2014-04-28

Phase 1 Final Report: Preamble Assisted Equalization for Aeronautical Telemetry (PAQ)

Michael Rice
mdr@byu.edu

Mohammad Saquib

Arlene Cole-Rhodes

Farzad Moazzami

Erik Perrins

Phase 1 Final Report

Preamble Assisted Equalization for Aeronautical Telemetry (PAQ)

Funded by the
Test Resource Management Resources Center (TRMC)
through a grant from the
Army PEO STRI Contracting Office
Contract Number W900KK-13-C-0026

Submitted by

Michael Rice (Brigham Young University)
Mohammad Saquib (University of Texas at Dallas)
Arlene Cole-Rhodes (Morgan State University)
Farzad Moazzami (Morgan State University)
Erik Perrins (University of Kansas)

April 2014

Executive Summary

To fully leverage the benefits of a periodically inserted preamble, the preamble is used to estimate the state of the channel. To this end the following algorithms have been developed, tested in simulation, and ported to the GPU-based real-time system:

1. The preamble detector, which scans the received samples searching for the presence of the preamble. The location of the preamble in the received samples is required to use the received samples to estimate the frequency offset, channel impulse response, and noise variance. The preamble detector works in the presence of an uncompensated frequency offset and unknown channel.

2. The frequency offset estimator, whose result is used to compensate for a large frequency offset in the RF carrier. The frequency offset estimator operates in the presence of an unknown channel.

3. The channel impulse response estimator, whose result is used by the zero-forcing and minimum mean-squared error equalizers to compute the equalizer filter coefficients. The channel impulse response estimate is also used by the CMA+AMA equalizer for initialization.

4. The noise variance estimator, whose result is used to estimate the signal-to-noise ratio parameter in computing the optimum minimum mean-squared error equalizer filter coefficients.

The performance of three equalizers has been evaluated in simulation using eleven test channels derived from channel sounding experiments at Edwards AFB, CA. The equalizers are the zero-forcing (ZF) equalizer, the minimum mean-squared error (MMSE) equalizer, and the combined constant modulus algorithm, alphabet matched algorithm (CMA+AMA) equalizer. The bit error rate performances of all three over the eleven test channels has been performed. A longitudinal comparison shows that the MMSE and CMA+AMA have almost equivalent bit error rate performance (and the performance of both is superior to that of the ZF equalizer). Other equalization options are also discussed (see Section 9).

Most of the hardware needed to implement the real-time demonstration system has been acquired. These items include the following:

- Modified L/S-band and C-band transmitters. The modification was the periodic insertion of the iNET preamble and ASM fields.

- A modified telemetry receiver that outputs inphase and quadrature samples at 2 samples/bit.

- Two NVidia GPUs for performing the computationally complex equalization algorithms. These cards reside in two rack-mounted host computers, that have also been purchased.
- An 8-channel bit error rate tester.
- Portable, shock-proof, racks for housing the equipment, along with the hardware, tools, connectors, cables, fans, etc. necessary to support the experiments.

Because the emphasis of Phase 1 was primarily algorithmic, the report focuses on the description, analysis, and performance of the algorithms that form the equalizers. The Phase 2 report will provide a more detailed description of the hardware, C/C++ code, and the software architecture used to implement the equalization algorithms.

Contents

1 Preliminaries

1.1 Signal Model, Notation, and High-Level Signal Processing

In this project, preamble and ASM fields are periodically inserted into a data bit stream. The format is illustrated in Figure 1 where the iNET preamble comprises eight repetitions of the 16-bit sequence CD98_{hex} and the ASM field is

$$034776\text{C}7272895\text{B}0_{\text{hex}}. \tag{1}$$

These bits modulate an SOQPSK-TG carrier. After propagating through a multipath channel, the received signal is filtered, amplified, down-converted, and sampled to produce a sample sequence $r(n)$. Let $s(n)$ be the sampled version of the transmitted signal and let $h(n)$ be the equivalent discrete-time channel between the transmitter and receiver. The sample rate for $s(n)$ and $h(n)$ is 2 samples/bit. The channel filter is assumed to have a non-causal component comprising N_1 samples and a causal component comprising N_2 samples as illustrated in Figure 2. Consequently, the channel is represented by a length-$(N_1 + N_2 + 1)$ FIR filter. The relationship between the samples of the transmitted signal $s(n)$ and samples of the received signal $r(n)$ is

$$r(n) = \Big[s(n) * h(n)\Big] e^{j\omega_0 n} + w(n) = \left[\sum_{k=-N_1}^{N_2} h(k)s(n-k)\right] e^{j\omega_0 n} + w(n) \tag{2}$$

where ω_0 is an unknown frequency offset due to oscillator uncertainty and Doppler, and $w(n)$ is a complex-valued, zero-mean discrete-time Gaussian random process with autocovariance function

$$\frac{1}{2}\text{E}\Big\{w(n)w^*(n-k)\Big\} = \sigma_w^2\delta(k). \tag{3}$$

The high-level signal processing requirements are illustrated in Figure 3. Here, the emphasis is on the preamble detector, frequency offset estimator, and channel estimator. The equalization algorithms are discussed more fully in Section 8. Because the frequency offset and channel are estimated from the preamble data, the first order of business is to find the start of the samples corresponding to the preamble in the received sample sequence $r(n)$. This is the role of the preamble detector block. The received samples are stored in a data buffer and processed by the preamble detector in parallel as shown. The preamble detector outputs the index i, which means the $r(i)$ is the received sample corresponding to the start of the preamble.

The received preamble samples are used to estimate the frequency offset. This estimate is

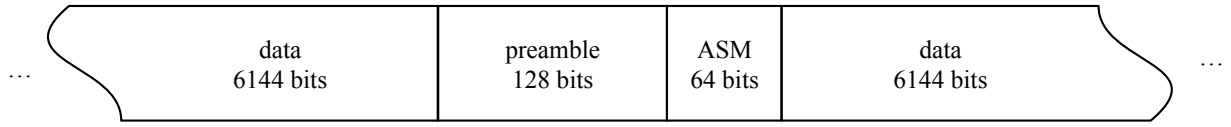

Figure 1: The data format used for the PAQ project.

performed by the frequency offset estimator block. The frequency offset $\hat{\omega}_0$ rads/sample is then used to "de-rotate" the received samples stored in the first data buffer. The de-rotated samples are stored in a second buffer.

The samples corresponding to the preamble and ASM fields in the de-rotated sample buffer are used to compute an estimate of the channel. The estimate of the channel is used to compute and/or initialize the equalizer algorithm as described in Section 8.

The signal processing flow outlined in Figure 3 shows that the preamble detector must be able to find the preamble in the presence of an unknown frequency offset an unknown channel. Furthermore, the frequency offset estimator must be capable of producing a reliable estimate in the presence of an unknown channel. These observations form the core requirements for the algorithms described in Sections 2 (preamble detector) and 3 (frequency offset estimator).

1.2 Test Channels

To perform the analysis and simulation described in the following sections, we used a set of eleven multipath channels derived from the M4A channel sounding experiments [1]. The channel impulse responses were computed as follows:

1. Resampling: The channel impulse response from the M4A experiments, $h_{\text{M4A}}(n)$, was sampled at 200 Msamples/s. We desire the same channel sampled at 2 samples/bit at the PAQ bit

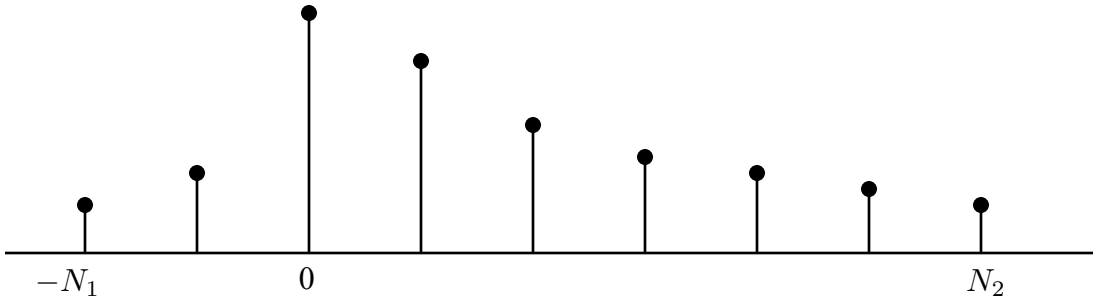

Figure 2: An illustration of the discrete-time channel of length $N_1 + N_2 + 1$ with a non-causal component comprising N_1 samples and a causal component comprising N_2 samples.

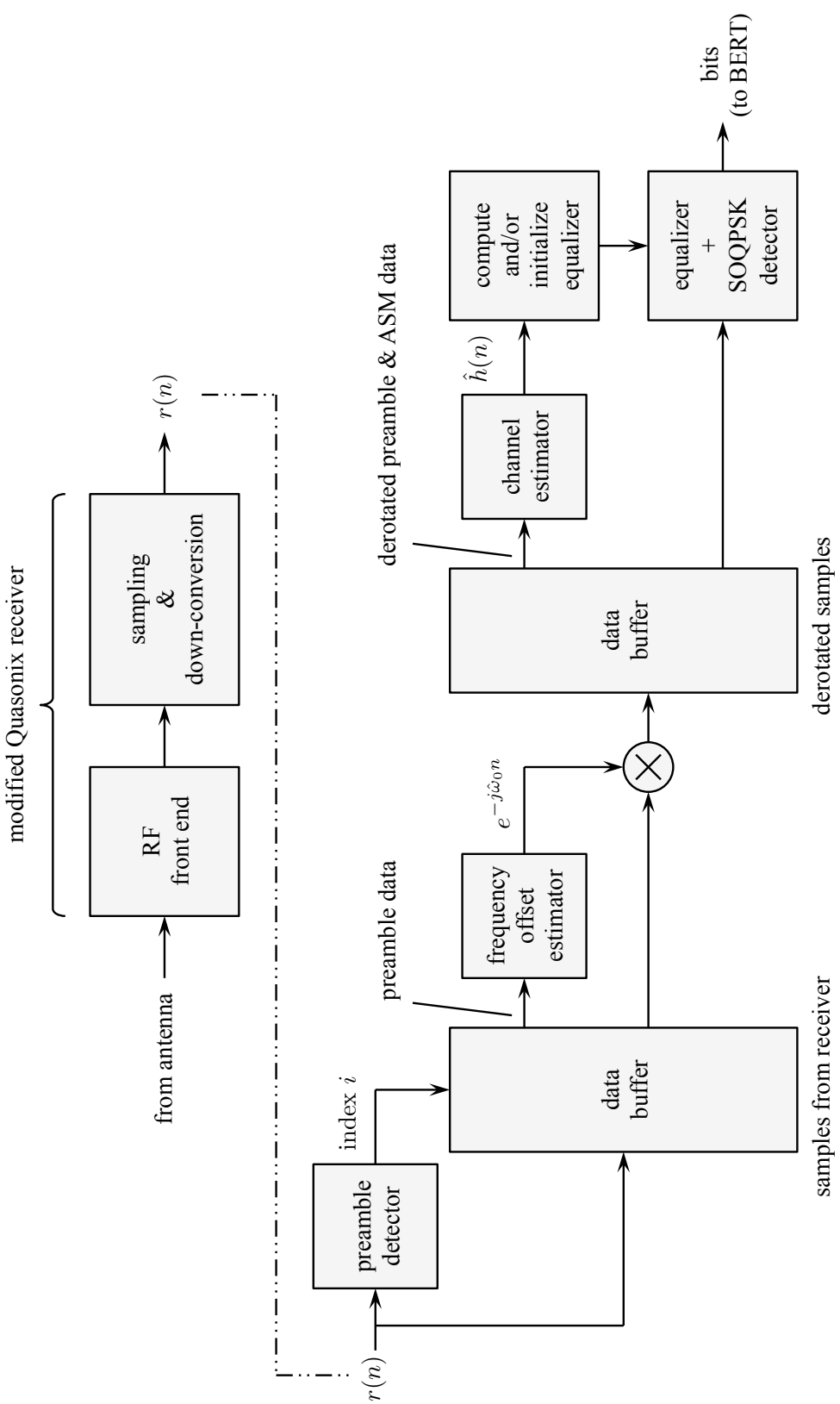

Figure 3: A high-level view of the signal processing requirements to support the equalizers. The emphasis here is on preamble detection, frequency offset estimation, and channel estimation. A more detailed description of the equalizers is given in Section 8.

rate of 10.3125 Mbits/s: that is the sample rate for the desired channel is 20.625 Msamples/s. Consequently, the first step is resampling $h_{\text{M4A}}(n)$ by the factor ${}^{33}/_{320}$. Call this $g(n)$.

2. Clean-up: To remove the noise from $g(n)$, we adopt a per-cent total power criterion. This criterion is applied as follows:

   (a) Form the temporary vector $T(i)$ where the elements in $T(i)$ are the elements of $g(n)$ ordered from largest to smallest in magnitude. That is, if

$$n_0 = \underset{-N_1 \le n \le N_2}{\operatorname{argmax}} \Big\{ g(n) \Big\},$$

$$n_1 = \underset{\substack{-N_1 \le n \le N_2 \\ n \ne n_0}}{\operatorname{argmax}} \Big\{ g(n) \Big\},$$

$$n_2 = \underset{\substack{-N_1 \le n \le N_2 \\ n \ne n_0, n_1}}{\operatorname{argmax}} \Big\{ g(n) \Big\}$$

   and so on, then $T(0) = g(n_0)$, $T(1) = g(n_1)$, $T(2) = g(n_2)$, etc.

   (b) Next, find the index I that captures 90% of the total energy in $g(n)$:

$$0.90 \times E_{\text{tot}} = \sum_{i=0}^{I-1} |T(i)|^2 \quad \text{where} \quad E_{\text{tot}} = \sum_{n=-N_1}^{N_2} |g(n)|^2.$$

   With the index I determined, define the set of indexes $\mathcal{N}_I$ as follows:

$$\mathcal{N}_I = \Big\{ n_0, n_1, \ldots n_{I-1} \Big\}.$$

   (c) Lastly "prune" $g(n)$ as follows:

$$h(n) = \begin{cases} g(n) & n \in \mathcal{N}_I \\ 0 & \text{otherwise.} \end{cases}$$

The eleven impulse responses are summarized in Table 1. The impulse responses and frequency responses for the eleven test channels are displayed in Figures 4 – 14.

Table 1: A summary of the eleven test channels derived from the M4A channel sounding experiments at Edwards AFB, CA.

channel	N_1	N_2	channel length	location
1	1	7	9	Taxiway E
2	2	17	20	Taxiway E
3	1	22	24	Taxiway E
4	6	12	19	Takeoff on 22L
5	1	1	3	Cords Road (W-E)
6	1	2	4	Cords Road (W-E)
7	0	4	5	Cords Road (W-E)
8	2	3	6	Black Mountain (W-E)
9	1	1	3	Black Mountain (E-W)
10	2	3	6	Final Approach/Land on 22L
11	3	2	6	Final Approach/Land on 22L

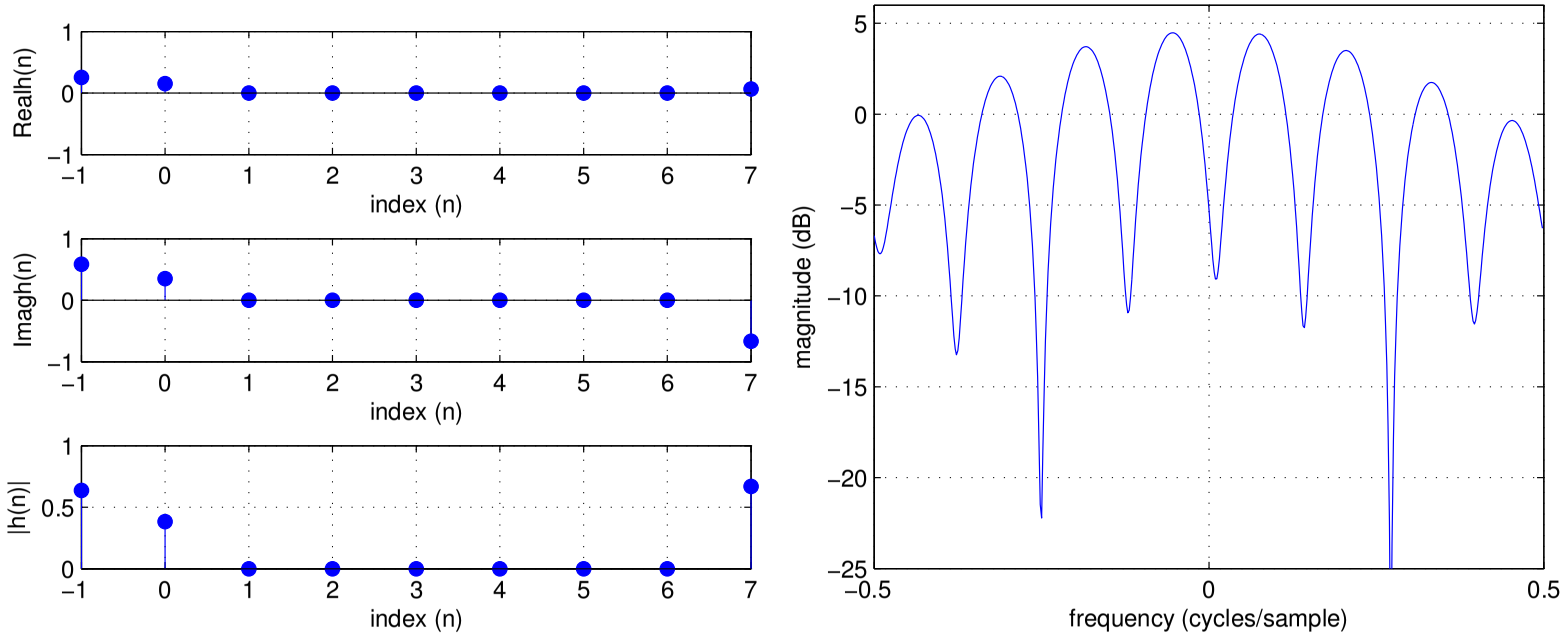

Figure 4: The impulse response (left) and frequency response (right) for test channel 1. This channel was measured on Taxiway E.

