TECHNICAL REPORT 1909
September 2003

Multi-*h* CPM Synchronization in Military Channels Phase 2: A Simulation Framework

J. C. Allen
B. E. Wahlen

Approved for public release;
distribution is unlimited.

SSC San Diego
San Diego, CA 92152-5001

Abstract

Continuous phase modulation (CPM) is a well-established signaling technology that attains spectral efficiency by smoothing the phase signal. The spectral efficiency of CPM makes it a natural candidate for communications where bandwidth is limited, such as the non-SATCOM channels in the littoral environments. However, CPM receivers require synchronization with respect to data, timing, phase, and frequency. Research on various CPM synchronizers has been undertaken at Space and Naval Warfare Systems Command (SPAWAR). This report sets out a simulation framework that allows these synchronizers to be swapped in and out of a CPM receiver model to assess performance in a credible end-to-end RF simulation.

Contents

1 The CPM Simulation Framework **1**

2 Multi-h Signals **4**
2.1 The Transmitted Multi-h Signal 4
2.2 The Received Multi-h Signal 5
2.3 Multi-h Spectra 7
2.4 A Multi-h Receiver 11

3 Synchronizers **18**
3.1 Timing Italian Style 19
3.1.1 Symbol Timing Estimation 19
3.1.2 A Superbaud Timing Estimator 24
3.2 2-h Timing 30
3.3 Frequency Synchronization 34
3.4 Discussion 39

4 Synchronizer Performance **40**
4.1 CPM1: A Simple CPM Receiver 40
4.2 CPM1: Timing and Phase Performance 46
4.3 Designs for Timing, Phase, and Frequency 49
4.4 CPM2: Timing, Phase, and Frequency 50
4.5 CPM2: Timing, Phase, and Frequency Performance 56
4.6 Discussion 57

5 Multi-h in Multipath **59**
5.1 Harbor Multipath 59
5.2 Multi-h in the Harbor Multipath 60
5.3 Summary 62

List of Figures

1.1 Phase pulse 1REC; $T = 1$. 2
1.2 Classical multi-h CPM receiver. 3

2.1 Spectral estimates of $[h_0 \;\; h_1] = [4 \;\; 5]/16$. 8
2.2 Spectral estimates of $[h_0 \;\; h_1] = [8 \;\; 14]/16$. 9
2.3 Spectral estimates of $[h_0 \; h_1] = [12 \; 15]/16$. 10
2.4 SNR performance of CPM0—perfect synchronization. 13
2.5 E_b/N_0 performance of CPM0—perfect synchronization; $\diamond$=Premji & Taylor [31], $\square$=Mazur & Taylor [21]. 14
2.6 Measuring the orthogonality of the $[n_1 \; n_2]/16$ indices. 15
2.7 Bandwidth and Orthogonality tradeoffs for $[n_1 \; n_2]/16$. 16
2.8 SNR performance of CPM0. 17
2.9 E_b/N_0 performance of CPM0. 17

3.1 $\Lambda_4(\mathbf{x}|\widetilde{\epsilon})$ at $\tau = 0$ using $\mathbf{x} \sim$ 5, 10, 20, and 30 symbols. 20
3.2 Real and imaginary part of the CPM signal in noise. 21
3.3 Robustness of the $\Lambda_4(\mathbf{x}|\epsilon)$ estimator. 22
3.4 Relating ϵ to its estimate $\widehat{\epsilon}$. 23
3.5 CMP signal in noise. 24
3.6 Symbol timing estimate. 25
3.7 Superbaud estimate. 25
3.8 CMP signal in noise; $\widetilde{\eta} = 1$. 26
3.9 Symbol timing estimate; $\widetilde{\eta} = 1$. 27
3.10 Superbaud estimate; $\widetilde{\eta} = 1$. 27
3.11 CMP signal in noise; $\widetilde{\eta} = 1$; 20 symbols. 28
3.12 Symbol timing estimate; $\widetilde{\eta} = 1$; 20 symbols. 29
3.13 Superbaud estimate; $\widetilde{\eta} = 1$; 20 symbols. 29
3.14 Delayed CPM and its noisy version observed over 100 symbols. 31
3.15 $\Lambda_P(\mathbf{x}|\widetilde{\tau})$ over a symbol interval. 32
3.16 $\Lambda_P(\mathbf{x}|\widetilde{\tau})$ over the observation interval. 33

3.17 Comparing the multi-h signal $s_B(t)$ and its noisy, frequency-shifted version $r_B(t)$. 35
3.18 A maximum likelihood estimate for the frequency offset. 36
3.19 A maximum likelihood estimate of the frequency offset—high SNR. . 37
3.20 A maximum likelihood estimate of the frequency offset—low SNR. . . 38

4.1 Transmitted and received CPM1 signals. 43
4.2 Transmitted and estimated CPM1 instantaneous phase and frequency. 44
4.3 CPM1 performance. 45
4.4 CPM1 SNR Performance; timing errors only. 46
4.5 CPM1 E_b/N_0 Performance; timing errors only. 47
4.6 CPM1 SNR Performance; timing and phase errors only. 48
4.7 CPM1 E_b/N_0 Performance; timing and phase errors only. 48
4.8 Transmitted and received CPM signals. 52
4.9 Transmitted and CPM2 estimate of the instantaneous phase and instantaneous frequency. 53
4.10 CPM2 spectra. 54
4.11 CPM2 performance. 55
4.12 CPM2 performance using only 100 symbols. 56
4.13 CPM2 performance using 200 symbols. 57
4.14 CPM2 performance using 500 symbols. 58

5.1 CPM in Harbor multipath. 61
5.2 CPM2 performance in Harbor multipath. 62

List of Tables

2.1 Sampling notation. 6
2.2 Optimal $[h_0 \ h_1]$ indices ranked by orthogonality (left-to-right corresponds to least-to-most orthogonality). 15

3.1 CPM Synchronization Literature. NDA=Nondata Aided. ML=maximum likelihood. 39

5.1 Two-Path Harbor Model [3]. 60
5.2 Multipath CPM Literature; NDA:=Nondata-Aided; DCD:=Data and Channel Demodulation; TDL:=Tapped Delay Line; DFSE:=Decision Feedback Sequence Equalization. 63

1

The CPM Simulation Framework

Continuous phase modulation (CPM) is a mature signaling technology that attains spectral efficiency by smoothing the phase signal. The tradeoff is that such a "slippery" signal can be hard to decode. From Sklar [34, page 631]:

> The bandwidth efficency of CPM is obtained by increasing the smoothness of the waveforms in the time domain. ... However, this smoothness in the time domain also tends to eliminate the symbol transition features upon which many symbol synchronization schemes depend.

In particular, CPM signal processing requires that the signal be synchronized with respect to

- timing
- frequency
- phase

before decoding is attempted. The (complex baseband) CPM signal has the form [34, Section 10.2.3]:

$$s_B(t) = \exp(j\psi_B(t)),$$

where the phase

$$\psi_B(t) = 2\pi \sum_{k=-\infty}^{\infty} h\alpha_k q(t - kT)$$

carries the information in the α_k's. These symbols modulate the continuous phase pulse $q(t)$. Figure 1.1 shows a specific phase pulse to illustrate the general shape of the phase pulse. Because the $q(t)$ is continuous, the phase $\psi_B(t)$ is continuous. The smoothness of $\psi_B(t)$ and the modulation index h determine the bandwidth of the

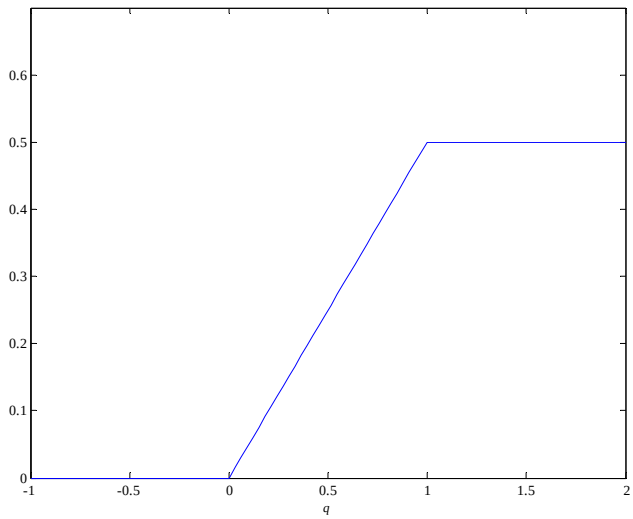

Figure 1.1: Phase pulse 1REC; $T = 1$.

CPM signal. The received CPM signal is corrupted by additive noise and demodulation errors:

$$r_B(t) = e^{j\Delta\phi} e^{j2\pi\Delta f_c t} s_B(t-\tau) + g_B(t).$$

The synchronizers must compensate for the phase $\Delta\phi$, frequency Δf_c, and timing τ offsets.

Multi-h CPM originated in 1978 paper of Anderson & Taylor [5]. Multi-h CPM generalizes single-h as

$$\psi_B(t) = 2\pi \sum_{k=-\infty}^{\infty} h_k \alpha_k q(t - kT).$$

Consequently, multi-h also requires synchronization on the *modulation indices*. Compared to single-h, multi-h codes can reduce the bandwidth [21] and increase the probability of detection [7, Section 3.2.3].

Synchronizers are required to process the CPM signals. The idea is that phase $\Delta\phi$, frequency Δf, and timing τ are compensated so that a good decoding of the phase is possible. Typically, each synchronizer is implemented with a Viterbi decoder. Several decoders may be used leading to complex CPM receivers [31]. Figure 1.2 shows such a layout for a multi-h CPM receiver [36], [21]. that handles phase, symbol timing, and superbaud timing in addition to the symbol decoding.

As expected, the literature on synchronization is large with papers filling in the matrix whose elements are indexed by synchronizers, waveforms, multi-h codes, receivers, channels, and applications. However, we are not interested in the particulars of a synchronizer but care only that it improve the end-to-end performance of the

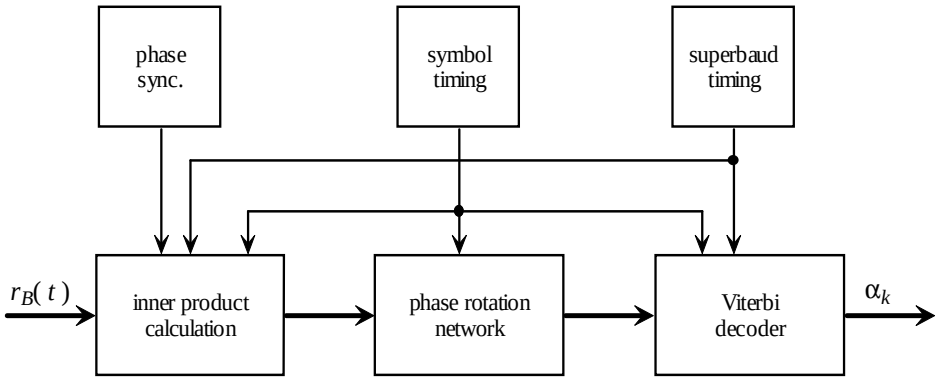

Figure 1.2: Classical multi-h CPM receiver.

CPM receiver. For example, suppose a timing synchronizer could be tweaked to get a 10% reduction in timing error. Is this significant? Only if this reduction in timing error improves performance of the CPM radio. So the goal of this report is to develop a simple CPM radio that allows us to swap out the various synchronizers.

To meet this goal, we developed of series of simple CPM receivers CMP0, CPM1, and CPM2 that permit us to "plug-n'-play" the synchronizers. Section 2 starts by reviewing the basics of the multi-h signals and sets out the very simple CPM0 receiver. Section 3 reviews selected synchronizers and reports on their performance at estimating the demodulation errors. Section 4 generalizes CPM0 to handle timing and phase in CPM1. Timing, phase, and frequency are handled in CMP2. Section 5 assess the performance of multi-h CPM in multipath. The channel considered is one of the military UHF models for a littoral zone. For the particular implementation of the multi-h, the multipath is essentially a noise source so communication is degraded. There exist several solutions (i.e., demodulate both the signal and channel). A review of these solutions and associated research topics concludes this report.

2

Multi-h Signals

This section sets the notation and nomenclature for processing the multi-h continuous phase modulation (CPM) signals. Sections 2.1 and 2.2 review the transmitted and received multi-h signal at baseband. The received signal is distorted by demodulation errors and corrupted by additive noise. The additive noise is set by the signal-to-noise ratio (SNR). However, receiver performance is typically measured by plotting the error rate against E_b/N_0 rather than the SNR. To map the SNR to E_b/N_0 requires that we know the spectra of the multi-h signals. The multi-h codes determine the multi-h spectra as illustrated in Section 2.3. These sections complete the simulation of the received multi-h signal.

Processing the multi-h signal requires synchronizing in spite of the demodulation errors. The processing requires compensating for the demodulation errors and can lead to complex CPM receivers. Because we want the end-to-end performance of the synchronizers, Section 2.4 offers a simple CPM receiver. The simplicity of this receiver allows us to swap out the various synchronizers to get the end-to-end comparisons.

2.1 The Transmitted Multi-h Signal

The baseband multi-h signal $s_B(t)$ has the form [10]:

$$s_B(t) = \exp(j\psi_B(t)); \quad \psi_B(t) = 2\pi \sum_{k=-\infty}^{\infty} h_{[k]}\alpha_k q(t - kT). \tag{2.1}$$

The *modulation indices* $\{h_0, h_1, \ldots, h_{H-1}\}$ typically cycle as

$$[k] = \mathrm{mod}(k, H).$$

The *symbol sequence* $\{\alpha_k\}$ consists of independent elements drawn from the alphabet

$$\alpha_k \in \{\pm 1, \pm 3, \ldots, \pm(M-1)\},$$

where M is even. Each symbol is transmitted over the *symbol period* T. The pulse $q(t)$ is known as the *phase pulse* [24], or the *pulse-shaping pulse* [10], or the *modulation phase response* [34]. The phase pulse $q(t)$ is the integral of the *frequency pulse* $q'(t)$ [24]:

$$q(t) = \int_{-\infty}^{t} q'(\tau) d\tau.$$

The frequency pulse is supported on the interval $[0, LT]$ and normalized so that

$$q(t) = \begin{cases} 0 & t \leq 0 \\ 1/2 & t \geq LT \end{cases},$$

where L is called the *correlation length* [34, page 633]. When $L = 1$, the signaling is called *full response* [34, page 633]. If $L > 1$, the phase pulses overlap and the signaling is called *partial response*.

2.2 The Received Multi-h Signal

The received baseband signal $r_B(t)$ is corrupted by additive noise and demodulation errors:

$$r_B(t) = e^{j\Delta\phi} e^{j2\pi\Delta f_c t} s_B(t - \tau) + g_B(t).$$

- $\Delta\phi :=$ phase error
- $\Delta f_c :=$ frequency error
- $\tau :=$ timing error

Jitters in the phase, the frequency, or the timing are not modeled. The additive noise $\{g_B(t)\}$ is zero-mean, complex-valued, wide-sense stationary bandlimited Gaussian noise with variance determined by the *signal-to-noise ratio*:

$$\text{SNR} = \frac{1}{\text{Var}[g_B]} = \frac{1}{E[|g_B(t)|^2]}.$$

We assume the received signal is sampled at N_{sps} samples per symbol. That is, the symbol period T is a multiple of the *sample period* T_s:

$$T = N_{\text{sps}} T_s.$$

The digital received signal is the vector

$$\mathbf{x} = [r_B(kT_s)] \qquad k = 0, 1, \ldots N_t - 1.$$

In this setup, the sample rate f_s is a multiple of the CPM "bandwidth"

$$f_s = \frac{N_{\text{sps}}}{T}$$

so that aliasing of $s_B(t)$ may be ignored. If the received signal is observed over N_{sym} symbols, a total of N_t samples is obtained. The total observation period is

$$N_t T_s = N_{\text{sym}} N_{\text{sps}} T_s = N_{\text{sym}} T.$$

Table 2.1 summarizes the sampling notation.

Table 2.1: Sampling notation.

N_{sym}	Number of symbols observed
N_t	Number of time samples
N_{sps}	Number of samples per symbol
T	Symbol period
T_s	Sampling period

The simulations take the noise $\{g_B(t)\}$ bandlimited to the frequency interval $[-f_s/2, f_s/2]$. Consequently, the sampled noise $\{g_B(kT_s)\}$ is a time series that is zero mean, complex-valued, IID Gaussian with variance $\text{Var}[g_B]$. As the sample rate f_s increases and the SNR is held constant, the noise in the CPM band decreases.

The ubiquitous E_b/N_0 is a normalized figure of merit for digital radios [34, Section 3.1.5]. The SNR is mapped to E_b/N_0 as [34, Eq. 3.30]:

$$\frac{E_b}{N_0} = \text{SNR} \times \frac{W}{R_b}, \tag{2.2}$$

where W and R_b denote the bandwidth (Hz) and bit rate (bits/sec), respectively. Computing the bit rate R_b is straightforward. With the alphabet $\{\pm 1, \pm 3, \ldots \pm (M-1)\}$, there are $\log_2(M)$ bits per symbol so the bit rate is [34, Eq. 9.14]:

$$R_b = \frac{\log_2(M)}{T} \quad \text{(Hz)}.$$

Computing the bandwidth is not as straightforward because "bandwidth does not have a unique definition" [37], [34, Figure 1.20]. Sundberg uses the 99% bandwidth f_{99} [37], as do Ho & Mclane [14]:

$$0.99 = \frac{\int_{-f_{99}}^{f_{99}} P_{ss}(f) df}{\int_{-\infty}^{\infty} P_{ss}(f) df}.$$

Here $P_{ss}(f)$ denotes the power spectrum of the CPM signal. We adopt f_{99} as the bandwidth of the multi-h CPM signal. Consequently, multi-h spectrum must be computed for each modulation index.

2.3 Multi-h Spectra

How the modulation indices determine the multi-h spectra is a fascinating topic that has led to a variety of computational methods [14], [7, Chapter 4], [37]. The *direct method* simply estimates the multi-h spectrum from a simulation [44]. The following figures plot these spectral estimates for selected multi-h indices. The simulations use symbol and sampling periods set to the nominal values:

$$T = 1, \quad T_s = 10.$$

The number of symbols is

$$N_{\text{sym}} = 1000.$$

Estimates of the 95% confidence intervals bound the spectrum. (These confidence intervals are computed pointwise under an asymptotic normality assumption [20].) Also reported on each plot is an estimate of the 99% bandwidth. This bandwidth is used to map the SNR to E_b/N_0.

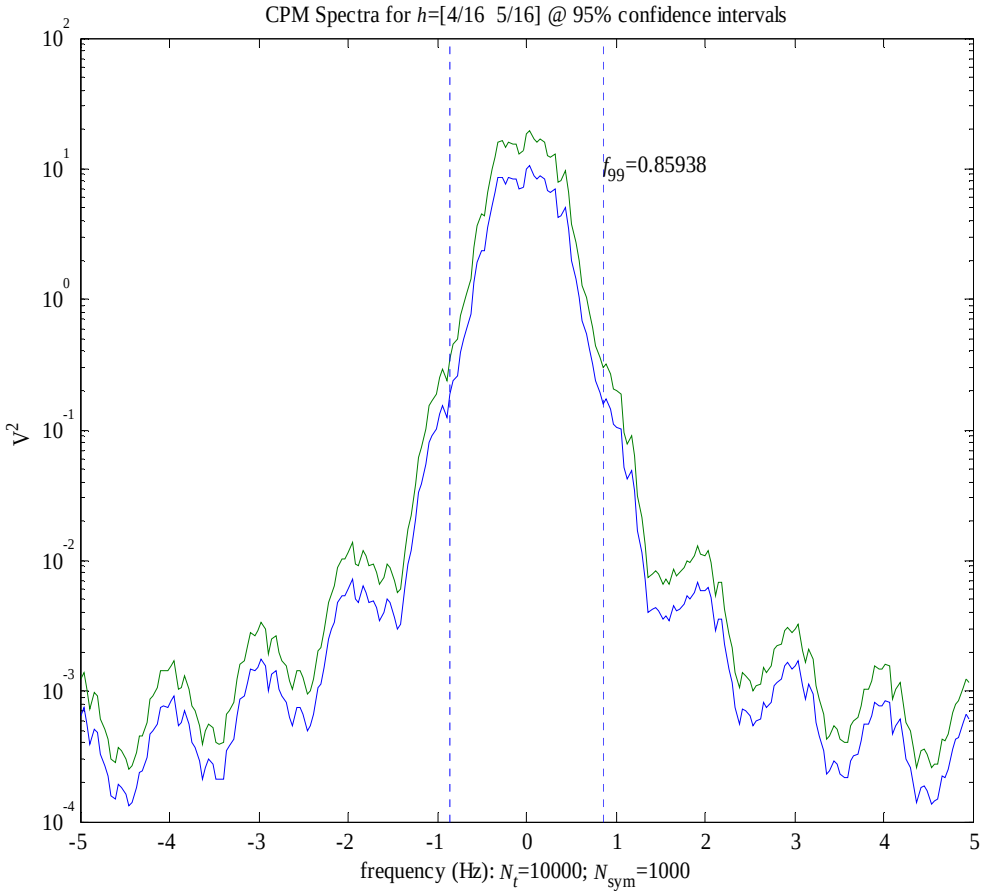

Figure 2.1: Spectral estimates of $[h_0 \;\; h_1] = [4 \;\; 5]/16$.

