(M-1)\pi h\hat{\varepsilon}}\lambda_2(\hat{\nu})\right\}, \tag{3.18}$$

respectively.

3.4.2 Implementation of the Frequency Offset Estimator

In the previous section, we derived simple closed-form expressions for estimation of phase and symbol timing. However, the frequency offset estimation requires computing the maximum of a one-dimensional function as defined in (3.16). $\lambda_1(\nu)$ and $\lambda_2(\nu)$ have the form of Fourier transforms of $r(t)$ and should be expected to have fluctuations due to the presence of noise, which results in several local maxima. Thus, a grid search is inevitable in order to find the correct frequency offset with confidence.

According to (3.10) and (3.11), computations of $\lambda_1(\nu)$ and $\lambda_2(\nu)$ require a different number of summations with different limits. In order to make both of them consistent, we define two new signals, i.e. $r_1[n]$ and $r_2[n]$, such that

$$r_1[n] = \begin{cases} r[n] & 0 \leq n < NL_0/4 \\ \exp[-j(M-1)\pi h L_0]r[n] & 3NL_0/4 \leq n < NL_0 \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

and

$$r_2[n] = \begin{cases} \exp[j(M-1)\pi h L_0/2]r[n] & NL_0/4 \leq n < 3NL_0/4 \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

The above modifications to $r[n]$ leads to similar forms for $\lambda_1(\nu)$ and $\lambda_2(\nu)$, where each one requires computation of one summation with NL_0 terms, i.e.,

$$\lambda_1(\tilde{\nu}) = \sum_{n=0}^{NL_0-1} r_1[n] e^{j(M-1)\pi hn/N} e^{-j2\pi n\tilde{\nu}} \tag{3.21}$$

and

$$\lambda_2(\tilde{\nu}) = \sum_{n=0}^{NL_0-1} r_2[n] e^{-j(M-1)\pi hn/N} e^{-j2\pi n\tilde{\nu}.} \tag{3.22}$$

The computations of (3.21) and (3.22) for different $\tilde{\nu}$ values resemble the discrete Fourier transform (DFT) operation, where $\tilde{\nu}$ is replaced by trial discrete frequencies. These operations can be performed efficiently using the fast Fourier transform (FFT). The FFT size will be equal to the summation length assuming NL_0 is a power of two. This process results in trial values for $\lambda_1(\tilde{\nu})$ and $\lambda_2(\tilde{\nu})$ such that $\tilde{\nu} \in [0, 1/NL_0, \ldots, (NL_0-1)/NL_0]$, which are then inserted in (3.16) in order to find $\hat{\nu}$. Therefore, the frequency offset estimate requires two FFTs of the same size.

The frequency estimation performance is limited by the resolution of the FFT operations, i.e. the distance between the discrete frequency components. A low resolution estimate may cause a ripple effect on the estimation performance of other parameters. In order to increase the accuracy of the frequency estimate, two approaches are considered. The first approach is to zero pad the FFT operands in (3.21) and (3.22) such that both FFTs have a size of $N_f = K_f NL_0$ where K_f is a power of two. This procedure results in a frequency resolution of $1/K_f L_0$ with respect to the symbol rate. The second approach is to employ an interpolator in order to estimate the true maximum of (3.16) between the discrete frequency values. In [60], it was shown that the Gaussian interpolator is superior to a parabolic one in terms of improving FFT resolution. The only added complexity is an extra look-up table for computation of the logarithm function. The Gaussian interpolation can be expressed as

$$\hat{\nu} = \hat{\nu}_0 + \frac{1}{2K_f NL_0} \frac{\log X(\hat{\nu}_{-1}) - \log X(\hat{\nu}_1)}{\log X(\hat{\nu}_{-1}) + \log X(\hat{\nu}_1) - 2\log X(\hat{\nu}_0)}, \tag{3.23}$$

where $\hat{\nu}_0$ represents the maximizing frequency resulting from (3.16). $\hat{\nu}_{-1}$ and $\hat{\nu}_1$ denote the

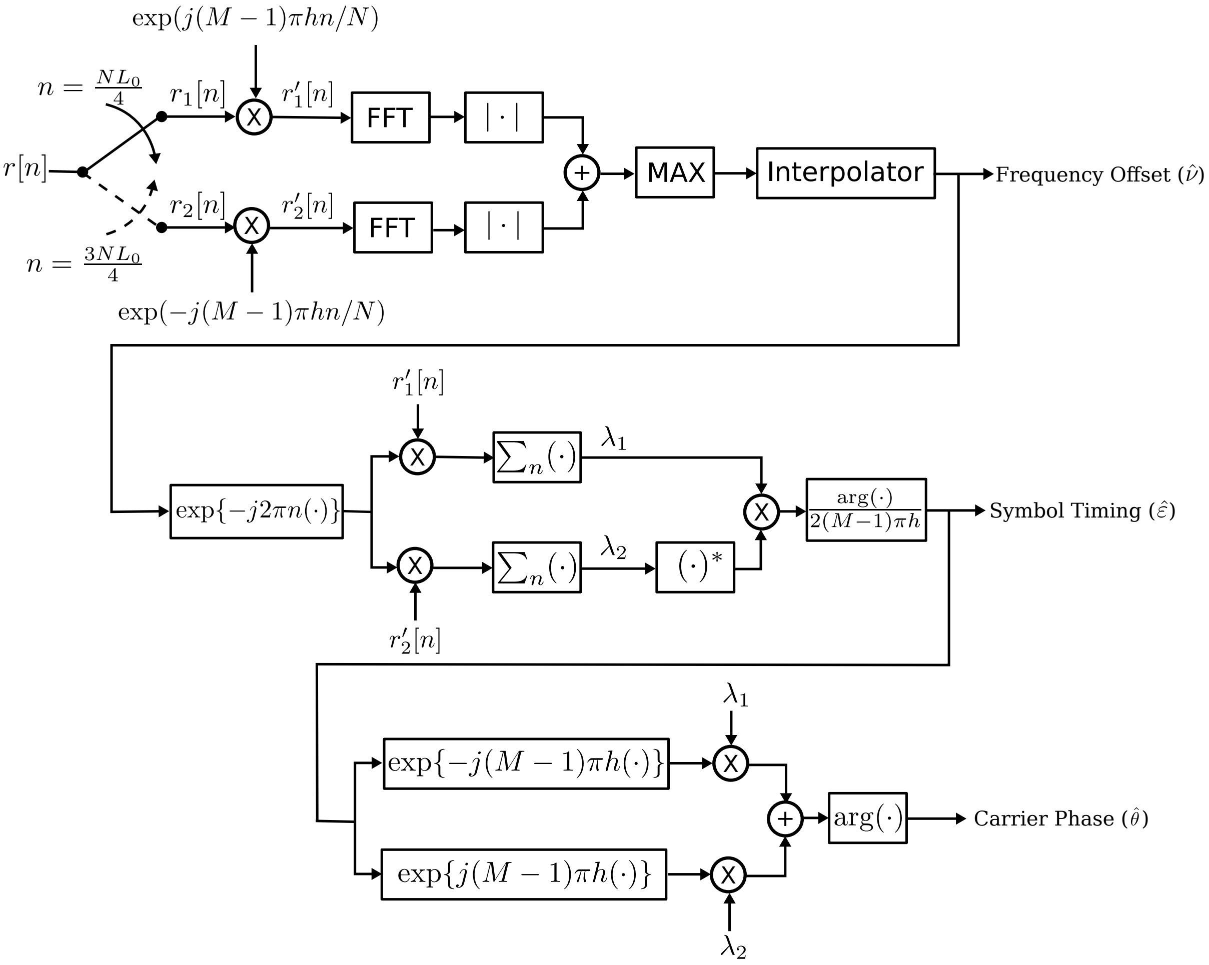


Figure 3.4. Block diagram of the feedforward joint frequency offset, symbol timing and carrier phase estimator.

discrete frequency components immediately before and after $\hat{\nu}_0$ respectively in terms of the FFT operation. The above operation can be regarded as a *fine search* while FFTs perform a *course search* on the frequency offset.

Based on DFT properties, FFT operations are periodic with a period of NL_0. Therefore, values of $1/2 \leq \hat{\nu} < 1$ represent negative frequency offsets, and hence, $\hat{\nu}$ is estimated over $[-1/2, 1/2)$. This limits the frequency estimation range to

$$-\frac{N}{2T_s} \leq \hat{f}_d < \frac{N}{2T_s}, \tag{3.24}$$

which can be increased by increasing the sampling frequency. Therefore, the proposed algorithm can easily handle applications in which the frequency offset is greater than the symbol rate.

The final design for our feedforward joint frequency offset, symbol timing and carrier phase estimator is illustrated in Figure 3.4. Based on (3.19) and (3.20), $r_1[n]$ and $r_2[n]$ should be multiplied by $\exp[-j(M-1)\pi h L_0]$ and $\exp[j(M-1)\pi h L_0/2]$ respectively. However, we have not shown this in Figure 3.4 for the sake of simplicity, and because the aforementioned factors are basically equal to one in our examples.

3.5 Frame Synchronization

So far, we have assumed the carrier and timing synchronization algorithm have the knowledge of the SoS within $\pm T_s/2$, which has to be carried out by the frame synchronization algorithm. In this work, we decompose the frame synchronization into two tasks: *SoS detection* and *SoS Estimation*. The SoS detector determines the arrival of a new burst such that the preamble is located within an *observation* or uncertainty window. The SoS estimation algorithm then tries to find the exact location of the SoS within that window. Using a reverse approach, we initially derive the SoS estimation algorithm. Based on its results, we propose a simple SoS detection algorithm.

3.5.1 SoS Estimation Algorithm

The framework for the SoS estimation algorithm is depicted in Figure 3.5 where an observation window of N_w samples is considered. The first δ samples contains only WGN, which correspond to the guard interval prior to the beginning of signal transmission. It is followed by N_p samples of the preamble. Finally, there are $N_w - \delta - N_p$ samples, which are assumed to be generated from a random CPM signal, and are associated with the UW and/or payload portion of the burst. The SoS estimation algorithm attempts to find the best estimate of δ according to the above observation window.

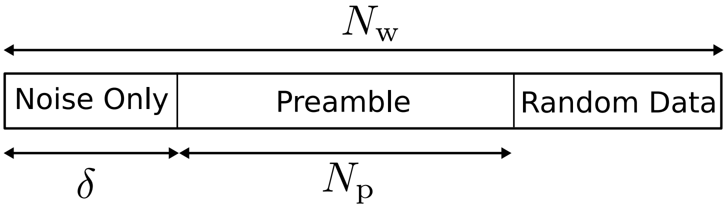


Figure 3.5. The observation window for the SoS estimation algorithm.

The received and sampled signal within the observation window can be expressed as

$$r[n] = \begin{cases} w[n] & 0 \leq n < \delta \\ e^{j(2\pi\nu n+\theta)}s[n-\delta] + w[n] & \delta \leq n < N_w, \end{cases} \tag{3.25}$$

where $w[n]$ is complex white Gaussian random sequence with a variance of $\sigma^2 = N(E_s/N_0)^{-1}$. Additionally, we have assumed $T_s = 1$ and $|s[n]| = 1$. It should be noted that θ in (3.25) is different from its value in (3.2) due to the frequency offset and different reference points. Finally, we denote the values of $r[n]$ within the observation window by $\mathbf{r}$.

Based on the ML rules, the best estimate of δ is the value which maximizes the likelihood function $p(\mathbf{r};\delta)$. However, let us first consider the likelihood function as a function of all unknown parameters, i.e.,

$$p(\mathbf{r};\delta,\nu,\theta,\boldsymbol{\alpha}_d) = \frac{1}{(\pi\sigma^2)^{N_w}} \exp\left(-\frac{1}{\sigma^2}\sum_{n=0}^{\delta-1}|r[n]|^2\right) \exp\left(-\frac{1}{\sigma^2}\sum_{n=\delta}^{N_w-1}|r[n]-s[n-\delta]e^{j(2\pi\nu n+\theta)}|^2\right), \tag{3.26}$$

where $\boldsymbol{\alpha}_d$ represents the random data sequence in the non-preamble portion of $s[n]$. If we omit constant factors in the likelihood function, it becomes

$$p(\mathbf{r};\delta,\nu,\theta,\boldsymbol{\alpha}_d) = \exp\left(\frac{\delta - N_w}{\sigma^2}\right) \exp\left(\frac{2}{\sigma^2}\sum_{n=\delta}^{N_w-1}\mathrm{Re}\left\{r^*[n]s[n-\delta]e^{j(2\pi\nu n+\theta)}\right\}\right). \tag{3.27}$$

In order to compute $p(\mathbf{r};\delta)$ from (3.27), we must either estimate or average out the *nuisance* parameters, i.e. ν, θ and $\boldsymbol{\alpha}_d$, which is not trivial due to the form of the above function. Instead, we initially approximate the exponential function with its second degree Taylor's series in the neighborhood of zero, i.e.,

$$
\begin{aligned}
p(\mathbf{r};\delta,\nu,\theta,\boldsymbol{\alpha}_d) \approx C(\delta)\Bigg(1 &+ \frac{2}{\sigma^2}\sum_{n=\delta}^{N_w-1}\mathrm{Re}\left\{r^*[n]s[n-\delta]e^{j(2\pi\nu n+\theta)}\right\} \\
&+ \frac{1}{\sigma^4}\sum_{n=\delta}^{N_w-1}\sum_{m=\delta}^{N_w-1}\mathrm{Re}\left\{r^*[n]r^*[m]s[n-\delta]s[m-\delta]e^{j(2\pi\nu(m+n)+2\theta)}\right\} \\
&+ \frac{1}{\sigma^4}\sum_{n=\delta}^{N_w-1}\sum_{m=\delta}^{N_w-1}\mathrm{Re}\left\{r^*[n]r[m]s[n-\delta]s^*[m-\delta]e^{j(2\pi\nu(n-m))}\right\}\Bigg),
\end{aligned}
\tag{3.28}
$$

where $C(\delta)$ represents $\exp(\frac{\delta-N_w}{\sigma^2})$ in (3.27), which is not a function of the nuisance parameters. However, we avoid using it in its original form because it can be very small and adversely affect the approximated likelihood function. Nevertheless, we will propose an approximation for $C(\delta)$ once the final form of the likelihood function becomes available.