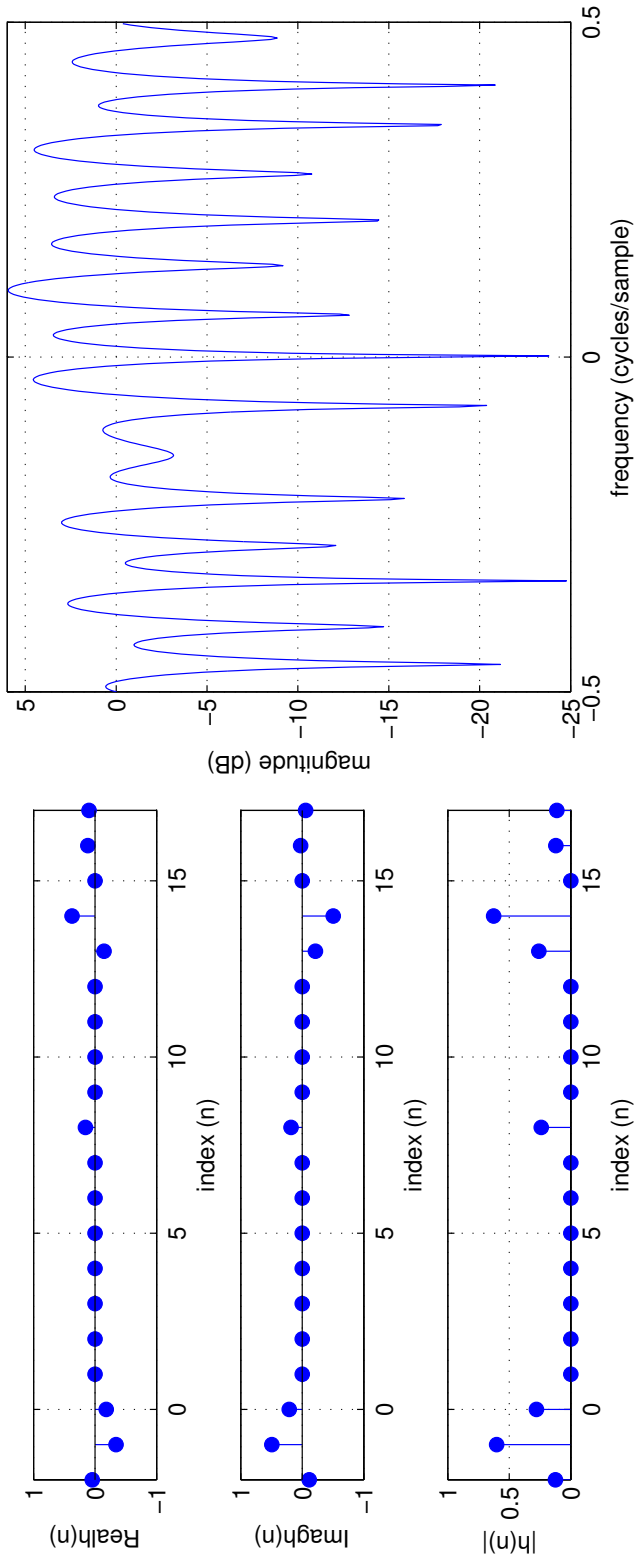

Figure 5: The impulse response (left) and frequency response (right) for test channel 2. This channel was measured on Taxiway E.

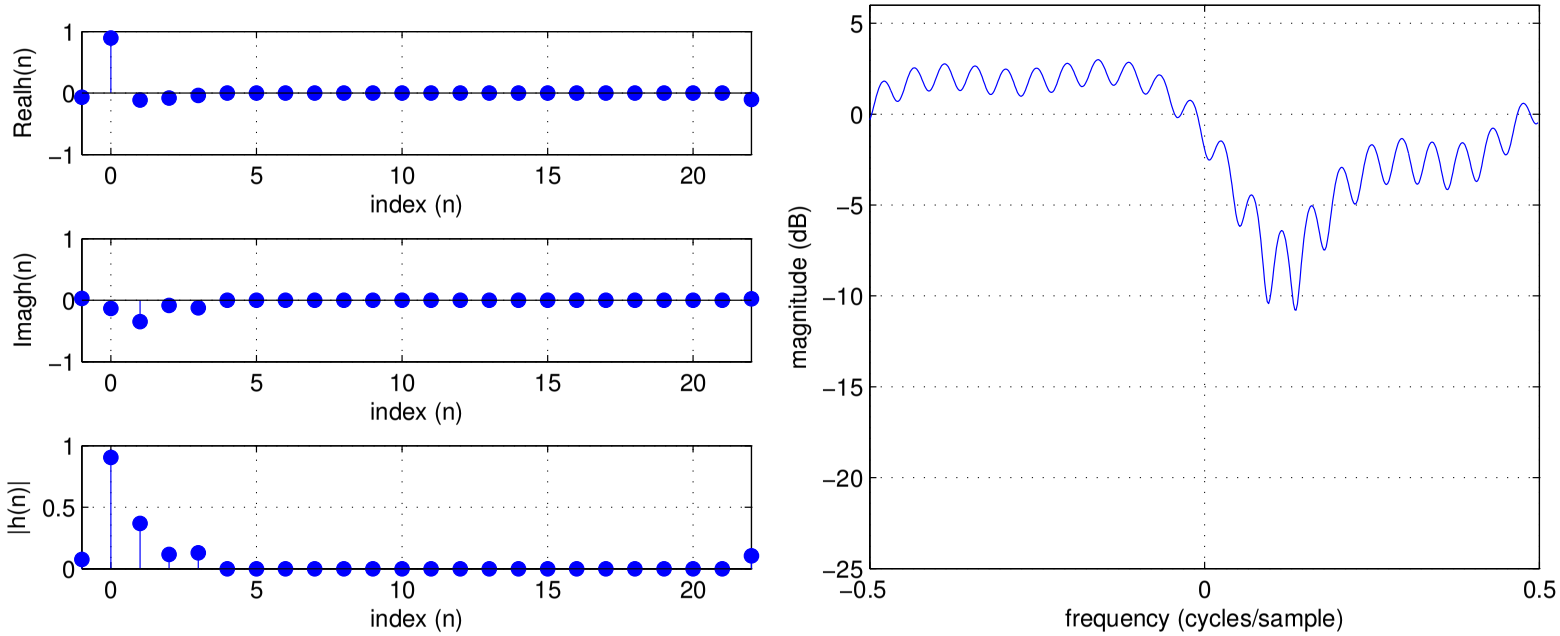

Figure 6: The impulse response (left) and frequency response (right) for test channel 3. This channel was measured on Taxiway E.

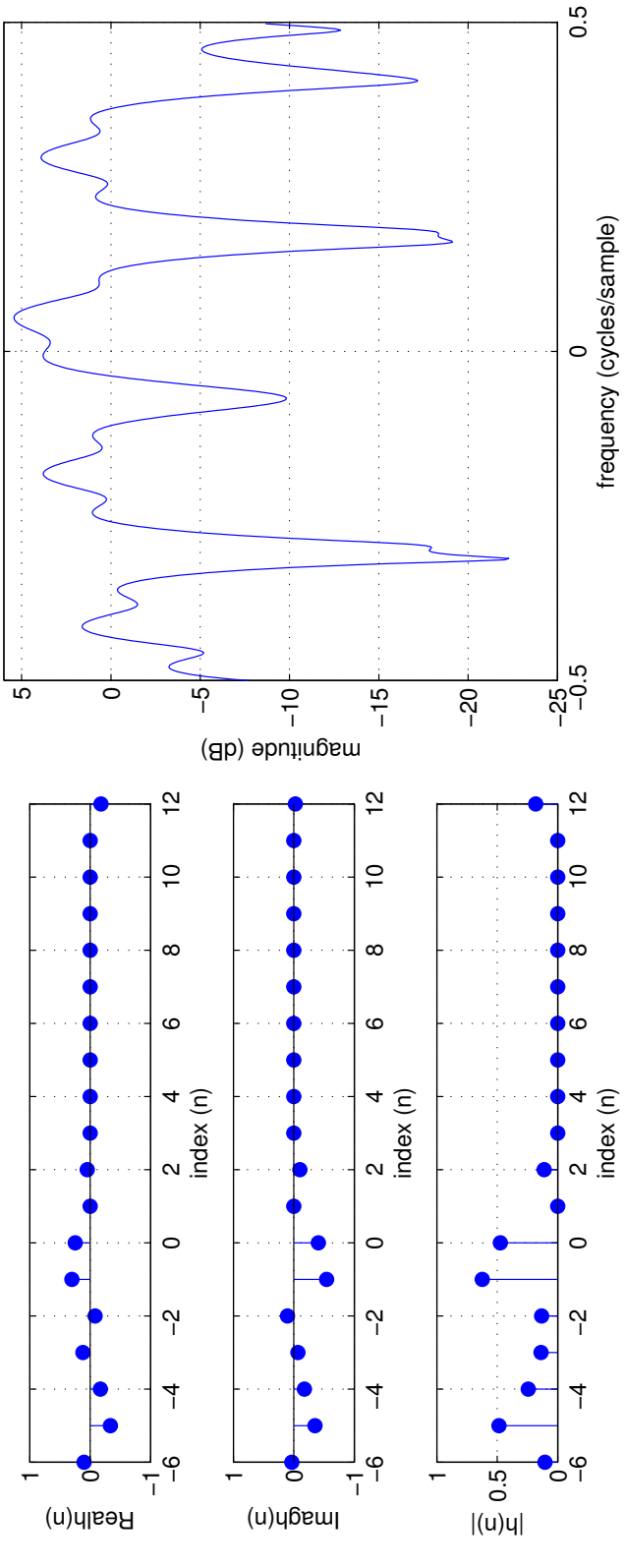

Figure 7: The impulse response (left) and frequency response (right) for test channel 4. This channel was measured at take-off on runway 22L.

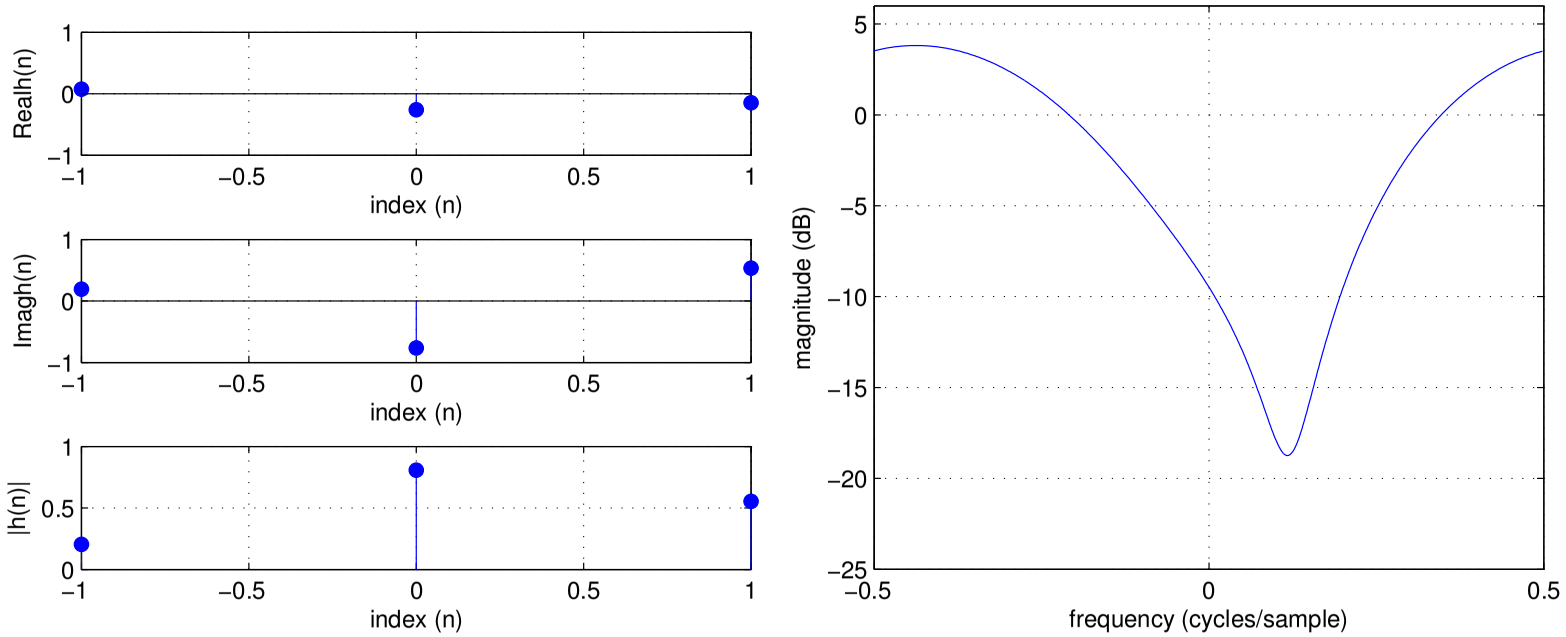

Figure 8: The impulse response (left) and frequency response (right) for test channel 5. This channel was measured on the Cords Road flight path.

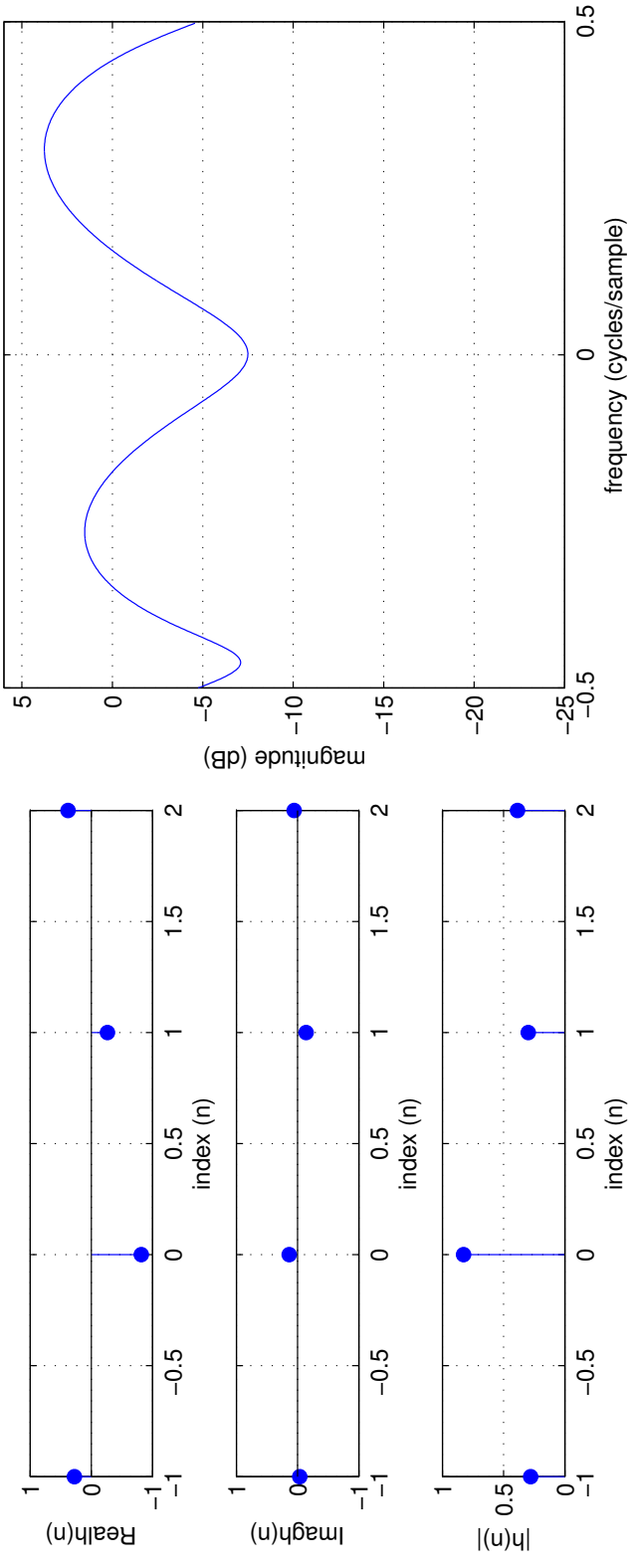

Figure 9: The impulse response (left) and frequency response (right) for test channel 6. This channel was measured on the Cords Road flight path.