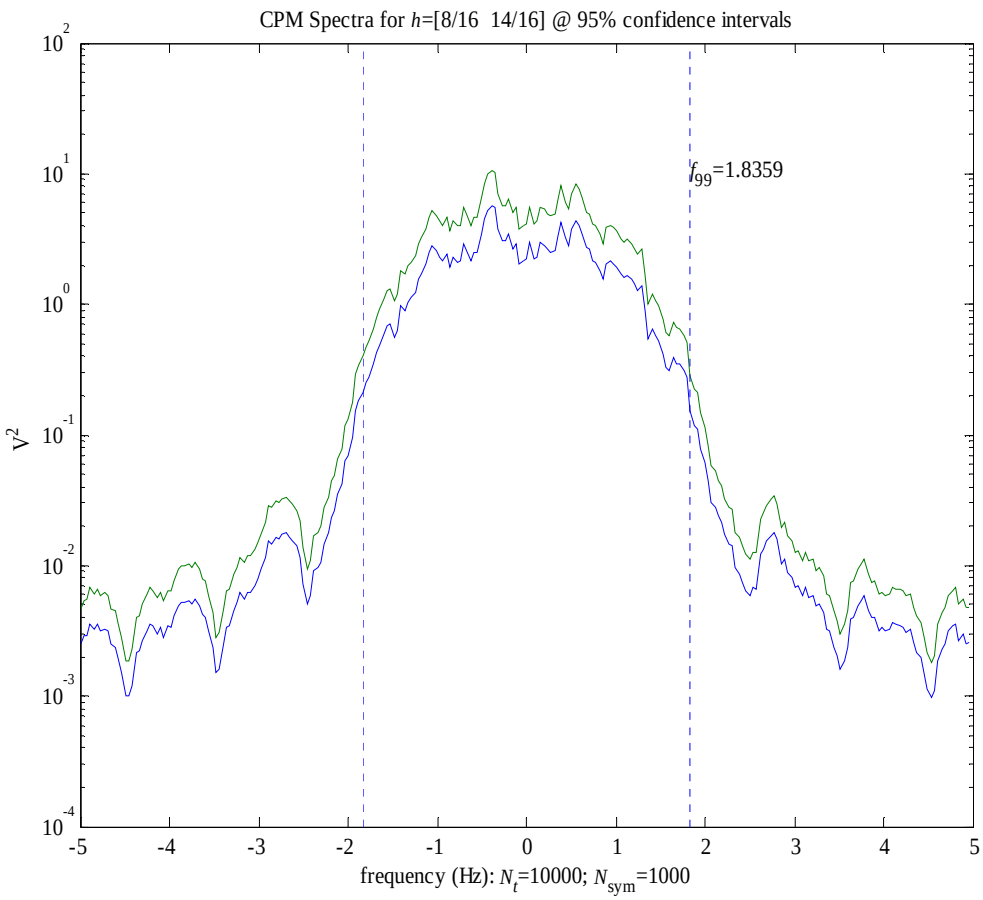

Figure 2.2: Spectral estimates of $[h_0 \;\; h_1] = [8 \;\; 14]/16$.

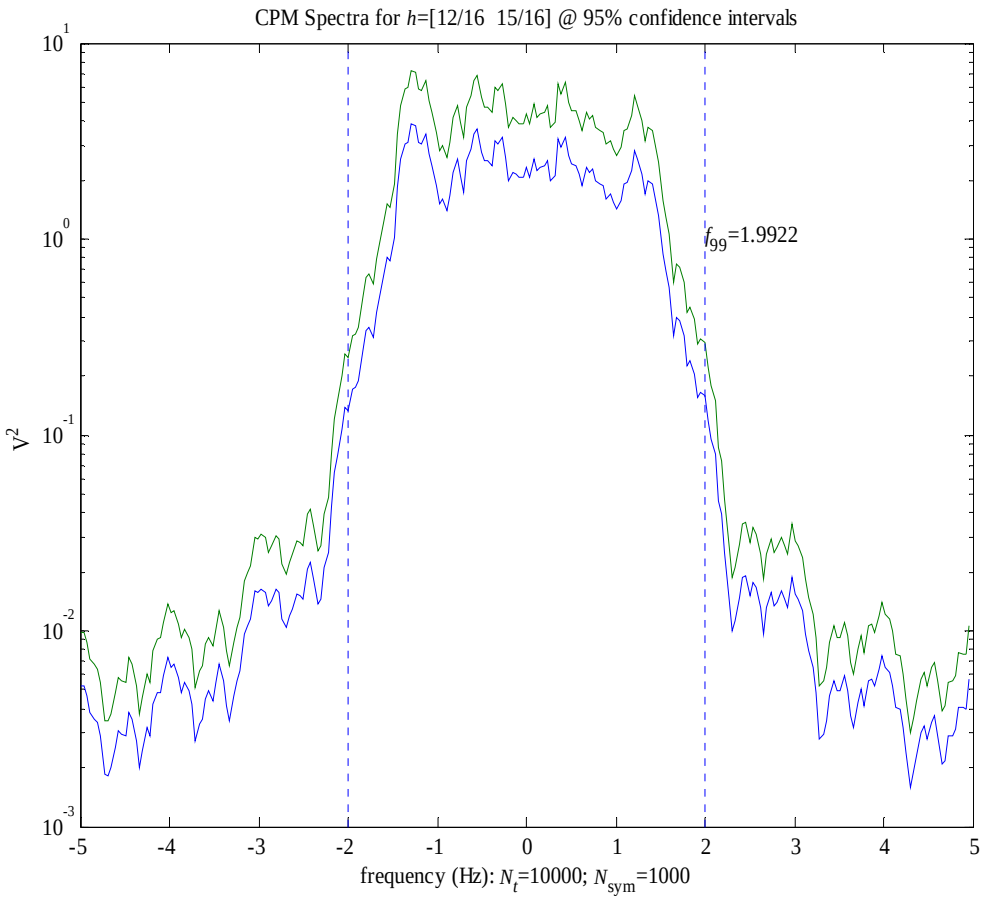

Figure 2.3: Spectral estimates of $[h_0\ h_1] = [12\ 15]/16$.

2.4 A Multi-h Receiver

Typical CPM radios are baroque designs [21], [31] with one or more Viterbi decoders devoted to each of the following tasks: symbol timing [6], superbaud timing [10], frequency estimation [4], phase estimation [23]—not to mention the Viterbi decoder for the data [19]. Instead of following these well-worn designs, we offer a simple CPM receiver whose transparent design lets us swap the synchronizers in and out, shows the effect of multipath, adapts to the Walsh receivers, and reveals particular difficulties of CPM processing.

We develop a series of CPM receivers, CMP0, CPM1, CPM2, ... of increasing capability. The starting point is a very simple CPM receiver. CPM0 assumes perfect synchronization in additive noise:

- No timing errors
- No phase errors
- No frequency errors

CPM0 assumes that N_{sym} symbols have been transmitted

$$s_B(t) = \exp(j\psi_B(t)); \quad \psi_B(t) = 2\pi \sum_{k=1}^{N_{\text{sym}}} h_{[k]}\alpha_k q(t - kT)$$

and that the received baseband signal is corrupted only by additive noise:

$$r_B(t) = s_B(t) + g_B(t).$$

The problem is to recover, and decode $\psi_B(t)$ from $r_B(t)$. CPM0 simply unwraps the phase, estimates the instantaneous frequency, and decodes by least squares:

CPM0-1 Estimate the phase of $s_B(t)$ from $r_B(t)$.

CPM0-2 Estimate the instantaneous frequency from the estimated phase.

CPM0-3 Estimate the alphabet from the instantaneous frequency.

Introduce the piecewise-linear function

$$\ell(t) := 2\pi \sum_{k=1}^{N_{\text{sym}}} f_k q(t - kT)2T.$$

With $L = 1$, the first derivative is

$$\ell'(t) := 2\pi \sum_{k=-\infty}^{\infty} f_k 1_{[0,T]}(t - kT).$$

Let $\mathcal{L}$ denote collection of all these piecewise linear functions—$\mathcal{L}$ is then a linear space of dimension N_{sym}. The CPM0 design is detailed as follows:

CPM0-1 Estimate the phase of $s_B(t)$ by phase unwrapping the received signal [20]:

$$\widehat{\psi}_B(t) := \text{unwrap}(\text{angle}(r_B(t))).$$

CPM0-2 Find the best piecewise linear approximation to the estimated phase:

$$\widehat{\ell} := \text{argmin}\{\|\widehat{\psi}_B - \ell\|_2 : \ell \in \mathcal{L}\}.$$

CPM0-3 Decode the phase. With

$$\widehat{\ell}(t) := 2\pi \sum_{k=1}^{N_{\text{sym}}} \widehat{f}_k q(t - kT)2T$$

decode by the best fit in each symbol interval:

$$\widehat{\alpha}_k := \text{argmin}\{|\widehat{f}_k/h_{[k]} - \alpha| : \alpha = \pm 1, \pm 3\}$$

The virtue of this CPM receiver is its relative simplicity—we can see exactly where the synchronizers fit. This approach is similar to the *polynomial-phase signaling* developed in the mid-1990s [8], [30], [15].

Figure 2.4 displays the performance of CPM0 for selected indices. These simulations use the symbol and sampling periods set to the nominal values:

$$T = 1, \quad T_s = 10.$$

The figure plots the probability of symbol error P_e as a function of SNR. The figure shows a considerable difference in performance as a function of the indices. However, most of the performance plots in the literature compare probability of bit error P_B to E_b/N_0. Thus, we need to convert this figure to make comparisons with the CPM literature.

Figure 2.5 displays the performance of CPM0 as a function of E_b/N_0. Equation 2.2 makes the conversion from the SNR of Figure 2.4 to E_b/N_0 but requires the bandwidth W of the multi-h signal. In Section 2.2, bandwidth was defined as $W = f_{99}$. The spectral plots of Section 2.3 show how the bandwidth is controlled by the multi-h indices and reports on our estimate of the f_{99} bandwidth. It is this estimate of f_{99} that maps the SNR to E_b/N_0. For comparison, two multi-h CPM receivers, also operating with perfect synchronization, are also plotted on Figure 2.5. These CPM receivers from the literature are measured using the bit error rate P_B. Assuming the bit errors are independent, the symbol error links to the bit error rate as:

$$\begin{aligned} P_e &= 1 - \text{Prob(no bit errors)} \\ &= 1 - \text{Prob}(1 - P_B)^{N_{\text{bits}}}, \end{aligned}$$

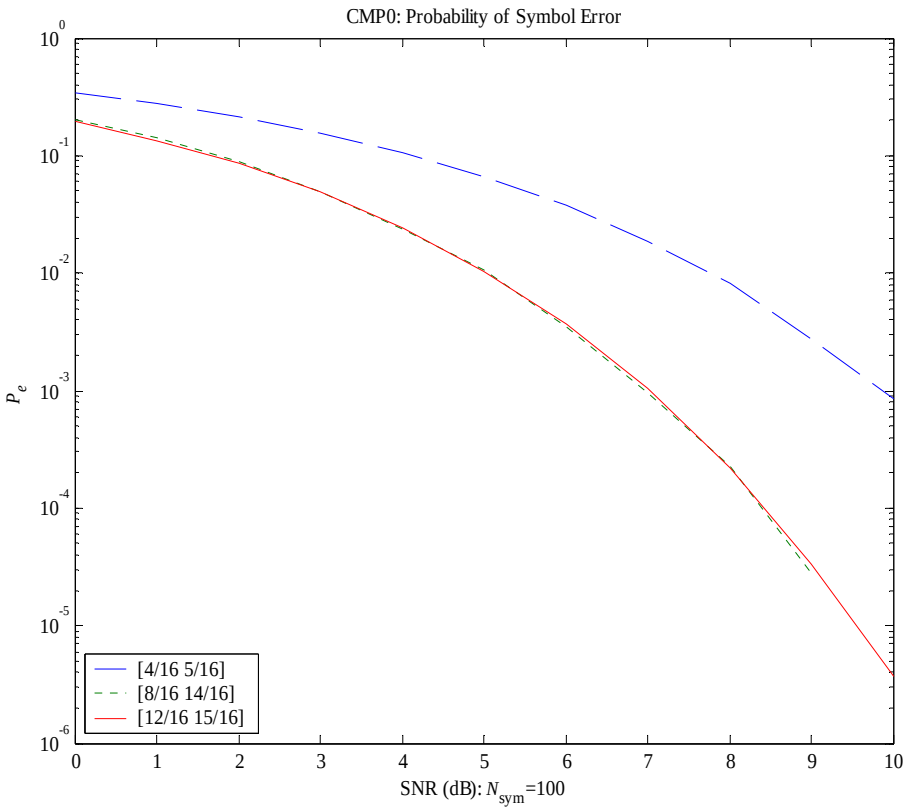

Figure 2.4: SNR performance of CPM0—perfect synchronization.

where N_{bits} is the number of bits per symbol. This is the formula we use to map between P_B and P_e. Figure 2.5 shows that the performance of Premji & Taylor's CPM receiver [31] is comparable to our simple CPM0 receiver. The extra complexity of Mazur & Taylor's receiver [21] pays a dividend of better performance.

We close this section with one way to rank the multi-h indices. This ranking follows from the Walsh receivers [38], [39]. Assume full response ($L = 1$). Each symbol and index determines a signal over a symbol interval:

$$s(\mathbf{h}; k, m; t) = \exp(j2\pi\alpha_m h_{[k]} q(t)); \quad (t \in [0, T]).$$

For matched filtering, ideal performance occurs if the collection of signals $\{s(\mathbf{h}; k, m)\}$ is orthogonal:

$$0 = \int_0^T \overline{s(\mathbf{h}; k, m; t)} s(\mathbf{h}; k', m'; t) dt \quad (k, m) \neq (k', m').$$

For sampled CPM signals, let $s(\mathbf{h}; k, m; \mathbf{t})$ denote vector of the samples of the (k, m) signal:

$$s(\mathbf{h}; k, m; \mathbf{t}) = [s(\mathbf{h}; k, m;\ nT_s) : n = 0, 1, \ldots, N_{\text{sps}} - 1].$$

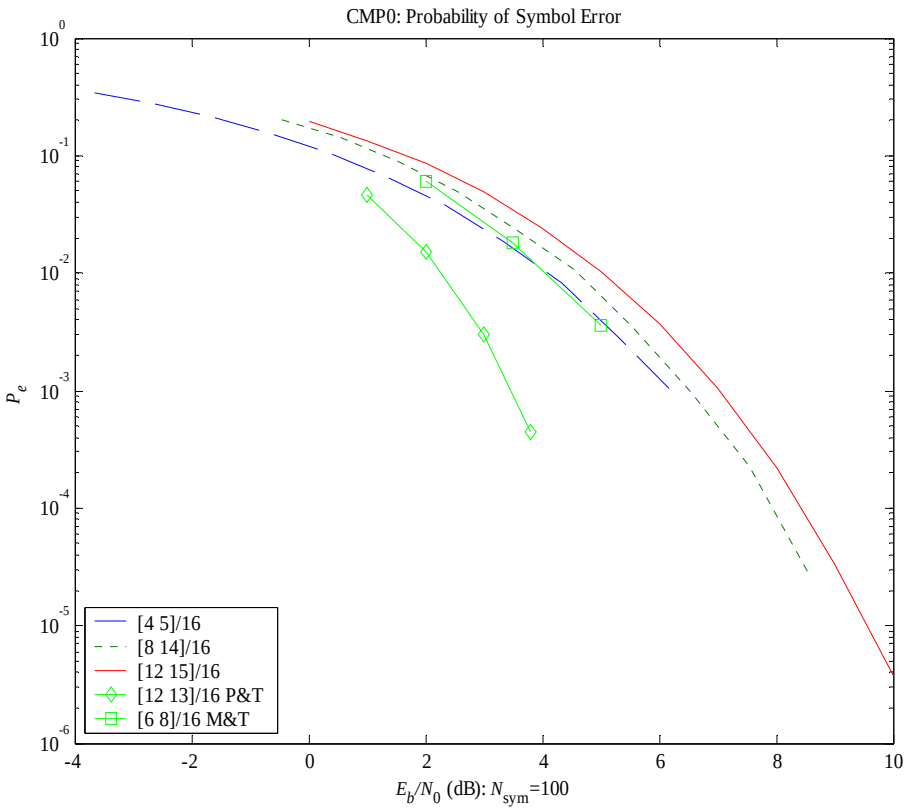

Figure 2.5: E_b/N_0 performance of CPM0—perfect synchronization; ◇=Premji & Taylor [31], □=Mazur & Taylor [21].

A measure of the orthogonality is determined by the singular-value decompositions (SVD) of the signal matrix

$$[s(\mathbf{h}; k, m; \mathbf{t}) : k = 0, 1; m = 1, 2, 3, 4].$$

Here the signal matrix is made specific to our setup. With $M = 4$ and using 2-h, there are 8 signal vectors. With $N_{\text{sps}} = 10$, the signal matrix is 10×8 and has singular values $s(1) \geq s(2) \geq \ldots \geq s(8) \geq 0$. If the 8 signal vectors were orthogonal, $s(8)/s(1) = 1$. If the signal matrix loses orthogonality, this ratio decreases. If the ratio decreases to zero, the signal matrix is rank deficient. Figure 2.6 sorts the $s(8)/s(1)$ ratio of the signal matrices as a function of the multi-h indices. The most orthogonal index is [12 15]/16 that corresponds to the maximal point on the curve. Table 2.2 reports the 11 most orthogonal indices.

However, ranking by orthogonality does not account for bandwidth. The spectral plots indicate that more orthogonality costs more bandwidth. Figure 2.7 makes this observation concrete by plotting bandwidth and the orthogonality for each $[h_0 \;\; h_1]$.

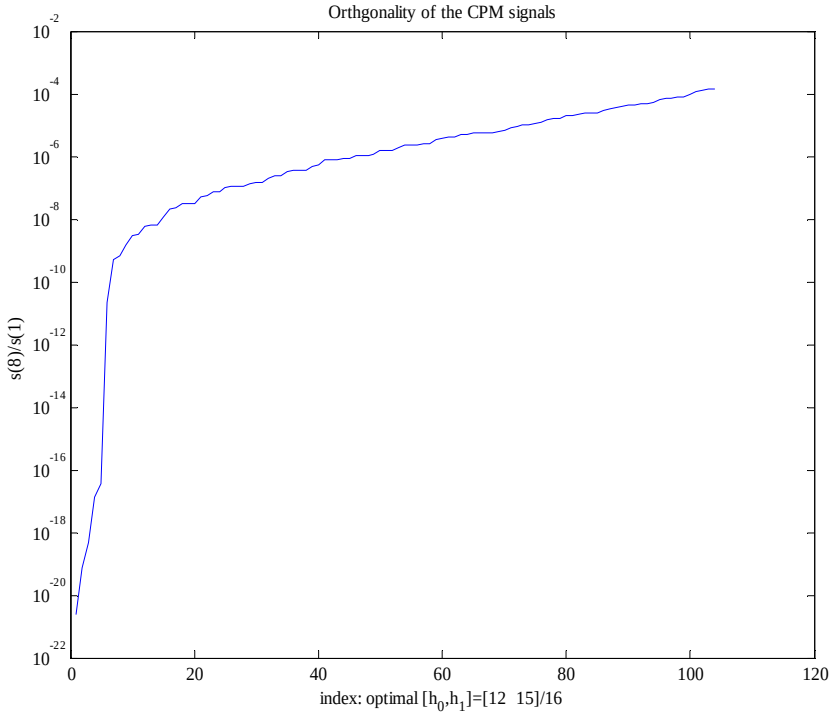

Figure 2.6: Measuring the orthogonality of the $[n_1\ n_2]/16$ indices.

A few points are listed with their indices. For example, this plot shows that $[h_0\ h_1] = [4\ 5]/16$ is near the "boundary". As such, it is one of the many "optimal" or Pareto points. This plot also suggest other candidate indices. For example, $[5\ 7]/16$ achieves almost 10 times the orthogonality over $[4\ 5]/16$ with only a modest increase in bandwidth. Thus, we expect the larger bandwidth lets $[5\ 7]/16$ outperform $[4\ 5]/16$ on the SNR plot. Moreover, we suspect that this better performance is not normalized away by a much larger bandwidth so expect $[5\ 7]/16$ outperforms $[4\ 5]/16$ on the E_b/N_0 plot.

The following figures verify these expectations. Figure 2.8 adds $[5\ 7]/16$ to the SNR performance plot and shows it lies midway between the SNR-optimal curves

Table 2.2: Optimal $[h_0\ h_1]$ indices ranked by orthogonality (left-to-right corresponds to least-to-most orthogonality).

$h_0 \times 16$	8	9	8	12	10	11	9	10	13	11	12
$h_1 \times 16$	14	14	14	14	14	14	15	15	15	15	15

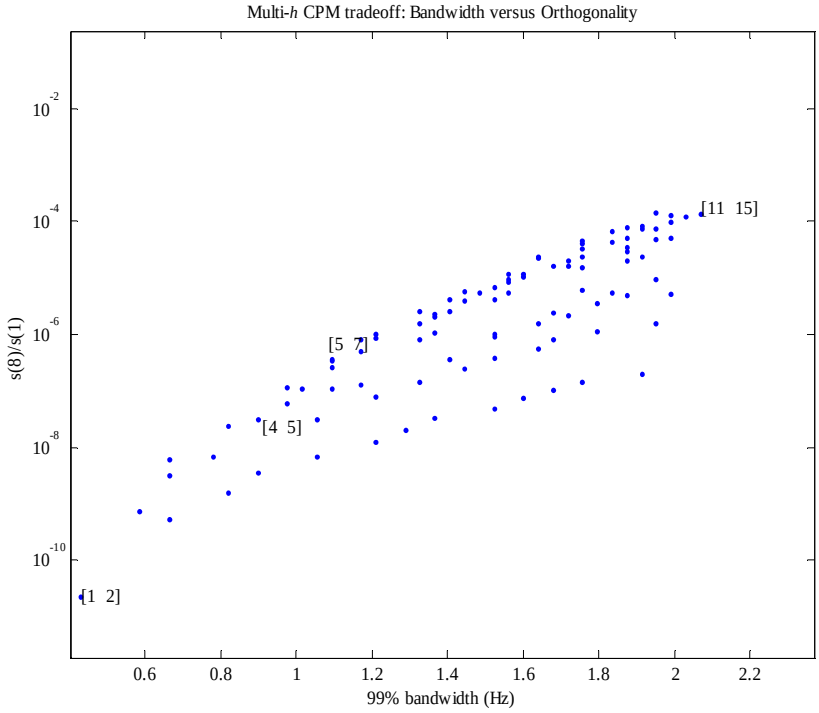

Figure 2.7: Bandwidth and Orthogonality tradeoffs for $[n_1\ n_2]/16$.

and the [4 5]/16 curve. Figure 2.9 adds [5 7]/16 to the E_b/N_0 performance plot. Compared to [4 5]/16, the slight increase in bandwidth is offset by the improved detection. However, all these error plots assume perfect synchronization—the real measure of any CPM receiver is how well it can extract the phase information when the received signal is not synchronized. Accordingly, Section 3 is devoted to synchronizers.