Assuming θ is uniformly distributed over $[-\pi,\pi]$, it can be eliminated from the likelihood function by averaging (3.28) over θ, i.e.,

$$
\begin{aligned}
p(\mathbf{r};\delta,\nu,\boldsymbol{\alpha}_d) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} p(\mathbf{r};\delta,\nu,\theta,\boldsymbol{\alpha}_d)\,d\theta \\
&\approx C(\delta)\frac{1}{\sigma^4}\sum_{n=\delta}^{N_w-1}\sum_{m=\delta}^{N_w-1}\mathrm{Re}\left\{r^*[n]r[m]s[n-\delta]s^*[m-\delta]e^{j(2\pi\nu(n-m))}\right\}.
\end{aligned}
\tag{3.29}
$$

Note that we have neglected 1 in (3.28) because it is much smaller than the forth term especially when noise variance is small. We also omit $1/\sigma^4$ from the above as it is a constant factor. If we denote $d = m - n$ in (3.29), it can be rearranged as

$$
\begin{aligned}
p(\mathbf{r};\delta,\nu,\boldsymbol{\alpha}_d) \approx C(\delta)\Bigg(&\sum_{n=\delta}^{N_w-1}|r[n]|^2 \\
&+2\sum_{d=1}^{N_w-\delta-1}\mathrm{Re}\left\{e^{-j2\pi d\nu}\sum_{n=\delta}^{N_w-d-1}r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta]\right\}\Bigg),
\end{aligned}
\tag{3.30}
$$

which allows us to investigate signal correlation due to the presence of random $\boldsymbol{\alpha}_d$. The computation of $E_{\boldsymbol{\alpha}_d}\{p(\mathbf{r};\delta,\nu,\boldsymbol{\alpha}_d)\}$ leads us to compute $E_{\boldsymbol{\alpha}_d}\{s[n-\delta]s^*[n+d-\delta]\}$, which, in our problem, is

$$E_{\boldsymbol{\alpha}_d}\{s[n-\delta]s^*[n+d-\delta]\} = \begin{cases} s[n-\delta]s^*[n+d-\delta] & \delta \leq n < N_p+\delta-d \\ 0 & N_p+\delta-d \leq n < N_p+\delta \\ R_{ss}(d) & n \geq N_p+\delta, \end{cases} \tag{3.31}$$

where $R_{ss}(d)$ is the autocorrelation function of the CPM signal normalized to the sample duration. $R_{ss}(d)$ can be computed numerically as described in [3, p. 208]. The first case in (3.31) corresponds to the preamble, which has no randomness. The second case is zero because $s[n-\delta]$ is deterministic whereas $s^*[n+d-\delta]$ is generated by the random data and its expected value is zero. Therefore, taking the expected value of (3.30) with respect to $\boldsymbol{\alpha}_d$ results in

$$p(\mathbf{r};\delta,\nu) \approx C(\delta)\Bigg(\sum_{n=\delta}^{N_w-1}|r[n]|^2 + 2\sum_{d=1}^{N_p-1}\mathrm{Re}\Bigg\{e^{-j2\pi d\nu}\Bigg(\sum_{n=\delta}^{N_p+\delta-d-1} r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta] \\ +R_{ss}(d)\sum_{n=N_p+\delta}^{N_w-d-1} r^*[n]r[n+d]\Bigg)\Bigg\}\Bigg). \tag{3.32}$$

In general, CPM autocorrelation function becomes zero for lag times greater than LT_s. Therefore, $R_{ss}(d)$ is zero except for its first few values.

The last step to obtain $p(\mathbf{r};\delta)$ is removing ν from (3.32). It can be verified that averaging (3.32) with respect to ν, uniformly distributed over [-0.5,0.5], completely eliminates the second summation. This indeed results in a poor ML estimate for δ because it ignores the knowledge of the known preamble. A better approach is to estimate ν by maximizing the second summation in the above. However, a closed-form solution seems to be unavailable due to the range of d.

Instead, we can derive different estimates $\hat{\nu}_d$ based on single terms inside the summation via

$$\hat{\nu}_d = \frac{1}{2\pi d} \arg \left\{ \sum_{n=\delta}^{N_p+\delta-d-1} r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta] + R_{ss}(d) \sum_{n=N_p+\delta}^{N_w-d-1} r^*[n]r[n+d] \right\}. \tag{3.33}$$

The above method is the basis for some well-known carrier frequency estimation algorithms, such as [61]. If we use $\hat{\nu}_d$ values and plug them back into (3.32), the likelihood function becomes independent of the frequency offset. Thus,

$$\begin{aligned} p(\mathbf{r};\delta) \approx C(\delta) \Bigg(\sum_{n=\delta}^{N_w-1} |r[n]|^2 \; + \; 2 \sum_{d=1}^{N_p-1} \Bigg| & \sum_{n=\delta}^{N_p+\delta-d-1} r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta] \\ & + R_{ss}(d) \sum_{n=N_p+\delta}^{N_w-d-1} r^*[n]r[n+d] \Bigg| \Bigg), \end{aligned} \tag{3.34}$$

which must be maximized with respect to δ in order to derive $\hat{\delta}$.

The computational complexity of (3.34) can be reduced by truncating the summation over d. This results in a sub-optimum, reduced-complexity estimator, i.e.

$$\begin{aligned} \hat{\delta} = \underset{\tilde{\delta}}{\operatorname{argmax}} \Bigg\{ C(\tilde{\delta}) \Bigg(\sum_{n=\tilde{\delta}}^{N_w-1} |r[n]|^2 \; + \; 2 \sum_{d=1}^{D} \Bigg| & \sum_{n=\tilde{\delta}}^{N_p+\tilde{\delta}-d-1} r^*[n]r[n+d]s[n-\tilde{\delta}]s^*[n+d-\tilde{\delta}] \\ & + R_{ss}(d) \sum_{n=N_p+\tilde{\delta}}^{N_w-d-1} r^*[n]r[n+d] \Bigg| \Bigg) \Bigg\}, \end{aligned} \tag{3.35}$$

where $1 \leq D < N_p$ is a design parameter, which allows a trade-off between complexity and performance.

As mentioned earlier, $C(\delta)$ needs to be adjusted based on the final form of the likelihood function. We note that (3.34) is dominated by the summation over d. If we ignore $R_{ss}(d)$ due to its short length, the computations inside the absolute value are performed over a sliding window that covers the hypothetical preamble. If this window is shifted to the left by one sample, one signal-plus-noise sample will be replace by one noise-only sample, which has a smaller energy compared to the former one. However, shifting the window to the right replaces it with a different signal-plus-noise sample. Therefore, we expect $p(\mathbf{r};\delta+1) > p(\mathbf{r};\delta-1)$, if δ is its true value. This makes the likelihood test biased, i.e. $\hat{\delta}$ is more likely to tend towards $\delta+1$ than $\delta-1$. We introduce a simple solution to this issue by proposing

$$C(\delta) \triangleq (N_w - \delta)^q \tag{3.36}$$

where $q \geq 0$ is another design parameter, which has to be chosen according to D. As we will see in the simulation results, $q = 1$ is a good choice for the full-complexity estimator, i.e. $D = N_p - 1$, while it has to be reduced for smaller values of D.

Choi and Lee [32] have presented a ML frame synchronization algorithm through a different path for PSK signals where the preamble is surrounded by random data. Despite similarities to (3.35), our estimator addresses a different scenario in which the preamble is preceded by the noise-only samples so that $C(\delta)$ was introduced. Additionally, the memory of CPM signals is handled via the presence of $R_{ss}(d)$. Finally, it should be mentioned that each of the summations $\sum_n r^*[n]r[n+d]s[n-\tilde{\delta}]s^*[n+d-\tilde{\delta}]$ in (3.35) is referred to as a *double-correlation* in [32].

3.5.2 SoS Detection Algorithm

As the last piece of our synchronization algorithm, we present a simple ML detection algorithm, which is closely related to our previous discussion. Let us assume a receiver which collects vectors of N_p samples using a sliding window. We denote this vector by $\mathbf{r}_p$. Additionally, consider two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. $\mathcal{H}_0$ is the hypothesis where the entire vector of samples in $\mathbf{r}_p$ are noise-only samples, which happens when no burst is received. On the other hand, $\mathcal{H}_1$ is the hypothesis where $\mathbf{r}_p$ is perfectly aligned with the preamble. We can distinguish these two hypotheses by performing a likelihood ratio test (LRT) according to

$$L(\mathbf{r}_p) = \frac{p(\mathbf{r}_p;\mathcal{H}_1)}{p(\mathbf{r}_p,\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma, \tag{3.37}$$

where $p(\mathbf{r}_p;\mathcal{H}_i)$ is the likelihood function under $\mathcal{H}_i$. Based on the above test, $\mathcal{H}_1$ is selected when $L(\mathbf{r}_p)$ is greater than a threshold γ. Otherwise, we select $\mathcal{H}_0$, i.e. no preamble is present.

Obviously, several other hypotheses also occur in between these two in which $\mathbf{r}_p$ contains only a fraction of the preamble, i.e. a *mixed-signal* scenario. However, all those scenarios can be considered as *irrelevant* hypotheses since we have already established a mechanism to estimate the exact location of the SoS. For instance, if we decide $\mathcal{H}_1$ while $\mathbf{r}_p$ covers only a portion of the preamble, it is still considered a successful detection because the SoS estimation algorithm is able to find the exact location of preamble given N_w is large enough, i.e. $N_w \geq 2N_p$. On the other hand, if $\mathcal{H}_0$ is selected under such circumstances, it is not a missed detection since $\mathcal{H}_1$ has not happened yet. Thus, we neglect mixed-signal scenarios in designing our LRT.

The LRT can be easily obtained from (3.34). In fact, $p(\mathbf{r}_p;\mathcal{H}_1)$ becomes equal to $p(\mathbf{r};\delta)$ when $\delta = 0$ and $N_w = N_p$. We also note that we can multiply $p(\mathbf{r}_p;\delta=0)$ by $\exp(-\frac{1}{\sigma^2}\sum_{n=0}^{N_p-1}|r[n]^2|)$ because it is not a function of δ. The latter factor is basically equal to $p(\mathbf{r}_p;\mathcal{H}_0)$. Thus, the LRT can be approximated by

$$L(\mathbf{r}_p) = \frac{p(\mathbf{r}_p;\delta=0)p(\mathbf{r}_p;\mathcal{H}_0)}{p(\mathbf{r}_p;\mathcal{H}_0)} \approx \sum_{n=0}^{N_p-1}|r[n]|^2 + 2\sum_{d=1}^{Np-1}\left|\sum_{n=0}^{Np-d-1} r^*[n]r[n+d]s[n]s^*[n+d]\right| \gtrless \gamma'. \tag{3.38}$$

Similar to (3.35), we propose a reduced-complexity test, i.e.,

$$L_{D'}(\mathbf{r}_p) \triangleq \sum_{d=1}^{D'}\left|\sum_{n=0}^{Np-d-1} r^*[n]r[n+d]s[n]s^*[n+d]\right| \gtrless \gamma_{D'}, \tag{3.39}$$

where $1 \leq D' < N_p$ is a design parameter and $\gamma_{D'}$ represents the test threshold for a given D'.

The threshold $\gamma_{D'}$ can be chosen based on the Neyman-Pearson criterion [62] in which the probability of *false alarm* is fixed. Here, the probability of false alarm is defined as $P_{\mathrm{FA}} = \Pr\{L_{D'}(\mathbf{r}_p) > \gamma_{D'}|\mathcal{H}_0\}$. Once the threshold is chosen, the probability of *missed detection* can be calculated via $P_{\mathrm{MD}} = \Pr\{L_{D'}(\mathbf{r}_p) < \gamma_{D'}|\mathcal{H}_1\}$. The probability of correct detection is $P_{\mathrm{D}} = 1 - P_{\mathrm{MD}}$. Exact closed-form expressions for P_{FA} and P_{D} may not be realized due to the magnitude operators and multiplications in (3.39). For instance, if we denote the output of each double-correlation as a random variable, i.e., $X_d = \sum_{n=0}^{N_p-d-1} r^*[n]r[n+d]s[n]s^*[n+d]$, a simple yet acceptable (for large $N_p - d$) approximation is to consider X_d as a complex Gaussian random variable (RV). This forces $|X_d|$ to become a Rayleigh RV under $\mathcal{H}_0$ and Rician RV under $\mathcal{H}_1$ due to the presence of signal. Thus, $L_{D'}(\mathbf{r}_p)$ can be approximated as sum of Rayleigh or Rician RVs depending on the hypothesis. In [63, 64], approximate CDFs are provided for such RVs. However, our investigations show that the approximation error is considerable because we are interested in regions where P_{FA} and P_{MD} are very low. Therefore, we resort to Monte-Carlo simulations with a large sample size in order to compute these probabilities, $\gamma_{D'}$, and the receiver operating characteristic (ROC).

3.6 Results and Discussion

3.6.1 Timing and Carrier Recovery Performance

In this section, we compute the error variances of frequency offset, carrier phase, and symbol timing for the proposed ML estimation algorithm using simulations. We have considered the three examples of Figure 3.3 along with MSK. In all examples, the optimum preamble with $L_0 = 64$ is deployed. In addition to AWGN, we apply ν, θ and ε that are uniformly distributed over $[-0.5, 0.5]$, $[0, 2\pi]$, and $[-0.5, 0.5]$ respectively.

The effect of the interpolation and FFT size on the estimator performance is demonstrated in Figure 3.6 via simulations. The GMSK ($BT_s = 0.3$) modulation is considered in this set of simulations in which $L_0 = 64$ and $N = 2$. The error variance for normalized frequency offset–with respect to symbol rate–is plotted for three different K_f values of 1, 2 and 4. Additionally, three scenarios of no interpolation, parabolic interpolation and Gaussian interpolation are studied. It can be seen that the error variance does not improve with respect to the SNR regardless of the zero padding factor when there is no interpolation. In fact, as K_f is increased, the variance is improved, however, the error due to the grid search, i.e. the FFT operation, still dominates the additive noise. This suggests that one has to carry out a very large FFT when no interpolation is utilized. On the other hand, the interpolation does not deliver any gains when

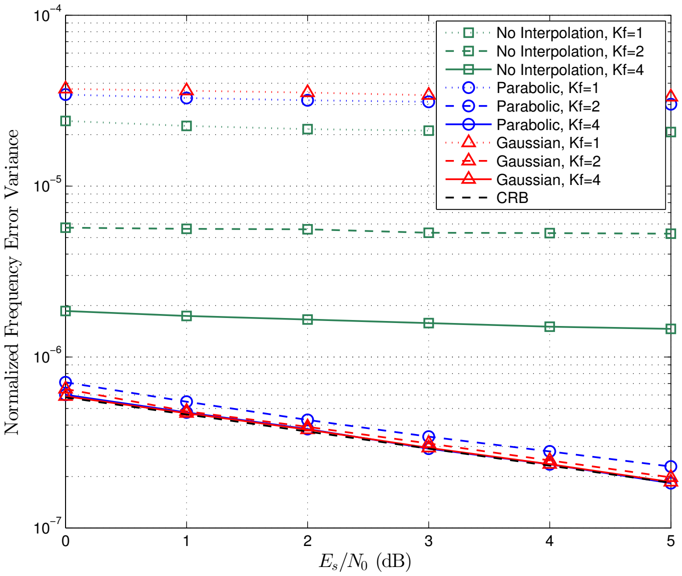
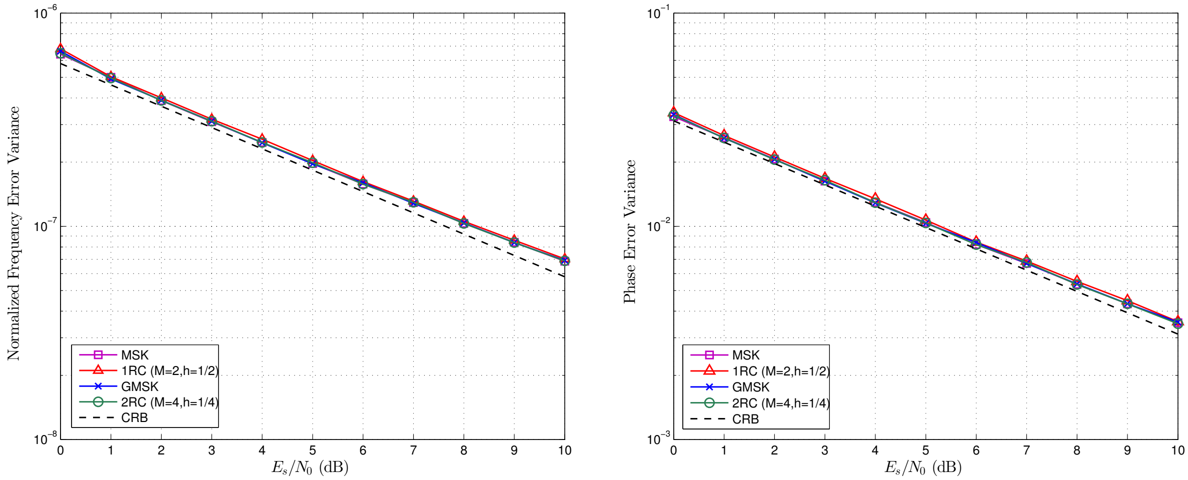


Figure 3.6. The effect of interpolation and zero padding on the frequency estimation for GMSK when $L_0 = 64$.