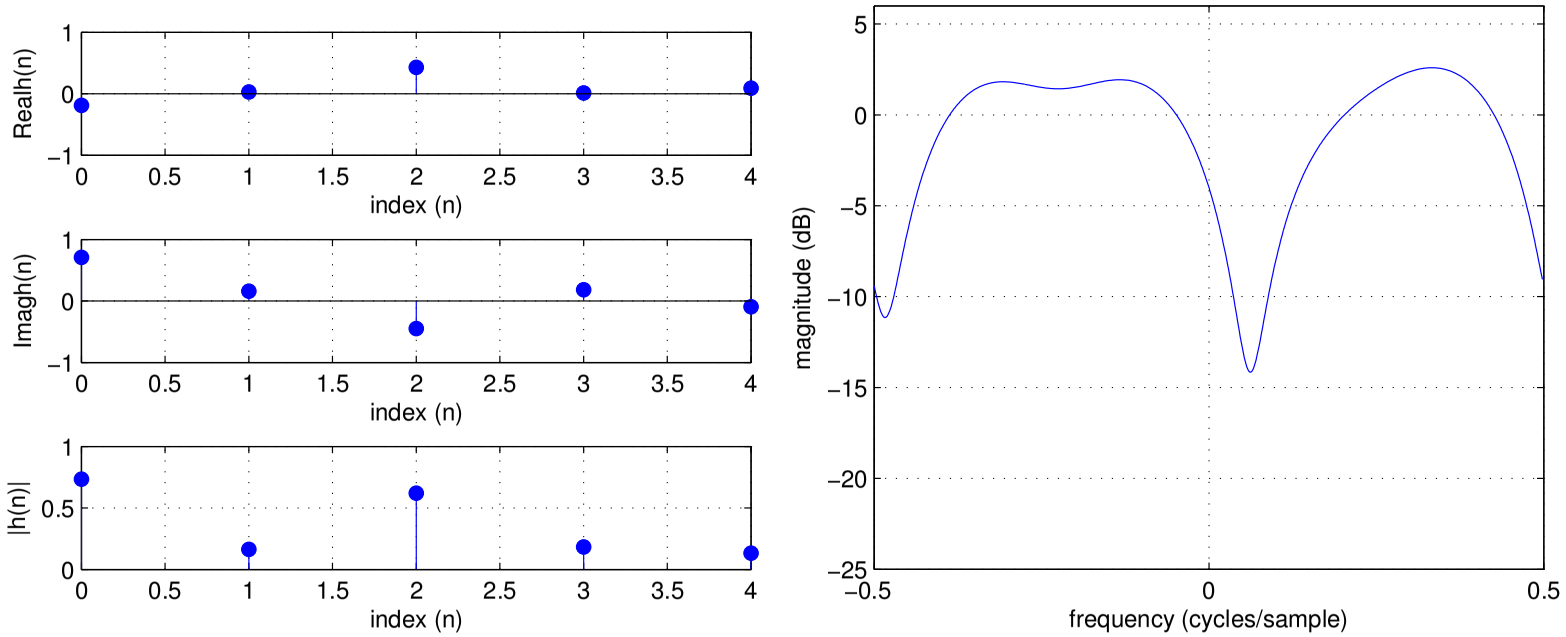

Figure 10: The impulse response (left) and frequency response (right) for test channel 7. This channel was measured on the Cords Road flight path.

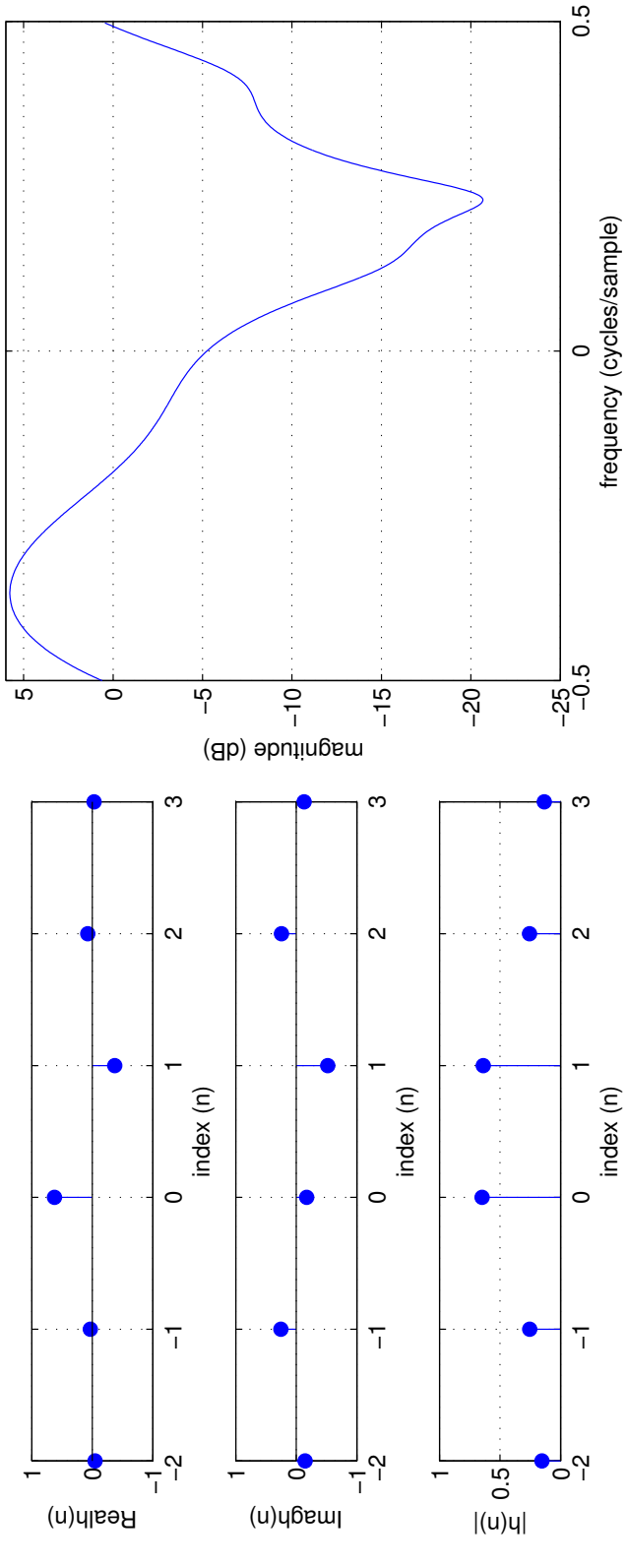

Figure 11: The impulse response (left) and frequency response (right) for test channel 8. This channel was measured on the Black Mountain flight path.

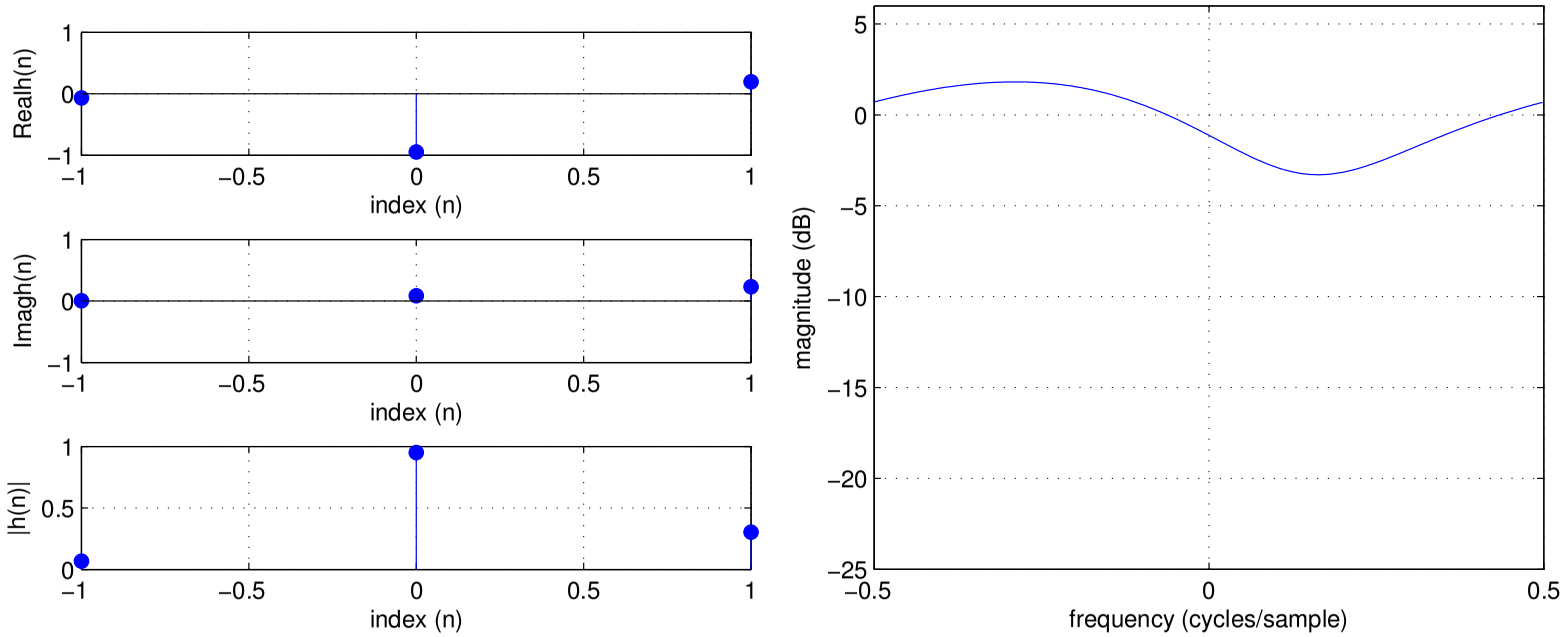

Figure 12: The impulse response (left) and frequency response (right) for test channel 9. This channel was measured on the Black Mountain flight path.

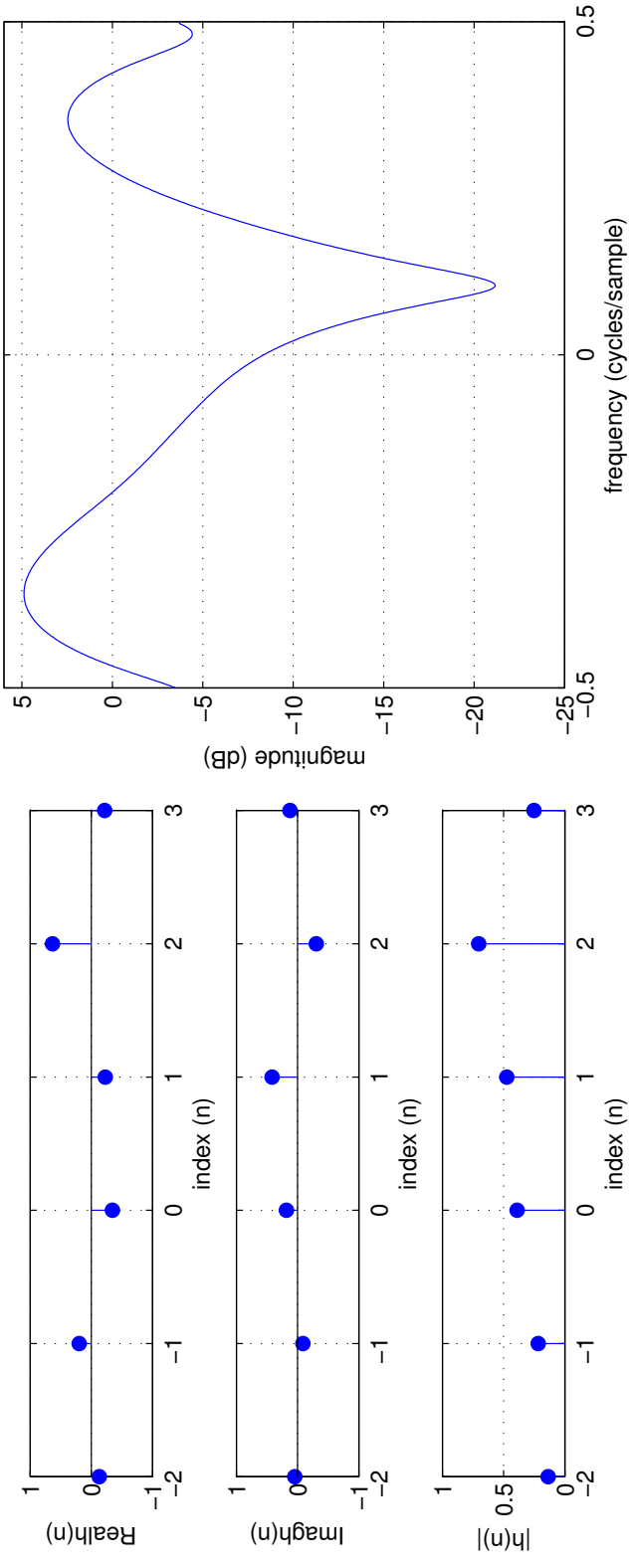

Figure 13: The impulse response (left) and frequency response (right) for test channel 10. This channel was measured on final approach and landing on runway 22L.

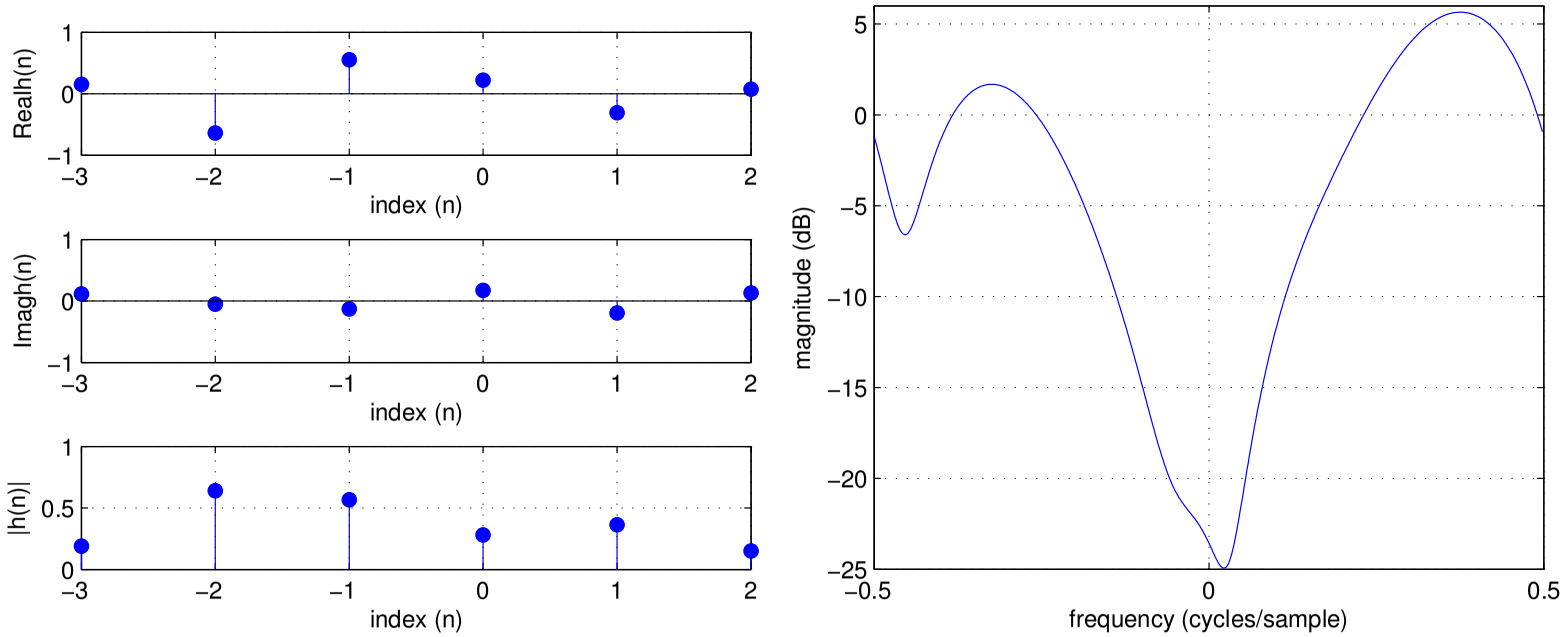

Figure 14: The impulse response (left) and frequency response (right) for test channel 11. This channel was measured on final approach and landing on runway 22L.