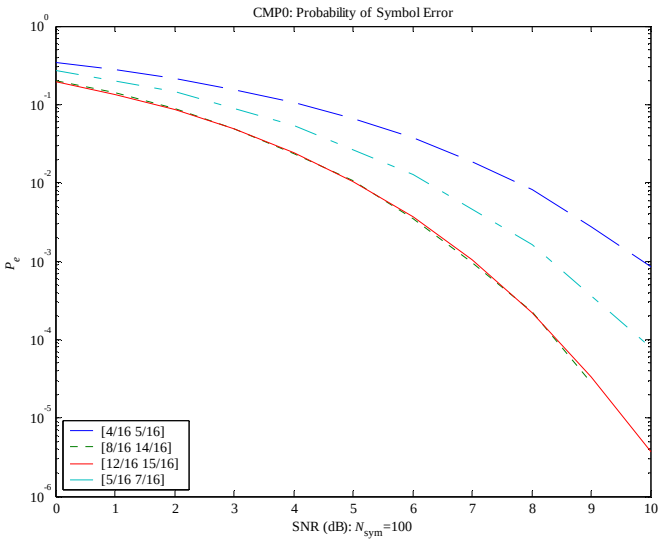

Figure 2.8: SNR performance of CPM0.

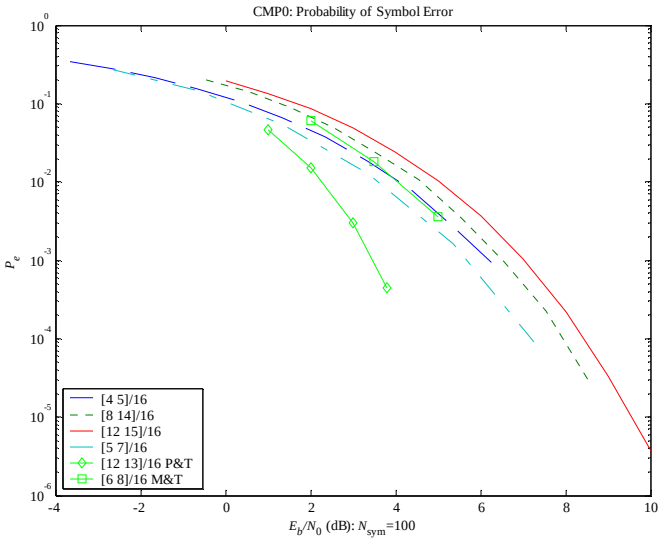

Figure 2.9: E_b/N_0 performance of CPM0.

3

Synchronizers

The received baseband CPM signal $r_B(t)$ is modeled as the transmitted baseband CPM signal $s_B(t)$ distorted by the demodulation errors—*phase* $\Delta\phi$, *frequency* Δf_c, and *timing* τ—and corrupted by additive noise:

$$r_B(t) = e^{j\Delta\phi} e^{j2\pi\Delta f_c t} s_B(t-\tau) + g_B(t).$$

A CPM synchronizer estimates $(\Delta\phi, \Delta f_c, \tau)$ to compensate for these demodulation errors. A multi-h CPM synchronizer not only estimates the delay—the superbaud timing—but also the location of the multi-h indices—the symbol timing. The most general synchronizer jointly estimates $(\Delta\phi, \Delta f_c, \tau)$. Because of mathematical difficulties, the parameters are also estimated separately: phase and data [19]; phase and timing [23]; frequency and data [24].

Synchronizers have been developed either ad hoc or a formal approach based on estimation theory. Both approaches have proven useful and, in many cases, the ad hoc synchronizers fit within an estimation theory framework. The formal approach is often based upon the *maximum likelihood* criterion of optimality—maximize a probability density function for the received samples conditioned upon the parameters to be estimated [41]. Because of compromises in both the analysis and receiver implementation, an actual synchronizer may only be an approximation to the maximum likelihood solution and consequently suboptimal.

Synchronizers are statistical estimators. A measure of "goodness" of an estimator is its variance, usually compared to the *Cramer-Rao* lower bounds [22]. We care only how well a synchronizer "works" as part of a CPM receiver to improve end-to-end performance. Consequently, the synchronizers are developed to be swapped in and out of the simple CPM receiver of Section 2.

The following sections develop a variety of synchronizers. Section 3.1 reviews an Italian approach to superbaud and symbol timing. Section 3.2 covers a timing synchronizer specialized to 2-h. Section 3.3 reviews a frequency synchronizer. The final section reviews the current literature on synchronizers and points out more advanced synchronizers for future extensions.

3.1 Timing Italian Style

The Italian school claims [10]: "coherent detection of multi-h CPM requires knowledge of the beginning of each index cycle." Their contribution is the representation of the delay as

$$\tau = (\eta + \epsilon)T.$$

- *Superbaud timing parameter*: $\eta = 0, 1, \ldots, H-1$.
- *Symbol timing parameter*: $-1/2 \leq \epsilon < 1/2$.

The delay τ is estimated by estimating the symbol timing parameter $\widehat{\epsilon}$ and then using $\widehat{\epsilon}$ to estimate η.

3.1.1 Symbol Timing Estimation

An estimator $\widehat{\epsilon}$ for symbol timing is pieced together from [10] and [6]. The maximum likelihood function is the quadratic form [10]:

$$\Lambda_4(\mathbf{x}|\widetilde{\epsilon}) = \mathbf{x}^H[F(k_1 - k_2)T_s, k_1 T_s - \widetilde{\epsilon}T)]\mathbf{x},$$

where F is given by [1, Eq. 3.2]:

$$F(\Delta t, t) := \prod_{k=-\infty}^{\infty} \frac{1}{M} \frac{\sin(2\pi M h_{[k]} q(\Delta t, t - kT))}{\sin(2\pi h_{[k]} q(\Delta t, t - kT))},$$

$$q(\Delta t, t) = q(t) - q(t - \Delta t).$$

The simulations show that $\Lambda_4(\mathbf{x}|\widetilde{\epsilon})$ tends to exhibit a single maximum so that

$$\widehat{\epsilon} = \operatorname{argmax}\{\Lambda_4(\mathbf{x}|\widetilde{\epsilon}) : \widetilde{\epsilon} \in [-1/2, 1/2)\}$$

can be referred to as "the" maximum likelihood estimate.

Figure 3.1 illustrates the performance of $\Lambda_4(\mathbf{x}|\widetilde{\epsilon})$ as the number of symbols in $\mathbf{x}$ is increased. For comparison, each estimate is scaled to the unit interval:

$$\Lambda_4(\mathbf{x}|\widetilde{\epsilon}) \mapsto \frac{\Lambda_4(\mathbf{x}|\widetilde{\epsilon}) - \min\{\Lambda_4(\mathbf{x}|\widetilde{\epsilon})\}}{\max\{\Lambda_4(\mathbf{x}|\widetilde{\epsilon})\} - \min\{\Lambda_4(\mathbf{x}|\widetilde{\epsilon})\}}.$$

The low-noise simulation shows 5 to 10 symbols barely work. To register a clean maximum, we see that 20 to 30 symbols are needed. This "need" brings us to an excellent research topic:

How many symbols are needed?

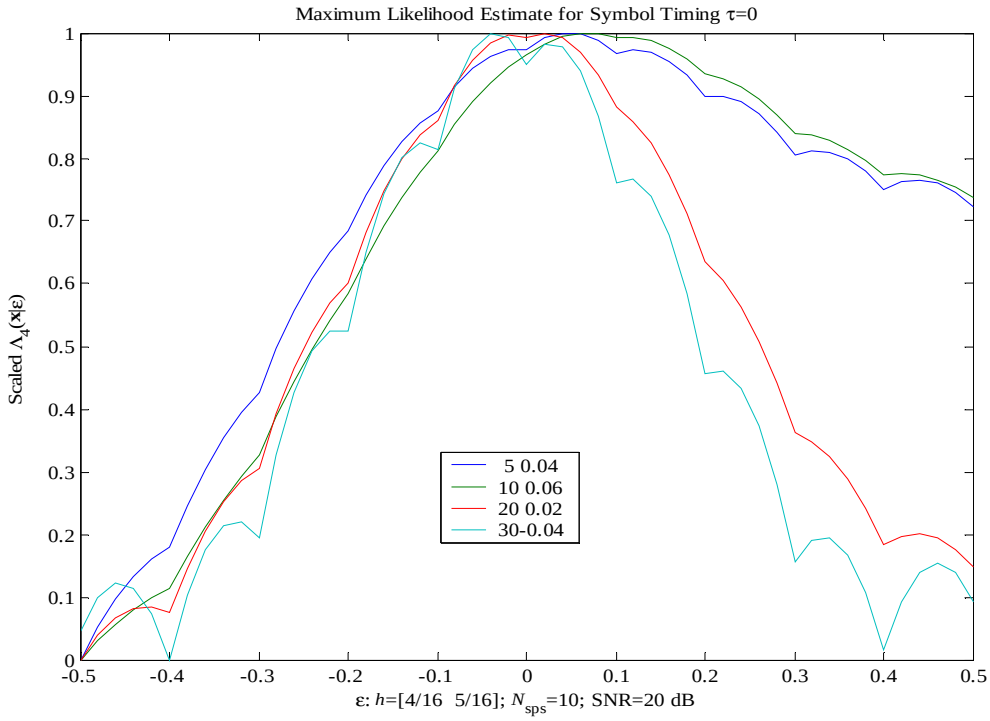

Figure 3.1: $\Lambda_4(\mathbf{x}|\widetilde{\epsilon})$ at $\tau = 0$ using $\mathbf{x} \sim$ 5, 10, 20, and 30 symbols.

This question offers many avenues of exploration. We can ask how the timing error behaves as a function of the number of symbols. We also ask what precision in $\widehat{\epsilon}$ is needed to synchronize. Thus, we are focused less on $\widehat{\epsilon}$ and more on its performance in a CPM receiver.

To show the effect of increasing noise, Figure 3.2 starts by comparing the CPM signal and its noisy version. Figure 3.3 shows the performance of the Λ_4 estimator when processing this noisy signal. What makes this plot suspect is that increasing the number of symbols does not register a cleaner maximum. To illustrate the performance over all symbol timings, Figure 3.4 displays the function:

$$\begin{bmatrix} \epsilon \\ \widetilde{\epsilon} \end{bmatrix} \mapsto \mathrm{Norm}[\Lambda_4(\mathbf{x}|\widetilde{\epsilon})],$$

where the normalization scales Λ_4 to the interval [0,1]:

$$\Lambda_4(\mathbf{x}|\widetilde{\epsilon}) \mapsto \frac{\Lambda_4(\mathbf{x}|\widetilde{\epsilon}) - \min\{\Lambda_4(\mathbf{x}|\widetilde{\epsilon})\}}{\max\{\Lambda_4(\mathbf{x}|\widetilde{\epsilon})\} - \min\{\Lambda_4(\mathbf{x}|\widetilde{\epsilon})\}}.$$

An ideal estimator would peak along the 45° reference line drawn on the image. Figure 3.4 shows that Λ_4 is a reasonable estimator for this large SNR.

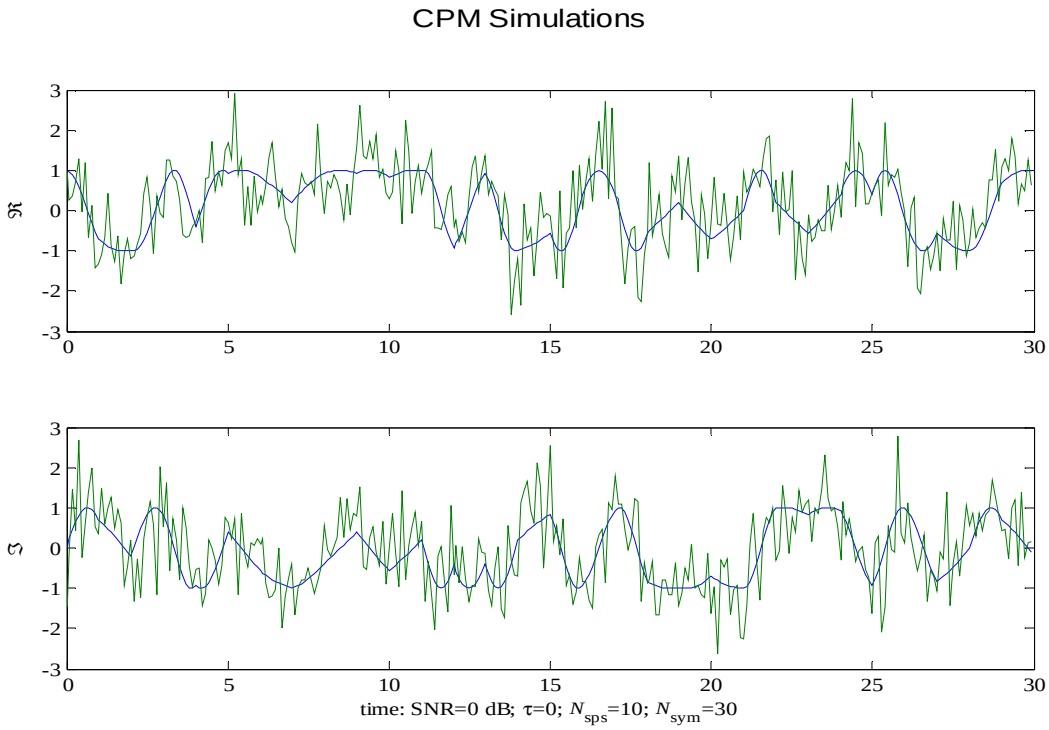

Figure 3.2: Real and imaginary part of the CPM signal in noise.

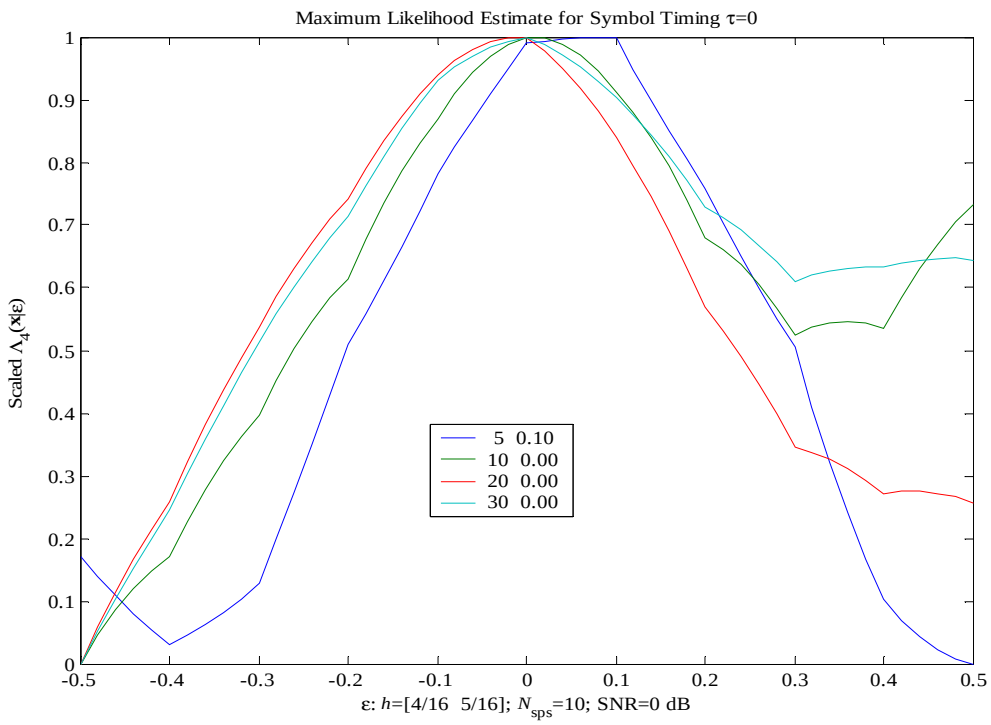

Figure 3.3: Robustness of the $\Lambda_4(\mathbf{x}|\epsilon)$ estimator.

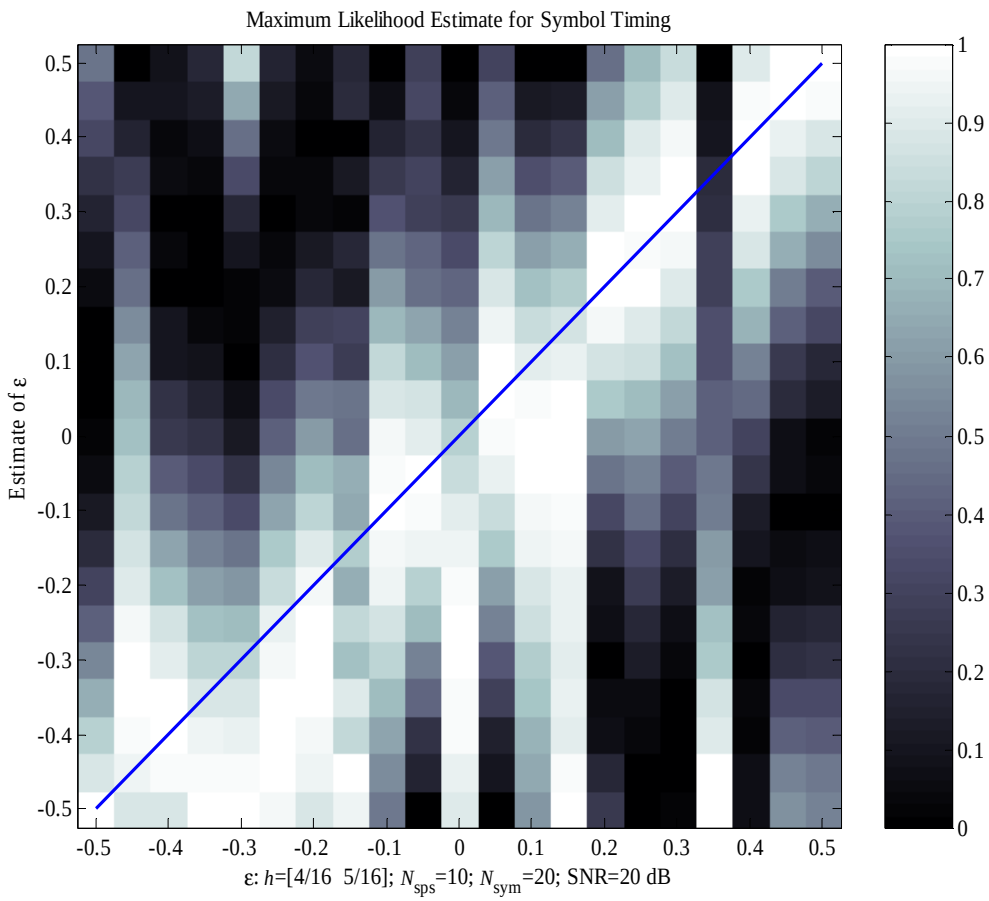

Figure 3.4: Relating ϵ to its estimate $\widehat{\epsilon}$.

3.1.2 A Superbaud Timing Estimator

The maximum likelihood function for the superbaud timing η=0, 1, ..., $H-1$ is [10, Eq. 4.1]:

$$\Lambda_4(\mathbf{x}, \widehat{\epsilon}|\widetilde{\eta}) = \mathbf{x}^H[F(k_1 - k_2)T_s, k_1 T_s - (\widetilde{\eta} + \widehat{\epsilon})T)]\mathbf{x},$$

where $\widehat{\epsilon}$ denotes the maximum likelihood estimate for ϵ.

Figure 3.5 displays the baseband received and transmitted CPM signals with the timing errors:

$$\widetilde{\tau} = (\widetilde{\eta} + \widetilde{\epsilon})T \qquad \widetilde{\eta} = 0; \;\; \widetilde{\epsilon} = -0.3.$$

Figure 3.5 shows the symbol timing estimator $\widehat{\epsilon}$ is tracking $\widetilde{\epsilon}$. Using this estimate $\widehat{\epsilon}$, Figure 3.5 presents the superbaud timing estimator where the lines are drawn between the integer values of η for clarity.

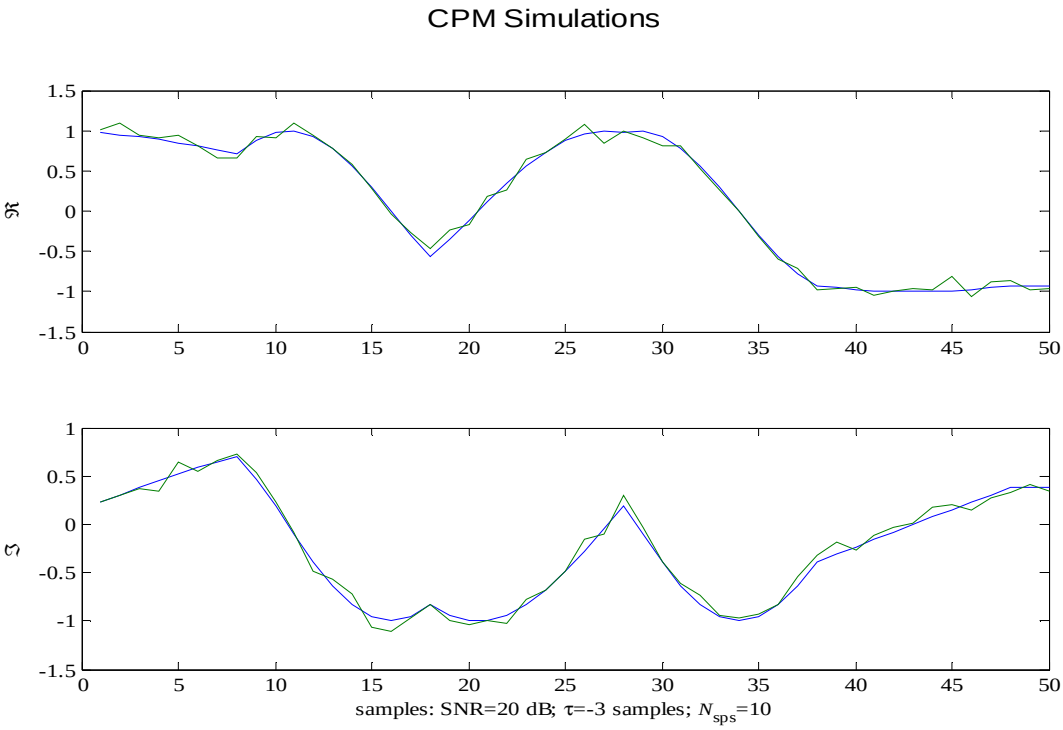

Figure 3.5: CMP signal in noise.