$K_f = 1$, i.e., there is no zero padding. When $K_f = 2$, both parabolic and Gaussian interpolators show their effect. The improvement is quite significant for the Gaussian interpolator where the error variance almost attains the CRB. Therefore, we have chosen the Gaussian interpolator with $K_f = 2$ for our estimator. It can be observed that both interpolators perform slightly better than the former case when $K_f = 4$, however, the increased FFT overhead prevents us from utilizing them.

The estimation error variances corresponding to the normalized frequency offset and carrier phase are depicted in Figures 3.7 (a) and 3.7 (b) respectively for different CPM schemes. The frequency estimation plots demonstrate that the proposed estimator performs with almost the same accuracy for all the schemes, that is less than 0.5 dB away from the CRB for low to moderate SNRs. As it was shown in Section 2.4, the frequency and phase estimation CRBs for the optimum training sequence are independent of the particular CPM scheme. Hence, only one CRB plot is shown in each Figure. Moreover, it is observed that the 1RC scheme performs slightly worse than the other schemes because it has the largest deviations from the 1REC template (refer to Figure 3.3). For the remaining schemes, the gap between the error variances



(a)

(b)

Figure 3.7. The error variance of frequency offset (a) and carrier phase (b) estimations for different CPM schemes when $L_0 = 64$. The frequency is normalized with respect to the symbol rate.

and the CRB is mostly due to the FFT precision and can be reduced by increasing K_f. This gap becomes more visible at high SNRs because errors due to the thermal noise are smaller than the FFT and interpolation precision.

The normalized timing error variances are plotted in Figure 3.8. It reveals that the proposed estimator reaches the CRB for the majority of the examples. The only exception is again the 1RC scheme as discussed above. For all other CPMs, the ML estimator attains the lower limit of the CRB despite the visible loss in the frequency estimation. This is because the optimum training sequence decouples timing from frequency in terms of the Fisher information matrix, which means that small errors in the frequency estimate do not affect the symbol timing estimate. The optimum training sequence does not decouple frequency and phase, and hence, errors in the frequency estimate leak into the phase estimator, which results in a slight performance degradation that is visible in Figure 3.7 (b).

It should be mentioned that the FFT operations will be replaced by simple correlations when $f_d = 0$. In such applications, (3.21) and (3.22) are computed for $\tilde{\nu} = 0$ without any need to perform the maximization of (3.16) and the interpolation. This leads to a joint symbol

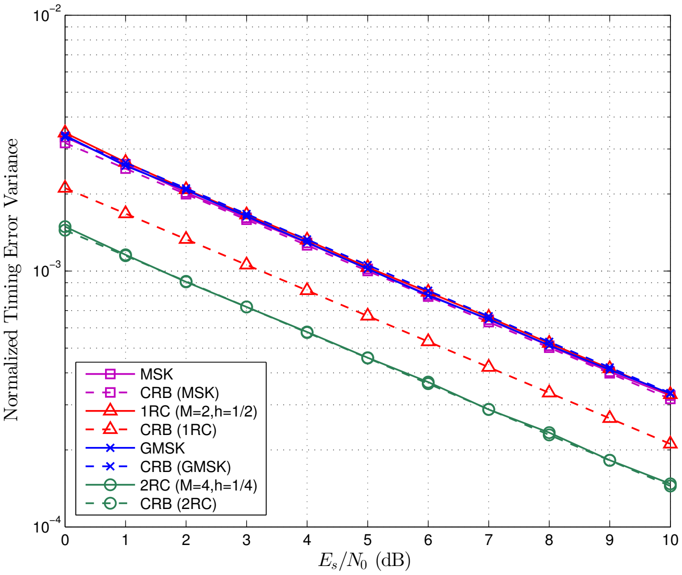


Figure 3.8. The variance of symbol timing estimation for different CPM schemes when $L_0 = 64$. The symbol timing is normalized with respect to the symbol duration.

timing and carrier phase estimator, which is efficient yet less complex compared to other DA works such as [18, 19, 20]. This simplicity is a direct result of unique structure of the optimized preamble.

3.6.2 Frame Synchronization Performance

The performance of the SoS estimation algorithm is characterized by the probability of *false lock*, which is $P_{\text{FL}} = \Pr\{\hat{\delta} \neq \delta\}$. This probability is computed given that the preamble is correctly detected and fully resides within the observation window.

The effect of the $C(\delta)$ as a function of q on P_{FL} are studied in Figure 3.9 using simulations. The GMSK scheme is used where $L_0 = N_p = 64$, $N_w = 96$ and $E_s/N_0 = 1$ dB. Additionally, we have varied q over the range of $[0, 2]$ and computed P_{FL} for several values of D. It is observed that the introduction of $C(\delta)$ reduces P_{FL} given that q is carefully selected. We observe that a value of $q = 1$ is suitable for $D = 63$, while it needs to be decreased for smaller values of D. In fact, $C(\delta)$ becomes less important for small values of D such as $D = 2$ and can simply be

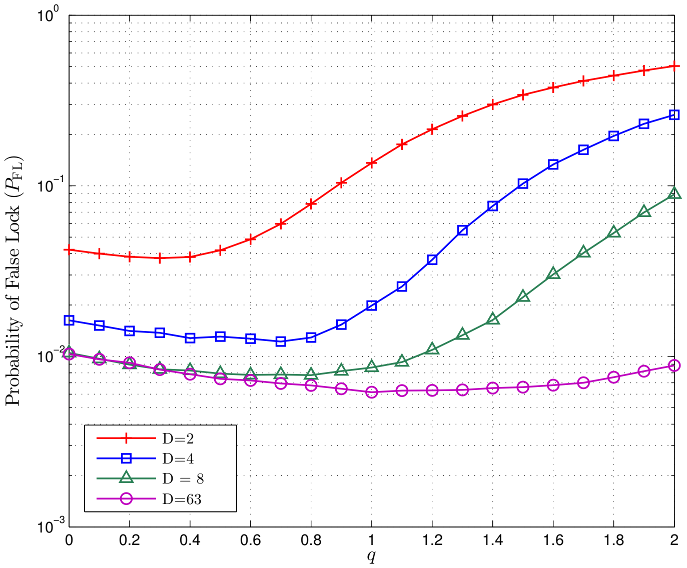


Figure 3.9. The effect of correction term $C(\delta)$ (Equation (3.36)) and its exponent q on P_{FL}. GMSK signaling is used when $N_p = 64$ and $E_s/N_0 = 1$ dB.

ignored, i.e. $q = 0$. Nevertheless, it visibly improves the performance for $D = 63$ such that it becomes superior to $D = 8$ only in the presence of $C(\delta)$. Our simulations also confirm that the SoS estimator becomes unbiased only for the optimized q, which was the main motivation for introduction of $C(\delta)$ as in (3.36).

The SoS estimator's performance with respect to SNR is shown in Figure 3.10 for two different preamble lengths and multiple values of D. We note that the proposed parameter of D allows us to avoid unwieldy complexity of $D = N_p - 1$. For instance, choosing $D = 4$ results in only a loss of 0.7 dB for $L_0 = 64$ in comparison with $D = 63$. Yet, the computational complexity is reduced by a factor of approximately 16. Another important observation that can be made is that increasing L_0 from 32 to 64 yields a gain of only a fraction of dB in terms of the SNR.

The performance of the SoS detection algorithm can be examined through the ROC plots. A few examples of the ROC are plotted in Figure 3.11 where P_{FA} and P_{D} are calculated using simulations by varying $\gamma_{D'}$. These ROCs are obtained for the GMSK scheme when $E_s/N_0 = 1$ dB, and $N = 1$ ($N_p = L_0$). It is observed that we are able to attain a very low P_{FA} at low SNR

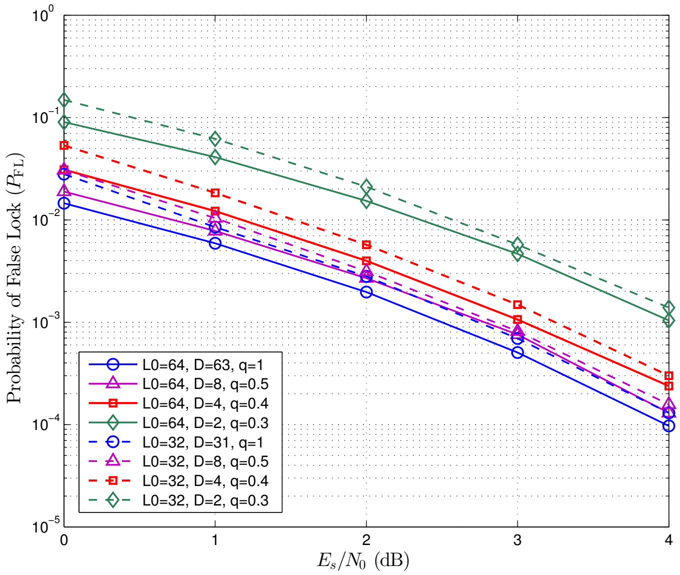


Figure 3.10. The probability of false lock versus SNR for different preamble lengths. The values of q are optimized for each case. The signal is sampled at $N = 1$, which results in $N_p = L_0$.

even with a relatively short preamble of $L_0 = 32$. It is also seen the improvement becomes less significant when D' is changed from 4 to 8. Therefore, a small value of D' looks sufficient to achieve a P_{D} that is close to the full-complexity detector, i.e. $D' = N_p - 1$. This is similar to the performance improvement of the SoS estimation algorithm versus D (Figures 3.9 and 3.10). On the other hand, the performance is improved substantially when $L_0 = 64$. For instance, $P_{\text{D}} = 1 - 5 \times 10^{-7} \approx 1$ and $P_{\text{FA}} = 4.86 \times 10^{-6}$ for $\gamma_2 = 40$. Comparing these metrics with P_{FL} in Figure 3.10 reveals that the performance of the frame synchronization algorithm is limited by the false locks rather than false alarms or missed detections.

3.6.3 BER Performance

In this section, we evaluate the overall BER performance of the proposed synchronization scheme using simulations. We have considered two examples of GMSK and 4-ary, $h = 1/4$ CPM with 2RC frequency pulse. Each burst consists of a preamble of L_0 symbols, a UW of 64 random but known bits and 4096 information bits. The UW is used in order to adjust the beginning

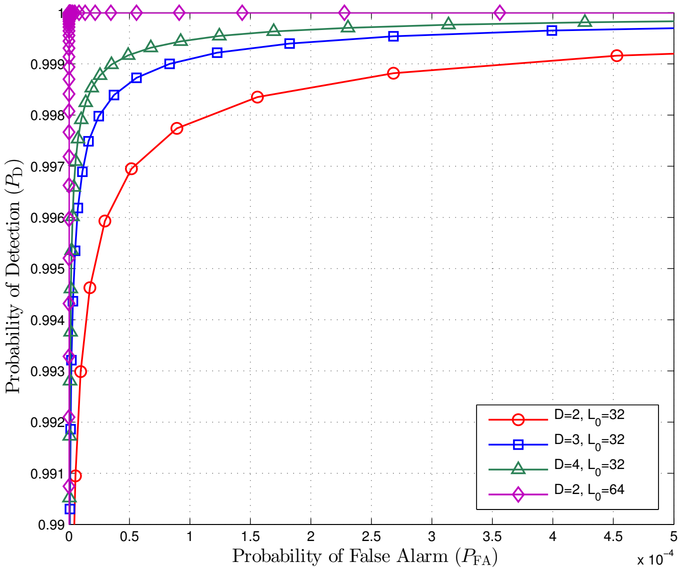
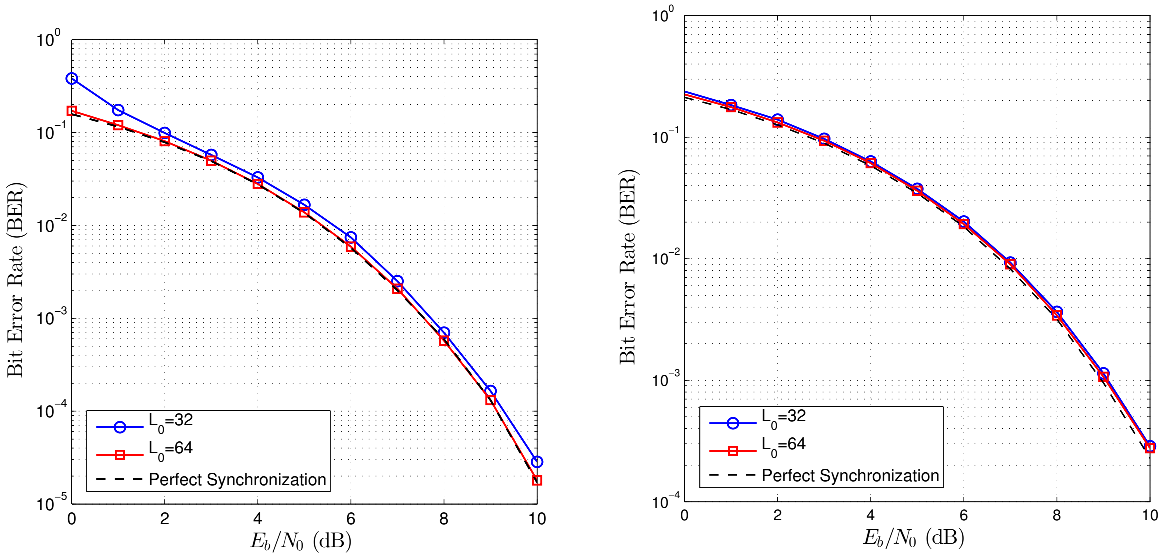


Figure 3.11. Receiver operating characteristics for the proposed detector. The optimum preamble is transmitted over an AWGN channel when $E_s/N_0 = 1$ dB and GMSK modulation is used.

of each burst by correlating it with the demodulated bits. In our simulations, the transmitter sends individual bursts that are preceded by a fixed but unknown amount of guard time. The AWGN is then added to the waveform along with random frequency and phase offsets. The received signal is sampled at $N = 2$ samples per symbol. The MLSD CPM demodulator is designed according to [65], which uses the Viterbi algorithm (VA). We have also employed a decision-directed phase and timing tracking loop [42] in which phase and timing error signals are generated according the decisions made inside the VA. The phase tracking loop is essential because even very small residual frequency offsets, after the DA synchronization, result in large phase rotations as the burst is being demodulated. The phase and timing loop bandwidths are both set to $10^{-3}/T_s$. Finally, we have set $D' = D = 4$ and $N_w = 2NL_0$ for the frame synchronization.

The BER performance of the burst-mode receiver for the GMSK scheme with two different preamble lengths is depicted in Figure 3.12 (a). It is observed that the receiver operates within less than 0.1 dB of the ideal synchronization for $L_0 = 64$ over the whole range of E_b/N_0.



(a) GMSK ($BT_s = 0.3$)

(b) 2RC ($h = 1/4, M = 4$)

Figure 3.12. BER for the burst-mode CPM receiver. L_0 is the preamble length in terms of data symbols.