2 Preamble Detector

The frequency offset and channel estimators outlined in this report require knowledge of the position of the preamble samples in the stream of received samples. The location is given by the index i, the index of the sample corresponding to the beginning of the preamble. But how is the position of preamble sequence known? The preamble detector is parses the sequence of received samples to find the first sample of the preamble. The preamble detector comprises two parts: a correlation or correlation-type function $L(u)$ that produces a real-valued quantity for each index u, and a decision function that uses the $L(u)$ to decide if the start of the preamble has been detected.

As an illustration, we consider the AWGN scenario where phase, frequency, and timing have all been established. Here the received samples are $r(n) = s(n)+w(n)$ and the optimum preamble detector is a simple correlator:

$$L(u) = \sum_{n=u}^{u+L_p-1} r(n)p^*(n-u) \tag{4}$$

where $p(\cdot)$ is a locally stored copy of the sequence of SOQPSK-TG samples corresponding to the preamble bits. When the $r(n)$ are aligned with the $p(\cdot)$, $L(u)$ is large and the resulting peak signals the presence of the preamble sequence at index $u = i$. Given the iNET packet structure, this means that $s(i)$ is the first sample of the preamble sequence in the transmitted signal. Consequently, we have

$$\begin{aligned} s(i) &= p(0) \\ s(i+1) &= p(1) \\ &\vdots \\ s(i+N_\mathrm{p}-1) &= p(L_\mathrm{p}-1) \end{aligned}$$

where $p(n)$ is the n-th sample of the SOQPSK-TG signal corresponding to the preamble bit sequence. Now, the i samples preceding $s(i)$ correspond to unknown data. To capture this, let

$$\begin{aligned} s(0) &= d_1(0) \\ s(1) &= d_1(1) \\ &\vdots \\ s(i-1) &= d_1(i-1) \end{aligned}$$

where $d_1(\cdot)$ is used to denote the unknown "data" samples preceding the preamble block of interest. Furthermore, the $L_{\text{asm}} = 128$ samples that follow $s(i + L_{\text{p}} - 1)$ are the SOQPSK-TG samples corresponding to the ASM field. To capture this, we write

$$\begin{aligned} s(i + L_{\text{p}}) &= a(0) \\ s(i + L_{\text{p}} + 1) &= a(1) \\ &\vdots \\ s(i + L_{\text{p}} + L_{\text{asm}} - 1) &= a(L_{\text{asm}} - 1) \end{aligned}$$

where $a(\cdot)$ represents the samples corresponding to the ASM bits. After the ASM bits, we encounter a second block of N_d unknown "data" samples which we denote $d_2(\cdot)$:

$$\begin{aligned} s(i + L_{\text{p}} + L_{\text{asm}}) &= d_2(0) \\ s(i + L_{\text{p}} + L_{\text{asm}} + 1) &= d_2(1) \\ &\vdots \\ s(i + L_{\text{p}} + L_{\text{asm}} + N_d - 1) &= d_2(N_d - 1) \end{aligned}$$

All this summarized nicely in Figure 15.

In our case, an unknown channel and unknown frequency offset alter the simple model $r(n) = s(n) + w(n)$ and the simple correlation function (7) will not work. As stated in Section 1, the relationship between the samples of the transmitted signal $s(n)$ and samples of the received signal $r(n)$ is

$$r(n) = \Big[s(n) * h(n)\Big] e^{j\omega_0 n} + w(n) = \left[\sum_{k=-N_1}^{N_2} h(k)s(n-k)\right] e^{j\omega_0 n} + w(n) \tag{5}$$

where where $h(n)$ is the equivalent discrete-time channel between the transmitter and the receiver, ω_0 rads/sample is the frequency offset, and $w(n)$ is a complex-valued, zero-mean discrete-time Gaussian random process with autocovariance function

$$\frac{1}{2}\text{E}\Big\{w(n)w^*(n-k)\Big\} = \sigma_w^2\delta(k). \tag{6}$$

The preamble detector must deal with the fact that the phase, frequency, timing offset and channel are all unknown. We investigate two preamble detection functions here. The first is the maximum

$$
\begin{array}{ccccccccccccc}
s(0) & \cdots & s(i-1) & s(i) & \cdots & s(i+L_{\mathrm{p}}-1) & s(i+L_{\mathrm{p}}) & \cdots & s(i+L_{\mathrm{p}}+L_{\mathrm{asm}}-1) & s(i+N_{\mathrm{p}}+N_{\mathrm{asm}}) & \cdots \\
d_1(0) & \cdots & d_1(i-1) & p(0) & \cdots & p(L_{\mathrm{p}}-1) & a(0) & \cdots & a(L_{\mathrm{asm}}-1) & d_2(0) & \cdots
\end{array}
$$

Figure 15: The relationship between the samples in $s(\cdot)$ and the samples of the constituent signals.

likelihood estimator and a simplified, low-complexity approximation.

2.1 The Maximum Likelihood Preamble Detector

The issue of preamble detection (or "frame synchronization") for binary data in Gaussian noise was first studied by Barker [2] who showed that good performance was achievable by maximizing a simple correlation function. Massey [3] derived the maximum likelihood (ML) preamble detector for binary data in Gaussian noise. The derivation accounted for random data preceding and following the preamble. Massey's preamble detector comprised Barker's simple correlation function and a bias term. Lui and Tan [4] generalized Massey's development to M-ary data ($M > 2$) and showed the same thing: the ML preamble detector maximizes a function comprising a simple correlation and a bias term. Gansman *et al.* [5] tackled the problem of unknown carrier frequency and phase. Treating the unknown frequency and phase as nuisance parameters with uniform prior distributions, they derive a preamble detector that is tolerant of both frequency and phase offsets. Choi and Lee [6] took an alternation approach to developing the ML detector in the presence of unknown frequency and phase offsets and derived a "double correlation" function capable of providing good performance over a larger frequency offset than the Gansman preamble detector [5].

Here we follow the same steps and apply the concepts of correlator and "double correlator" to samples of the received waveform. Let $p(n)$ for $0 \leq n < L_p$ be the locally stored copy of the sequence of SOQPSK-TG samples corresponding to the preamble bits. The relationship between the indexes of the transmitted signal, the received samples, and the locally stored copy of the preamble samples is illustrated in Figure 16.

For the simple synchronous AWGN case, we have $r(n) = s(n) + w(n)$ and the preamble detector is one that maximizes the absolute value of the simple correlation function:

$$L_{\text{AWGN}}(u) = \left| \sum_{n=u}^{u+L_p-1} r(n)p^*(n-u) \right|. \tag{7}$$

Let i be the index in $r(n)$ corresponding to the beginning of the preamble sequence. When the $r(n)$ are aligned with the $p(\cdot)$, $L_{\text{AWGN}}(u)$ is large and the resulting peak signals the presence of the preamble sequence at index $u = i$.

In the case of a phase and frequency offset, the received samples are given by

$$r(n) = s(n)e^{j(\omega_0 n+\theta)} + w(n). \tag{8}$$

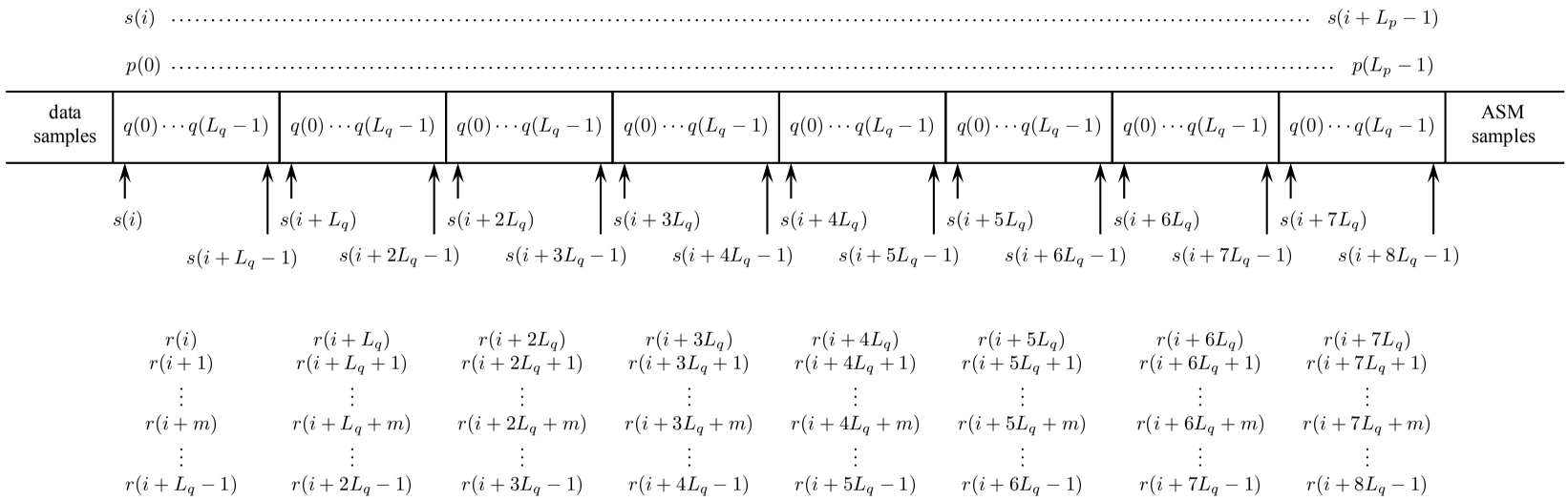

Figure 16: A graphical illustration of the relationship between the indexes of the received samples $r(n)$, the signal samples $s(n)$, the preamble samples $p(n)$ and the short sequences $q(n)$ that constitute the preamble sequence.

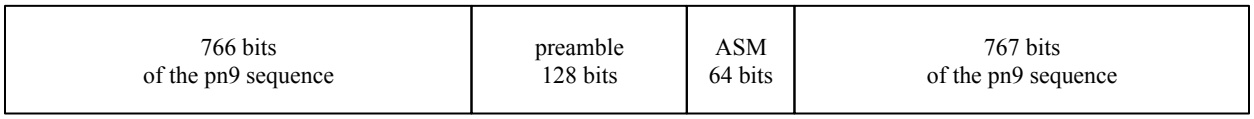

Figure 17: The data format used for to test the behavior of $L(u)$.

In this case, correlation function (7) is inadequate in the presence of a frequency offset because the phase rotations caused by the frequency offset lead to destructive cancellation in the summation. The result is a small value for the summation, which when compared to a threshold, gives at best ambiguous information. The Choi and Lee correlation function (called $L_1(u)$ in [6]) is

$$L_{\mathrm{CL}}(u) = \sum_{i=0}^{L_p-1} \left\{ \left| \sum_{k=0}^{L_p-i-1} r^*(u+k)p(k)r(u+k+i)p^*(k+i) \right| - \sum_{k=u+i}^{u+L_p-1} |r(k)|\,|r(k-i)| \right\}. \quad (9)$$

This function works amazingly well, but is computationally complex. Each sample of $L_{\mathrm{CL}}(u)$ requires

$$N_{\mathbb{C}\times\mathbb{C}} = 3 \times \frac{L_p(L_p+1)}{2} \quad (10)$$

complex-by-complex multiplications and

$$N_{\mathbb{R}\times\mathbb{R}} = \frac{L_p(L_p+1)}{2} \quad (11)$$

real-by-real multiplications. The total number of real-by-real multiplications for each sample of $L_{\mathrm{CL}}(u)$ is $N_{\mathbb{C}\times\mathbb{C}} + N_{\mathbb{R}\times\mathbb{R}}$. For example, at an equivalent sample rate of 2 samples/bit (the lowest sample rate that satisfies the sampling theorem), we have $L_p = 256$ and the number of real-by-real multiplications is $427{,}648$.

In the aeronautical telemetry, the transmitted signal experiences multipath propagation in addition to a phase and frequency offset. Here, the received samples are given by (5). The performance of $L_{\mathrm{CL}}(u)$ in the presence of a frequency offset and the multipath channel is the open question.

To answer the question, samples of an SOQPSK-TG signal corresponding to the bit sequence shown in Figure 17 were generated at an equivalent sample rate of 2 samples/bit and used as the input to $L_{\mathrm{CL}}(u)$. A plot of $L_{\mathrm{CL}}(u)$ for the case of no noise, no frequency offset, a perfect channel [$h(n) = \delta(n)$], is shown in Figure 18. In this figure, the index axis is shifted so that the true starting index of the preamble corresponds to $u = 0$. A series of correlation peaks centered about $u = 0$ is clearly evident. The presence of multiple peaks is due to the periodic property of the preamble (i.e., the 8-times repetition of the 16-bit sequence). Even though there are multiple peaks, the

maximum peak occurs at $u = 0$ and correctly marks the start of the preamble. The behavior of $L_{\text{CL}}(u)$ over the eleven test channels is displayed in Figures 19 – 29. In each case, the highest peak occurs at the correct location. This shows that the Choi-Lee function $L_{\text{CL}}(u)$ is capable of correctly identifying the preamble in the presence of corruption by multipath channels likely to be encountered in aeronautical telemetry.

The behavior of $L_{\text{CL}}(u)$ in the presence of a frequency offset of 50 kHz in the case of no noise and an ideal channel is shown in Figure 30. The values produced by $L_{\text{CL}}(u)$ in the presence of a 50 kHz offset are quite similar to those produced by $L_{\text{CL}}(u)$ with no frequency offset (cf. Figure 18). This shows that the function $L_{\text{CL}}(u)$ is not impacted by frequency offsets in the range typically encountered in aeronautical telemetry.

The performance of $L_{\text{CL}}(u)$ in the presence of both a 50 kHz frequency offset and a multipath channel is illustrated by the plots in Figures 31 – 41. In all cases, $L_{\text{CL}}(u)$ correctly identifies the start of the preamble.

In summary, the Choi-Lee correlation function $L_{\text{CL}}(u)$ correctly identifies the start of the preamble in the conditions encountered in aeronautical telemetry, namely a frequency offset and multipath channel. However, as pointed out above, this is accomplish at the expense of computational complexity. Motivated by the high computational burden of computing $L_{\text{CL}}(u)$ for each sample, we develop a low-complexity, sub-optimal preamble detector in the next section.