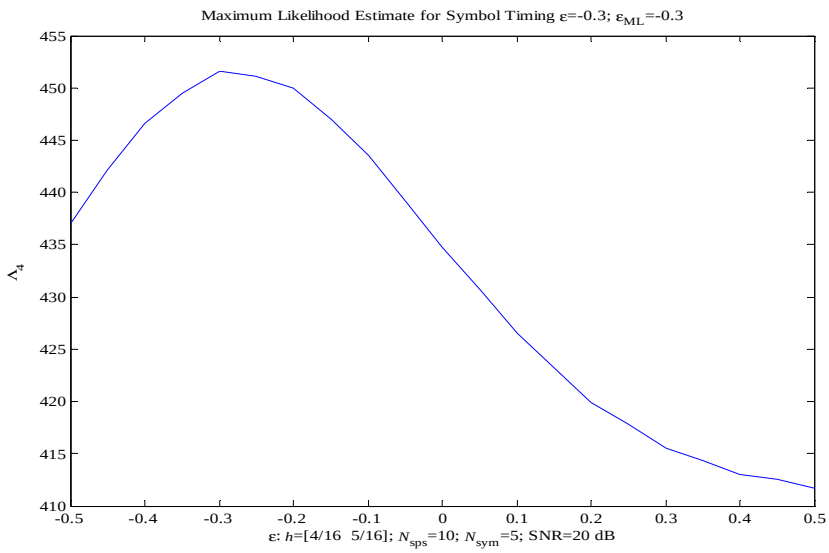

Figure 3.6: Symbol timing estimate.

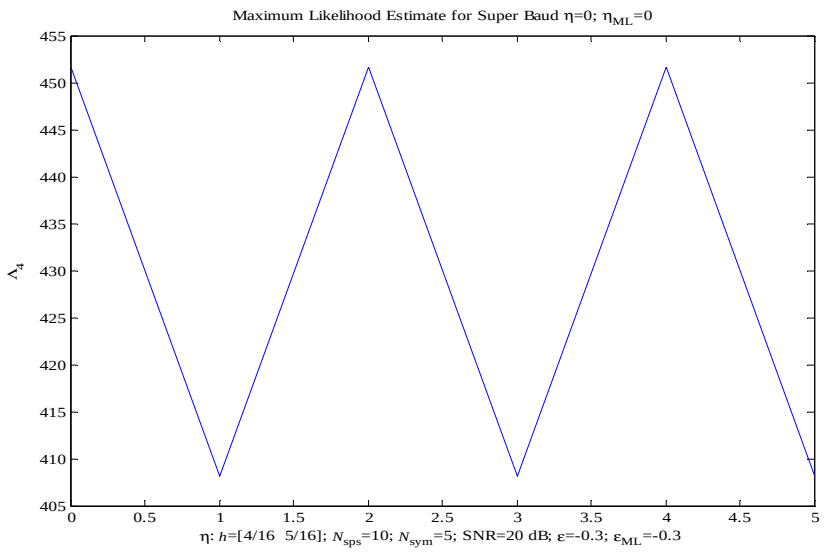

Figure 3.7: Superbaud estimate.

Encouraged by this correct result, we experimented by changing the superbaud timing to $\widetilde{\eta} = 1$. Figures 3.8, 3.9, and 3.10 show the time series, the failed symbol time estimate, and the resulting failure of the superbaud timing estimate.

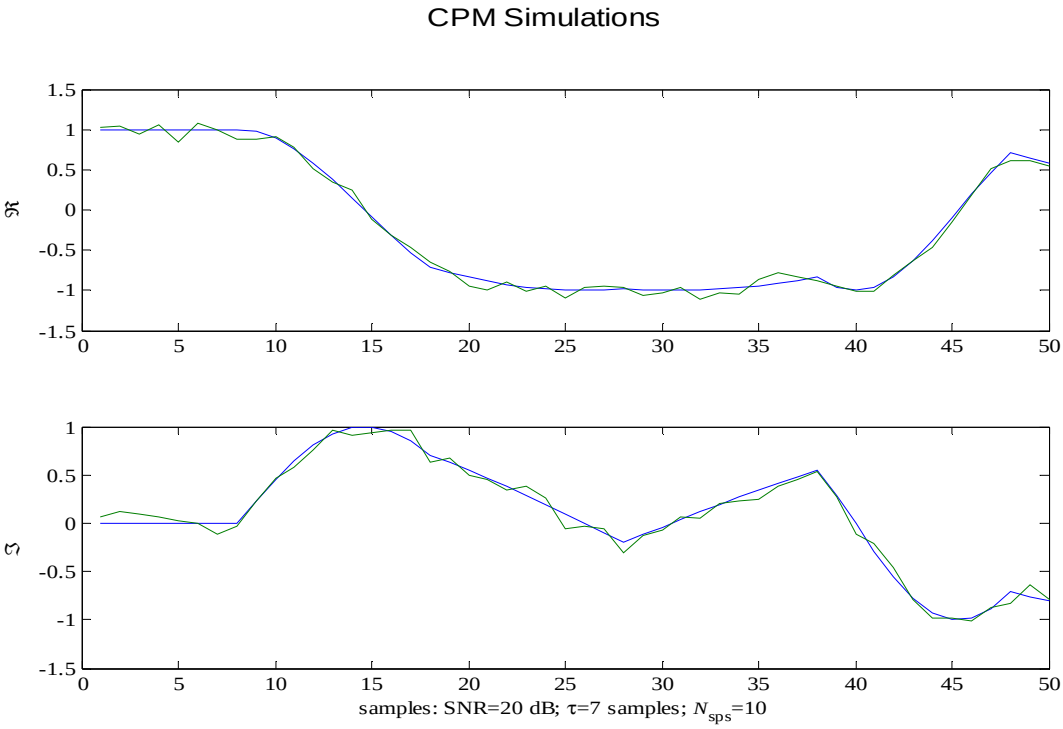

Figure 3.8: CMP signal in noise; $\widetilde{\eta} = 1$.

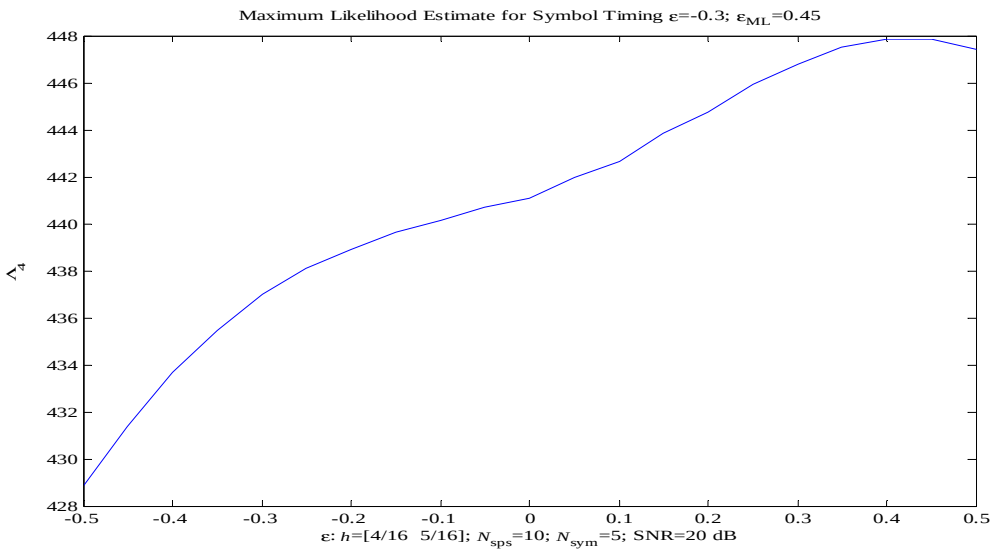

Figure 3.9: Symbol timing estimate; $\widetilde{\eta} = 1$.

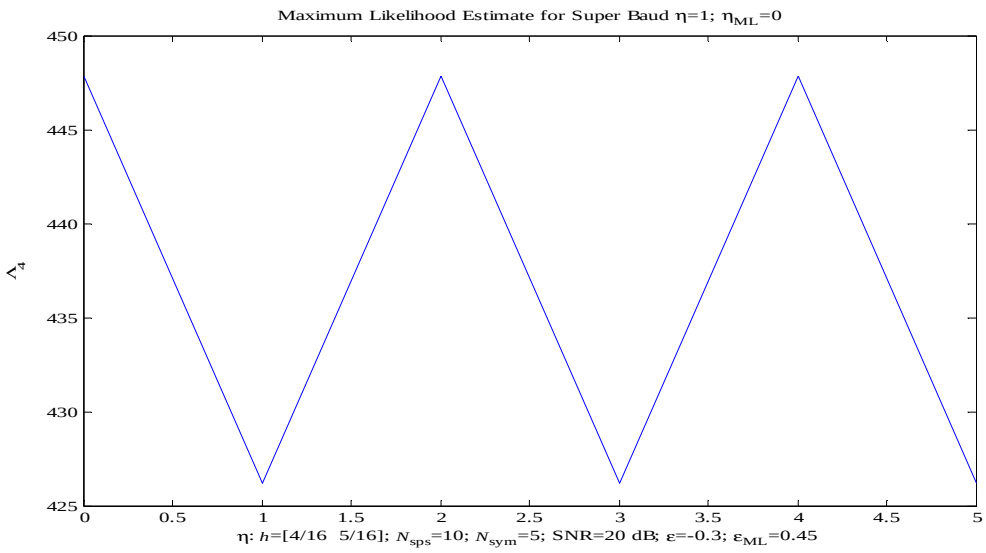

Figure 3.10: Superbaud estimate; $\widetilde{\eta} = 1$.

It is worth asking—do more symbols improve this estimator? Instead of using only 5 symbols, the number of symbols was increased to 20. Figures 3.11, 3.12, and 3.13 show the time series, the failed symbol time estimate, and the resulting failure of the superbaud timing estimate. This result may explain the running update of $\widehat{\eta}$ found in [10].

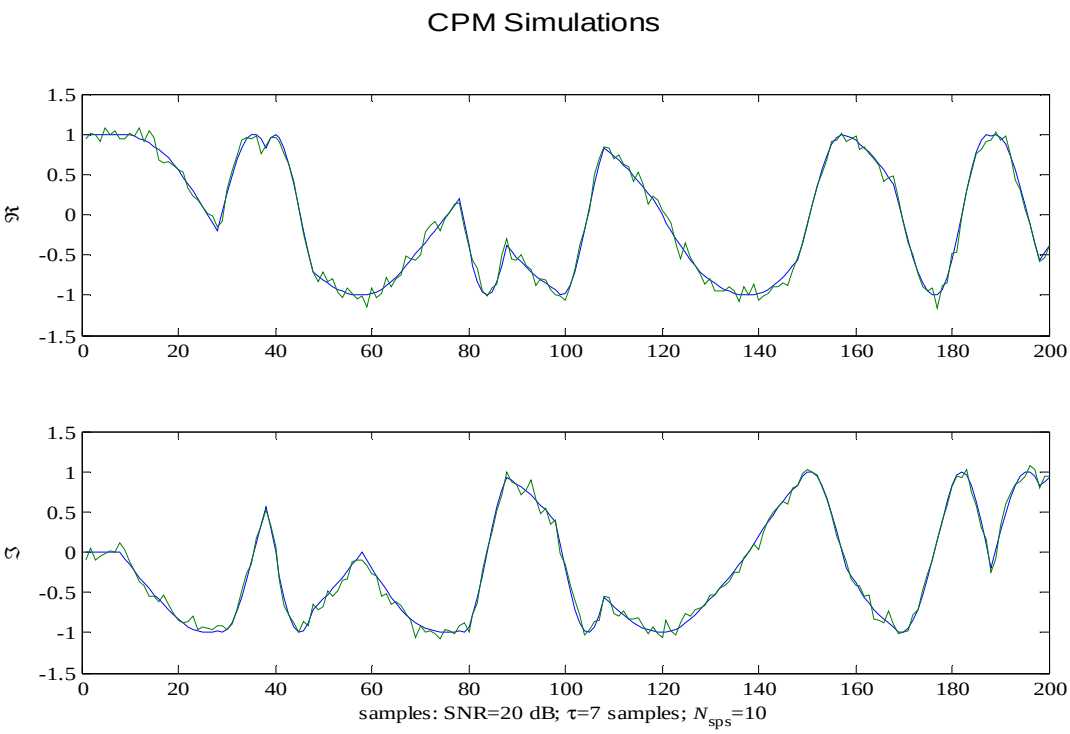

Figure 3.11: CMP signal in noise; $\widetilde{\eta} = 1$; 20 symbols.

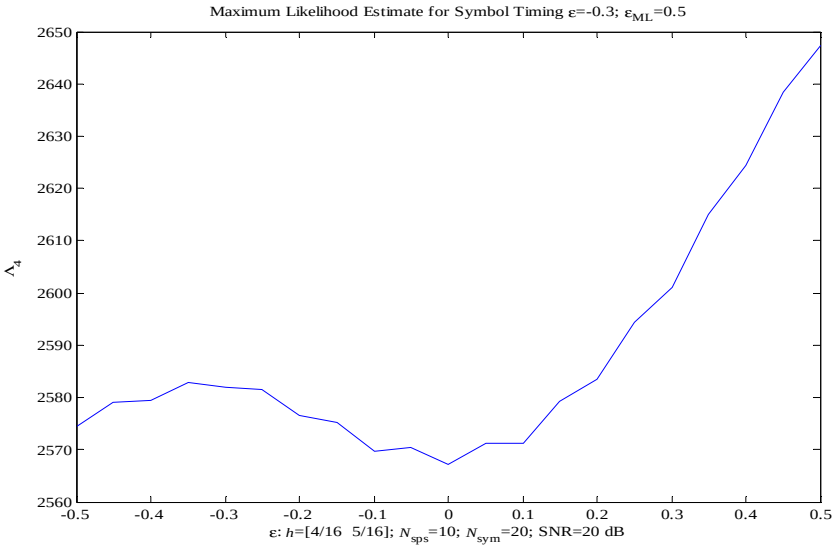

Figure 3.12: Symbol timing estimate; $\widetilde{\eta} = 1$; 20 symbols.

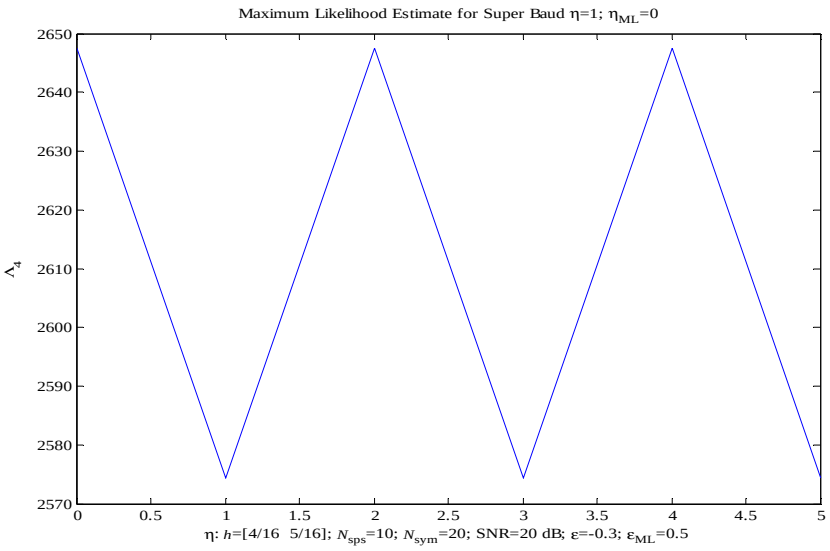

Figure 3.13: Superbaud estimate; $\widetilde{\eta} = 1$; 20 symbols.

3.2 2-h Timing

An approach specialized to 2-h delay is the maximum-likelihood estimator [29, Eq. 16]:

$$\Lambda_P(\mathbf{x}|\widetilde{\tau}) := |X|^2; \qquad X := \sum_{k=1}^{N_t} x_B(kT_s) F(h_0, h_1, kT_s - \widetilde{\tau}),$$

where the ubiquitous F function is [29, Eq. 15]:

$$\begin{aligned} F(h_0, h_1, t) \quad &:= \quad \prod_{k=0}^{N_{\text{sym}}/2-1} \cos(2\pi h_0 q(t - 2kT)) \cos(4\pi h_0 q(t - 2kT)) \\ &\times \quad \cos(2\pi h_1 q(t - (2k+1)T)) \cos(4\pi h_1 q(t - (2k+1)T)). \end{aligned}$$

The product indices are found on [29, page 17]. The "region of uncertainty" for $\widehat{\tau}$ is the interval $[0, T]$ so the maximum-likelihood estimator [29, page 5]:

$$\widehat{\tau} := \text{argmax}\{\Lambda_P(\mathbf{x}|\widetilde{\tau}) : \widetilde{\tau} \in [0, T]\}$$

is actually an estimate of the symbol timing parameter ϵ shifted by $T/2$.

The robustness of $\widehat{\tau}$ is problematic. Figure 3.14 displays a CPM signal and its noise-corrupted version. To test the estimator to its best advantage, the delay is restricted to $\tau \in [0, T]$ and, as the figure shows, the noise is very small.

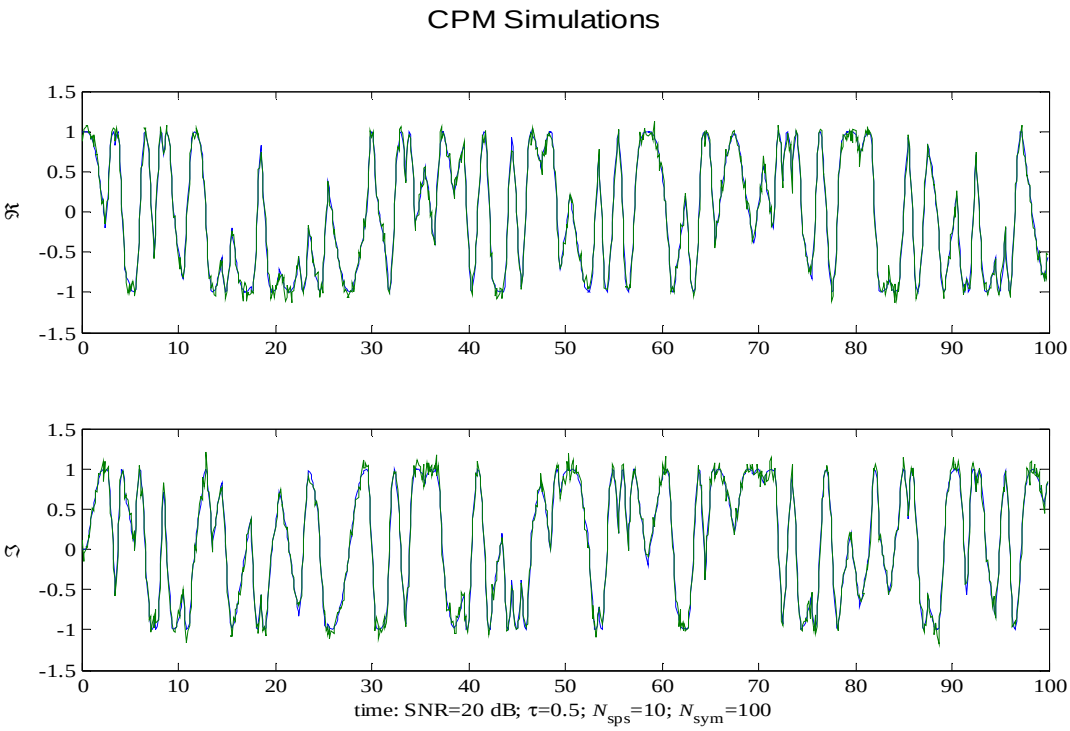

Figure 3.14: Delayed CPM and its noisy version observed over 100 symbols.

Figure 3.15 plots $\Lambda_P(\mathbf{x}|\widetilde{\tau})$ over a symbol interval $[0, 1]$. The actual value of τ is marked on the plot. The monotonic shape of $\Lambda_P(\mathbf{x}|\widetilde{\tau})$ is explained by opening up the plot.

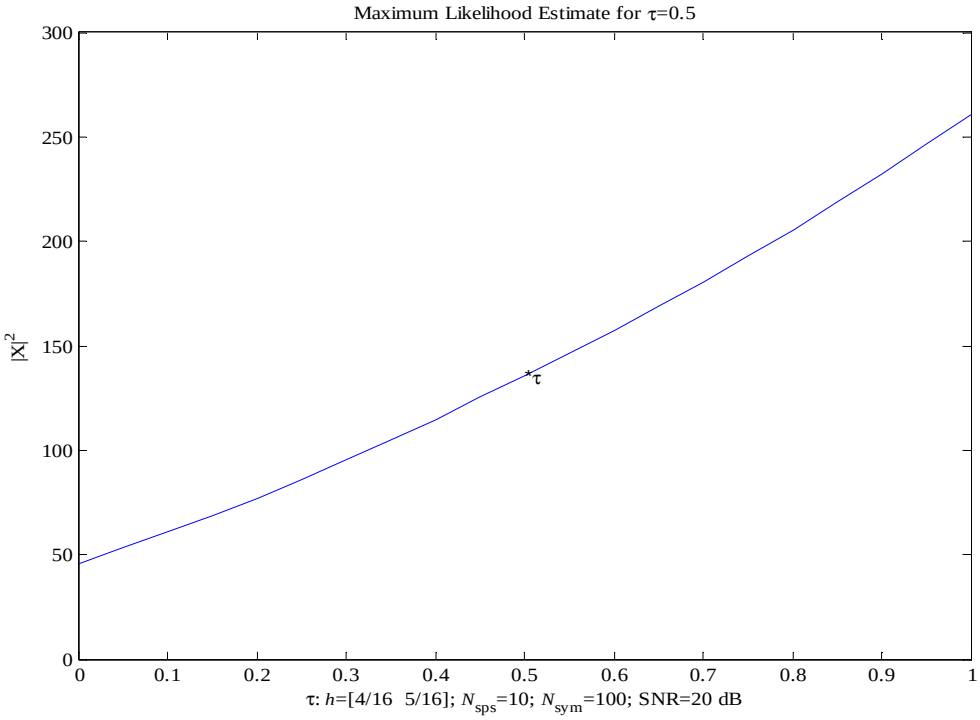

Figure 3.15: $\Lambda_P(\mathbf{x}|\widetilde{\tau})$ over a symbol interval.

Figure 3.16 plots $\Lambda_P(\mathbf{x}|\widetilde{\tau})$ over the full observation interval. The lack of periodicity precludes various foldings to $[0, T]$ to rescue this estimator.