However, there is a substantial BER degradation at the low SNR region for the short preamble of $L_0 = 32$. Our simulation results show that this is mainly due to the SoS false locks that are more likely to happen at low SNRs and short preamble lengths. False locks reduce the accuracy of the timing and carrier recovery algorithm, which impact the BER. At higher SNRs, there is about 0.2 dB gap that is caused by estimation errors, which are increased when L_0 is reduced.

The BER performance for the 2RC scheme is reported in Figure 3.12 (b). Similar to GMSK, $L_0 = 64$ performs almost ideal and within about 0.1 dB of perfect synchronization. However, the preamble of $L_0 = 32$ shows slightly different behavior than that of GMSK, where no BER degradation at low SNRs is visible. This is because $E_s = 2E_b$ for the 4-ary scheme and both Figures 3.12 (a) and 3.12 (b) are expressed in E_b/N_0. In other words, E_s/N_0 for GMSK is 3 dB less than for the 2RC, and hence, P_{FL} becomes larger. In fact, 2RC with $L_0 = 32$ should be compared to GMSK with $L_0 = 64$ in order to have a fair comparison where both preambles contain 64 bits. We also note there is no visible difference between the two preambles in terms of the BER, and hence, $L_0 = 32$ is an adequate length in practice. Finally, this scheme, i.e. non-binary and partial-response, is known to be prone to false locks when DD timing estimation algorithms such as [18] or [42] are implemented. Here, we showed that our proposed DA algorithm with a short preamble can be another method to solve the false lock problem while it significantly reduces the acquisition time.

3.7 Conclusions

In this Chapter, we addressed the synchronization problem for CPM signals in burst-mode transmissions. Thanks to the unique structure of the optimized synchronization preamble, we developed a DA ML algorithm, which jointly estimates the frequency offset, carrier phase and symbol timing. The proposed algorithm, which is implemented in a feedforward manner, estimates the frequency offset via two FFT operations. Once the frequency estimate is available, the carrier phase and symbol timing are easily computed via simple closed-form expressions. Our method can be applied to the whole range of CPM signals. The computed MSEs demonstrate that its performance is within 0.5 dB of the CRB for all three synchronization parameters for various examples. Moreover, it operates at frequency offsets as large as half of the sampling frequency without sacrificing the estimation accuracy.

In the second part of this chapter, we addressed the frame synchronization issue in burst-mode CPM transmissions using ML principles. We developed a simple test for detection of the SoS after which the exact location of the SoS is estimated via a one dimensional search. We numerically computed the ROCs for the SoS detector along with the false lock probabilities for the SoS estimator. The frame synchronization allowed us to implement a realistic burst-mode CPM receiver. The simulated BER curves demonstrated an almost ideal performance for preambles as short as 64 bits and SNRs as low as 0 dB.

Chapter 4

Applications to SOQPSK

4.1 Key Points of the Chapter

In this chapter, we introduce a complete synchronization strategy for burst-mode SOQPSK signals in AWGN channels. Due to the similarity of SOQPSK and CPM, we extensively use our techniques in the previous chapters by tailoring them the SOQPSK's characteristics. We first derive the optimum training sequence such that it minimizes the CRBs for frequency offset, carrier phase and symbol timing estimations. Our optimization method of the training sequence is *ad hoc*, however, it is validated by exhaustive computer search. Additionally, we propose a ML DA algorithm for joint estimation of the synchronization parameters for SOQPSK signals. We show that the proposed algorithm for the optimum training sequence can easily be extended to the preamble defined in the iNET standard [43]. This demonstrates one immediate application of this work as the first feedforward ML synchronization algorithm for the iNET standard. We compute and compare the MSE of the proposed algorithm for the optimum and iNET preambles, and for different synchronization parameters. Finally, we compare the overall performance of our proposed training sequence and synchronization algorithm for different sequence lengths by simulating a burst-mode SOQPSK receiver. This allows us to employ the right preamble length based on the desired complexity and BER performance.

4.2 Introduction

SOQPSK is a physical-layer waveform that has seen extensive use in serial streaming telemetry (SST), and has been selected for future use in the iNET system. A key difference between SST and iNET is that iNET uses burst-mode transmission. The synchronization task becomes more challenging in burst-mode transmissions because there is little time for acquiring and locking onto the signal. SOQPSK waveform can be viewed as a CPM modulation signal, and hence, we extend our techniques in the previous chapters to this type of modulation.

The burst-mode synchronization problem for CPM signals in general and SOQPSK in particular have not been studied well in the published literature. For example, [66] and [67] present timing recovery algorithms specifically designed for SOQSPK signals. However, these methods are designed based on a PLL structure, which typically needs a rather long acquisition time. Therefore, they are not suitable for a burst-mode receiver. As the name of SOQPSK suggests, it shares important similarities with OQPSK modulation, which belongs to the MSK-type modulations, i.e. binary CPMs with a modulation index of 1/2. A few published works have addressed burst-mode synchronization of MSK-type modulations using DA [24] or (NDA) algorithms [25, 26, 27]. For instance, [24] presents a DA *ad hoc* feedforward synchronization algorithm based on an MSK preamble of repeating "1100" data symbols. Although MSK synchronization techniques may be applied to SOQPSK, this approach is suboptimal due to the approximations, and hence, it would result in a poor performance.

In this chapter, we introduce a ML algorithm for feedforwrd synchronization of SOQPSK signals based on an optimized training sequence. We consider the burst-mode transmission over an AWGN channel where frequency offset, carrier phase and symbol timing are to be jointly estimated. The optimization of the training sequence is based on the CRB method for CPM signals where SOQPSK's properties are taken into the consideration. Moreover, we apply our joint estimation algorithm of Chapter 3 to the SOQPSK synchronization problem. This is explained in more details for the iNET's proposed preamble, which differs from our optimized one. Therefore, we show that our synchronization algorithm is applicable to a wider group of preambles, although it becomes suboptimum. Finally, we compare these two candidate preambles in terms of estimation error variances and BER performance using simulations.

The remainder of this chapter is organized as follows. Section 4.3 introduces the transmission model using SOQPSK signaling. In Section 4.4, we address the optimum training sequence design based on the CRB criteria for CPM signals. Section 4.5 explains the joint ML algorithm for both the optimum and iNET preambles. Section 4.6 illustrates the performance of the studied training sequences and the synchronization algorithm via simulations. Finally, the conclusions are drawn in Section 4.7.

4.3 SOQPSK Signal Representation

The SOQPSK signal can be viewed as a CPM signal with the baseband representation

$$s(t;\boldsymbol{\alpha}) = \sqrt{\frac{E_s}{T_s}}\exp\{j\phi(t;\boldsymbol{\alpha})\}, \tag{4.1}$$

where E_s is the energy per transmitted symbol and T_s is the symbol duration. The phase signal is defined as

$$\phi(t;\boldsymbol{\alpha}) = 2\pi h \sum_i \alpha_i q(t - iT_s), \tag{4.2}$$

where $\alpha_i \in \{-1, 0, 1\}$ is the transmitted ternary symbol and $h = 1/2$ is called the modulation index. The waveform $q(t)$ is the *phase response* of SOQPSK and in general is represented as the integral of the *frequency pulse* $g(t)$ with a duration of LT_s. There are currently two different versions of SOQPSK defined by their own frequency pulses. The first one known as the SOQPSK-MIL [68] is a full-response ($L = 1$) scheme with a rectangular-shaped frequency pulse. The second form is the telemetry group version [69], i.e. SOQPSK-TG, which is partial-response ($L = 8$) with a custom frequency pulse. The latter version of SOQPSK has been adopted in the iNET standard. According to the CPM definition, $q(t)$ is zero for $t < 0$ and becomes equal to $1/2$ for $t > LT_s$. The phase responses of the aforementioned SOQPSKs are illustrated in Figure 4.1.

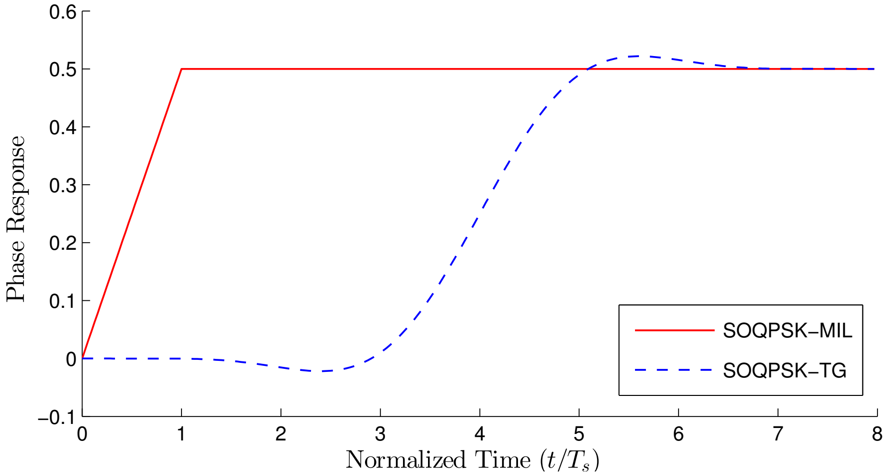


Figure 4.1. Phase response $q(t)$ for SOQPSK-MIL ($L = 1$) and SOQPSK-TG ($L = 8$).

The complexity in detection of CPM signals in general and SOQPSK in particular is exponentially related to L, which makes optimum detection of SOQPSK-TG quite complex. However, we can take advantage of a simplified model for SOQSPK-TG [70], which truncates the phase response into one symbol duration at the receiver. Therefore, the phase can be approximated as

$$\phi(t;\boldsymbol{\alpha}) \approx \pi\alpha_n q_{\mathrm{PT}}(t-nT_s) + \frac{\pi}{2}\sum_{i=0}^{n-1}\alpha_i \tag{4.3}$$

for $nT_s \leq t < (n+1)T_s$. The truncated phase response is denoted by $q_{\mathrm{PT}}(t)$ for which $L = 1$. The second term in Equation (4.3) is viewed as the phase state $\theta_{n-1} \in \{0, \pi/2, \pi, 3\pi/2\}$ when taken modulo-2π. It should be mentioned that the phase of SOQPSK-MIL can also be represented by (4.3) in which $q_{\mathrm{PT}}(t)$ is simply replaced by the original phase response of SOQPSK-MIL. Unlike SOQPSK-TG, there is no approximation in the latter case.

The SOQPSK modulator can be characterized as a precoder connected to a CPM modulator. The precoder converts binary information bits $a_n \in \{0, 1\}$ to ternary symbols by means of

$$\alpha_n = (-1)^{n+1}(2a_{n-1}-1)(a_n - a_{n-2}) \tag{4.4}$$

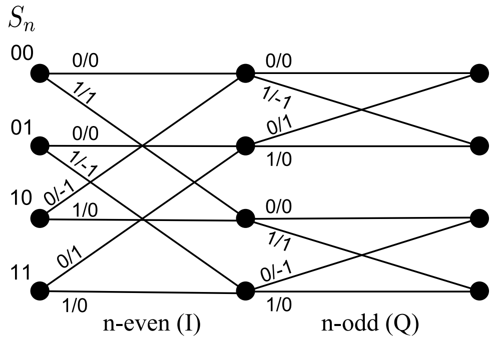


Figure 4.2. Four state time varying trellis. The labels on the branches indicate the input bit/output symbol based on the precoder of (4.4).

in order to impose OQPSK-like characteristics on the CPM signal. This important role can be identified as $\alpha_{n-1} = 1$ cannot be immediately followed by $\alpha_n = -1$ and vice versa. In other words, α_n is selected from either of $\{0, 1\}$ or $\{0, -1\}$ depending on α_{n-1}. This restriction on the sequence of data symbols increases the bandwidth efficiency of SOQPSK signals compared to binary CPMs with $h = 1/2$.

One can interpret the output of the precoder as a function of current bit a_n and three state variables: a_{n-1}, a_{n-2} and n-even/n-odd, which leaves us with an eight state trellis diagram. However, we can remove the time index from state variables and represent its function as a time-varying four state trellis as shown in Figure 4.2. The state variables, labeled as $S_n \in \{00, 01, 10, 11\}$, have a one-to-one mapping with the CPM phase state θ_{n-1} [70, Fig. 4].

4.4 Best Training Sequence

In this section we derive the optimum training sequence for joint estimation of carrier phase, frequency offset, and symbol timing of SOQPSK signals based on the CRB criterion. Since SOQPSK waveform is a special case of CPM, the CRB computations in Section 2.3 can be directly applied to SOQPSK. We only need to work with ternary symbols rather than information bits as the known training sequence $\boldsymbol{\alpha}$. The same discussion applies to the optimum training sequence for which the CRBs are minimized.

Let us recall our optimization method in Section 2.4. The data sequence, which minimizes the symbol timing CRB, is the solution to

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}}\, C \qquad \text{subject to} \quad A = B = 0, \tag{4.5}$$

where A, B, and C are defined in (2.27), (2.28), and (2.38) respectively. Additionally, setting $A = B = 0$ automatically minimizes the frequency and phase CRBs. The ternary symbol set can be viewed as the union of ± 1 symbols (binary) and zero symbols. Therefore, we consider the optimum training sequence for binary CPM as a candidate for SOQPSK. This sequence satisfies $A = B = 0$, and hence, we must only check whether it maximizes C among all ternary candidates as well. Based on (2.38), C is the summation of $\alpha_i \alpha_{i+n}$ terms when $0 \leq i < L_0$ and $n \in \{0, 1\}$. Thus, we need to avoid $\alpha_i = 0$ in maximizing C as long as a ternary $\boldsymbol{\alpha}$ is concerned. Therefore, the optimum binary training sequence is also the solution to (4.5) for the ternary $\boldsymbol{\alpha}$ because it does not contain any zero symbols. However, an issue arises with the proposed sequence due to the constraints on SOQPSK symbols where a -1 symbol cannot be immediately followed by a $+1$ symbol and vice versa. We can approach this problem in two different ways. The first solution is to bypass the SOQPSK's pre-coder during the transmission of the training sequence such that the exact sequence of Figure 4.3 (a) is fed to the CPM modulator. Another approach is to insert zero symbols in the locations that there is a sign transition. The second approach violates our condition on A. However, we opt for the second approach due to the ease of implementation and the fact that there are only two transitions. This results in a sequence which is only different by two symbols from the optimum sequence. The optimized ternary sequence is shown in Figure 4.3 (b). We note that the optimality of this sequence is subject to certain assumptions and approximations. However, it can be shown that the above sequence is asymptotically optimum following CPM methods in Chapter 2. In order to be consistent in the rest of our discussion, we refer to the sequence of Figure 4.3 (b) as the optimum training sequence whether it is optimum or near-optimum.