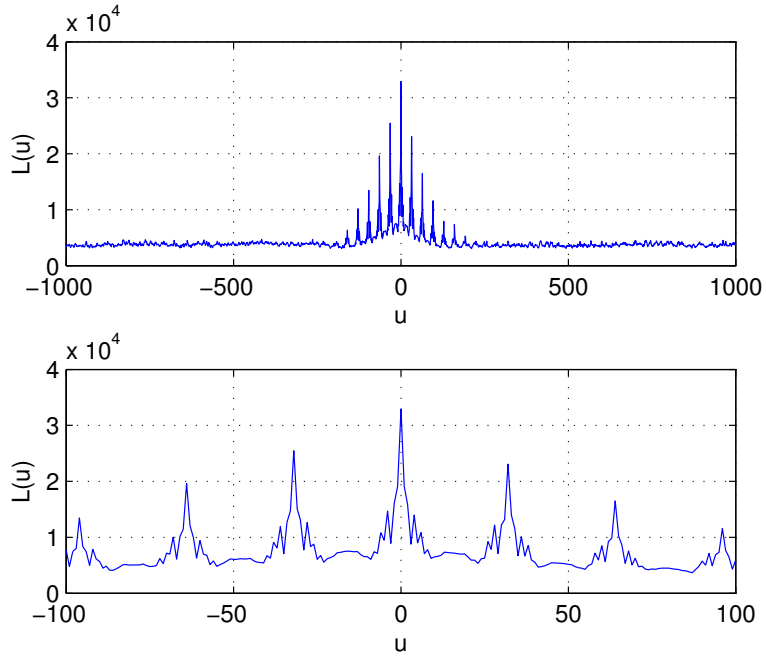

Figure 18: A plot of $L_{\mathrm{CL}}(u)$ for an ideal channel and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

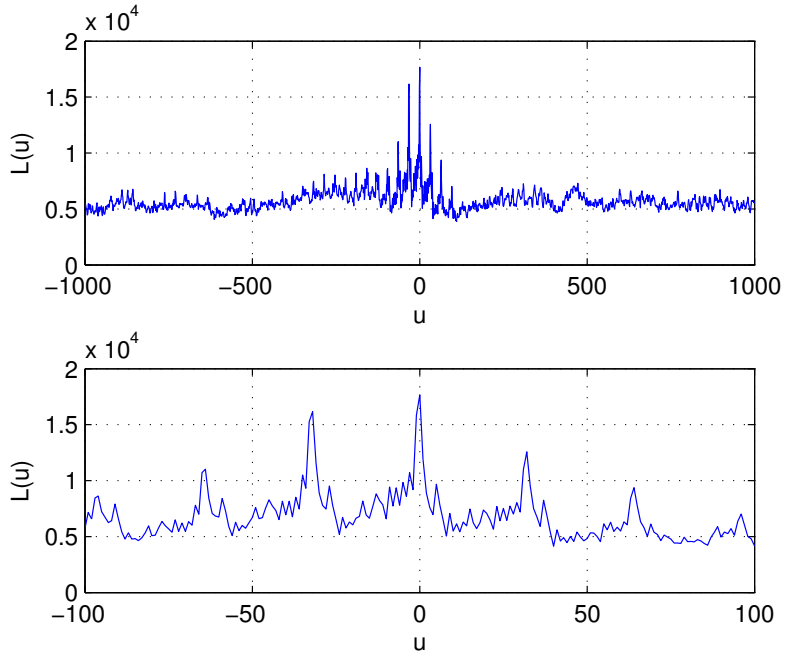

Figure 19: A plot of $L_{\mathrm{CL}}(u)$ for test channel 1 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

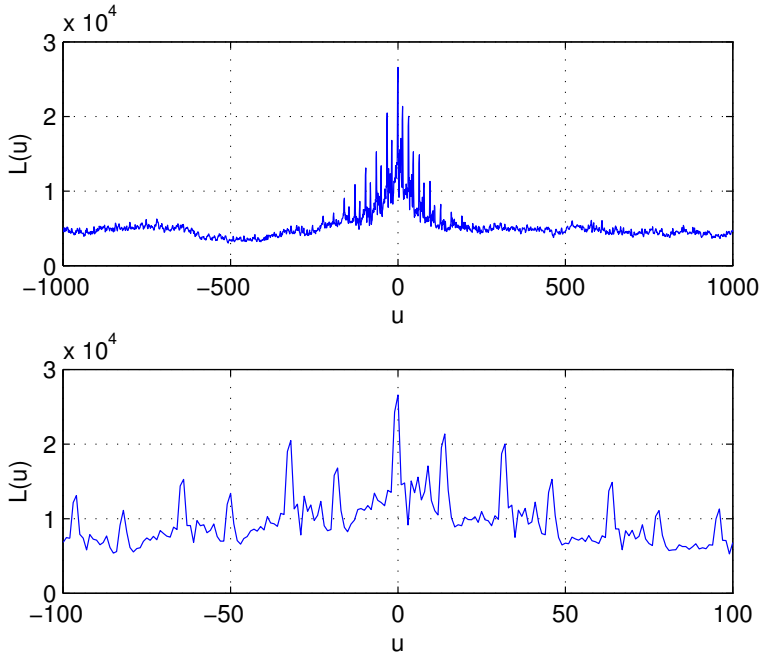

Figure 20: A plot of $L_{\mathrm{CL}}(u)$ for test channel 2 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

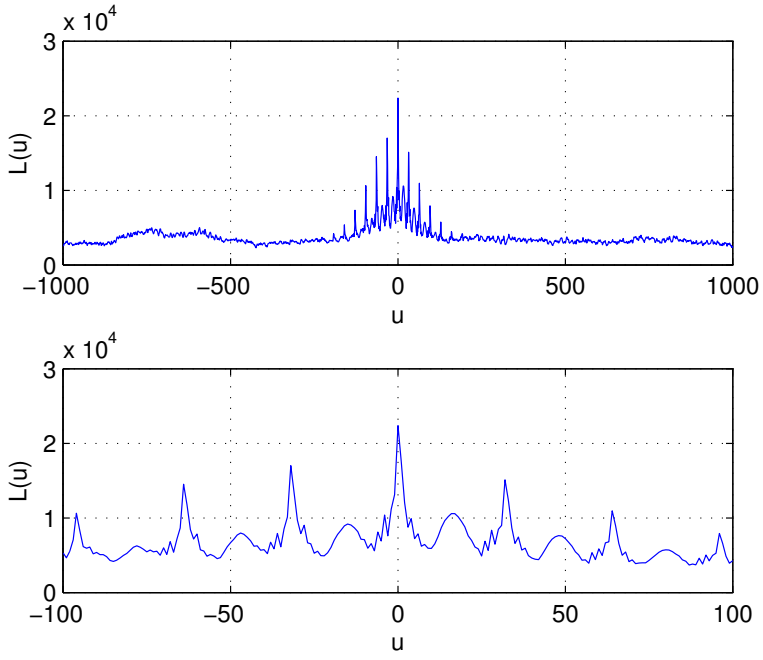

Figure 21: A plot of $L_{\mathrm{CL}}(u)$ for test channel 3 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

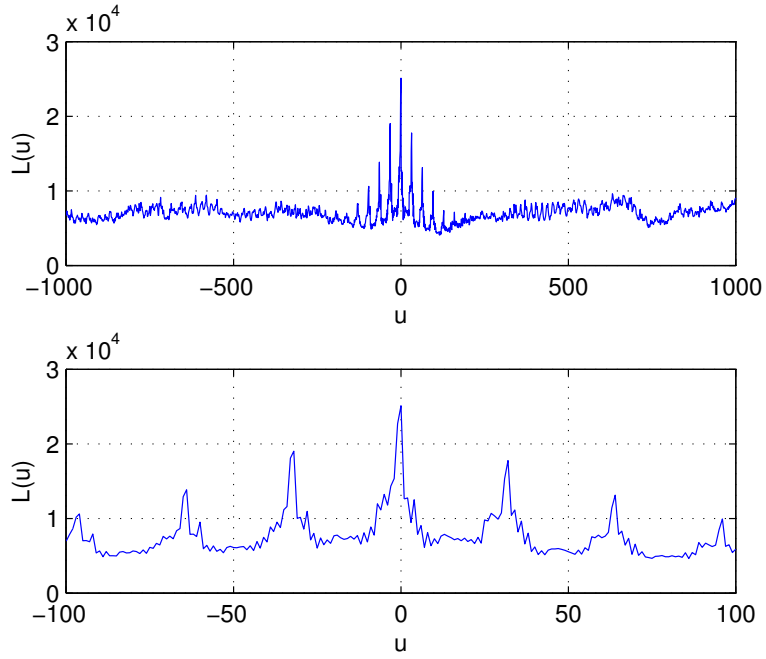

Figure 22: A plot of $L_{\mathrm{CL}}(u)$ for test channel 4 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

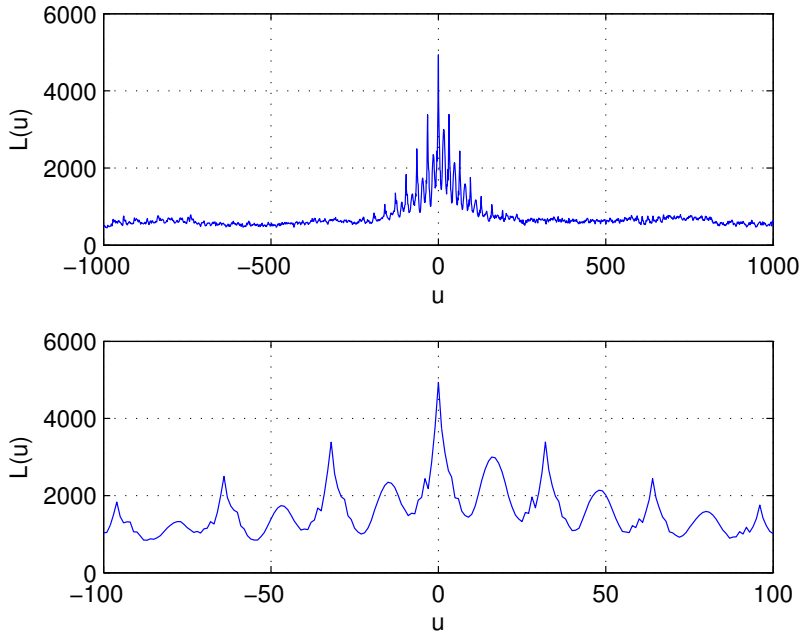

Figure 23: A plot of $L_{\mathrm{CL}}(u)$ for test channel 5 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

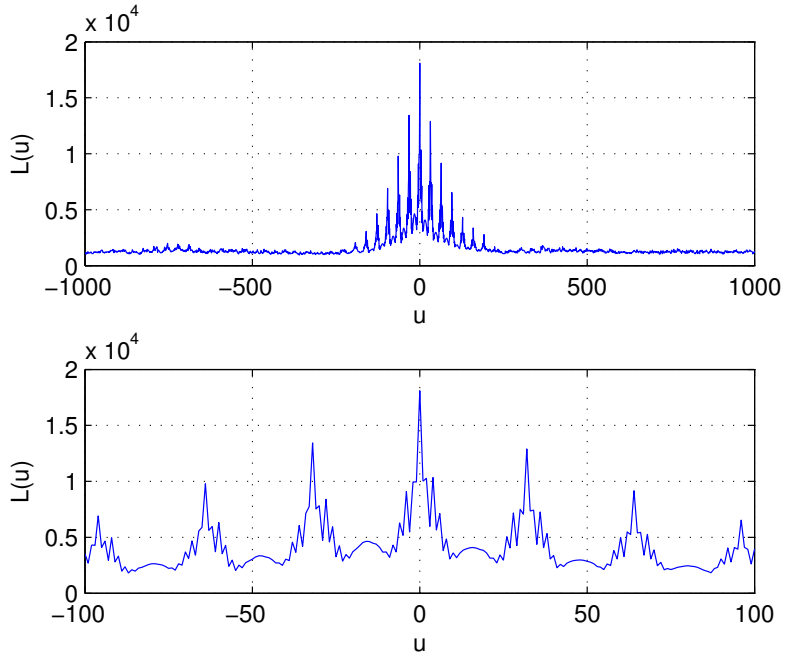

Figure 24: A plot of $L_{\mathrm{CL}}(u)$ for test channel 6 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

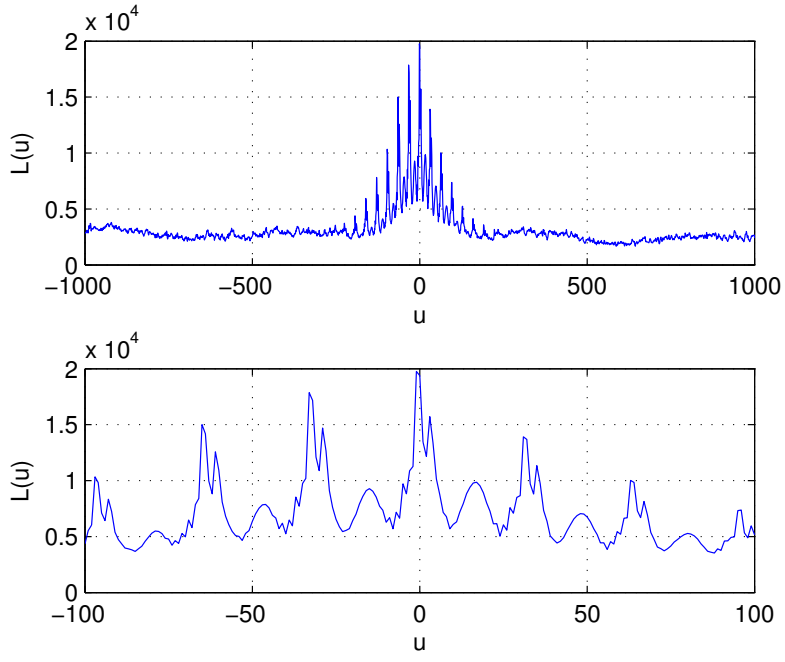

Figure 25: A plot of $L_{\mathrm{CL}}(u)$ for test channel 7 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

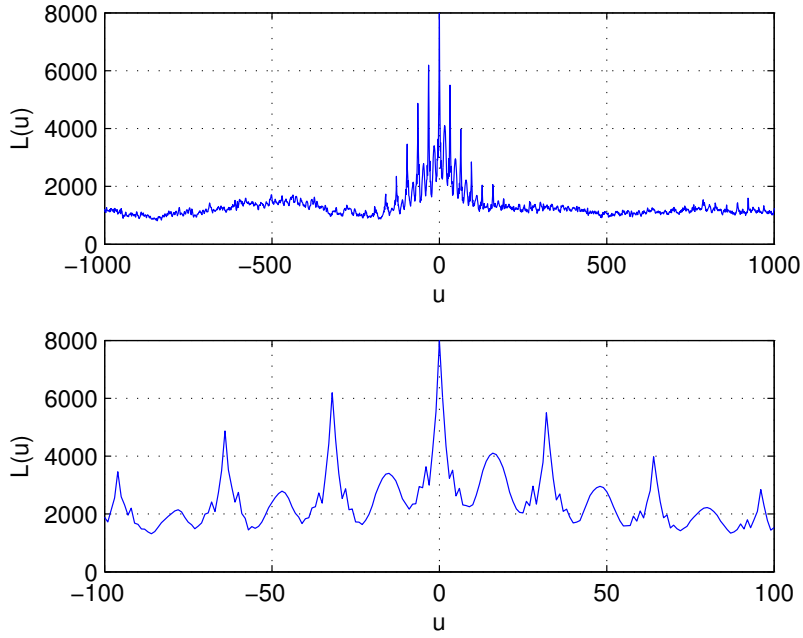

Figure 26: A plot of $L_{\mathrm{CL}}(u)$ for test channel 8 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

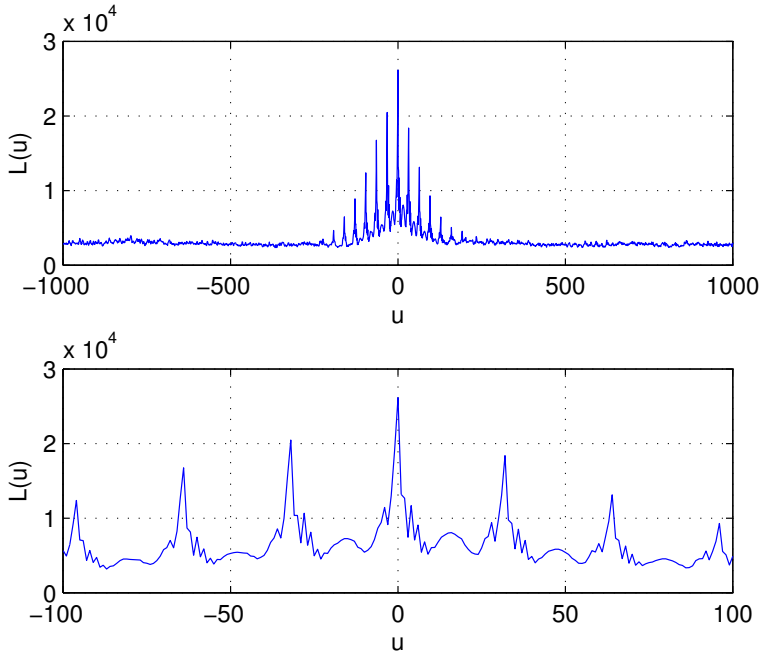

Figure 27: A plot of $L_{\mathrm{CL}}(u)$ for test channel 9 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