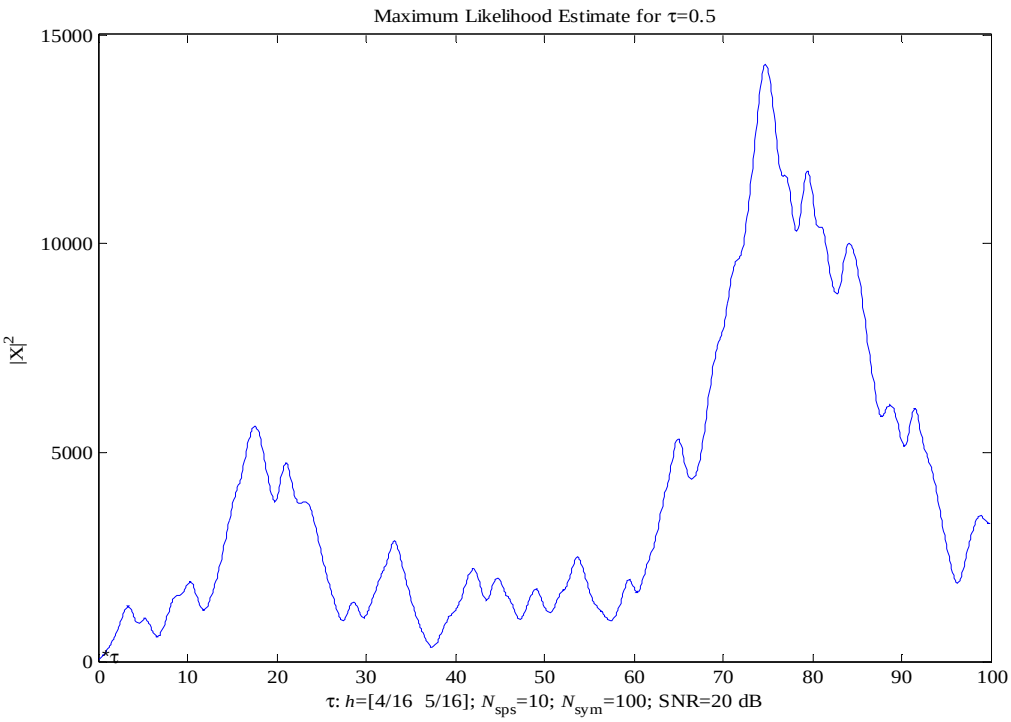

Figure 3.16: $\Lambda_P(\mathbf{x}|\widetilde{\tau})$ over the observation interval.

3.3 Frequency Synchronization

The maximum likelihood estimator for frequency is the quadratic form [28, Eq. 16]:

$$\Lambda_{\mathrm{PW}}(\mathbf{x}|f) := (\mathbf{z} \odot \mathbf{x})^T \, \mathcal{H} \, \overline{(\mathbf{z} \odot \mathbf{x})}.$$

Here $\mathbf{x}$ is the vector of time samples of the noisy CPM signal

$$\mathbf{x} := r_B(\mathbf{t}) \quad (\mathbf{t} := [kT_s]).$$

Recall that the number of time samples N_t is the product of the number of observed symbols and the number of samples per symbol: $N_t = N_{\mathrm{sym}} \times N_{\mathrm{sps}}$. The complex sinusoid is

$$\mathbf{z} := \exp(-j2\pi f \mathbf{t}).$$

The "$\odot$" denotes the Hadamard or pointwise vector product. The matrix $\mathcal{H}$ is $N_t \times N_t$ with [28, Eq. 17]:

$$\begin{aligned} \mathcal{H}(k_1, k_1) \quad := \quad & \frac{1}{2T} \int_0^{2T} \prod_{n=-\infty}^{\infty} \frac{\sin(8\pi h_0 p(k_2 T_s - t - 2nT, (k_2 - k_1)T_s))}{4\sin(2\pi h_0 p(k_2 T_s - t - 2nT, (k_2 - k_1)T_s))} \\ & \frac{\sin(8\pi h_1 p(k_2 T_s - t - 2nT, (k_2 - k_1)T_s))}{4\sin(2\pi h_1 p(k_2 T_s - t - 2nT, (k_2 - k_1)T_s))} dt, \end{aligned}$$

where [28, Eq. 18]:

$$p(t, \Delta t) := q(t) - q(t - \Delta t).$$

With some work, one can verify that $\mathcal{H}$ is real, symmetric, Toeplitz, and positive. Observe that the conjugation is twisted in this form. Taking the complex conjugate converts the twisted form into a proper quadratic form. However, we use the original form in the simulations.

The following figures illustrate this maximum likelihood estimator. The received signal is

$$r_B(t) = e^{j2\pi f_0 t} s_B(t - \tau) \; + \; g(t).$$

As in the preceding simulations, the symbol period T and sampling period T_s are set as

$$T = 1, \quad T_s = 0.1$$

so that there are 10 samples per symbol.

Figure 3.17 compares the transmitted and received multi-h signals. The upper and lower panels display the real ($\Re$) and imaginary ($\Im$) parts of these baseband signals. The noise is significant and the number of symbols is relatively small.

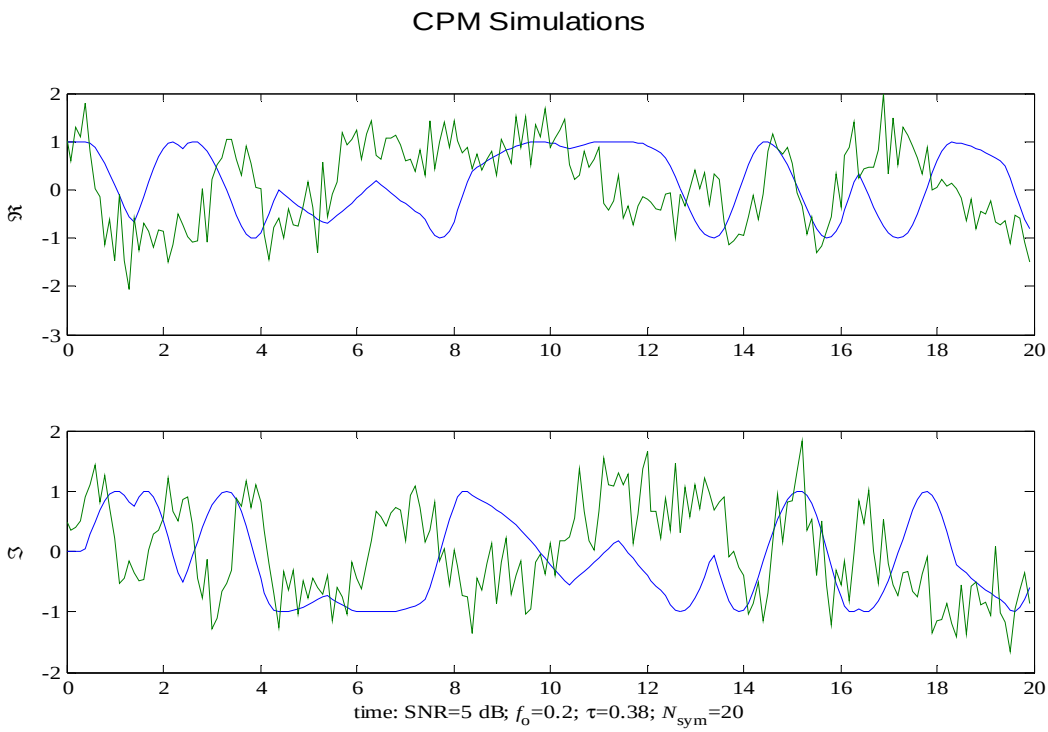

Figure 3.17: Comparing the multi-h signal $s_B(t)$ and its noisy, frequency-shifted version $r_B(t)$.

Although Figure 3.17 shows the received signal $r_B(t)$ is corrupted by noise and shifted in frequency, the maximum likelihood estimate shown in Figure 3.18 clearly registers a single maximum that is relatively close to the frequency offset. This

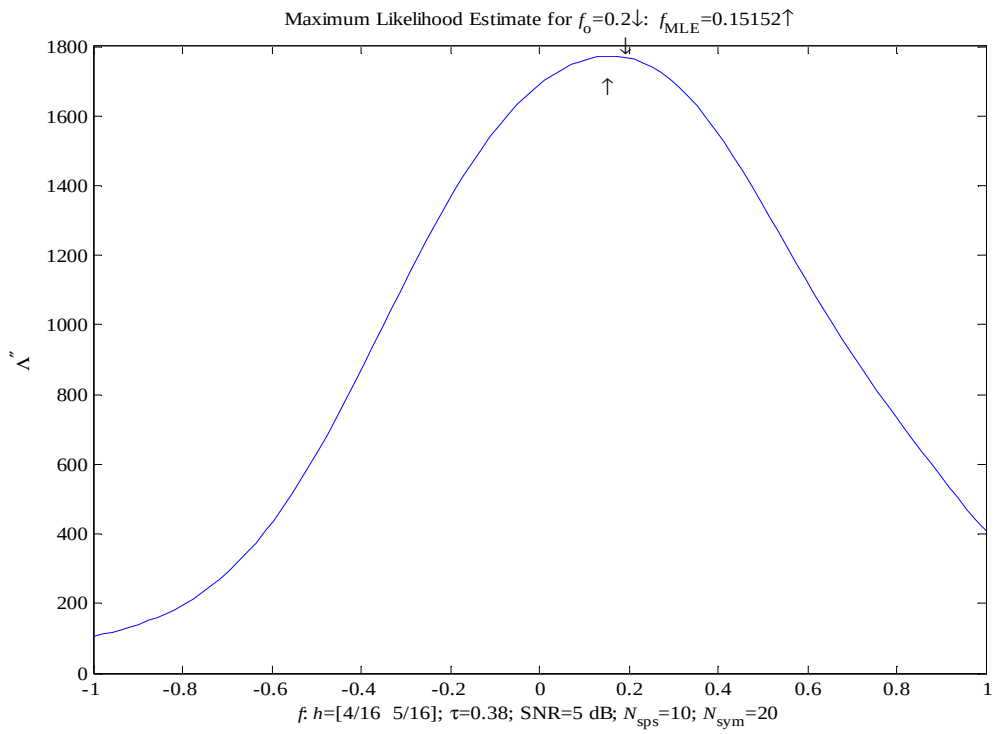

Figure 3.18: A maximum likelihood estimate for the frequency offset.

plot indicates that this frequency estimator is robust, accurate, and requires only a relatively small number of symbols.

To explore this claim, Figure 3.19 plots the maximum likelihood estimate f_{ML} as a function of the frequency offset. The solid line on this plot displays the target of the

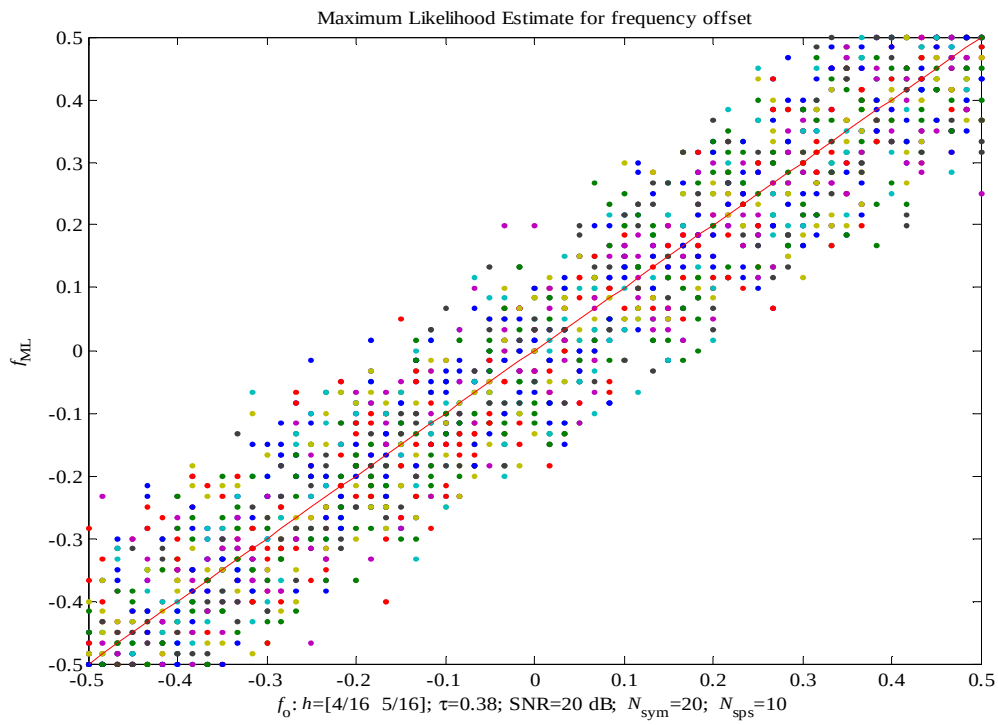

Figure 3.19: A maximum likelihood estimate of the frequency offset—high SNR.

f_{MLE}'s. Over each frequency offset, the column of dots are the f_{MLE}'s generated for several trials. The "grid" pattern is visible because the f_{MLE}'s are generated on the same grid as the frequency offsets. The variance in the f_{MLE}'s appears to be relatively constant and small over the frequency range. Thus, the claims of an accurate and robust estimator are validated—at this high SNR.

Figure 3.20 runs the same simulations using a low SNR. As before, the variance in the f_{MLE}'s appears to be relatively constant over the frequency range. Thus, this maximum likelihood estimate of the frequency offset appears to be robust and accurate on relatively short data segments.

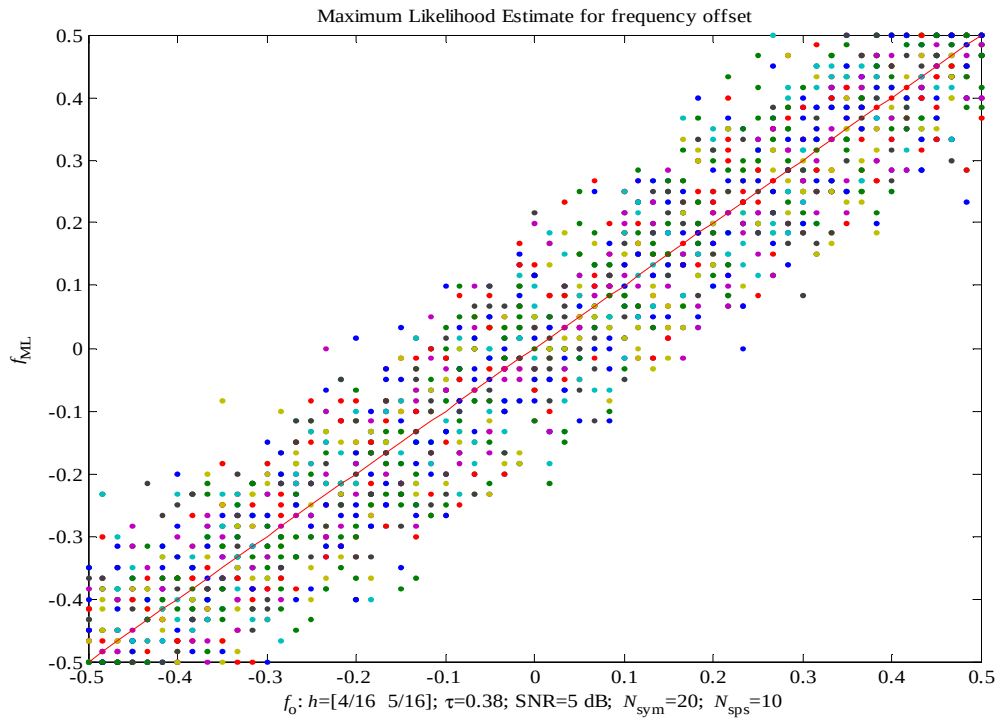

Figure 3.20: A maximum likelihood estimate of the frequency offset—low SNR.

3.4 Discussion

CPM synchronizers have been under development for decades and a considerable literature is available. A standard approach is to start from the joint maximum likelihood estimator and factor the joint estimator into several single-parameter estimators using independence assumptions. The preceding synchronizers all estimate a single parameter. For future efforts, Table 3.1 highlights a few of the joint estimators. Synchronization can also key off a preamble or specified data sequence. Such

Table 3.1: CPM Synchronization Literature. NDA=Nondata Aided. ML=maximum likelihood.

Multi-H	NDA	Channel	Parameters	Algorithm	Reference
	$\surd$	AWGN	delay+phase	ML	[6], [23]
$\surd$	$\surd$	AWGN	phase+data		[19]
$\surd$	$\surd$	AWGN	delay+phase+data		[31]
$\surd$	$\surd$	AWGN	delay+phase	ML	[15]
	$\surd$	AWGN	frequency, delay		[42]
	$\surd$	AWGN	delay+phase	ML/Walsh	[39]
	$\surd$	AWGN	frequency		[4]

synchronizers are called data-aided. The synchronizers covered in this report are nondata-aided. As the table shows, the synchronization literature is focused on the additive noise channels.

4

Synchronizer Performance

This section generalizes the simple CPM receiver (CMP0) of Section 2 to handle timing, phase, and frequency. Section 4.1 extends CPM0 to handle the phase and timing (CPM1). Section 4.2 reports on the performance of CPM1, and we conclude that phase and timing for multi-h CPM in Gaussian noise are readily handled with this simple design of CPM1. Thus, a more complex CPM receiver must beat CPM1 by several dB to justify trading complexity for performance.

Section 4.3 considers several approaches for frequency synchronization. One simple approach of estimating the frequency offset is developed in Section 4.4. This generalizes CPM1 to handle timing, phase, and frequency shifts (CPM2). Section 4.5 illustrates the performance and limitations of CPM2. If too few symbols are observed, an inaccurate estimate of the frequency shift forces a noise floor in CPM2. This observation raises excellent engineering questions:

- How sensitive is a CPM receiver to frequency errors? That is, how good does the frequency synchronizer have to be?
- What is the tradeoff between frequency synchronization and the error rate? That is, given a desired error rate, how accurate does the frequency synchronizer have to be?

Section 4.6 closes with a discussion of these questions and the usage of these CPM receivers to assess synchronizer performance.

4.1 CPM1: A Simple CPM Receiver

Section 2 compared the performance of the simple CPM receiver (CPM0) against designs in the IEEE literature—assuming perfect synchronization. The performance of CPM0 was credible. This section generalizes CPM0 to CPM1 that handles timing τ and phase errors $\Delta\phi$. Credible performance is reported in the following section. Frequency errors are treated later so we assume $\Delta f_c = 0$ for CPM1.

Assume that N_{sym} symbols have been transmitted:

$$s_B(t) = \exp(j\psi_B(t)); \quad \psi_B(t) = 2\pi \sum_{k=1}^{N_{\text{sym}}} h_{[k]}\alpha_k q(t - kT).$$

Assume the received baseband signal $r_B(t)$ is corrupted by additive noise and demodulation errors of the following form:

$$r_B(t) = e^{j\Delta\phi} e^{j2\pi\Delta f_c t} s_B(t - \tau) + g_B(t).$$

The received phase then has the following form:

$$\psi_R(t) = \Delta\phi + 2\pi\Delta f_c t + \psi_B(t - \tau) + \psi_G(t),$$

where the last term is the phase noise induced by the additive noise $\{g_B(t)\}$. The problem is to recover and synchronize $\psi_B(t)$ from $r_B(t)$.

To introduce the CPM1 design, recall that $\ell(t)$ denotes the piecewise linear function

$$\ell(t) := 2\pi \sum_{k=-\infty}^{\infty} f_k q(t - kT)2T.$$

With $L = 1$, the first derivative is

$$\ell'(t) := 2\pi \sum_{k=-\infty}^{\infty} f_k 1_{[0,T]}(t - kT).$$

Recall also that $\mathcal{L}$ denotes collection of all these piecewise linear functions. The CPM1 design is a generalization of the CPM0 design where $\ell(t - \tau)$ is fit to the phase $\psi_R(t)$ so the instantaneous frequencies f_k's can be read off from $\ell'(t)$.

CPM1-1 Estimate the phase of $s_B(t)$

1. Phase unwrap [20]:

$$\psi_R(t) := \text{unwrap}(\text{angle}(r_B(t))).$$

2. Compute the instantaneous frequency to remove the phase error:

$$\psi'_R(t) = 2\pi\Delta f_c + \psi'_B(t - \tau) + \psi'_G(t).$$

   By assumption, $\Delta f_c = 0$. If the additive noise is small,

$$\psi'_R(t) \approx \psi'_B(t - \tau).$$

3. Integrate to get an estimate of $\psi_B(t - \tau)$:

$$\widehat{\psi}_B(t - \tau) := \int_0^t \psi'_R(t')dt'.$$

CPM1-2 Synchronize:

1. Estimate the symbol timing error as

$$\{\widehat{\epsilon}, \widehat{\ell}\} := \operatorname{argmin}\{\|\widehat{\psi}_B(\circ - \tau) - \ell(\circ - \epsilon T)\|_2 : \epsilon \in [-1/2, 1/2], \ell \in \mathcal{L}\}.$$

   That is, $\widehat{\epsilon}$ is a symbol time that best explains the phase as a piecewise linear function.

2. Estimate the instantaneous frequency from the best piecewise linear minimizer ($L = 1$):

$$\widehat{f}_B(t) := \frac{1}{2\pi}\widehat{\ell}'(t - \widehat{\epsilon}T).$$

3. Estimate the superbaud error η: If the symbol timing is correct,

$$f_k \approx \alpha_k h_{[k+\eta]}$$

   so choose the delay that minimizes

$$\widehat{\eta} := \operatorname{argmin}\left\{\sum_{k=1}^{N_{\text{sym}}} \|f_k/h_{k+\eta} - \mathbf{a}\|_{-\infty} : \eta = 0, 1\right\}, \quad \mathbf{a} = \begin{bmatrix} 3 \\ 1 \\ -1 \\ -3 \end{bmatrix}.$$

CPM1-3 Decode:

$$\widehat{\alpha}_k := \operatorname{argmin}\{|f_k/h_{k+\widehat{\eta}} - \alpha| : \alpha = \pm 1, \pm 3\}$$

For orientation, the following figures illustrate CPM1 in a typical simulation. Figure 4.1 compares the transmitted (no noise) CPM signal $s_B(t)$ against the noisy, time-shifted, and phase-rotated baseband received signal $r_B(t)$. Although the noise is still relatively small, the phase does distort the received signal.