Finally, we have derived the optimum training sequence via a computer brute force search when $L_0 = 20$. The results are shown for SOQPSK-TG and SOQPSK-MIL standards in

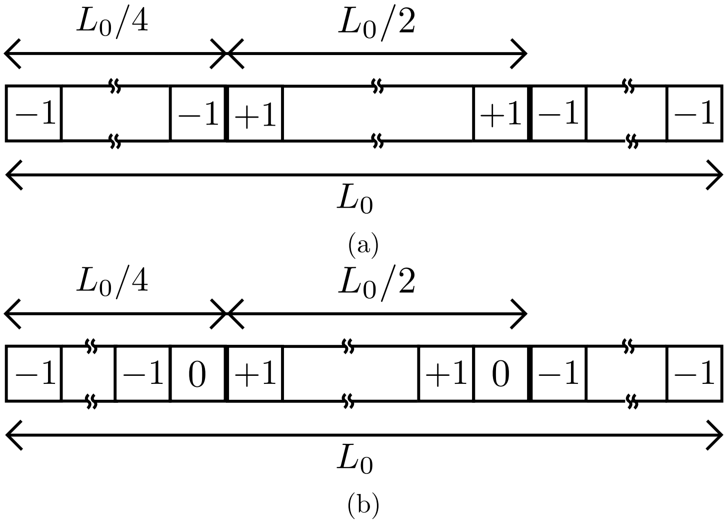
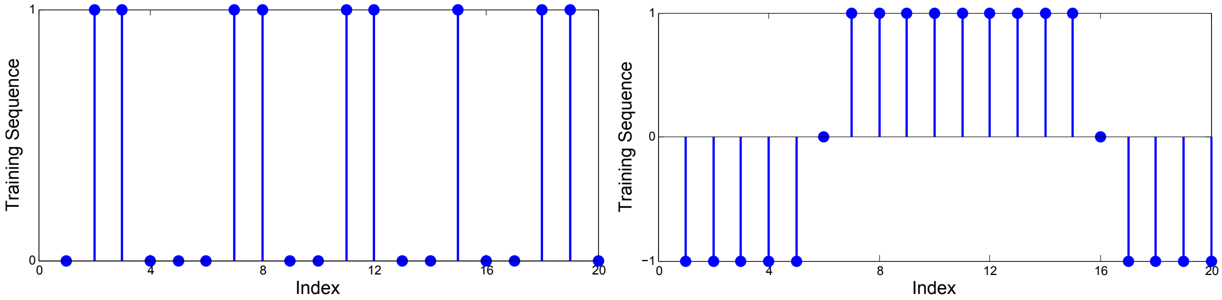


Figure 4.3. The optimum sequence for symbol timing estimation in binary CPM signals (a) and the modified version for SOQSPK signals (b). The negative of this sequence is also optimum.

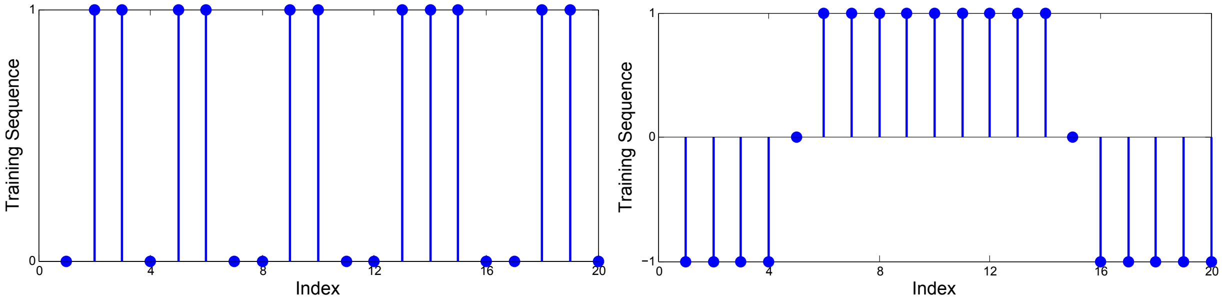
Figures 4.4 and 4.5 respectively. It can be seen that the computer search results follow our proposed sequence of Figure 4.3 (b), which validate the accuracy of our approximations for short sequences. Careful reader may notice that the transition points are shifted by one symbol for the SOQPSK-TG, which is caused by its long phase response. Although we can adjust the transition points for a given L_0 using the method of Figure 2.2 for partial-response CPMs, we assume the training sequence of Figure 4.3 (b) for all versions of SOQPSK. This makes the following discussion consistent and simple. Moreover, we have presented the actual information bits which are fed to the SOQPSK modulator. It is interesting to interpret the optimum training sequence in terms of the QPSK constellation points. It can be seen that the optimum training sequence alternates between two QPSK constellation points that are 180° apart for the first quarter of the sequence. In the next half, it alternates between two other points, which are again 180° apart. Finally, it alternates between the initial points for the remaining quarter of the sequence.



(a) Information Bits

(b) Ternary symbols

Figure 4.4. The computer search results for optimum training sequence for SOQPSK-TG when $L_0 = 20$.



(a) Information Bits

(b) Ternary symbols

Figure 4.5. The computer search results for optimum training sequence for SOQPSK-MIL when $L_0 = 20$.

4.5 Timing and Carrier Recovery

In this section, we derive an ML algorithm for joint estimation of frequency offset, carrier phase and symbol timing for SOQPSK signals using the training sequence of Figure 4.3 (b). Due to the similarities with CPM in terms of waveform and training sequence, we only highlight the differences. Later in Section 4.5.2, we will extend our algorithm to the iNET preamble in more details.

4.5.1 ML Estimation for Optimum Training Sequence

We begin by studying the phase response of SOQSPK schemes to our optimized training sequence, which is shown in Figure 4.6 for $L_0 = 32$. It is observed that the phase response consists of three major parts in which $\phi(t; \boldsymbol{\alpha})$ varies with a constant rate of $\pm\pi/2$ radians

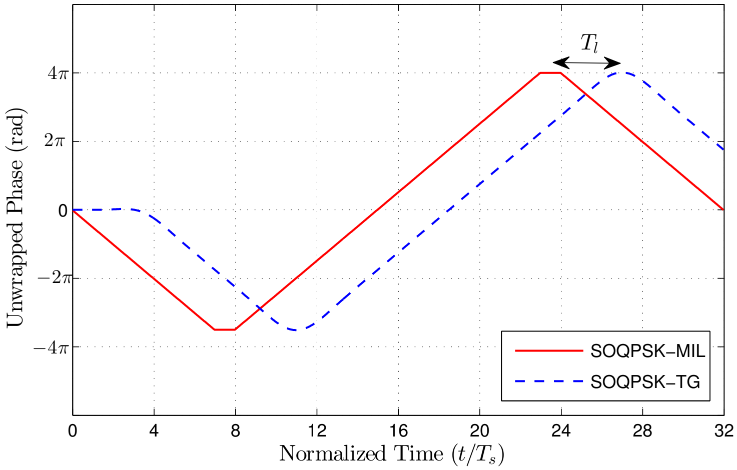


Figure 4.6. The unwrapped phase response of SOQPSK-MIL and SOQPSK-TG schemes to the optimum training sequence when $L_0 = 32$.

per symbol. Each of these parts corresponds to consequent $+1$ or -1 symbols in the training sequence. More importantly, the phase response of SOQPSK-TG varies with a fixed rate within each part despite its non linear phase response (see Figure 4.1). This is resulted by the overlap of frequency pulses when adjacent symbols are the same. We also note that the overall response of SOQPSK-TG is delayed by $T_l = 3.5T_s$, which is caused by its partial response behavior. Another difference is that SOQPSK-TG exhibits smoother transitions because of its phase response shape, which also makes it more bandwidth efficient. Therefore, we use the phase response of SOQPSK-MIL as the *template* and approximate other versions of SOQPSK with a delayed version of it. This is the main idea of the proposed estimation algorithm.

Based on the above discussion, we can mathematically express the phase response of SOQPSK to the optimum training sequence $\boldsymbol{\alpha}^*$, i.e.,

$$\phi(t, \boldsymbol{\alpha}^*) \approx \begin{cases} -\frac{\pi t}{2T_s} & T_l < t \leq \frac{T_0}{4} + T_l \\ \frac{\pi(t - T_0/2 + T_s)}{2T_s} & \frac{T_0}{4} + T_l < t \leq \frac{3T_0}{4} + T_l \\ -\frac{\pi(t-T_0)}{2T_s} & \frac{3T_0}{4} + T_l < t \leq T_0 + T_l \\ 0 & \text{otherwise,} \end{cases} \tag{4.6}$$

where T_l is the lag time in the phase response present in partial response SOQPSK such as SOQPSK-TG. We proved in Appendix **??** that $T_l = (L-1)/2$. We can easily recognize the similarities between (4.6) and its CPM counterpart in (3.6) when $h = 1/2$ and $M = 2$. The only difference is that $\phi(t, \boldsymbol{\alpha}^*)$ in the second case of (4.6) is advanced by $\pi/2$ radians, which is equal to T_s seconds. This is explained according to Figure 4.6 and SOQPSK-MIL template as follows. It is seen that the phase response has in fact five parts while (4.6) expresses it in three segments. This is another approximation we make in which the phase response during the two transitions (constant-phase) intervals is approximated by the same function as its previous part, i.e. we assume $\phi(t, \boldsymbol{\alpha}^*)$ is reduced by $\pi/2$ during $7T_s < t < 8T_s$. However, the phase of the signal during the transition interval does not change in reality because it corresponds to transmission of a zero symbol. We compensate this by adding $\pi/2$ radians to the second case of (4.6). The third case is left unchanged as the second transition cancels the first one. This assumption simplifies our derivations in the following discussion. We do not expect a noticeable loss in the performance of the proposed estimator as each of these transition intervals last only for one symbol time, which is much smaller than the sequence duration especially for longer training sequences. Our simulation results in Section 4.6 confirm this prediction.

We take advantage the above representation to derive the joint LLF and its maximization, which is similar to the CPM's in Chapter 3. The modification of (4.6) shows its effect on the *pre-processing* of the received and sampled signal $r[n]$, i.e.,

$$r_1[n] = \begin{cases} r[n] & 0 \le n < NL_0/4 \\ \exp[-j\pi L_0/2] r[n] & 3NL_0/4 \le n < NL_0 \\ 0 & \text{otherwise}, \end{cases} \tag{4.7}$$

and

$$r_2[n] = \begin{cases} \exp[j\pi(L_0/4 - 1/2)] r[n] & NL_0/4 \le n < 3NL_0/4 \\ 0 & \text{otherwise}. \end{cases} \tag{4.8}$$

After computing above signals, they are used in (3.21) and (3.22) to compute $\lambda_1(\hat{\nu})$ and $\lambda_2(\hat{\nu})$, respectively, when $h = 1/2$ and $M = 2$. The results are then applied to (3.16), (3.17), and (3.18) in order to estimate frequency offset, symbol timing and carrier phase respectively.

4.5.2 ML Estimation for iNET Preamble

The proposed preamble for iNET [71] has a length of $L_0 = 128$. This preamble is periodic and it consists of repeating a sequence of 16 information bits 8 times as follows,

$$\left.\begin{array}{lcl} a_{2k} & = & 1,0,1,0,1,0,1,0 \\ a_{2k+1} & = & 1,0,1,1,0,1,0,0 \end{array}\right\} \text{for } k = 0,\ldots,7 \tag{4.9}$$

where $\{a_i\}$ is fed into a precoder in which $a_{-2} = a_{-1} = 0$. This results in a sequence of ternary symbols which is depicted in Figure 4.7 for one period of the preamble. We denote this preamble by $\mathring{\boldsymbol{\alpha}}$ in the rest of our discussion.

Let us investigate $\mathring{\boldsymbol{\alpha}}$ and its phase response more carefully. We notice that $\mathring{\boldsymbol{\alpha}}$, within each period, can be divided into two segments, each of which having the same symbols of either $+1$ or -1. This pattern causes the signal's phase to change with a uniform rate of approximately $\pi/2$ radians per symbol in the same direction within each segment. This behavior is illustrated in Figure 4.8 by plotting the unwrapped phase response of SOQPSK-MIL and SOQPSK-TG when $\mathring{\boldsymbol{\alpha}}$ for its first 32 symbols is utilized. More importantly, SOQPSK-TG's phase response follows a straight line within each part, similar to SOQPSK-MIL, despite the short length of each period. This is indeed the same behavior as we observed for $\boldsymbol{\alpha}^*$ and SOSPKS-TG. Therefore, we can apply a similar method for timing and carrier synchronization using the iNET preamble.

Based on the above discussion, we approximate the phase response of SOQPSK-TG to $\mathring{\boldsymbol{\alpha}}$ with a delayed version of SOQPSK-MIL's response to the same preamble sequence. Its

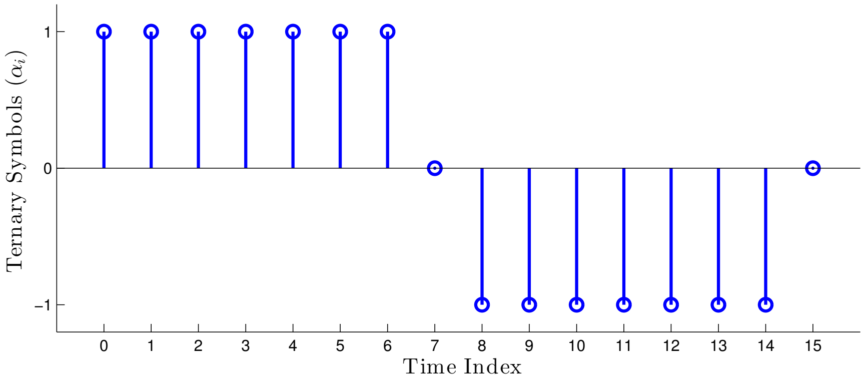


Figure 4.7. A length-16 period of the ternary symbols in the iNET preamble for SOQPSK-TG. The full length-128 preamble is formed by repeating above sequence 8 times.

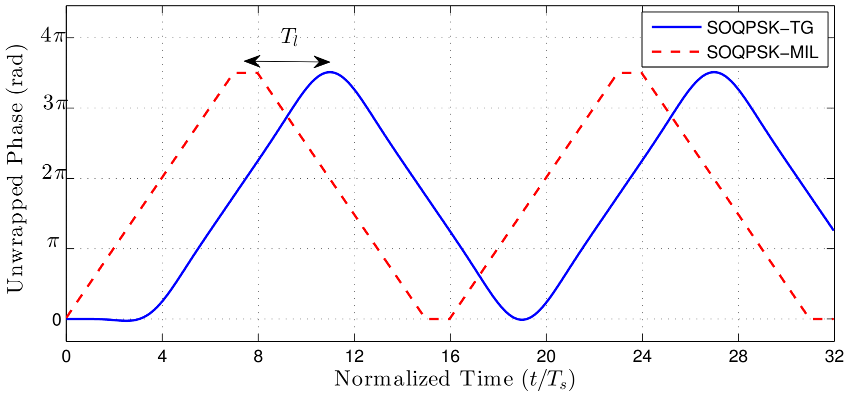


Figure 4.8. The unwrapped phase response of SOQPSK-MIL and SOQPSK-TG schemes for the first 32 symbols of the iNET preamble.

approximated phase response $\phi(t, \mathring{\boldsymbol{\alpha}})$ can be mathematically expressed as

$$\phi(t; \mathring{\boldsymbol{\alpha}}) \approx \begin{cases} \frac{\pi(t-16kTs-T_l)}{2T_s} & 16kT_s + T_l < t \leq (16k+8)T_s + T_l \\ -\frac{\pi(t-(16k+15)T_s-T_l)}{2T_s} & (16k+8)T_s + T_l < t \leq (16k+16)T_s + T_l \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

for $k = 0, \ldots, 7$. $T_l = 3.5T_s$ is the lag time in the phase response of SOQPSK-TG, which is fixed and known to the receiver. Unlike our discussion on CPM signals, we continue using continuous time representations, and apply the sampling to the final results. We use (4.10) to express the baseband SOQPSK-TG signal $s(t)$ during the training sequence transmission as

$$s(t) \approx \begin{cases} \exp[+j\frac{\pi t}{2T_s}] & 16kT_s < t \leq (16k+8)T_s \\ \\ \exp[-j(\frac{\pi t}{2T_s} + \frac{\pi}{2})] & (16k+8)T_s < t \leq (16k+16)T_s. \end{cases} \tag{4.11}$$