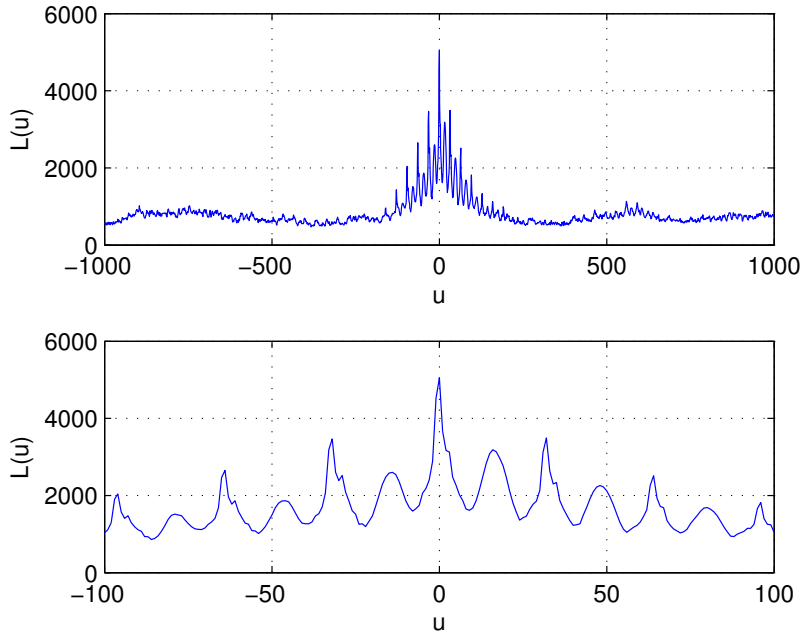

Figure 28: A plot of $L_{\mathrm{CL}}(u)$ for test channel 10 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

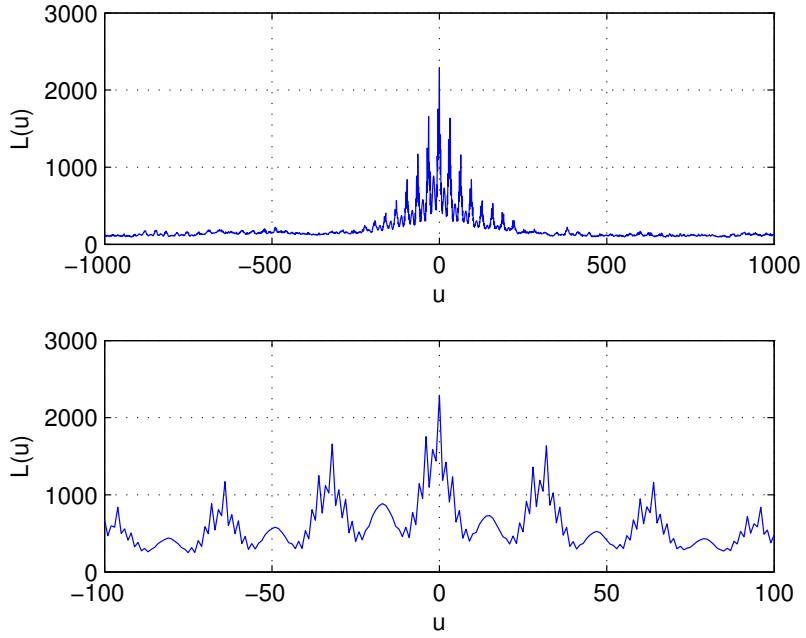

Figure 29: A plot of $L_{\mathrm{CL}}(u)$ for test channel 11 and no frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

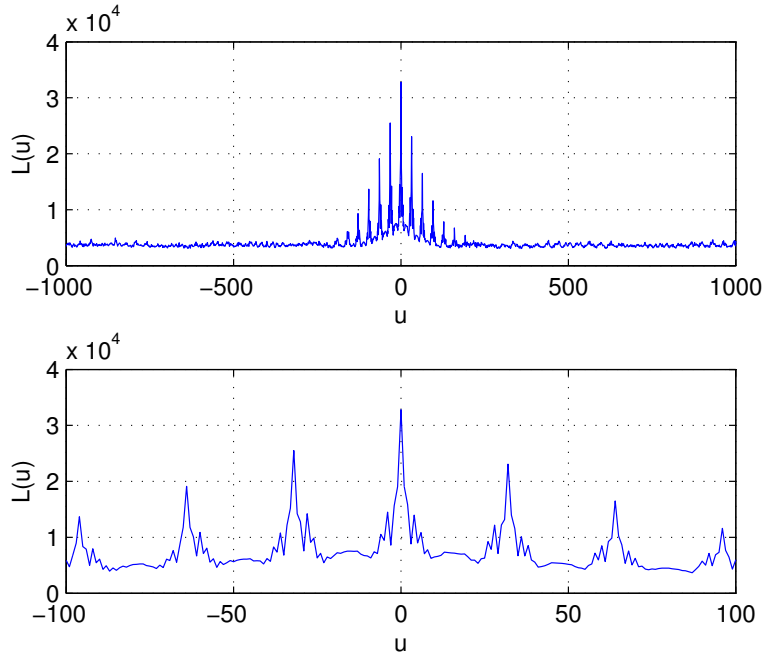

Figure 30: A plot of $L_{\mathrm{CL}}(u)$ for an ideal channel and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

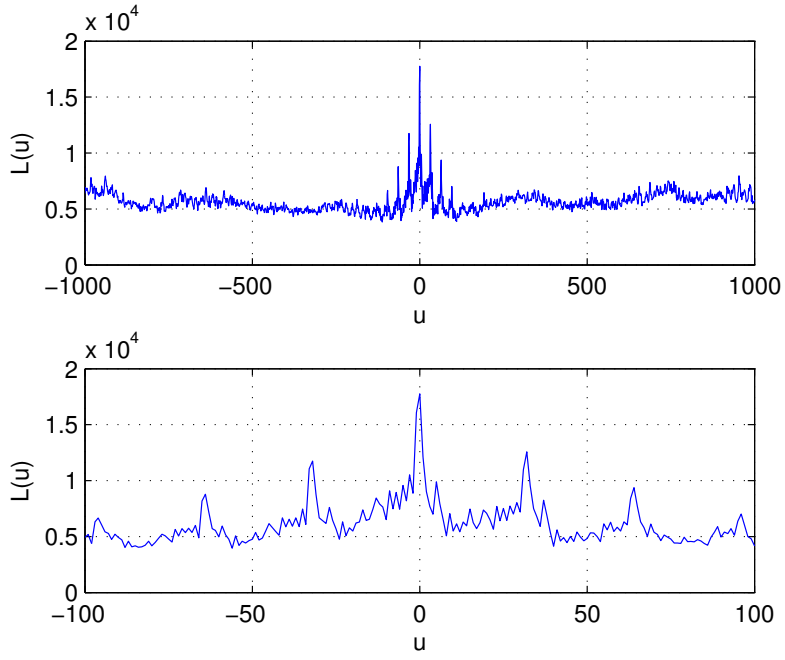

Figure 31: A plot of $L_{\mathrm{CL}}(u)$ for test channel 1 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

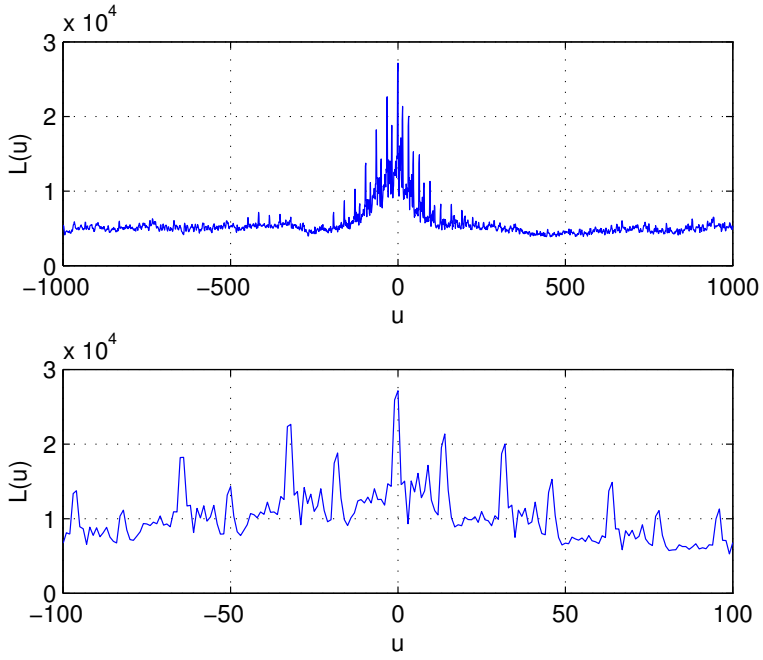

Figure 32: A plot of $L_{\mathrm{CL}}(u)$ for test channel 2 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

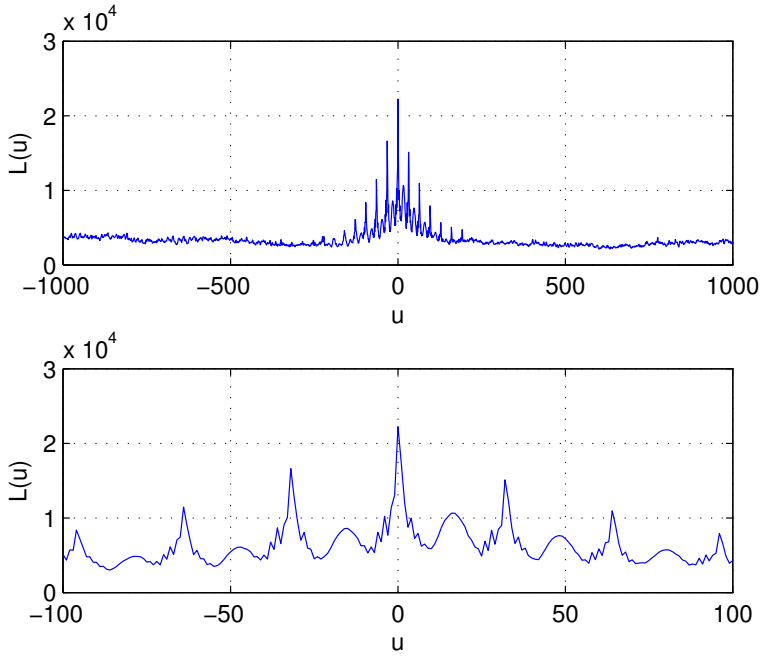

Figure 33: A plot of $L_{\mathrm{CL}}(u)$ for test channel 3 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

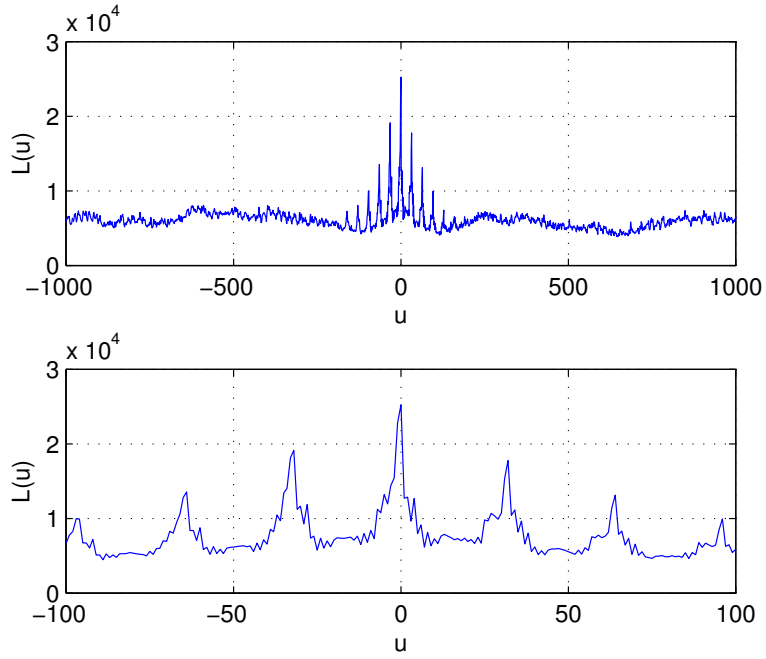

Figure 34: A plot of $L_{\mathrm{CL}}(u)$ for test channel 4 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

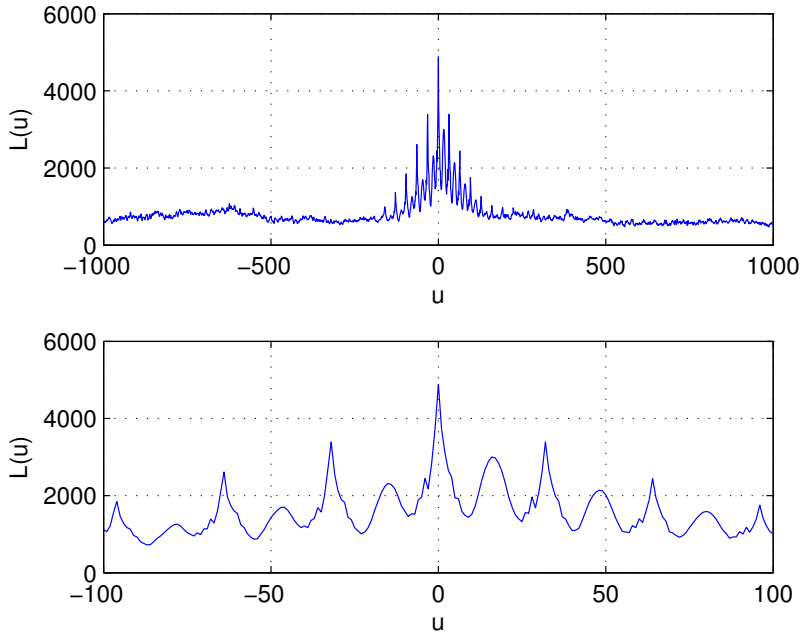

Figure 35: A plot of $L_{\mathrm{CL}}(u)$ for test channel 5 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

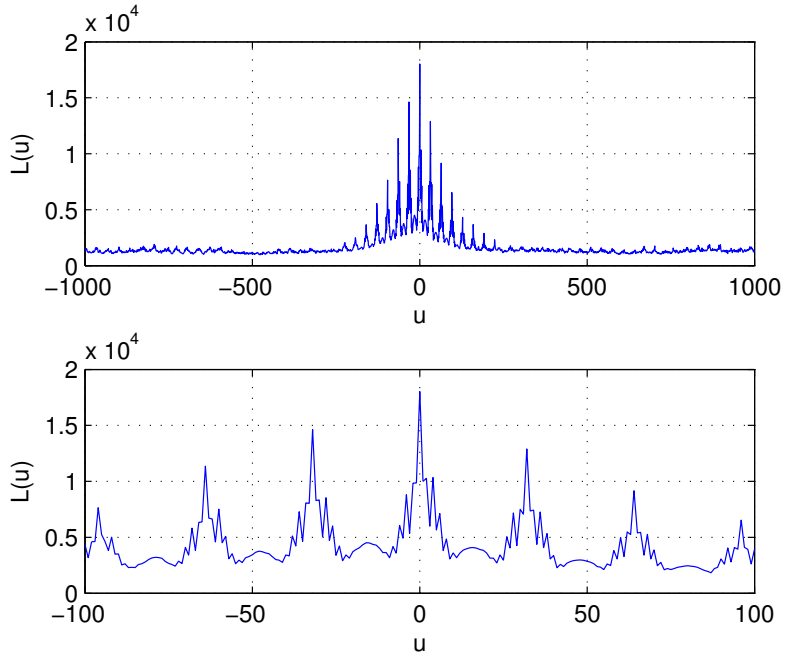

Figure 36: A plot of $L_{\mathrm{CL}}(u)$ for test channel 6 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

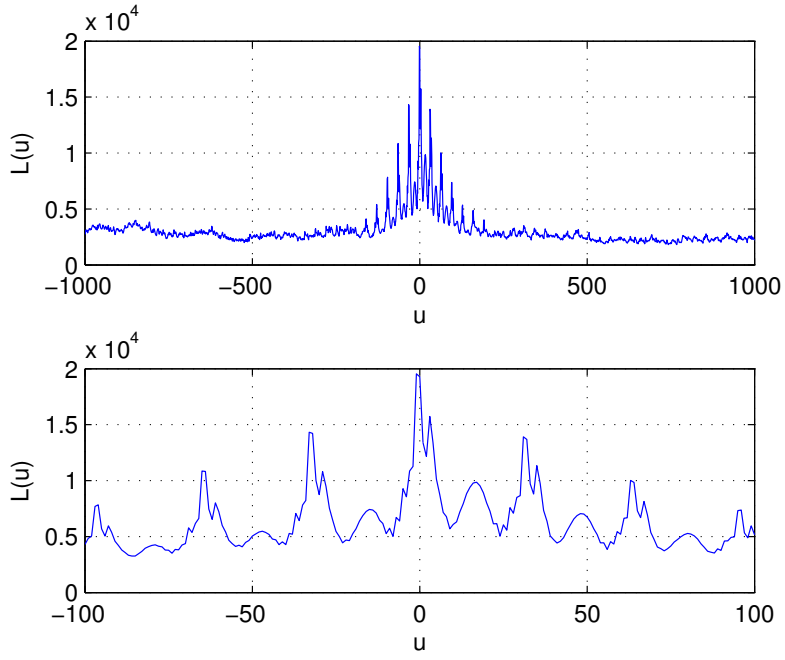

Figure 37: A plot of $L_{\mathrm{CL}}(u)$ for test channel 7 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