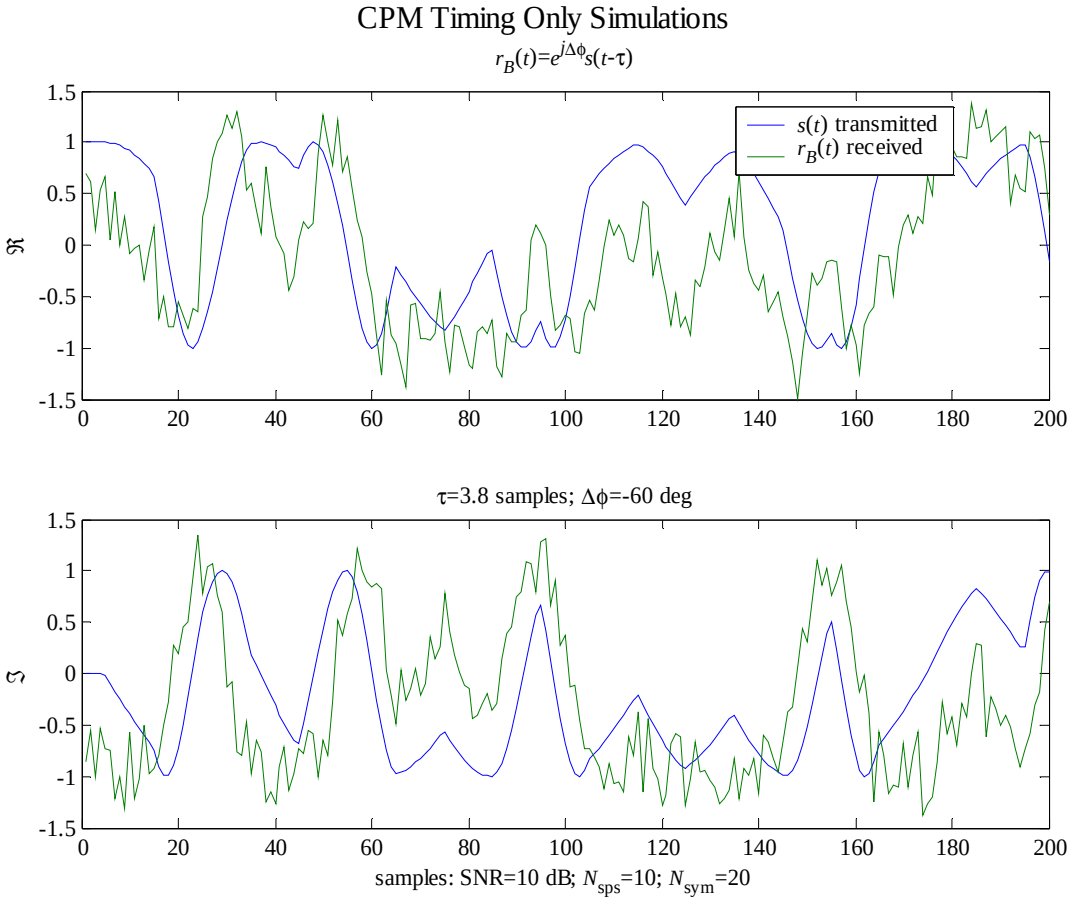

Figure 4.1: Transmitted and received CPM1 signals.

Figure 4.2 makes the corresponding comparison on the phases and instantaneous frequencies. Even though the SNR is large, and phases are close, the derivative amplifies the phase noise in the instantaneous frequency.

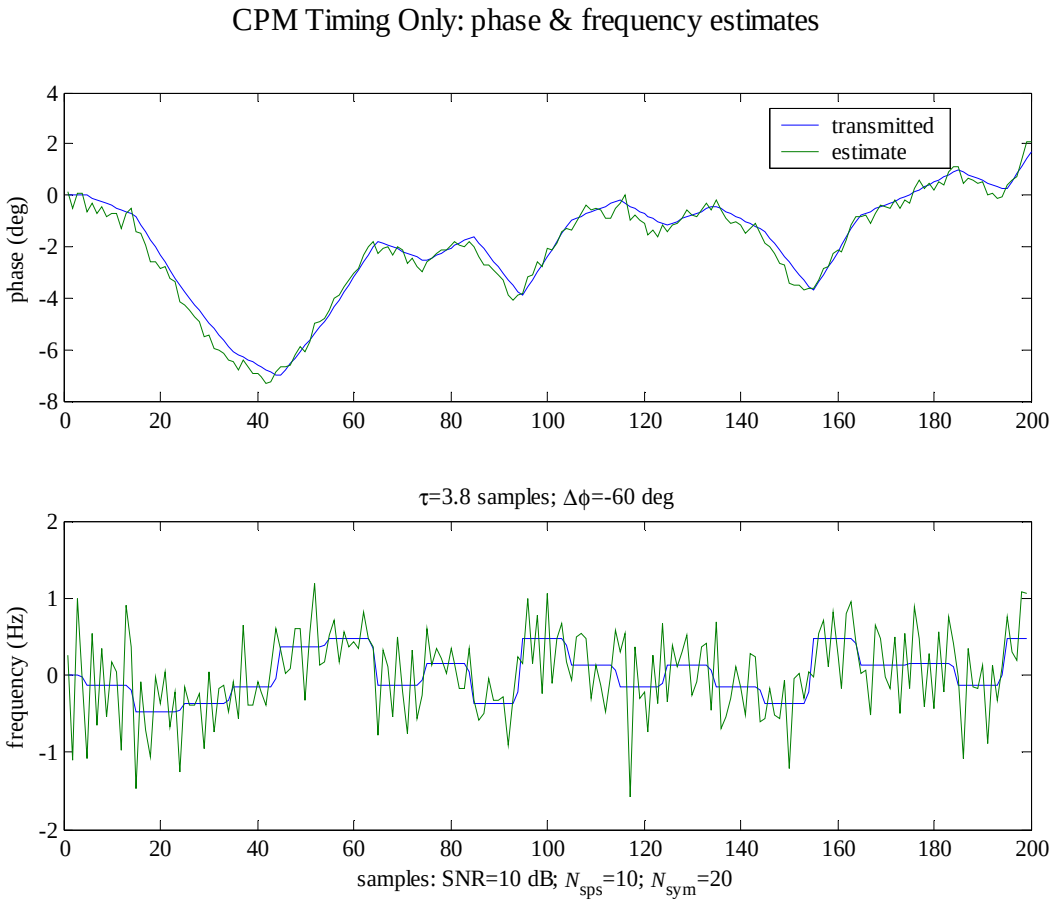

Figure 4.2: Transmitted and estimated CPM1 instantaneous phase and frequency.

Figure 4.3 shows that the symbol timing, the superbaud timing, and the decoding has been successful despite the noise in the instantaneous frequency. The step

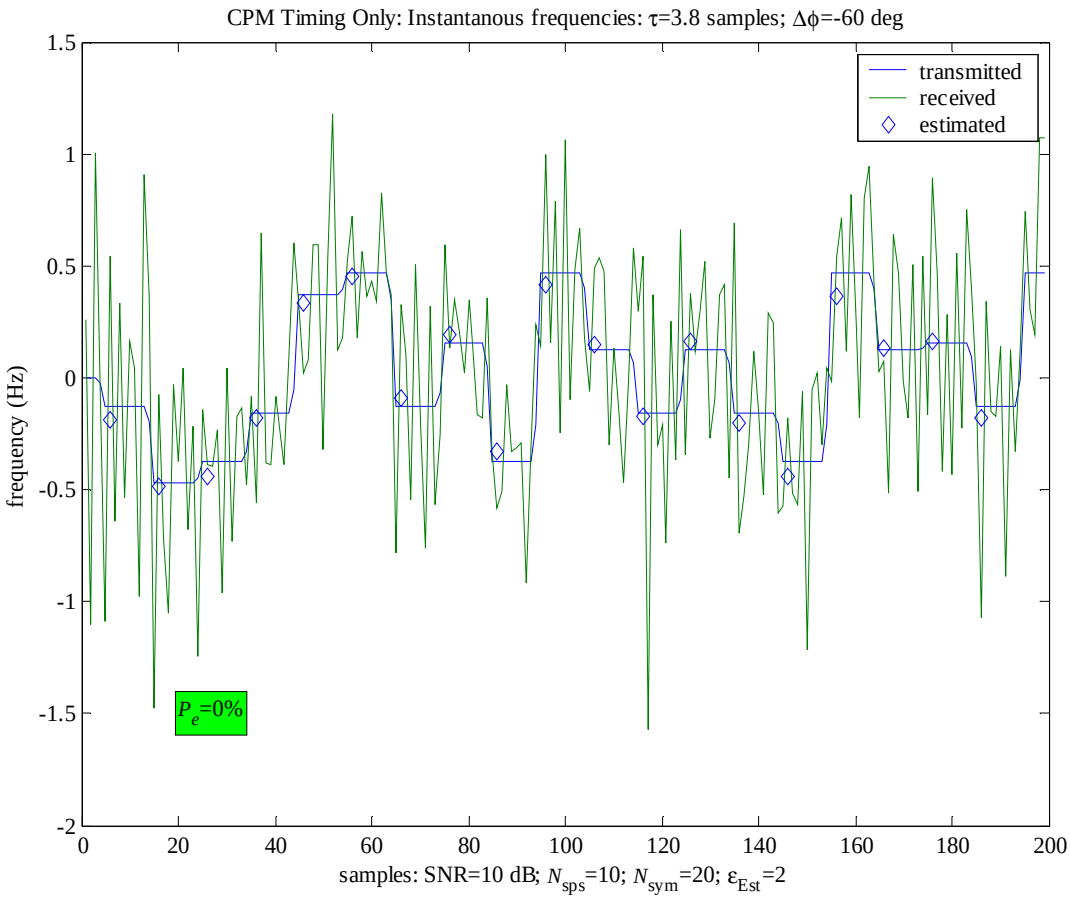

Figure 4.3: CPM1 performance.

function (blue for color viewers) plots the instantaneous frequency of the transmitted signal: $\psi'_B(t)$. The jagged (green) line is the received phase after correcting for phase, frequency, and superbaud timing. The (blue) diamonds mark the instantaneous frequency estimated from the received phase. Figure 4.3 reports that estimated probability of symbol error is zero: $P_e = 0$. From that observation, we infer that the estimate of the symbol timing did also work.

4.2 CPM1: Timing and Phase Performance

The performance of CPM1 is illustrated in the following plots. Figures 4.4 and 4.5 plot estimates of the probability of symbol error as a function of the SNR and E_b/N_0, respectively. Equation 2.2 is used to map the SNR to E_b/N_0. In both plots, the timing

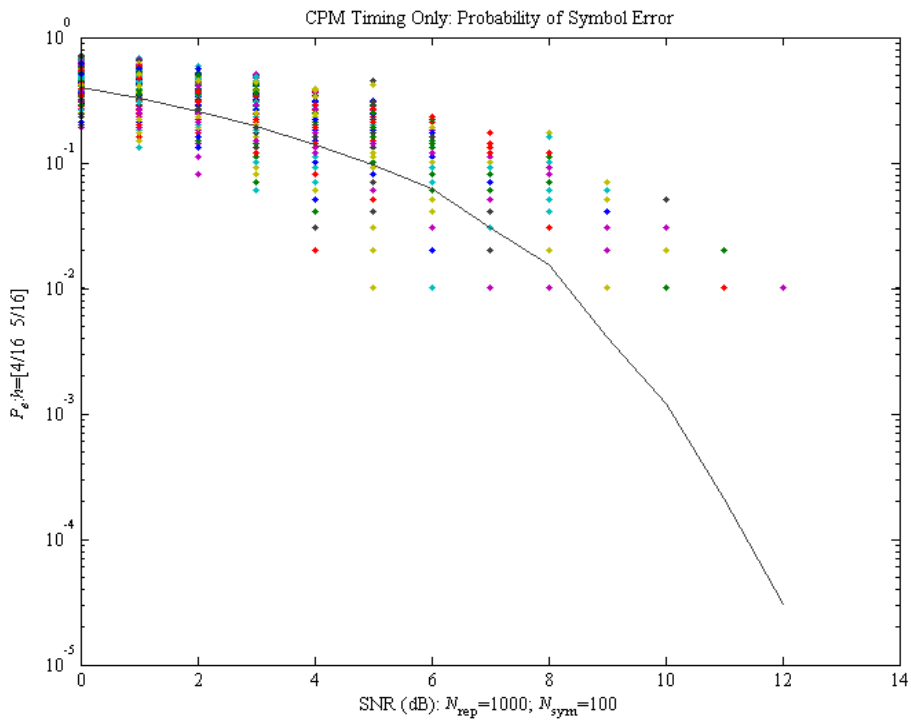

Figure 4.4: CPM1 SNR Performance; timing errors only.

error is held fixed at $\tau = 3.8$ samples and the CPM simulations contain $N_{\text{sym}} = 100$ symbols. Both plots present each simulation as a dot and draw the mean probability of error as line. These plots demonstrate that CPM1 is at least functional with respect to timing. A nice feature of the CPM1 design is that the phase error $\Delta\phi$ is eliminated by the derivative. Figures 4.6 and 4.7 verify that CPM1 functions regardless of the phase error.

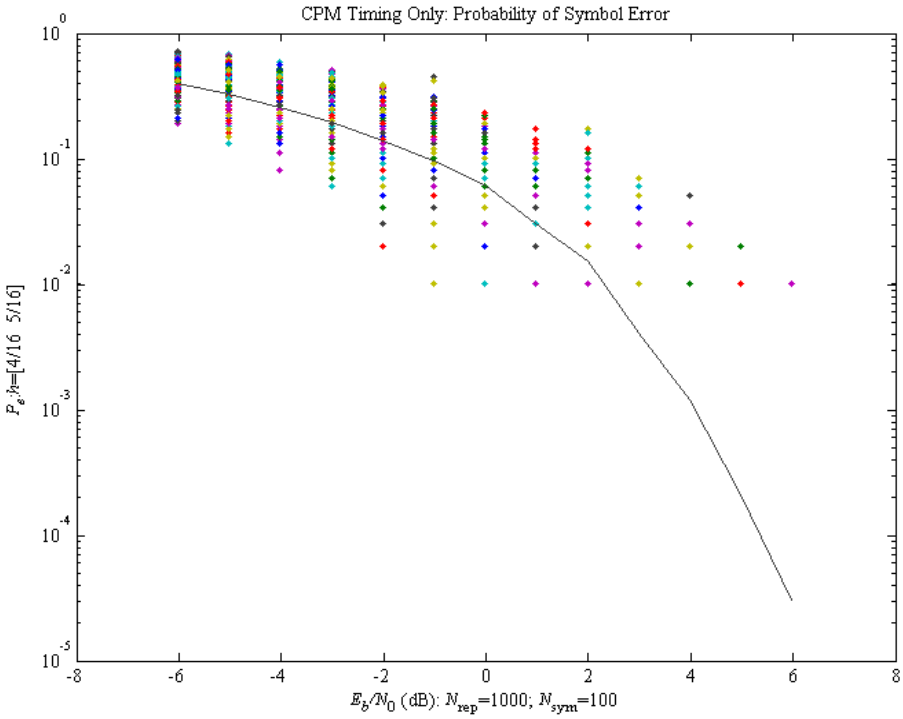

Figure 4.5: CPM1 E_b/N_0 Performance; timing errors only.

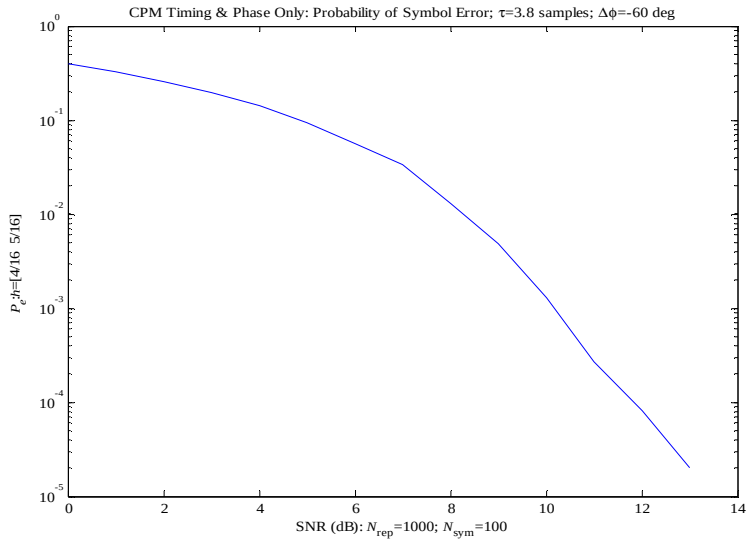

Figure 4.6: CPM1 SNR Performance; timing and phase errors only.

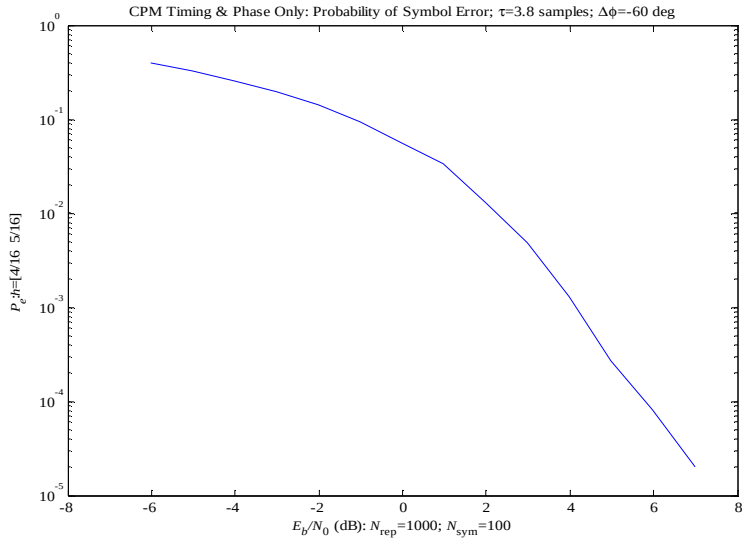

Figure 4.7: CPM1 E_b/N_0 Performance; timing and phase errors only.

4.3 Designs for Timing, Phase, and Frequency

The reason that CPM1 handled the phase error $\Delta\phi$ is that taking the derivative of the received phase

$$\psi_R'(t) = \psi_B'(t-\tau) + \psi_G'(t)$$

automatically removes the phase error—assuming that no frequency error Δf_c is present. Recall that $\psi_G(t)$ denotes the phase error induced by the additive noise. If the frequency error is present, the instantaneous frequency of the received CPM signal $r_B(t)$ is

$$\psi_R'(t) = 2\pi\Delta f_c + \psi_B'(t-\tau) + \psi_G'(t).$$

Thus, it is natural to take another derivative to remove the frequency error:

$$\psi_R''(t) = \psi_B''(t-\tau) + \psi_G''(t).$$

The problem is that using the 1REC phase pulse,

$$q'(t) = \frac{1}{2T}1_{[0,T]}(t) \quad \Longrightarrow \quad q''(t) = \frac{1}{2T}\left\{\delta(t) - \delta(t-T)\right\}.$$

That is, most of the signal is also wiped out and the information is carried in two noise-corrupted estimates of the delta functions that now overlap with the preceding and following pulse.

Simulations verified that using two derivatives with the 1REC pulse did compensate for the frequency offset—but the symbol error rate was too large. Nevertheless, several schemes to compensate for the frequency shift come to mind.

- Instead of using the 1REC phase, use a phase pulse with a non-vanishing second derivative (i.e., the chirps in [7, Fig. 2.4]).
- Estimate Δf_c from the sample mean of $\psi_B'(t)$.
- Simultaneously estimate Δf_c and the f_k's using a Viterbi decoder.
- Use the frequency estimator of Section 3.3.

Because we are looking for the simplest design, the next section explores the CPM receiver that uses the sample mean.

4.4 CPM2: Timing, Phase, and Frequency

CPM2 generalizes CPM1 to handle frequency offsets using the sample mean of the received phase. The received signal

$$r_B(t) = e^{j\Delta\phi} e^{j2\pi\Delta f_c t} s_B(t-\tau) + g_B(t)$$

has instantaneous frequency

$$\psi'_R(t) = 2\pi\Delta f_c + \psi'_B(t-\tau) + \psi'_G(t)$$

with expected value

$$E[\psi'_R(t)] = 2\pi\Delta f_c + E[\psi'_B(t-\tau)] + E[\psi'_G(t)] = 2\pi\Delta f_c,$$

using the circularity of the noise and assuming all symbols $\alpha_k = \pm 1, \pm 3$ are equally likely. If a *sufficiently* large number of symbols are observed, a *sufficiently* accurate estimate of the frequency offset is obtained from the sample mean. The CPM2 design is a generalization of the CPM1 design using this estimate of the frequency offset.

CPM2-1 Estimate the phase of $s_B(t)$

1. Phase unwrap [20]:

$$\psi_R(t) := \text{unwrap}(\text{angle}(r_B(t))).$$

2. Compute the instantaneous frequency (IF) to remove the phase error:

$$\psi'_R(t) = 2\pi\Delta f_c + \psi'_B(t-\tau) + \psi_G(t).$$

3. Estimate the frequency error:

$$\widehat{\Delta f_c} := \frac{1}{2\pi}\text{mean}[\psi'_R(t)].$$

4. Integrate to get a clean phase:

$$\widehat{\psi}_B(t-\tau) := \int_0^t \left\{\psi'_R(t') - 2\pi\widehat{\Delta f_c}\right\} dt'.$$

CPM2-2 Synchronize:

1. Estimate the symbol timing error as

$$\{\widehat{\epsilon}, \widehat{\ell}\} := \text{argmin}\{\|\widehat{\psi}_B(\circ - \tau) - \ell(\circ - \epsilon T)\|_2 : \epsilon \in [-1/2, 1/2], \ell \in \mathcal{L}\}.$$

2. Estimate the instantaneous frequency from the best piecewise linear minimizer ($L = 1$):

$$\widehat{f}_B(t) := \frac{1}{2\pi}\widehat{\ell}'(t - \widehat{\epsilon}T).$$

3. Estimate the superbaud error η: If the symbol timing is correct,

$$f_k \approx \alpha_k h_{[k+\eta]}$$

so choose the delay that minimizes

$$\widehat{\eta} := \operatorname{argmin}\left\{\sum_{k=1}^{N_{\text{sym}}} \|f_k/h_{k+\eta} - \mathbf{a}\|_{-\infty} : \eta = 0, 1\right\}, \quad \mathbf{a} = \begin{bmatrix} 3 \\ 1 \\ -1 \\ -3 \end{bmatrix}.$$

CPM2-3 Decode:

$$\widehat{\alpha}_k := \operatorname{argmin}\{|f_k/h_{k+\widehat{\eta}} - \alpha| : \alpha = \pm 1, \pm 3\}$$

The following figures illustrate CPM2 in a typical simulation. Figure 4.8 compares the transmitted (no noise) CPM signal $s_B(t)$ against the noisy, time-shifted, phase-shifted, and frequency-shifted baseband received signal $r_B(t)$. The noise is small but the frequency offset is spinning the received signal.