We take advantage of the above approximation in order to simplify the LLF and its maximization algorithm. Using (4.11) in (3.4) results in a simplified form for the LLF when $\mathring{\boldsymbol{\alpha}}$ is transmitted, i.e.,

$$\begin{aligned} \mathring{\Lambda}[r(t); \tilde{f}_d, \tilde{\theta}, \tilde{\tau}] \approx \mathrm{Re}\Big\{ e^{-j\tilde{\theta}} \sum_{k=0}^{7} \Big[\int_{16kT_s}^{(16k+8)T_s} & e^{-j2\pi f_d t} r(t) e^{-j\pi(t-\tau)/2T_s} dt \\ + \int_{(16k+8)T_s}^{(16k+16)T_s} & e^{-j2\pi f_d t} r(t) e^{j\pi(t-\tau)/2T_s} e^{j\pi/2} dt \Big] \Big\}, \end{aligned} \tag{4.12}$$

where $\mathring{\Lambda}[\cdot]$ represents the joint LLF given $\mathring{\boldsymbol{\alpha}}$. It is evident from (4.12) that the symbol timing is now decoupled from the frequency offset and can be moved outside the integrals of the LLF. Hence, the joint LLF can be summarized as

$$\mathring{\Lambda}[r(t); \tilde{f}_d, \tilde{\theta}, \tilde{\tau}] \approx \mathrm{Re}\Big\{ e^{-j\tilde{\theta}} \big[e^{j\pi\tilde{\tau}/2T_s} \lambda_1(\tilde{f}_d) + e^{-j\pi\tilde{\tau}/2T_s} \lambda_2(\tilde{f}_d) \big] \Big\}, \tag{4.13}$$

where

$$\lambda_1(\tilde{f}_d) = \sum_{k=0}^{7} \int_{16kT_s}^{(16k+8)T_s} e^{-j2\pi f_d t} r(t) e^{-j\pi t/2T_s} \, dt, \tag{4.14}$$

and

$$\lambda_2(\tilde{f}_d) = e^{j\pi/2} \sum_{k=0}^{7} \int_{(16k+8)T_s}^{(16k+16)T_s} e^{-j2\pi f_d t} r(t) e^{j\pi t/2T_s} \, dt. \tag{4.15}$$

Because the estimation parameters are now decoupled, the maximization of the LLF becomes straightforward. Based on (4.13), we define the normalized symbol timing with respect to the symbol duration as $\varepsilon = \tau/T_s$, which is used in the rest of our discussion. Let us proceed by denoting the term in (4.13) that corresponds to symbol timing and frequency offset as

$$\Gamma(\tilde{f}_d, \tilde{\varepsilon}) = e^{j(\pi/2)\tilde{\varepsilon}}\lambda_1(\tilde{f}_d) + e^{-j(\pi/2)\tilde{\varepsilon}}\lambda_2(\tilde{f}_d). \tag{4.16}$$

It is observed that for any trial value of $(\tilde{f}_d, \tilde{\varepsilon})$, $\mathring{\Lambda}(\cdot)$ is maximized by choosing $\tilde{\theta}$ such that it rotates $\Gamma(\tilde{f}_d, \tilde{\varepsilon})$ towards the real axis, i.e.,

$$\tilde{\theta} = \arg\{\Gamma(\tilde{f}_d, \tilde{\varepsilon})\}, \tag{4.17}$$

which reduces the LLF to $|\Gamma(\tilde{f}_d, \tilde{\varepsilon})|$. Thus, the ML estimates of $\tilde{f}_d$ and $\tilde{\tau}$ are found by maximizing

$$|\Gamma(\tilde{f}_d, \tilde{\varepsilon})|^2 = |\lambda_1(\tilde{f}_d)|^2 + |\lambda_2(\tilde{f}_d)|^2 + 2\mathrm{Re}\Big[e^{-j\pi\tilde{\varepsilon}}\lambda_1^*(\tilde{f}_d)\lambda_2(\tilde{f}_d)\Big] \tag{4.18}$$

with respect to $(\tilde{f}_d, \tilde{\varepsilon})$. The first two terms on the right-hand side of (4.18) do not depend on $\tilde{\varepsilon}$. Using a similar argument as $\tilde{\theta}$, the third term is maximized by selecting $\tilde{\varepsilon}$ according to

$$\tilde{\varepsilon} = \frac{\arg\{\lambda_1^*(\tilde{f}_d)\lambda_2(\tilde{f}_d)\}}{\pi} \tag{4.19}$$

such that the term inside the real part operator of (4.18) becomes purely real and equal to $|\lambda_1^*(\tilde{f}_d)\lambda_2(\tilde{f}_d)|$. Therefore, the maximization of the LLF is now a one dimensional problem that results in the ML estimate of frequency offset, i.e. $\hat{f}_d$. This can be expressed mathematically in the form of

$$\hat{f}_d = \underset{\tilde{f}_d}{\operatorname{argmax}} \Big\{ X(\tilde{f}_d) = |\lambda_1(\tilde{f}_d)| + |\lambda_2(\tilde{f}_d)| \Big\}, \tag{4.20}$$

which leads to the ML estimates of the normalized symbol timing $\hat{\varepsilon}$ and phase offset $\hat{\theta}$ via

$$\hat{\varepsilon} = \frac{\arg\{\lambda_1^*(\hat{f}_d)\lambda_2(\hat{f}_d)\}}{\pi}, \tag{4.21}$$

and

$$\hat{\theta} = \arg\Big\{e^{j(\pi/2)\hat{\varepsilon}}\lambda_1(\hat{f}_d) + e^{-j(\pi/2)\hat{\varepsilon}}\lambda_2(\hat{f}_d)\Big\}, \tag{4.22}$$

respectively.

According to (4.14) and (4.15), each of $\lambda_1(f_d)$ and $\lambda_2(f_d)$ requires computation of 8 integrals with different limits. In order to make them consistent, we define two new signals, i.e. $r_1(t)$ and $r_2(t)$, such that

$$r_1(t) = \begin{cases} r(t) & 16kT_s < t \leq (16k+8)T_s \\ 0 & \text{otherwise}, \end{cases} \tag{4.23}$$

and

$$r_2(t) = \begin{cases} e^{j\pi/2}r(t) & (16k+8)T_s < t \leq (16k+16)T_s \\ 0 & \text{otherwise}. \end{cases} \tag{4.24}$$

for $0 \leq k \leq 7$. The above modifications to $r(t)$ lead to similar forms for $\lambda_1(f_d)$ and $\lambda_2(f_d)$, where each one requires computation of one integral with a duration of $[0, T_0]$.

In practice, $r(t)$ is sampled at N times per symbol time. This results in $r_1[n]$ and $r_2[n]$, which are discrete versions of (4.23) and (4.24) respectively. These signals are then phase rotated and sent to the FFT modules, i.e.

$$\lambda_1(\tilde{\nu}) = \sum_{n=0}^{NL_0-1} r_1[n]e^{-j\frac{\pi n}{2N}}e^{-j2\pi n\tilde{\nu}}, \tag{4.25}$$

and

$$\lambda_2(\tilde{\nu}) = \sum_{n=0}^{NL_0-1} r_2[n]e^{j\frac{\pi n}{2N}}e^{-j2\pi n\tilde{\nu}}, \tag{4.26}$$

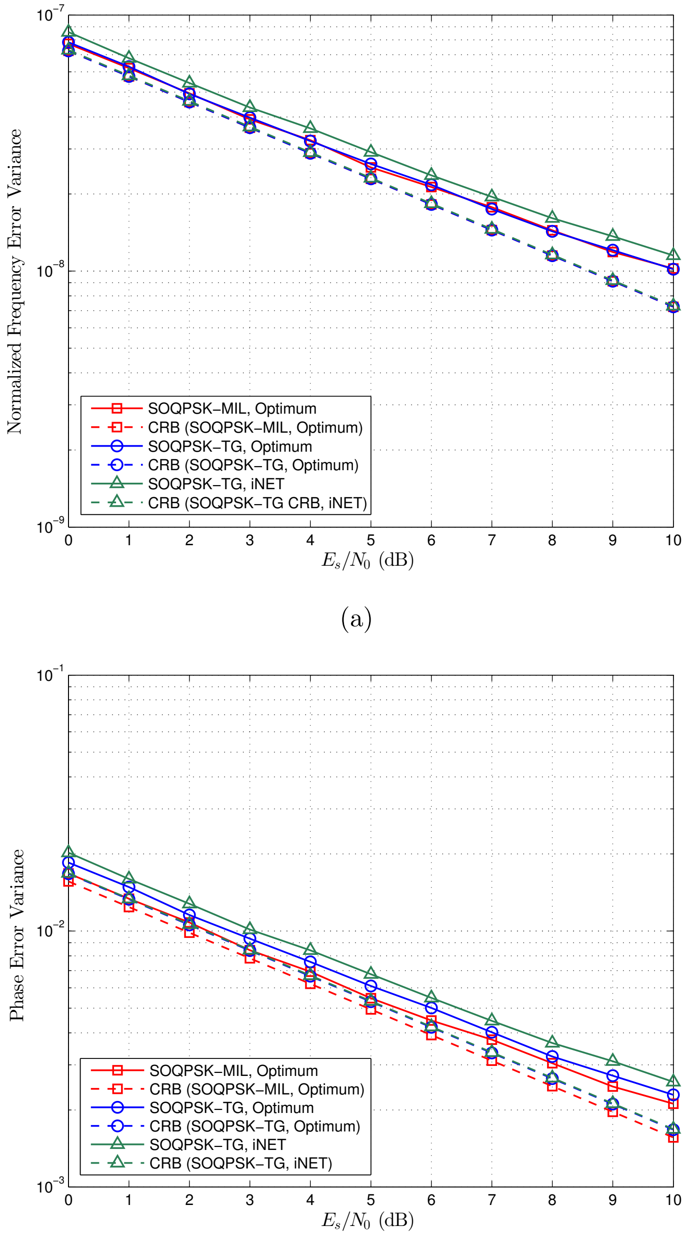
which are then used in (4.21) in order to estimate the frequency offset. Similar to the CPM, we zero pad $r_1[n]$ and $r_2[n]$ with a factor of K_f, and perform the Gaussian interpolation on the FFT results. Once a fine estimate of ν is available, we recompute $\lambda_1(\hat{\nu})$ and $\lambda_2(\hat{\nu})$, which are inserted in (4.21) and (4.22) for estimation of ε and θ respectively.

4.6 Discussion and Results

The estimation error variances of frequency offset, carrier phase, and symbol timing are depicted in Figures 4.9 (a), 4.9 (b), and 4.10 respectively. These set of plots are computed using Monte Carlo simulations in which $L_0 = 128$, $K_f = 2$, and $N = 2$. Moreover, we have compared the variances with their corresponding CRB for both optimum and iNET training sequences.

Figure 4.9 (a) shows that SOQPSK-MIL and SOQPSK-TG with the optimum training sequence perform the same in terms of frequency estimation as their CRBs are equal too. However, there is a performance gap between the variance and its CRB, which is due to the FFT and interpolation resolution. This gap becomes larger at high SNRs where the errors due to thermal noise become smaller than the FFT precision. In terms of phase estimation, it is seen that SOQPSK-MIL exhibit lower error variance compared to the SOQPSK-TG, which follows their CRBs. This in fact demonstrates the close relation between the CRB and ML estimation error variances. The phase estimation error variance degrades at high SNRs because of the close relation between frequency and phase, i.e. errors in frequency estimation leads to undesirable phase rotations. Finally, timing error variances show that both versions of SOQPSK perform the same as each other, and the CRB is reached. This situation is similar to the CPM where the errors in frequency estimation does not impact symbol timing estimates as far as the optimum training sequence is utilized.

Comparison of the optimum and iNET preambles for SOQPSK-TG reveals the superiority of our proposed preamble in terms of the estimation error variance. It is seen that the optimum training sequence delivers an approximate SNR gain of 0.5 dB for frequency and phase estimations despite the fact that both sequences have the same CRB. Figure 4.10 shows that the iNET preamble's symbol timing CRB is approximately 1 dB worse than that of the optimum preamble. However, its estimation error variance demonstrates a dramatic loss especially at high SNRs. This can be explained based on the approximations we made in representing the LLF (4.12). Following the piecewise linear representation of signal phase, we separated the integral involved in computation of the LLF. In a strict sense, each of these integrals must be



(a)

(b)

Figure 4.9. The error variance of frequency offset (a) and carrier phase (b) estimations for different SOQPSK schemes when $L_0 = 128$. The frequency is normalized with respect to the symbol rate.

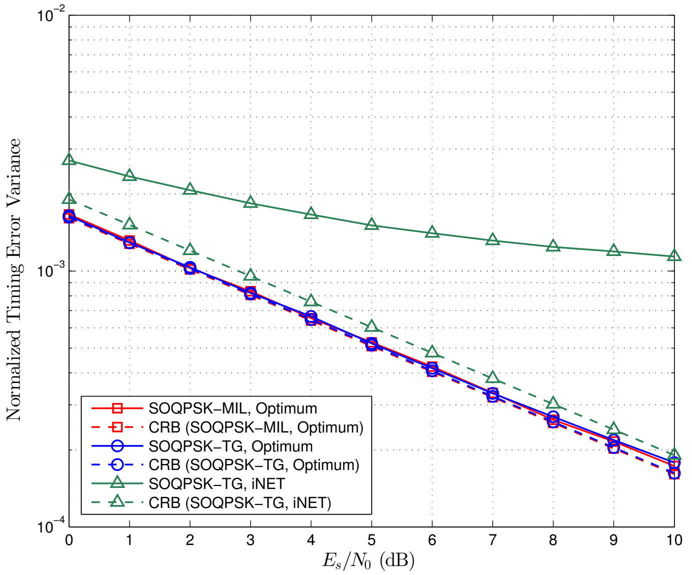


Figure 4.10. The error variance of symbol timing estimation for different SOQPSK schemes. The symbol timing is normalized to the symbol duration.

perfectly aligned the received signal, i.e. the integral limits should be adjusted based on $\tilde{\tau}$. However, we ignored $\tilde{\tau}$ in the integral limits to be able to proceed with our method. This is indeed a very good approximation for the optimum preamble because $|\tilde{\tau}|$ is much smaller than the integral limits for the optimum preamble and for moderate to large values of L_0. However, it becomes comparable to the integral duration in the LLF of the iNET preamble, which is $8T_s$. Our simulation results in Figure 4.11 also confirms that the estimation error is increased for larger values of $|\tau|$ in case of the iNET preamble. A similar behavior has also been reported in [19], where a DA ML algorithm was developed for joint estimation of symbol timing and phase. The authors assume an arbitrary training sequence in their work and separated the LLF for every single symbol. That is a worst case scenario, and as reported, its timing estimation performance is only good for small $|\tau|$. On the contrary, our work takes into account the structure of the optimized method, and hence, it works perfectly for all possible values of τ. Finally, it should be mentioned that the performance of our joint estimation algorithm for the iNET preamble is still acceptable as CPM detection is less sensitive to timing errors rather than errors in the phase [1].