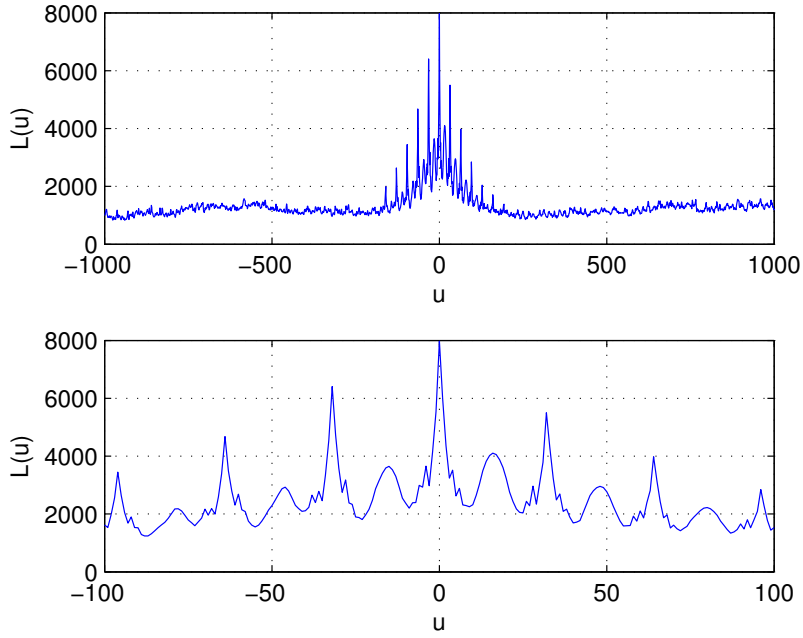

Figure 38: A plot of $L_{\mathrm{CL}}(u)$ for test channel 8 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

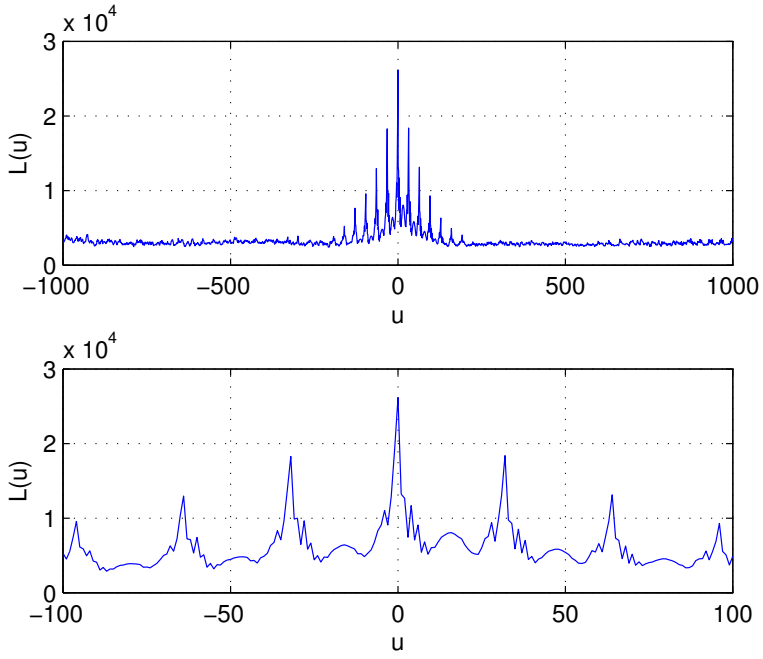

Figure 39: A plot of $L_{\mathrm{CL}}(u)$ for test channel 9 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

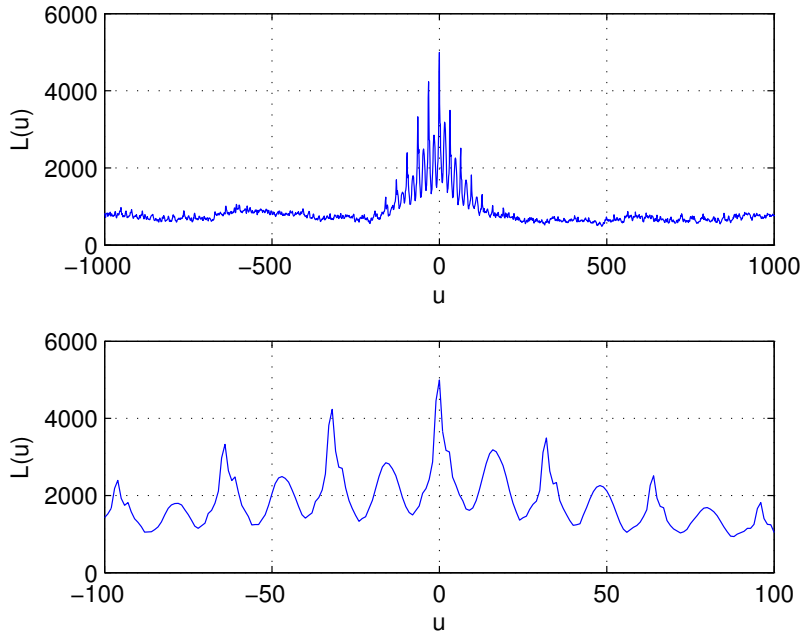

Figure 40: A plot of $L_{\mathrm{CL}}(u)$ for test channel 10 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

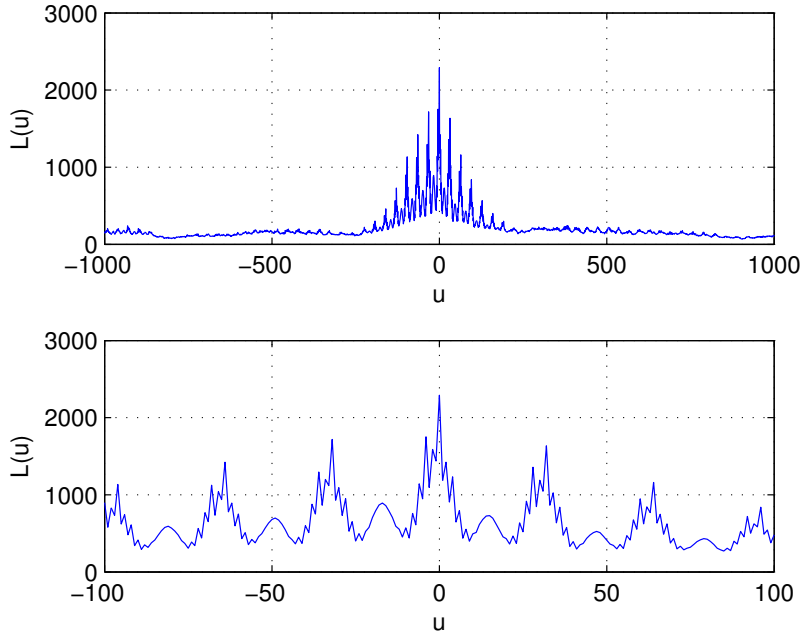

Figure 41: A plot of $L_{\mathrm{CL}}(u)$ for test channel 11 and a 50 kHz frequency offset: (top): $L_{\mathrm{CL}}(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

2.2 A Low-Complexity Preamble Detector

As a consequence of the high computational burden required by the Choi-Lee preamble detector, low-complexity preamble detectors are of interest. One of the most interesting is a simplified version of the "non-coherent post-detection integration" (NCPDI) described by Pedone *et al* [7]. By writing the length of the correlation interval as $L_p = L_{\text{PDI}}L_{\text{coh}}$, the Pedone-Corazza approach partitions the correlation into L_{PDI} coherent correlations of length L_{coh} and combines each of these length-L_{coh} correlations *noncoherently*. In equation form, the function is

$$L_{\text{PC}}(u) = \sum_{k=0}^{L_{\text{PDI}}-1} \left| \sum_{m=kL_{\text{coh}}}^{(k+1)L_{\text{coh}}-1} r(u+m)p^*(m) \right|^2 . \tag{12}$$

Here L_{PDI} and L_{coh} are design parameters. Typically, L_{coh} is defined so that the loss in coherent correlation due to the frequency offset is negligible. Here, we chose L_{coh} based on the structure of the iNET preamble to reduce computational complexity: we set $L_{\text{coh}} = L_q$ which forces $L_{\text{PDI}} = 8$. In this case $p(m) = q(m - kL_q)$ for $kL_{\text{coh}} \le m \le (k+1)L_{\text{coh}} - 1$ so that we may write[1]

$$L_{\text{PC}}(u) = \sum_{k=0}^{7} \left| \sum_{\ell=0}^{L_q-1} r(u + kL_q + \ell)q^*(\ell) \right|^2 . \tag{13}$$

Here, each sample of $L_{\text{PC}}(u)$ requires $8L_q = 256$ complex-by-complex multiplications and 16 real-by-real multiplications [equivalently 1040 real-by-real multiplications] — a reduction by more than two orders of magnitude over that required by (9).

Even further reductions are possible. A plot of $q(\ell)$ for $0 \le \ell < L_q$ is shown in Figure 42. The points are all on the unit circle and are approximately clustered about ± 1, $\pm j$, and $1/\sqrt{2}\,(\pm 1 \pm j)$. Quantizing the values of $q(\ell)$ to these values produces the quantized version of $q(\ell)$, $q_Q(\ell)$, plotted in Figure 43. Using the notation $r(n) = r_R(n) + jr_I(n)$, we may write

$$L(u) = \sum_{k=0}^{7} I^2(u,k) + Q^2(u,k) \tag{14}$$

where

[1] To see that this is so, let $\ell = m - kL_q$. This means $m = kL_q + \ell$ so that $r(u+m) = r(u + kL_q + \ell)$. Now determine the the limits on the inner summation: when $m = kL_{\text{coh}} = kL_q$, we have $\ell = kL_q - kL_q = 0$; when $m = (k+1)L_{\text{coh}} - 1 = (k+1)L_q - 1$, we have $\ell = (k+1)L_q - 1 - kL_q = L_q - 1$.

$$
\begin{aligned}
I(u,k) \approx & \sum_{i\in\mathcal{I}_1} r_R(u+kL_q+i) - \sum_{i\in\mathcal{I}_2} r_R(u+kL_q+i) \\
& + \sum_{i\in\mathcal{I}_3} r_I(u+kL_q+i) - \sum_{i\in\mathcal{I}_4} r_I(u+kL_q+i) \\
& + 0.7071 \left[\sum_{i\in\mathcal{I}_5} r_R(u+kL_q+i) - \sum_{i\in\mathcal{I}_6} r_R(u+kL_q+i) \right. \\
& \left. + \sum_{i\in\mathcal{I}_7} r_I(u+kL_q+i) - \sum_{i\in\mathcal{I}_8} r_I(u+kL_q+i) \right], \quad (15)
\end{aligned}
$$

$$
\begin{aligned}
Q(u,k) \approx & \sum_{i\in\mathcal{I}_1} r_I(u+kL_q+i) - \sum_{i\in\mathcal{I}_2} r_I(u+kL_q+i) \\
& - \sum_{i\in\mathcal{I}_3} r_R(u+kL_q+i) + \sum_{i\in\mathcal{I}_4} r_R(u+kL_q+i) \\
& + 0.7071 \left[\sum_{i\in\mathcal{I}_5} r_I(u+kL_q+i) - \sum_{i\in\mathcal{I}_6} r_I(u+kL_q+i) \right. \\
& \left. - \sum_{i\in\mathcal{I}_7} r_R(u+kL_q+i) + \sum_{i\in\mathcal{I}_8} r_R(u+kL_q+i) \right] \quad (16)
\end{aligned}
$$

with

$$
\begin{aligned}
\mathcal{I}_1 &= \{0, 8, 16, 24\} \\
\mathcal{I}_2 &= \{4, 20\} \\
\mathcal{I}_3 &= \{2, 10, 14, 22\} \\
\mathcal{I}_4 &= \{6, 18, 26, 30\} \\
\mathcal{I}_5 &= \{1, 7, 9, 15, 17, 23, 25, 31\} \\
\mathcal{I}_6 &= \{3, 5, 11, 12, 13, 19, 21, 27, 28, 29\} \\
\mathcal{I}_7 &= \{1, 3, 9, 11, 12, 13, 15, 21, 23\} \\
\mathcal{I}_8 &= \{5, 7, 17, 19, 25, 27, 28, 29, 31\}.
\end{aligned} \quad (17)
$$

These equations show that each sample of $L(u)$ may be approximated by a function that requires 4 real-by-real multiplications!

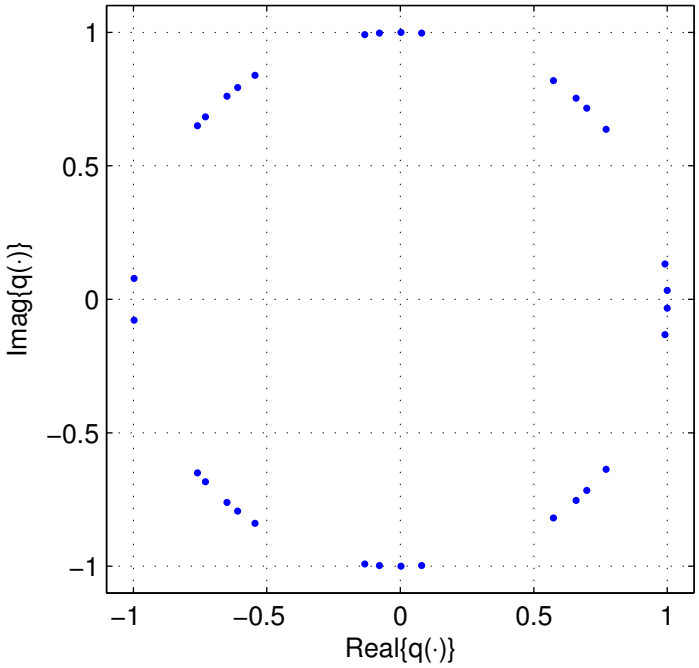

Figure 42: A plot of the complex values of $q(\ell)$ for $0 \leq \ell < L_q$.

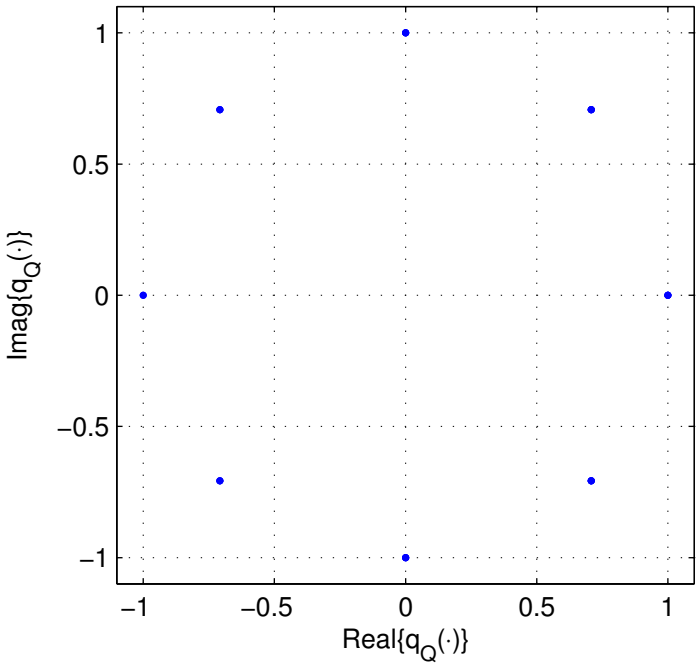

Figure 43: A plot of the complex values of $q_Q(\ell)$ for $0 \leq \ell < L_q$.

As an example of the behavior of this low-complexity function (14), we apply the samples of an SOQPSK-TG signal corresponding to the bit sequence shown in Figure 17. As before, the SOQPSK-TG samples are generated at a sample rate corresponding to 2 samples/bit. A comparison of $L_{\text{CL}}(u)$ given by (9), the Pedone-Corazza correlation function $L_{\text{PC}}(u)$ given by (13), and the low-complexity correlation function $L(u)$ given by (14) are shown in Figure 44. All three correlation functions exhibit the multiple "peaking" behavior due to the repetitive nature of the iNET preamble. The off-center peaks of $L_{\text{PC}}(u)$ and $L(u)$ are slightly higher than those of $L_{\text{CL}}(u)$ and this could cause a problem in very low signal-to-noise ratio conditions. Interestingly, $L_{\text{PC}}(u)$ and $L(u)$ are very similar — in fact almost indistinguishable. This shows that the approximation illustrated in Figure 43 does not seriously impact the behavior of $L(u)$.