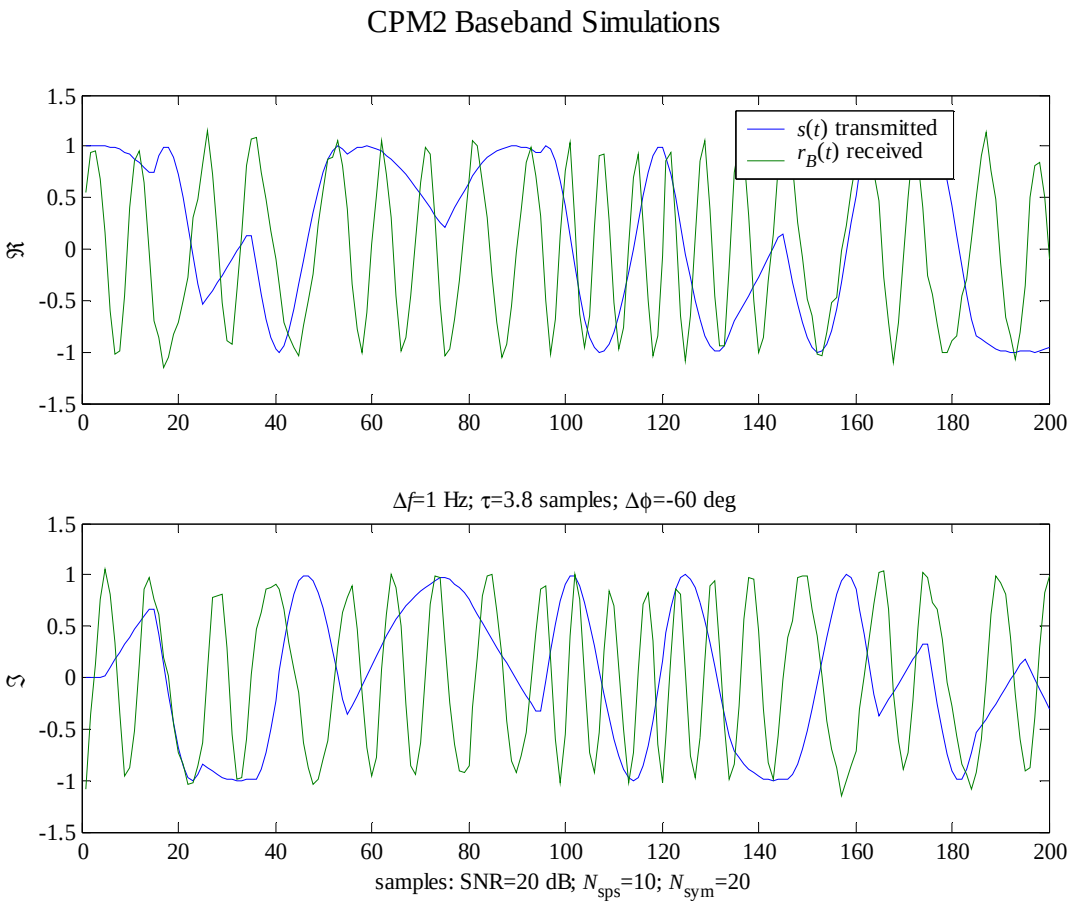

Figure 4.8: Transmitted and received CPM signals.

Figure 4.9 makes the corresponding comparison on the phases and instantaneous frequencies. Even though the SNR is large, and phases are close, the derivative amplifies the phase noise in the instantaneous frequency.

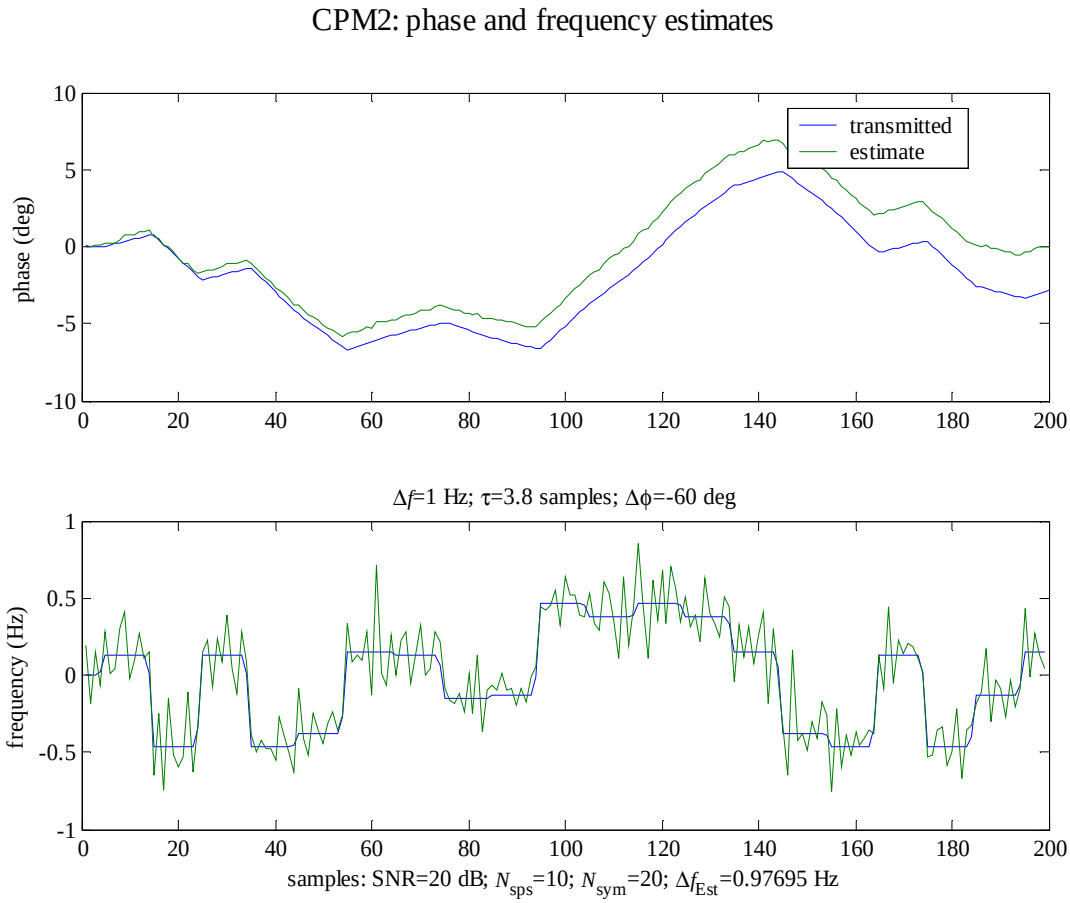

Figure 4.9: Transmitted and CPM2 estimate of the instantaneous phase and instantaneous frequency.

Figure 4.10 compares the spectra of the transmitted CPM signal $s_B(t)$, the received signal $r_B(t)$, and the estimated signal

$$s_E(t) := \exp(j\widehat{\psi}_B(t-\tau)).$$

The spectrum of received signal is registering the frequency shift. The frequency estimate appears to be close to the frequency shift. We see that the spectrum of the estimated signal overlaps the spectrum of the transmitted signal.

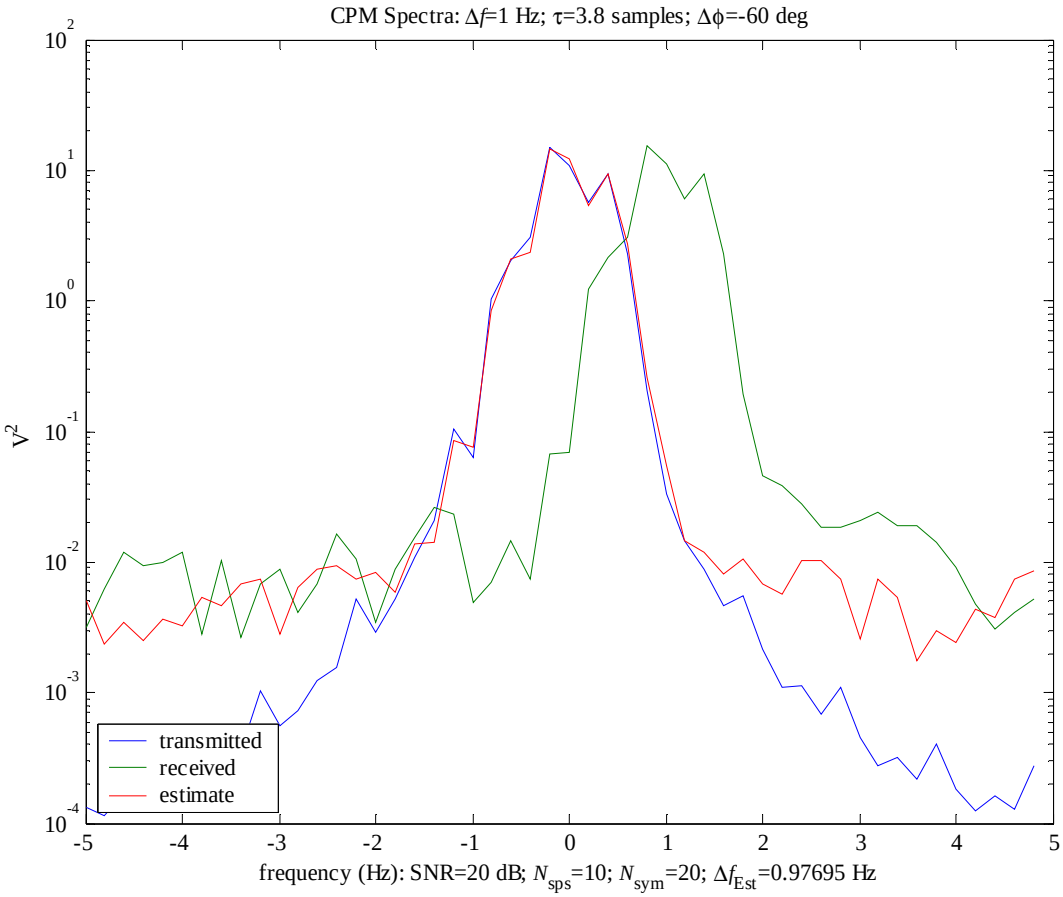

Figure 4.10: CPM2 spectra.

Figure 4.11 shows that the symbol timing, the superbaud timing, and the decoding has been successful despite the noise in the instantaneous frequency. The step

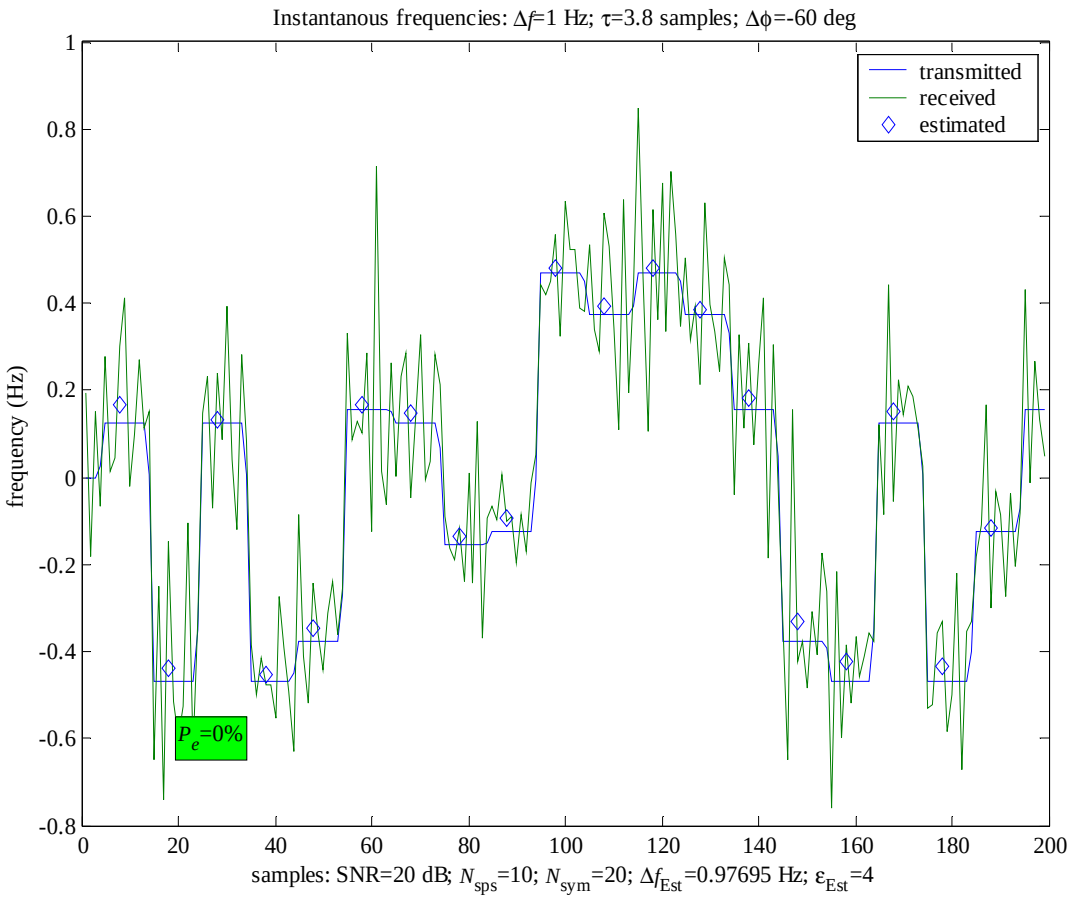

Figure 4.11: CPM2 performance.

function (blue line segments) shows the instantaneous frequency of the transmitted signal: $\psi'_B(t)$. The noisy (green) line is the received phase after correcting for phase, frequency, and superbaud timing. The (blue) diamonds mark the instantaneous frequency estimate of that estimated symbol period obtained from the noisy received phase. That the estimated probability of symbol timing is zero ($P_e = 0$) means that the symbol timing functioned.

4.5 CPM2: Timing, Phase, and Frequency Performance

Figure 4.12 reports on the performance of CPM2. As in the preceding simulations, the delay is 3.8 sample periods ($\tau = 3.8 \times T_s$) and the phase shift is $\Delta\phi = -60°$. The frequency shift is scaled to the sample period of $T_s = 0.1$. Using seconds, we shift the CPM signal by $\Delta f = 1$ Hertz. The effect of the frequency shift on the

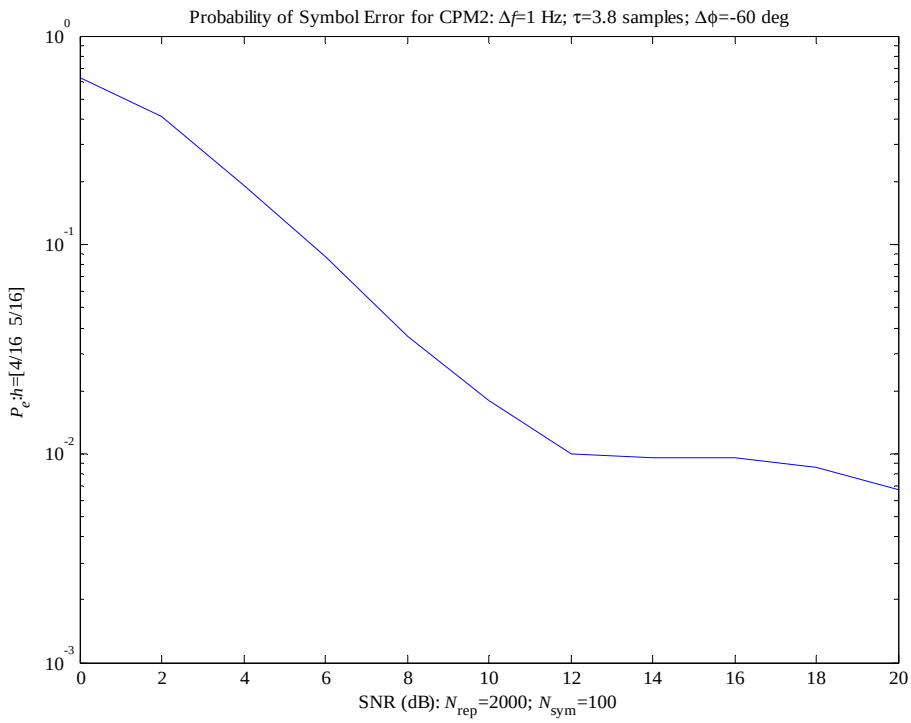

Figure 4.12: CPM2 performance using only 100 symbols.

simple synchronizer is clear: As the noise decreases, a noise floor is revealed. Thus, the frequency estimate is too coarse for a credible receiver. However, the frequency estimator was gotten from 100 symbols. Would taking more symbols improve the performance of CMP2? Figure 4.13 shows collecting 200 symbols does improve the performance of CPM2. Doubling the number of symbols pushed the error floor down 10 dB. However, an error floor still exists and is still too high. Figure 4.14 reports that opening the observation interval to 500 symbols continues to improve the performance of CPM2.

Before we leave this section, it is worthwhile to put these frequency shifts in context. A frequency shift of $\Delta f = 1$ Hertz is commensurate with the symbol period $T = 1$ second. Scaled to a real-world system using $T = 1$ μs, the frequency shift is $\Delta f = 1$ MHz. This exceeds typical demodulation and frequency errors by several

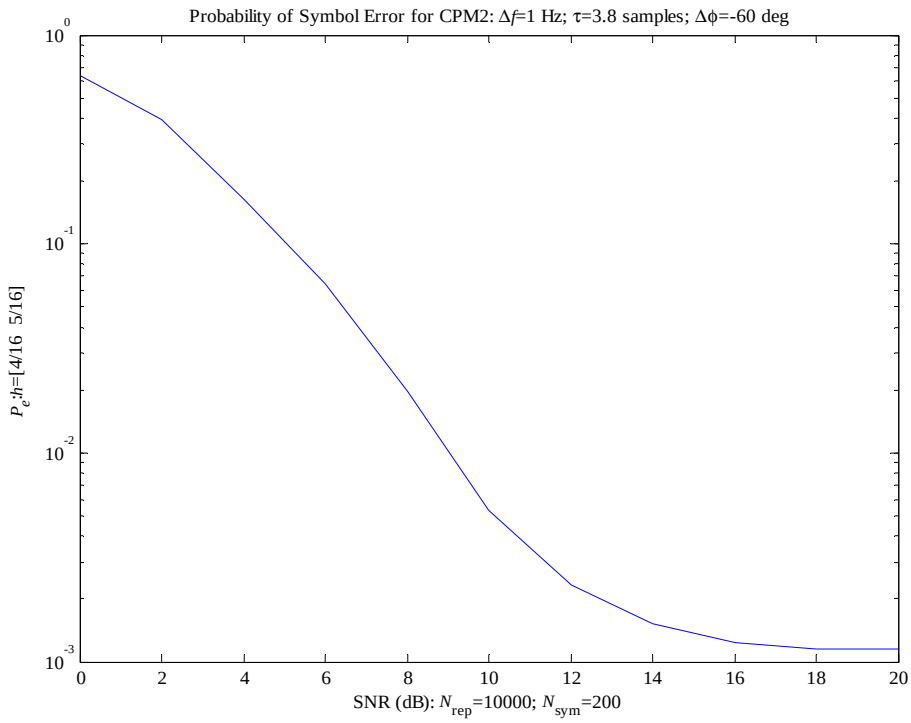

Figure 4.13: CPM2 performance using 200 symbols.

orders of magnitude [2].

4.6 Discussion

The progression of CPM0, CMP1, and CPM2 shows how to generalize the very simple multi-h CPM receiver to handle timing, phase, and frequency synchronization. The progression also shows that the synchronizer's performance should be measured in the context of a receiver. For example, increasing the accuracy of the frequency synchronizer by 10% is meaningless until the 10% improvement maps to an improvement in its CPM receiver.

The simplicity of these CPM designs also lets us swap out various synchronizers. For example, the frequency synchronizer of Section 3.3 can estimate the frequency $\widehat{\Delta f_c}$ used in CPM2-1. We can ask: Does the accuracy of the more complex estimator let us shorten the observation interval? More generally, we could omit a frequency estimator entirely. Indeed, we could simply hand the CPM receiver the true frequency Δf_c corrupted by Gaussian noise:

$$\widehat{\Delta f_c} = \Delta f_c + \Delta f.$$

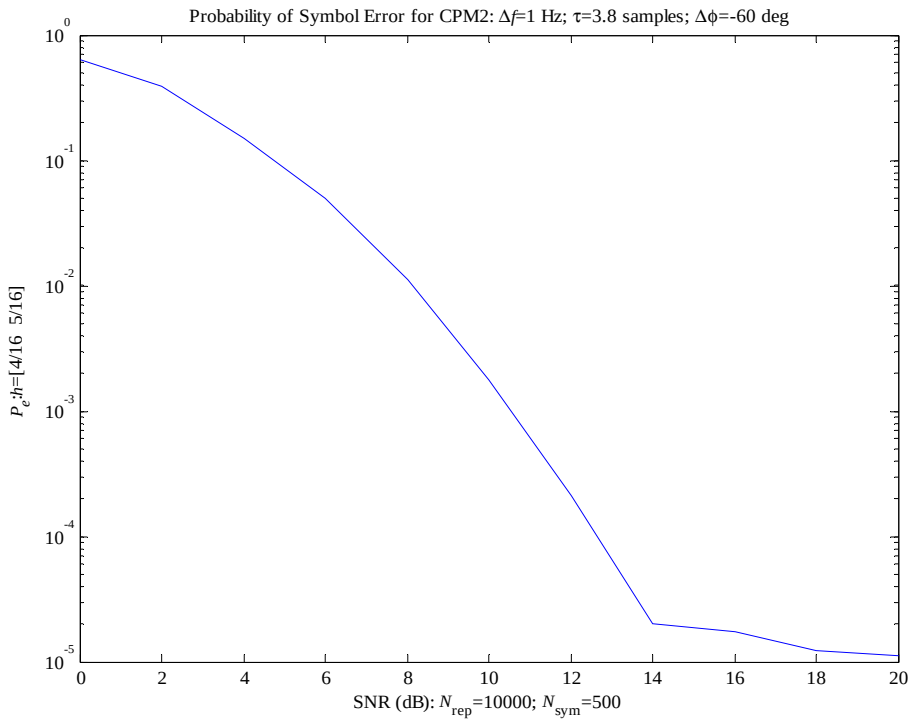

Figure 4.14: CPM2 performance using 500 symbols.