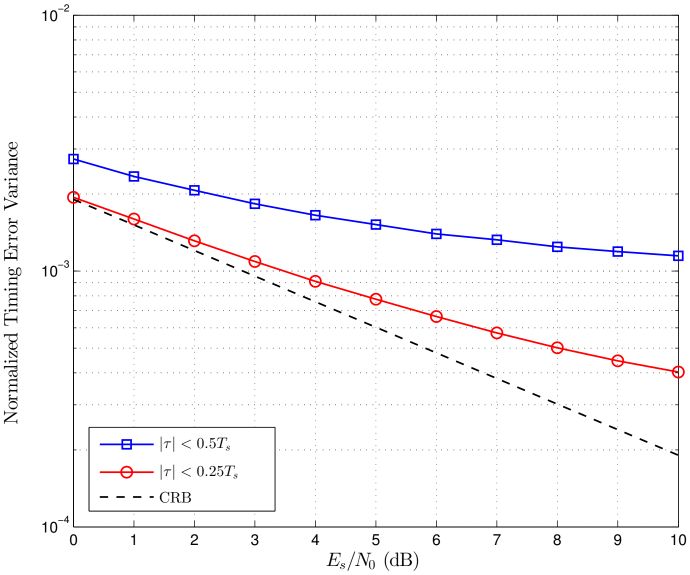


Figure 4.11. The comparison of error variance of normalized symbol timing estimation for SOQPSK-TG with iNET preamble for different variations of symbol timing error.

Based on our discussion in this chapter and frame synchronization of Section 3.5, we implemented a burst-mode SOQPSK-TG receiver in MATLAB in order to evaluate the overall performance of our synchronization method. A high-level block diagram of this receiver is depicted in Figure 4.12. The channel observation is first sent to an anti-aliasing filter (AAF) after which it is sampled at $N = 2$ samples per symbol. The frame synchronization module observes $r[n]$ using a sliding window such that it detects the arrival of new bursts, and then estimates the location of the SoS. The parameters for this module are $D = D' = 4$ and $N_w = 2NL_0$. Once the SoS is identified, a vector of NL_0 samples, which correspond to the preamble, are used for timing and carrier recovery. The frequency estimator works at $K_f = 2$ and utilizes a Gauassian interpolator. After estimation of the synchronization parameters, the phase of received signal is corrected according to $\hat{\nu}$ and $\hat{\theta}$. Next, a parabolic interpolator [72] corrects the sample timing error based on $\hat{\varepsilon}$. Unlike CPM, the synchronized signal cannot be directly used by the Viterbi demodulator because the SOQPSK-TG trellis is time-varying, which results in ambiguity. Thus, we use a symbol-by-symbol detection filter (DF) [73] over an uncertainty region where the UW

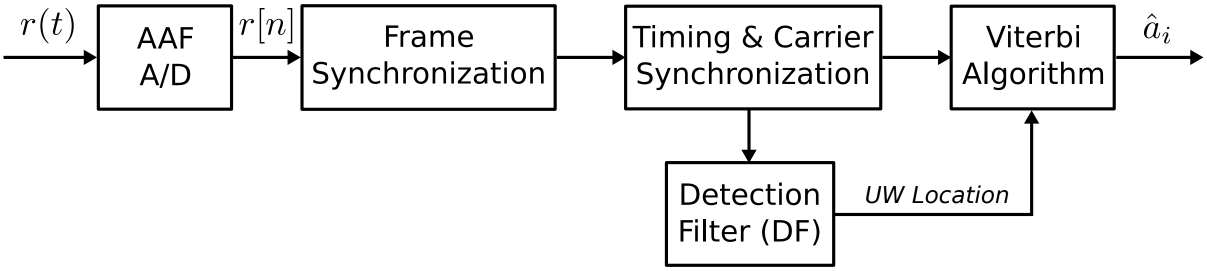


Figure 4.12. The block diagram of the burst-mode SOQPSK-TG receiver.

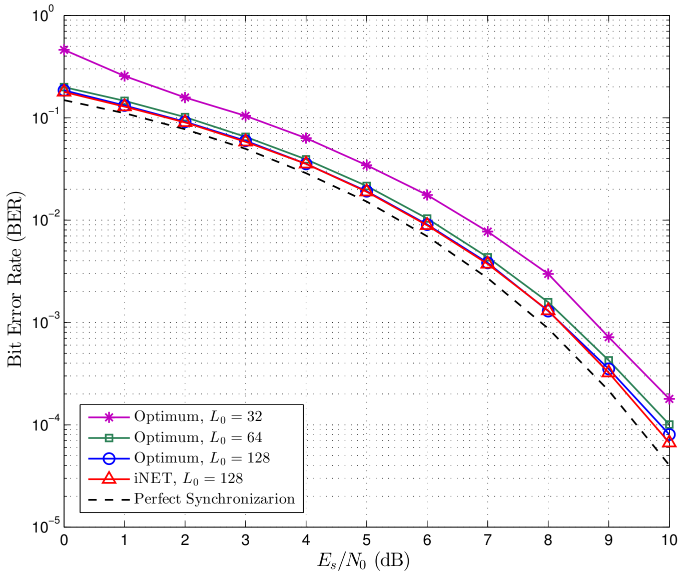


Figure 4.13. The BER performance of the burst-mode SOQPSK-TG receiver for different preambles.

is supposedly located. The resulted symbols are correlated with the known UW symbols so that its exact location is identified. This resolves the even/odd timing ambiguity and allows the Vitrebi algorithm to correctly demodulate the received burst according to the trellis of Figure 4.2. Finally, we have employed a phase tracking loop inside the Viterbi demodulator using a method known as the per-survivor-processing (PSP) [1, Section 6.5.2]. The PSP corrects the phase of each branch in the trellis prior to making the decisions by the Viterbi algorithm. The PSP allows the receiver to handle any residual frequency offset as well as fading channels.

The BER performance of the SOQPSK-TG burst-mode receiver is shown in Figure 4.13 for different preambles using simulations. Each burst consists of 6144 information bits, a UW of 64 pseudo-random bits, and a preamble of L_0 bits. Since SOQPSK is inherently binary, one bit is equivalent to one symbol. Comparison of the iNET and optimum preambles shows that both of them perform quite closely despite considerable loss in symbol timing estimation of the iNET's in Figure 4.10. This can be viewed as SOQPSK's demodulation (similar to CPM's) is rather robust to small timing errors, i.e. both preambles are able to attain an error variance that is well below 10^{-2} at all SNRs. Nevertheless, we can still observe that the optimum preamble outperforms the iNET preamble at the high SNR region in which the iNET's symbol timing error variance levels off. We also note that both these preambles result in only 0.25 dB SNR loss at the BER of 10^{-3} when they are compared to the perfect synchronization scenario. This shows the effectiveness of our proposed synchronization algorithm. We have also studied the BER performance of the optimum preamble for $L_0 = 64$ and $L_0 = 32$. It is observed that the preamble of $L_0 = 64$ makes an additional SNR loss of about 0.2 dB, which is still quite acceptable recalling the complexity reduction that it delivers. The performance loss is more significant in the case of $L_0 = 32$, that is slightly less than 1 dB away from the perfect synchronization scenario. There are two reasons for the performance loss when L_0 is reduced. The first one is the increase of initial estimation error variances as the CRBs are inversely proportional to L_0 (L_0^3 for frequency). Additionally, we have to increase the phase tracking loop's bandwidth for smaller values of L_0 in order to cope with the increased residual frequency offset. This in turn introduces more noise to the phase loop, which impacts the demodulation quality. Finally, we note that the BER loss is larger for $L_0 = 32$ at very low SNRs. This behavior is resulted by false locks at the frame synchronization level, which are more likely to happen for small values of L_0.

4.7 Conclusions

In this chapter, we studied the training sequence design and feedforward DA synchronization of burst-mode SOQPSK transmissions over AWGN channels. We presented a systematic technique for the design of optimum or near-optimum training sequences for joint estimation of frequency offset, carrier phase and symbol timing based on the CRB computations, which we had derived for CPM in Chapter 2. We showed that the SOQPSK's optimum training sequence is very similar to the CPM's, and hence, the same method of piecewise linear phase approximation can be applied to obtain the synchronization algorithm. We extended this idea to the iNET's preamble, which has more variations compared to the optimum one. Using simulations, we calculated the MSE of the joint estimation algorithm for both preambles, which revealed that the optimum preamble attains the CRB at all SNRs while the iNET's timing MSE levels off at high SNRs. However, the estimation errors for both preambles are such that they both deliver very close BERs at low to moderate SNRs. We also investigated the BER performance of SOQPSK-TG for the optimum preamble with shorter lengths. It was observed that a preamble of 64 bits is quite reasonable in which the SNR loss is less than 0.5 dB away from the ideal synchronization case.

Chapter 5

Conclusions

This work addresses feedforward synchronization of CPM signals for burst-mode communications. This area had not been studied adequately before, due to the complexities in CPM signals. In order to assist the synchronization task, we analytically derived the optimum training sequence for CPM, which had been previously available only for linear modulations such as PSK. We utilized the structure of our optimized training sequence for joint timing and carrier recovery. Moreover, we developed a frame synchronization scheme for CPM signals in burst-mode transmissions and frequency uncertainty. Thus, it can be stated that a complete feedforward DA synchronization scheme for general CPM signals in burst-mode transmission was developed in this work. This is the essence of this work, however, this contribution can be divided into separate parts as follows.

5.1 Contributions

1. The best training sequence for synchronization of CPM signals was proposed. We computed the CRB for joint estimation of frequency offset, carrier phase, and symbol timing for CPM signals in AWGN. We developed the asymptotically optimum training sequence such that it minimizes the CRB for all three estimation parameters simultaneously. We also presented closed-form expressions for the UCRB, which enabled us to compare the

theoretical performance of the optimum training sequence and a randomly selected one.

2. Based on the optimized training sequence, we designed a DA ML synchronization scheme, which jointly estimates frequency offset, carrier phase and symbol timing. To the best of our knowledge, all previous publications in this field had either ignored one of the synchronization parameters or they were limited to a specific class of CPM such as MSK modulation. We computed the MSE of the proposed algorithm for different CPM examples including M-ary and/or partial-response ones using simulations. It was observed that our algorithm is capable of performing quite close to the theoretical CRB for all three synchronization parameters and SNRs as low as 0 dB.

3. We also extended the frame synchronization work of Choi & Lee to our scenario by considering the CPM autocorrelation as well as the guard interval prior to the preamble in burst-mode transmissions. This enabled us to simulate a burst-mode CPM receiver by which we derived BER plots. It was observed that the proposed synchronization scheme–including the training sequence, timing & carrier recovery, and frame synchronization–performs within 0.1 dB of a perfectly synchronized receiver at all SNRs with a relatively short preamble of 64 bits.

4. Finally, we extended our techniques to SOQPSK signals, which are practical ternary CPMs. As an immediate application, we applied our synchronization algorithm to the preamble defined in the iNET standard. We showed that this preamble along with our algorithm can perform as well as the optimum one in terms of the delivered BER. However, we suggested a half-sized preamble of 64 bits in order to save bandwidth and computations while its SNR loss is only 0.2 dB compared to the standard one.

5.2 Areas of Future Study

The optimum training sequence design can be investigated for other transmission environments such as frequency selective channels. In such applications, the CRB becomes a function of the channel response as well as the training sequence and/or synchronization parameters. Due to the increased complexity of the optimization problem, we may use computer search methods such as the GA, which was touched on in this work.

Additionally, work can be done on DA synchronization of CPM signals in frequency selective channels where multiple replicas of the preamble can arrive at the receiver. An interesting research direction will be a joint synchronization and channel estimation algorithm such that channel coefficients are estimated along with frequency offset and symbol timing.

Page left intentionally blank.

Appendix A

Exact Computation of $\mathbf{C}_{\mathbf{x}_2}$

Let us begin by recalling X_i according to (2.57),

$$X_i = \alpha_1 v_1^{(i)} + \alpha_2 v_2^{(i)} + \cdots + \alpha_{L_0} v_{L_0}^{(i)}, \tag{A.1}$$

where $v_k^{(i)}$ is the k-th element of the i-th eigenvector $\mathbf{v}_i$. For the sake of simplicity we assume binary modulation, i.e., $\alpha_k = \pm 1$ with equal probabilities. Therefore,

$$X_i^2 = \sum_{k=l} \alpha_k \alpha_l v_k^{(i)} v_l^{(i)} + \sum_{k \neq l} \alpha_k \alpha_l v_k^{(i)} v_l^{(i)} = 1 + \sum_{k \neq l} \alpha_k \alpha_l v_k^{(i)} v_l^{(i)} \tag{A.2}$$

because $\alpha_k^2 = 1$ and $\|\mathbf{v}_i\| = 1$. Similarly,

$$X_i^4 = 1 + \sum_{k \neq l} \alpha_k \alpha_l v_k^{(i)} v_l^{(i)} + \sum_{m \neq n} \alpha_m \alpha_n v_m^{(i)} v_n^{(i)} + \sum_{k \neq l} \sum_{m \neq n} \alpha_k \alpha_l \alpha_m \alpha_n v_k^{(i)} v_l^{(i)} v_m^{(i)} v_n^{(i)}. \tag{A.3}$$

If we take the expectation of (A.3), the second and third terms on the right hand side become equal to zero because data symbols are assumed to be uncorrelated and have a zero mean. Thus,

$$E\{X_i^4\} = 1 + \sum_{k \neq l} \sum_{m \neq n} E\{\alpha_k \alpha_l \alpha_m \alpha_n\} v_k^{(i)} v_l^{(i)} v_m^{(i)} v_n^{(i)} \tag{A.4}$$

in which the expectation term is non-zero only when $(k,l) = (m,n)$ or $(k,l) = (n,m)$ so that $E\{\alpha_k \alpha_l \alpha_m \alpha_n\} = E\{\alpha_k^2 \alpha_l^2\} = 1$. Finally, one can obtain

$$E\{X_i^4\} = 1 + 2 \sum_{k \neq l} \left[v_k^{(i)}\right]^2 \left[v_l^{(i)}\right]^2. \tag{A.5}$$

The above expression is the exact value for $E\{X_i^4\}$. However, we note that the summation in (A.5) increases only slightly if we include $l = k$ terms. This is due to the fact the summation of the latter terms are proportional (approximately) to L_0 while the whole summation is proportional to L_0^2. Hence, one can make the following approximation

$$E\{X_i^4\} \approx 1 + 2 \sum_{k,l} \left[v_k^{(i)}\right]^2 \left[v_l^{(i)}\right]^2 = 1 + 2 \left(\sum_k \left[v_k^{(i)}\right]^2 \right) \left(\sum_l \left[v_l^{(i)}\right]^2 \right) = 3. \tag{A.6}$$

Following a similar approach as of (A.2) to (A.5), one can obtain

$$E\{X_i^2 X_j^2\} = 1 + 2 \sum_{k \neq l} v_k^{(i)} v_k^{(j)} v_l^{(i)} v_l^{(j)}, \tag{A.7}$$

which can be approximated by

$$E\{X_i^2 X_j^2\} \approx 1 + 2 \sum_{k,l} v_k^{(i)} v_k^{(j)} v_l^{(i)} v_l^{(j)} = 1 + 2 \left(\sum_k v_k^{(i)} v_k^{(j)} \right) \left(\sum_l v_l^{(i)} v_l^{(j)} \right) = 1 \tag{A.8}$$

because the eigenvectors are orthogonal to each other. It is seen that the approximated values of (A.6) and (A.8) match with (2.60) in which we used a Gaussian random variable approximation. Nonetheless, one is able to obtain the exact entries of $\mathbf{C_{x_2}}$ based on (A.5) and (A.7).

Appendix B

Derivation of T_l

We start by assuming transmission of K "$M-1$ symbols" when the phase response length is L. The CPM phase at $t = KT_s$ when $K > L$ can be written as

$$\phi(KT_s) = 2\pi h \sum_{i=0}^{K-1} (M-1) q(KT_s - iT_s) = \pi h(M-1)(K-L+1) + 2\pi h(M-1) \sum_{l=1}^{L-1} q(lT_s), \quad \text{(B.1)}$$

where the second equality holds since $q(mT_s) = 1/2$ for $m \geq L$. Without loss of generality we assume L is odd. Additionally, we consider frequency pulses which have even symmetry around $LT_s/2$. Therefore, the second term on the right hand side of (B.1) can be expressed as

$$\begin{aligned}
\sum_{l=1}^{L-1} q(lT_s) &= \sum_{k=1}^{(L-1)/2} q(kT_s) + q((L-k)T_s) \\
&= \sum_{k=1}^{(L-1)/2} \int_0^{(L/2)T_s} g(t)dt - \int_{kT_s}^{(L/2)T_s} g(t)dt + \int_0^{(L/2)T_s} g(t)dt + \int_{(L/2)T_s}^{(L-k)T_s} g(t)dt \\
&= \sum_{k=1}^{(L-1)/2} \frac{1}{2} = \frac{L-1}{4}. \qquad \text{(B.2)}
\end{aligned}$$

The last equality is true due to the following equalities for symmetric $g(t)$,

$$\int_0^{(L/2)T_s} g(t)dt = \frac{1}{2}\int_0^{LT_s} g(t)dt = \frac{1}{4}, \tag{B.3}$$

$$\int_{kT_s}^{(L/2)T_s} g(t)dt = \int_{(L/2)T_s}^{(L-k)T_s} g(t)dt, \tag{B.4}$$

where $k < L/2$. Thus, (B.1) is simplified to

$$\phi(KT_s) = \pi h(M-1)[K - \frac{L-1}{2}]. \tag{B.5}$$

It can be shown that the above results hold for even values of L as well. It is observed that the signal phase in (B.5) is equal to the phase of a CPM signal with 1REC pulse shape, same h and data sequence at $t = (K - \frac{L-1}{2})T_s$. The latter signal is basically the approximated phase response, and hence,

$$T_l = KT_s - (K - \frac{L-1}{2})T_s = \frac{L-1}{2}T_s. \tag{B.6}$$

References

[1] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers.* Plenum Press, 1997.

[2] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory.* Prentice Hall, 1993.

[3] J. G. Proakis, *Digital Communications.* McGraw-Hill, fourth ed., 2000.

[4] T. Aulin and C. Sundberg, "Continuous phase modulation–part I: full response signaling," *IEEE Transactions on Communications*, vol. 29, pp. 196–209, Mar. 1981.

[5] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, Apr. 1967.

[6] R. Dabora, J. Goldberg, and H. Messer, "Inherent limitations in data-aided time synchronization of continuous phase-modulation signals over time-selective fading channels," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1470–1482, June 2002.

[7] Y. Jiang, F.-W. Sun, and J. S. Baras, "On the true Cramer-Rao lower bound for the DA joint estimation of carrier phase and timing offsets," in *IEEE International Conference on Communications Proceedings*, vol. 1, pp. 331–335, IEEE, 2000.

[8] Y. Jiang, F.-W. Sun, and J. S. Baras, "On the performance limits of data-aided synchronization," *IEEE Transactions on Information Theory*, vol. 49, pp. 191–203, Jan. 2003.

[9] R. Johnson, M. Jorgenson, and B. Moreland, "An algorithmic approach to preamble sequence optimization," in *IEEE Military Communications Conference Proceedings*, vol. 2, pp. 993–997, IEEE, 1999.

[10] R. Mehlan and H. Meyr, "Efficient preamble design for digital DMSK packet synchronization," in *IEEE International Conference on Communications Proceedings*, pp. 918–922, IEEE, May 1994.

[11] M. Miller, M. Harris, and D. Stephens, "An innovative synchronization preamble for UHF MILSATCOM," in *IEEE Military Communications Conference Proceedings*, vol. 2, pp. 1338–1342, IEEE, 1999.

[12] M. Rice and E. Perrins, "On the performance of estimators for burst-mode offset QPSK," in *IEEE Military Communications Conference Proceedings*, pp. 1–6, IEEE, Oct. 2009.

[13] C. Shaw and M. Rice, "Optimum training sequences for data-aided synchronization," in *IEEE Global Telecommunications Conference Proceedings*, pp. 1–6, IEEE, Dec. 2010.

[14] G. N. Tavares and L. M. Tavares, "Sequence design for data-aided estimation of synchronization parameters," *IEEE Transactions on Communications*, vol. 55, pp. 670–677, Apr. 2007.

[15] G. Tavares, L. Tavares, and A. Petrolino, "On the true Cramer-Rao lower bound for data-aided carrier-phase-independent frequency offset and symbol timing estimation," *IEEE Transactions on Communications*, vol. 58, pp. 442–447, Feb. 2010.

[16] Y.-D. Kim, J. Lim, C. Suh, and Y. Lee, "Designing training sequences for carrier frequency estimation in frequency-selective channels," *IEEE Transactions on Vehicular Technology*, vol. 55, pp. 151–157, Jan. 2006.

[17] H. Minn, X. Fu, and V. Bhargava, "Optimal periodic training signal for frequency offset estimation in frequency-selective fading channels," *IEEE Transactions on Communications*, vol. 54, pp. 1081–1096, June 2006.

[18] J. Huber and W. Liu, "Data-aided synchronization of coherent CPM-receivers," *IEEE Transactions on Communications*, vol. 40, pp. 178–189, Jan. 1992.

[19] W. Tang and E. Shwedyk, "ML estimation of symbol timing and carrier phase for CPM in Walsh signal space," *IEEE Transactions on Communications*, vol. 49, pp. 969–974, June 2001.

[20] Q. Zhao and G. Stuber, "Robust time and phase synchronization for continuous phase modulation," *IEEE Transactions on Communications*, vol. 54, pp. 1857–1869, Oct. 2006.

[21] R. Dabora, J. Goldberg, and H. Messer, "Training-based time-delay estimation for CPM signals over time-selective fading channels," *IEEE Transactions on Communications*, vol. 52, pp. 1169–1177, July 2004.

[22] M. Maiolo, G. Boccolini, and M. Luise, "Soft carrier phase recovery for continuous phase modulations," in *European Wireless Conference (EW) Proceedings*, pp. 624–630, IEEE, 2010.

[23] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, Mar. 1974.

[24] M. Nezami, "Techniques for acquiring and tracking MIL-STD-181B signals," in *IEEE Military Communications Conference Proceedings*, vol. 1, pp. 224–231, IEEE, 2002.

[25] R. Mehlan and H. Meyr, "A fully digital feedforward MSK demodulator with joint frequency offset and symbol timing estimation for burst mode mobile radio," *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 434–443, Nov. 1993.

[26] M. Morelli and U. Mengali, "Feedforward carrier frequency estimation with MSK-type signals," *IEEE Communications Letters*, vol. 2, pp. 235–237, Aug. 1998.

[27] M. Morelli and G. M. Vitetta, "Feedforward joint phase and timing estimation for MSK-type signals," *European Transactions on Telecommunications*, vol. 12, pp. 327–336, July 2001.

[28] M. Luise and R. Reggiannini, "Carrier frequency recovery in all-digital modems for burst-mode transmissions," *IEEE Transactions on Communications*, vol. 43, no. 2, pp. 1169–1178, 1995.

[29] M. Morelli and A. A. D'Amico, "Maximum likelihood timing and carrier synchronization in burst-mode satellite transmissions," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 1, pp. 1–8, 2007.

[30] J. Gunther and T. Moon, "Burst mode synchronization of QPSK on AWGN channels using kurtosis," *IEEE Transactions on Communications*, vol. 57, pp. 2453–2462, Aug. 2009.

[31] J. Massey, "Optimum frame synchronization," *IEEE Transactions on Communications*, vol. 20, pp. 115–119, Apr. 1972.

[32] Z. Y. Choi and Y. H. Lee, "Frame synchronization in the presence of frequency offset," *IEEE Transactions on Communications*, vol. 50, pp. 1062–1065, July 2002.

[33] Y. Koo and Y. H. Lee, "A joint maximum likelihood approach to frame synchronization in presence of frequency offset," in *IEEE International Conference on Communications Conference Proceedings*, vol. 3, pp. 1546–1550, Ieee, 2002.

[34] M. Chiani and M. Martini, "On sequential frame synchronization in AWGN channels," *IEEE Transactions on Communications*, vol. 54, no. 2, pp. 339–348, 2006.

[35] M. Chiani, A. Giorgetti, and E. Paolini, "Optimum synchronization of ternary preamble sequences in Gaussian noise," in *IEEE 5th International Symposium on Wireless Pervasive Computing Proceedings*, pp. 146–150, Ieee, 2010.

[36] J. Gansman, M. Fitz, and J. Krogmeier, "Optimum and suboptimum frame synchronization for pilot-symbol-assisted modulation," *IEEE Transactions on Communications*, vol. 45, no. 10, pp. 1327–1337, 1997.

[37] C. Herzet, K. Woradit, H. Wymeersch, and L. Vandendorpe, "Code-aided maximum-likelihood ambiguity resolution through free-energy minimization," *IEEE Transactions on Signal Processing*, vol. 58, pp. 6238–6250, Dec. 2010.

[38] H. Huh and J. Krogmeier, "A unified approach to optimum frame synchronization," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 3700–3711, Dec. 2006.

[39] H. Wymeersch, H. Steendam, H. Bruneel, and M. Moeneclaey, "Code-aided frame synchronization and phase ambiguity resolution," *IEEE Transactions on Signal Processing*, vol. 54, pp. 2747–2757, July 2006.

[40] F.-W. Sun, Y. Jiang, and L.-N. Lee, “Frame synchronization and pilot structure for second generation DVB via satellites,” *International Journal of Satellite Communications and Networking*, vol. 22, pp. 319–339, May 2004.

[41] X. Xu, H. Ma, H. Zhang, Z. Song, and Y. Cai, “Phase and timing recovery based on frame synchronization and fractional-spaced frequency domain equalization for CPM,” in *9th International Symposium on Communications and Information Technology Proceedings*, pp. 1477–1482, Ieee, Sept. 2009.

[42] M. Morelli, U. Mengali, and G. Vitetta, “Joint phase and timing recovery with CPM signals,” *IEEE Transactions on Communications*, vol. 45, pp. 867–876, July 1997.

[43] M. Geoghegan, “Challenges of implementing an iNET transceiver for the radio access network standard (RANS),” in *International Telemetering Conference Proceedings*, 2011.

[44] M. Mitchell, *An Introduction to Genetic Algorithms.* The MIT Press, 1996.

[45] C. E. Tan and I. J. Wassell, “Near-optimum training sequences for OFDM systems,” in *9th Asia-Pacific Conference on Communications Proceedings*, pp. 119–123, IEEE, 2003.

[46] T. Koike and S. Yoshida, “Genetic designing of near-optimal training sequences for spatial multiplexing transmissions,” in *The 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceeding*, vol. 1, pp. 474–478, IEEE, 2004.

[47] N. Noels, H. Wymeersch, H. Steendam, and M. Moeneclaey, “True Cramer-Rao bound for timing recovery from a bandlimited linearly modulated waveform with unknown carrier phase and frequency,” *IEEE Transactions on Communications*, vol. 52, pp. 473–483, Mar. 2004.

[48] M. Moeneclaey, “On the true and the modified Cramer-Rao bounds for the estimation of a scalar parameter in the presence of nuisance parameters,” *IEEE Transactions on Communications*, vol. 46, pp. 1536–1544, Nov. 1998.

[49] H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Part I.* John Wiley & Sons, Inc., 2001.

[50] P. Laurent, “Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP),” *IEEE Transactions on Communications*, vol. 34, pp. 150–160, Feb. 1986.

[51] A. D’Andrea, U. Mengali, and R. Reggiannini, “The modified Cramer-Rao bound and its application to synchronization problems,” *IEEE Transactions on Communications*, vol. 42, pp. 1391–1399, Feb. 1994.

[52] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* Prentice Hall, 2000.

[53] D. A. Harville, *Matrix Algebra: Excercises and Solutions.* Springer, 2001.

[54] C. B. Chang and R. W. Miller, “A modified Cramér-Rao bound and its applications (Corresp.),” *IEEE Transactions on Information Theory*, vol. 24, pp. 398–400, May 1978.

[55] Y.-l. Huang, S. Member, K.-d. Fan, and C.-c. Huang, “A fully digital noncoherent and coherent GMSK receiver architecture with joint symbol timing error and frequency offset estimation,” *IEEE Transactions on Vehicular Technology*, vol. 49, pp. 863–874, May 2000.

[56] W. Shen, M. Zhao, P. Qiu, and A. Huang, “Data aided symbol timing estimation in space-time coded CPM systems over Rayleigh fading channels,” in *IEEE 66th Vehicular Technology Conference Proceedings*, pp. 556–560, IEEE, Sept. 2007.

[57] a.N. D’Andrea, A. Ginesi, and U. Mengali, “Digital carrier frequency estimation for multilevel CPM signals,” in *IEEE International Conference on Communications Proceedings*, vol. 2, pp. 1041–1045, Ieee, 1995.

[58] P. Bianchi, P. Loubaton, and F. Sirven, “On the blind estimation of the parameters of continuous phase modulated signals,” *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 944–962, May 2005.

[59] R. Pedone, M. Villanti, A. Vanelli-Coralli, G. Corazza, and P. Mathiopoulos, “Frame synchronization in frequency uncertainty,” *IEEE Transactions on Communications*, vol. 58, pp. 1235–1246, Apr. 2010.

[60] M. Gasior and J. L. Gonzalez, “Improving FFT frequency measurement resolution by parabolic and Gaussian spectrum interpolation,” in *AIP Conference Proceedings*, vol. 732, (Knoxville, TN), pp. 276–285, AIP, 2004.

[61] M. Fitz, “Planar filtered techniques for burst mode carrier synchronization,” in *IEEE Global Telecommunications Conference Proceedings*, pp. 365–369, IEEE, 1991.

[62] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory.* Prentice Hall, 1993.

[63] J. Hu and N. Beaulieu, “Accurate simple closed-form approximations to Rayleigh sum distributions and densities,” *IEEE Communications Letters*, vol. 9, no. 2, pp. 109–111, 2005.

[64] J. Hu and N. Beaulieu, “Accurate closed-form approximations to Ricean sum distributions and densities,” *IEEE Communications Letters*, vol. 9, pp. 133–135, Feb. 2005.

[65] J. B. Anderson, T. Aulin, and C.-E. Sundberg, *Digital Phase Modulation.* New York: Plenum Press, 1986.

[66] P. Chandran and E. Perrins, “Symbol timing recovery for CPM with correlated data symbols,” *IEEE Transactions on Communications*, vol. 57, pp. 1265–1270, May 2009.

[67] P. Chandran and E. Perrins, “Decision-directed symbol timing recovery for SOQPSK,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, pp. 781–789, Apr. 2009.

[68] M. J. Dapper and T. J. Hill, “SBPSK: a robust bandwidth-efficient modulation for hard-limited channels,” in *IEEE Military Communications Conference Proceedings*, (Los Angeles, CA), pp. 458–463, 1984.

[69] “IRIG Standard 106-04: Telemetry Standards,” 2004.

[70] E. Perrins and M. Rice, “Reduced-complexity approach to iterative detection of coded SOQPSK,” *IEEE Transactions on Communications*, vol. 55, pp. 1354–1362, July 2007.

[71] E. Hosseini and E. Perrins, “Synchronization of SOQPSK-TG in burst-mode transmissions,” in *International Telemetering Conference Proceedings*, Oct. 2013.

[72] M. Rice, *Digital Communications: A Discrtete-Time Approach.* Prentice Hall, 2009.

[73] M. Geoghegan, “Optimal linear detection of SOQPSK,” in *International Telemetering Conference Proceedings*, Oct. 2002.