The performance of the CPM receiver could then be known as function of the variance of Δf. Knowing this function sets the accuracy requirements of *any* frequency estimator. However, tinkering with frequency estimators in additive noise is much less of a problem than multipath channels.

5

Multi-h in Multipath

The received signal in multipath is a sum of delayed and faded versions of the transmitted signal:

$$r_B(t) = \sum_{m=1}^{M} a_m(t) s_B(t - \tau_m)$$

Each fading process $\{a_m(t)\}$ modulates the CPM signal. The fading processes $\{a_m(t)\}$ under consideration have power levels ranging from 0 to -10 dB. We have seen that the CPM receivers need a signal in excess of 10 dB over additive noise. Consequently, some fading processes may inject too much noise for these simple CPM receivers. Sections 5.1 and 5.2 verify this noise limitation using a UHF channel model of the San Diego Harbor. Good exploitation of multipath may require simultaneously estimating both the channel and signal [2]. Section 5.3 formalizes this observation and concludes this report.

5.1 Harbor Multipath

A communications channel can be modeled as a stochastic, time-varying linear filter $h(t, \tau)$ that maps the baseband transmitted signal $s_B(t)$ to the received signal $r_B(t)$ as [9]:

$$r_B(t) = \int_{-\infty}^{\infty} h(t, t - \tau) s_B(\tau) d\tau.$$

A standard multipath model is the Quadrature Modulation Fading Simulator (QMFS) [32]:

$$h(t, \tau) = \sum_{m=1}^{M} a_m(t) \delta(\tau - \tau_m).$$

The received signal is a sum of delayed and faded versions of the transmitted signal:

$$r_B(t) = \sum_{m=1}^{M} a_m(t) s_B(t - \tau_m)$$

Typically, the fading processes $\{a_m(t)\}$ are complex-valued, jointly wide-sense stationary (JWSS), narrow-band, Gaussian random processes:

$$a_m(t) = e^{j2\pi f_D t}\{\gamma_m + g_m(t)\}.$$

The Dopper shift is denoted by f_D. The specular component models coherent reflection using the complex-valued reflection coefficient γ_m. The diffuse component is modeled by the narrow-band, zero-mean, Gaussian random process $\{g_m(t)\}$ with variance σ_g^2. The K factor for the mth path is

$$K_m = \frac{\gamma_m^2}{\sigma_g^2}.$$

Random phases are simultaneously absorbed in γ_m and $\{g_m(t)\}$. For the delays, general models take the τ_m's as time-varying. Constant delays sufficed for modeling the channels in the Extended Littoral Battlefield (ELB) [2]. The delays and statistics of the fading processes determine the channel model. Table 5.1 lists the statistics for the Two-Path Harbor Model. All the measurements are referenced to Path 1.

Table 5.1: Two-Path Harbor Model [3].

Path	Delay μs	Power dB	Doppler shift Hz	Fade rate Hz	Fading
1	0.0	0	0	1.6	Rican $K = 136$
2	2.0	-5	-5	3.8	Rican $K = 14$

5.2 Multi-h in the Harbor Multipath

This section pushes the multi-h CPM signal through the Harbor multipath of Table 5.1 and then attempts to processes the received signal. To match the bandwidth of the ELB measurements, 10^6 symbols per second are transmitted. The symbol interval is then

$$T = 1 \quad (\mu\text{s})$$

If the observation interval is 100 symbols, the large K factors and the slow fade rates make the Harbor multipath almost time invariant:

$$\begin{aligned} r_B(t) &= a_1(t)s_B(t-\tau_1) + a_2(t)s_B(t-\tau_2) \\ &= \{\gamma_1 + g_1(t)\}s_B(t) + e^{-j2\pi f_D t}\{\gamma_2 + g_2(t)\}s_B(t-\tau_2) \\ &\approx \gamma_1 s_B(t) + e^{-j2\pi f_D t}\gamma_2 s_B(t-\tau_2). \end{aligned}$$

Because the observation interval is only 100μs, the 5-Hertz Doppler is absorbed into γ_2. The Two-Path Harbor Model, assuming 100 symbols, then admits the approximation

$$r_B(t) = \gamma_1 s_B(t) + \gamma_2 s_B(t - \tau_2).$$

With a 2-symbol delay, Path 2 looks like 5 dB noise to Path 1. The preceding section demonstrated that 5 dB noise will kill the CPM receiver by forcing a probability of symbol error P_e of approximately 15%. Path 2 forces a similar effect.

Figure 5.1 shows the transmitted and multipath received signals. Even though the additive noise is 20 dB down from Path 1, Path 2 is only 5 dB down. Compared

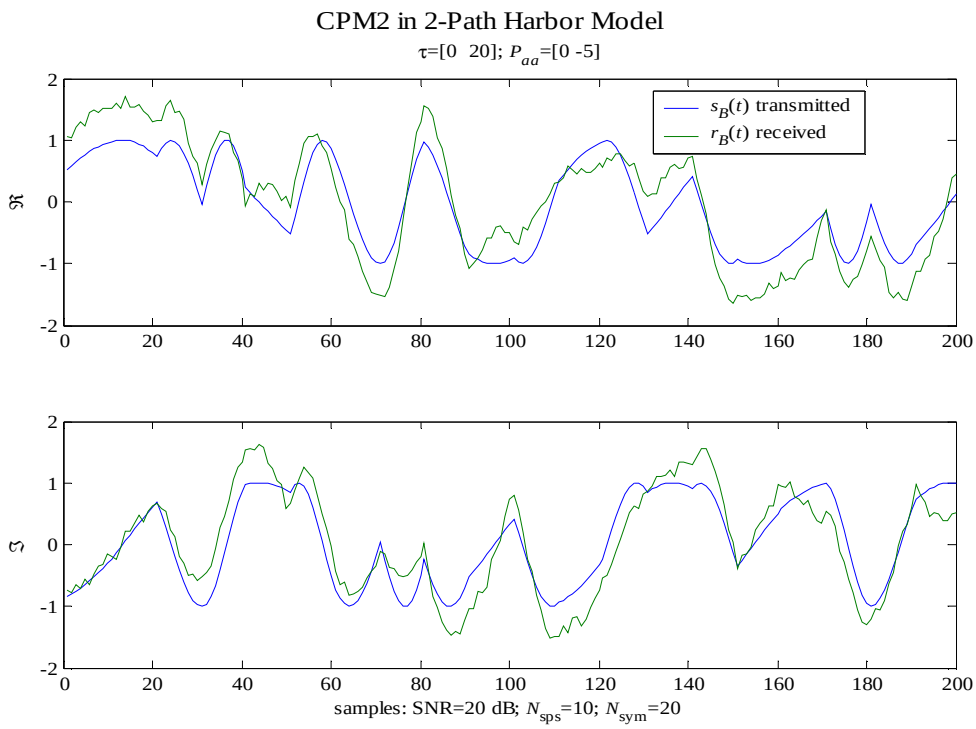

Figure 5.1: CPM in Harbor multipath.

to the low-noise plots of Section 4, Path 2 is a significant noise term. Figure 5.2 compares instantaneous frequency of the transmitted signal to the multipath version. Path 2 has injected large spikes that degrade the CPM receiver (CMP2) and returns a probability of symbol error commensurate with 5 dB noise. Additional simulations verify that the power in Path 2 is equivalent to adding noise at that power. In contrast to the multipath that actually improved the performance of an LPI receiver [3], the 2-symbol delay in Path 2 degrades the CPM receiver. Consequently, exploitation of multipath may require the channel and signal estimation to use these CPM receivers [2].

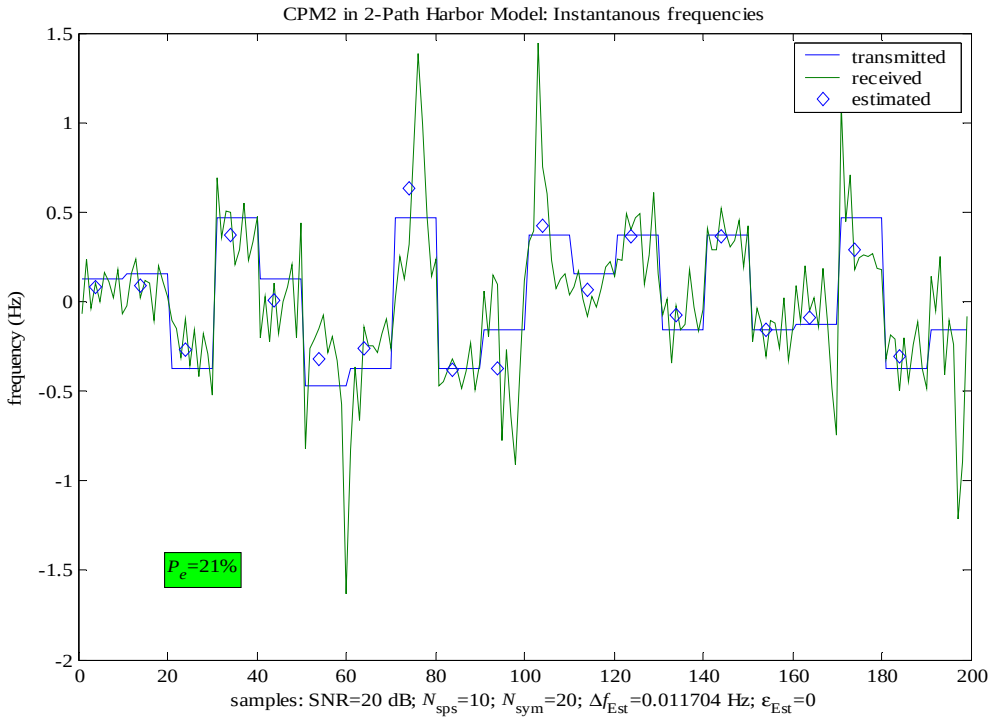

Figure 5.2: CPM2 performance in Harbor multipath.

5.3 Summary

This report sets out a simulation framework that allows phase, timing, and frequency synchronizers to be swapped in and out of a CPM receiver to assess end-to-end performance. The additive noise simulations demonstrated the functionality of the basic CPM receivers of this report. However, the preceding multipath simulation demonstrated these CPM receivers require additional processing to function in multipath. Table 5.2 organizes selected papers on CPM performance over RF channels. The table shows few papers consider multi-h CPM in the full-up multipath. Consequently, the simulation efforts of Phase 3 should focus on extending the CPM receiver to handle multi-h in multipath. The simplest approach is to simultaneously estimate both the CPM signal and channel [2]. The received and demodulated CPM signal can then be processed by these basic CPM receivers The research efforts of Phase 3 can attack the following:

Q-1 How many symbols are needed for synchronization?

Q-2 How does synchronization depend on the precision of the estimators?

Q-3 Does there exist a general lower bound on the number of symbols to synchronize?

Table 5.2: Multipath CPM Literature; NDA:=Nondata-Aided; DCD:=Data and Channel Demodulation; TDL:=Tapped Delay Line; DFSE:=Decision Feedback Sequence Equalization.

Multi-H	NDA	Channel	Algorithm	Reference
	$\surd$	QMFS	DCD	[48]
	$\surd$	TDL	MLSE	[47], [13]
	$\surd$	Rican	Trellis	[1]
	$\surd$	TDL	T	[16]
	$\surd$	TDL	DFSE	[13]
$\surd$	$\surd$	TDL	MLSD	[17]
	$\surd$	Rayleigh+CCI	1-DPD	[18]
	$\surd$	TDL	unwrapping	[26]
$\surd$	$\surd$	MILSATCOM	DFSE	[27]
$\surd$	$\surd$	Satellite-Mobile	MLSE	[33]
	$\surd$	Rayleigh	Trellis	[25]
$\surd$	$\surd$	CCI	MLSE	[45]
$\surd$	$\surd$	Rican-Mobile	MLSE	[46]

Q-4 When does multipath destroy synchronization?

Q-5 Would demodulation of both the channel and signal improve CPM performance in multipath?

Q-6 Can a hybrid CPM system overcome multipath?

Q-1 suggests that the CPM receiver should be made adaptive to the channel. The question implicitly asks how the performance depends on the SNR. Q-2 brings up the issue of the intertwining to the various synchronizers in the full CPM radio. For example, it makes little sense to improve the symbol timing estimator if the CPM receiver is limited by superbaud estimator. Q-3 seeks a general design bound to benchmark any synchronizer. Q-4 is answered by the Harbor multipath example in preceding section. The final questions are points-of-departure for Phase 3. Q-5 is a straight-forward question based on the demodulation results of [2]. Q-6 is more ambitious and based on the mix of CPM and the excellent multipath properties of OFDM [40].

Bibliography

[1] Abrishamkar, F. [1988] Class of Compact Spectrum Coded Signals over Multipath Fading Channels, *MilCom88*, Volume 1, pages 265–270.

[2] Allen, J.; M. Reuter; R. North [1998] *RF Channel Characterization & Estimation: Sequence-Based Methods*, `http://bobcat.spawar.navy.mil/hdrlos`.

[3] Allen, J. & M. Reuter [2000] *The Extended COTS LPI Communication System Phase 3: Performance Analysis*, Technical Report 1828, SPAWAR System Center, San Diego, CA.

[4] D'Andrea, Aldo; Alberto Ginesi; Umberto Mengali [1995] Frequency Detectors for CPM Signal, *IEEE Transactions on Communications*, 43(2/3/4).

[5] Anderson, J. B. & D. P. Taylor [1978] A Bandwidth-Efficient Class of Signal-Space Codes, *IEEE Transactions on Information Theory*, IT-24(11).

[6] D'Andrea, Aldo; Umberto Mengali; Michele Morelli [1996] Symbol Timing Estimation CPM Modulation, *IEEE Transactions on Communications*, 44(10).

[7] Anderson, John B.; Tor Aulin, Carl-Erik Sundberg [1986] *Digital Phase Modulation*, Plenum Press, New York.

[8] Barbarossa, Sergio; Anna Scaglione; Georgios B. Giannakis [1998] Product High-Order Ambiguity Function for Multicomponent Polynomial-Phase Signal Modeling, *IEEE Transaction on Signal Processing*, 46(3).

[9] Bella, Philip A. [1963] Characterization of Randomly Time-Variant Linear Channels, *IEEE Transactions on Communications Systems*, CS-11.

[10] Ginesi, Alberto; Umberto Mengali; Michele Morelli [1999] Symbol and Superbaud Timing Recovery in Multi-H CPM, *IEEE Transactions on Communications*, 47(5).

[11] Geoghean, M. [2000] Description and Performance Results for a Multi-h CPM Telemetry Waveform, *MilCom 2000*, Volume 1, pages 353–357.

[12] Graser, S. J. [2001] Techniques for Improving Power and Bandwidth Efficiency of UHF MILSATCOM Waveforms, *MilCom 2001*, Volume 1.

[13] Guren, Hans C. & Nils Holte [1993] Decision Feedback Sequence Estimation for Continuous Phase Modulation on a Linear Multipath Channel, *IEEE Transactions on Communications*, 41(2).

[14] Ho, Paul, K. & Peter J. Mclane [1988] Spectrum, Distance, and Receiver Complexity of Encoded Continuous Phase Modulation, *IEEE Transactions on Information Theory*, 34(5).

[15] Hamila, Ridha; Jussi Vesma; Markku Renfors [2002] Polynomial-Based Maximum-Likelihood Technique for Synchronization in Digital Filters, *IEEE Transactions on Circuits and Systems–II*, 49(8).

[16] Ince, E. A.; Y. J. Guo; S. K. Barton [1997] T-Algorithm Detection of Partial Response Continuous Phase Modulated Signals Over Multipath Channels, PIMRC97, Volume 3, pages 1135–1139.

[17] Jacquemin, P.; A. J. Rodrigues; L. Vandenorpe [1995] Performance of Multi-H DS-CDMA in Multipath Rayleigh Fading Channels with Multi-User Interference and Uplink Diversity, *PIMRC'95*, Volume 1, pages 178–182.

[18] Korn, I. [1991] GMSK with Frequency Selective Rayleigh Fading and Co-Channel Interference, *GLOBECOM'91*, Volume 2, pages 792–796.

[19] Liebetreu, John M. [1986] Joint Carrier Phase Estimation and Data Detection Algorithms for Multi-h CPM Data Transmission, *IEEE Transactions on Communications*, COM-34(9).

[20] MATLAB [1996] *Signal Processing Toolbox User's Guide, Version 4*, The MathWorks, Inc., 24 Prime Park Way, Natick, MA.

[21] Mazur, Brian A. & Desmond P. Taylor [1981] Demodulation and Carrier Synchronization of Multi-h Phase Codes, *IEEE Transactions on Communications*, COM-39(3).

[22] Mengali, U. & A. N. D. D'Andrea [1997] *Synchronization Techniques for Digital Receiver*, Plenum Press, New York, NY.

[23] Morelli, Michele; Umberto Mengali; Giorgio M. Vitetta [1997] Joint Phase and Timing Recovery with CPM Signals, *IEEE Transactions on Communications*, 45(7).

[24] Mengali, Umberto & Michele Morelli [1999] Joint Frequency and Timing Recovery for MSK-Type Modulation, *IEEE Transactions on Communications*, 47(6).

[25] Narayanan, Krishan R. [1999] Iterative Demodulation and Decoding of Trellis Coded CPM, 0-7803-5538-5/99.

[26] Neugebauer, Shawn; Gary Ford; Michael Ready [2000] Characterization of Multipath-Distorted and Frequency Discriminated CPM Signals, *Asilomar 34*, Volume 2, pages 819–822.

[27] Peterson, Bror & Donald R. Stephens [2002] DFSE Equalization of Dual-H CPM Over UHF MILSATCOM Channels, *MilCom'02*, Volume 2, pages 1406–1411.

[28] Pettit, R. H. & Bruce E. Wahlen [2000] *A Maximum-Likelihood-Based Frequency Synchronizer for Dual-H, Full Response 4-ARY Continuous Phase Modulation (CPM)*, Space and Naval Warfare Systems Center Technical Report 3094.

[29] Pettit, R. H. & Bruce E. Wahlen [2002] *A Joint Maximum-Likelihood-Based Phase and Timing Synchronizer for Dual-h, Full-Response 4-ary CPM*, Space and Naval Warfare Systems Center Technical Report 3143.

[30] Porat, Boaz & Benjamin Friedlander [1996] Blind Deconvolution of Polynomial-Phase Signal Using the High-Order Ambiguity Function, *Signal Processing*, 53, pages 149–163.

[31] Premji, Al-Nasir & Desmond P. Taylor [1987] A Practical Receiver Structure for Multi-h CPM Signals, *IEEE Transactions on Communications*, COM-35(9).

[32] Proakis, John G. & Masoud Salehi [1994] *Communications Systems Engineering*, Prentice Hall, New Jersey.

[33] Rodrigues, A. J. & A. A. Albuquerque [1996] Diversity Techniques with Multi-H CPM for Satellite Mobile Systems, *IEEE 46th Vehicular Technology Conference*, Volume 1, pages 551–555.

[34] Skalar, Bernard [2000] *Digital Communications*, second edition, Prentice-Hall, Inc., Upper Saddle River, NJ.

[35] Sabel, Lesley Phillip [1993] *A Maximum Likelihood Approach to Symbol Timing Recovery in Digital Communications*, Ph.D. Thesis, School of Electronic Engineering, University of South Australia, The Levels, South Australia, 5095.

[36] Sasase, Iwao & Mori Shinsaku [1991] Multi-h Phase-Coded Modulation, *IEEE Communications Magazine*, 29(12).

[37] Sundberg, Carl-Erik [1986] Continuous Phase Modulation, *IEEE Communications Magazine*, 24(4).

[38] Tang, Weiyi & Ed Shwedyk [2000] A Quasi-Optimum Receiver for Continuous Phase Modulation, *IEEE Transactions on Communications*, 48(7).

[39] Tang, Weiyi & Ed Shwedyk [2001] ML Estimation of Symbol Timing and Carrier Phase for CPM Walsh Signal Space, *IEEE Transactions on Communications*, 49(6).

[40] Tasadduq, I. A. & R. K. Rao [2002] Detection of OFDM-CPM Signals over Multipath Channels, *IEEE International Conference on Communications*, Volume 3, pages 1651–1655.

[41] Van Trees, H. L. [1968] *Detection, Estimation, and Modulation: Part I*, Wiley, New York, NY.

[42] Villares, Javier & Gregori Váquez [2002] Optimal Quadratic Non-Assisted Parameter Estimation for Digital Synchronization, *2002 International Zurich Seminar on Broadband Communications*, pages 46-1 to 46-6.

[43] Wade, G.; M. Fu; R. Jakobs; J. Nign; M. Tomlinson; A. Ambroze [2002] On CPM System Design and Simulations, *First International Conference on Information Technology & Applications* (ICITA 2002), Bathurst, Australia.

[44] Wilons, Stephen G. & Richard C. Gaus [1981] Power Spectra of Multi-h Phase Codes, *IEEE Transactions on Communications* COM-29(3).

[45] Xiong, Fuqin & Vivek Shivananda [1996] Performance of 1REC-MHPM in the Presence of Adjacent Channel Interference, *IEEE Transactions on Communications*, 44(12).

[46] Xiong, Fuqin & Sachin Bhatmuley [1997] Performance of MHPM in Rician and Rayleigh Fading Mobile Channels, *IEEE Transactions on Communications*, 45(3).

[47] Yiin, Lihbor & Cordaon L Stüber [1997] MLSE and Soft-Output Equalization for Trellis-Coded Continuous Phase Modulation, *IEEE Transactions on Communications*, 45(6).

[48] Zeger, Linda M. & Hisashi Kobayashi [1999] MLSE for CPM Signal in a Fading Multipath Channel, *IEEE Pacific Rim Conference on CCSP*.

Approved for public release; distribution is unlimited.