The comparison for the case of a 50 kHz frequency offset is shown in Figure 45. The magnitude of $L_{\text{PC}}(u)$ and $L(u)$ are reduced from their values in the case of no frequency offset, but $L_{\text{CL}}(u)$ maintains its value. (This was observed in the previous section). But the reduction is modest and will probably not constitute a major performance issue.

Plots of $L(u)$ in the presence of a 50 kHz frequency offset over the eleven test channels are plotted in Figures 46 – 56. In all cases, the maximum peak in $L(u)$ correctly identifies the start of the preamble.

In summary, the low-complexity correlation function $L(u)$ given by (14) correctly identifies the start of the preamble in the conditions encountered in aeronautical telemetry, namely a frequency offset and a multipath channel. And it accomplishes this with very low computational complexity.

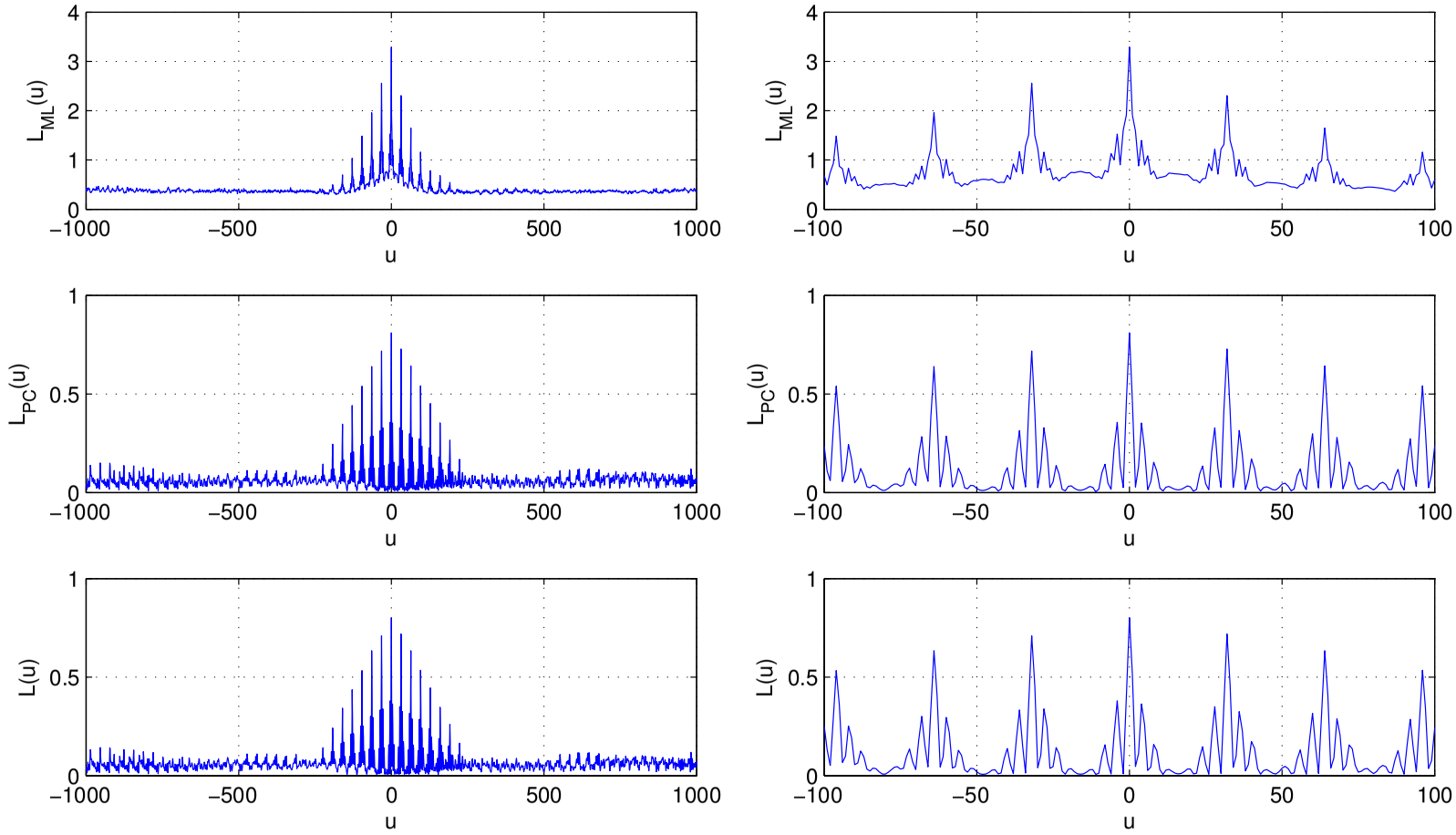

Figure 44: A comparison of the maximum likelihood correlation function $L_{\text{CL}}(u)$ given by (9), the Pedone-Corazza correlation function $L_{\text{PC}}(u)$ given by (13), and the low-complexity correlation function $L(u)$ given by (14) for the case of no noise, no frequency offset, and an ideal channel. (Left): the three correlation functions for all the SOQPSK-TG samples corresponding to the bit sequence shown in Figure 17; (Right): a zoomed in view.

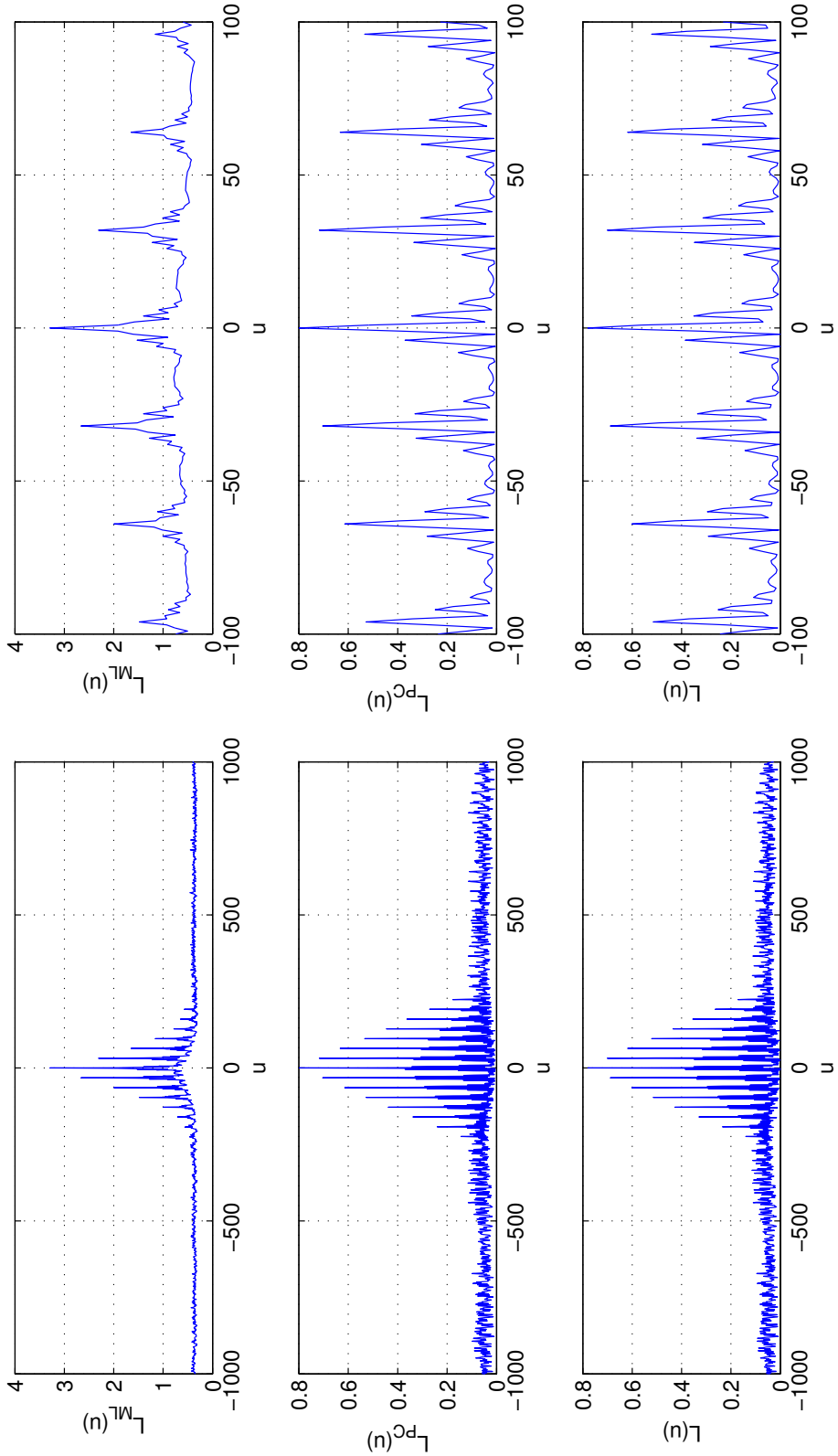

Figure 45: A comparison of the maximum likelihood correlation function $L_{\mathrm{CL}}(u)$ given by (9), the Pedone-Corazza correlation function $L_{\mathrm{PC}}(u)$ given by (13), and the low-complexity correlation function $L(u)$ given by (14) for the case of no noise, a 50 kHz frequency offset, and an ideal channel. (Left): the three correlation functions for all the SOQPSK-TG samples corresponding to the bit sequence shown in Figure 17; (Right): a zoomed in view.

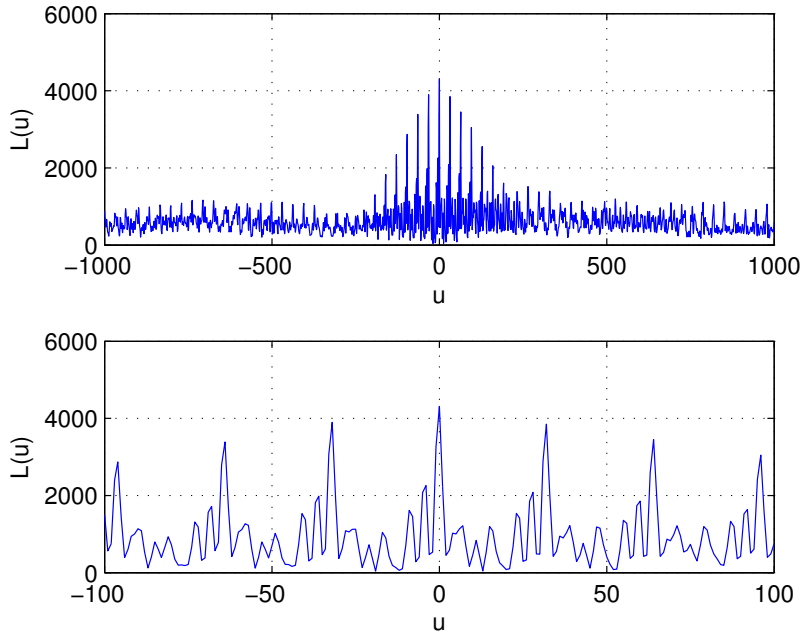

Figure 46: A plot of $L(u)$ for test channel 1 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

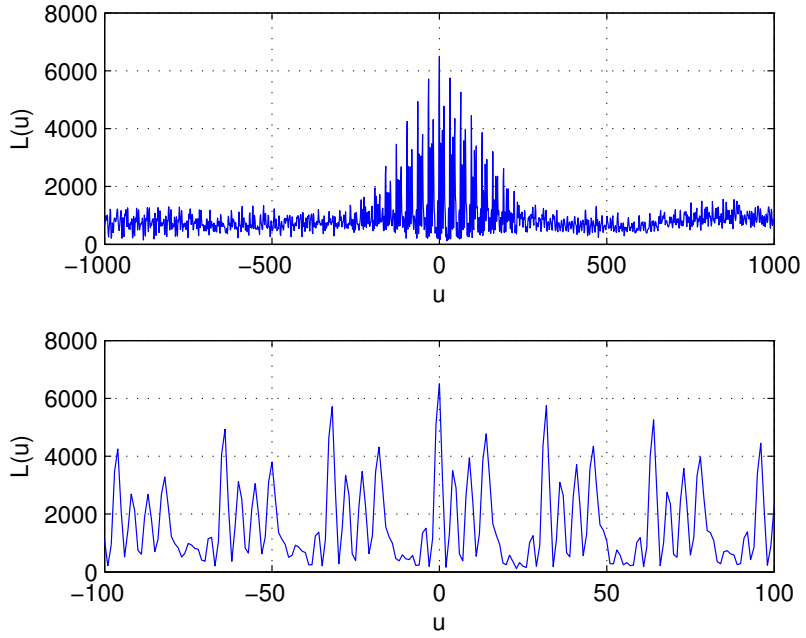

Figure 47: A plot of $L(u)$ for test channel 2 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

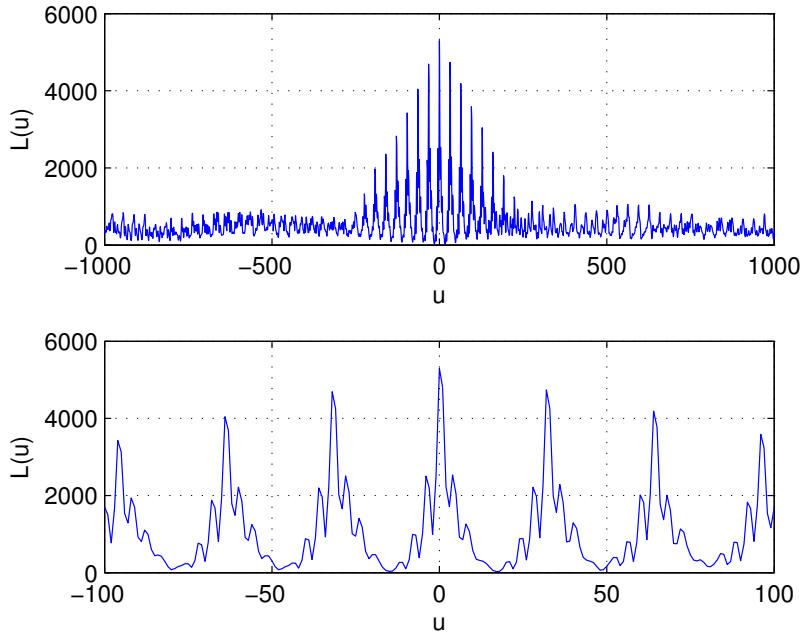

Figure 48: A plot of $L(u)$ for test channel 3 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

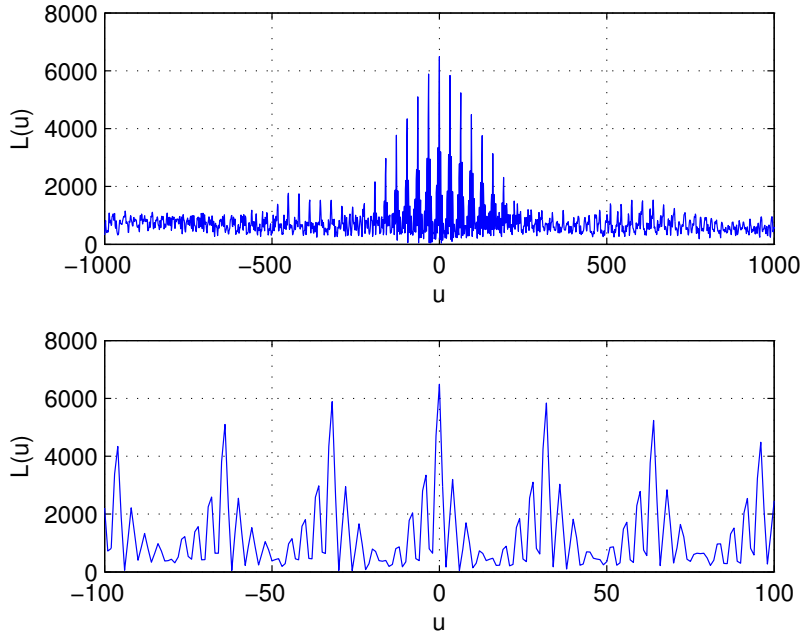

Figure 49: A plot of $L(u)$ for test channel 4 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

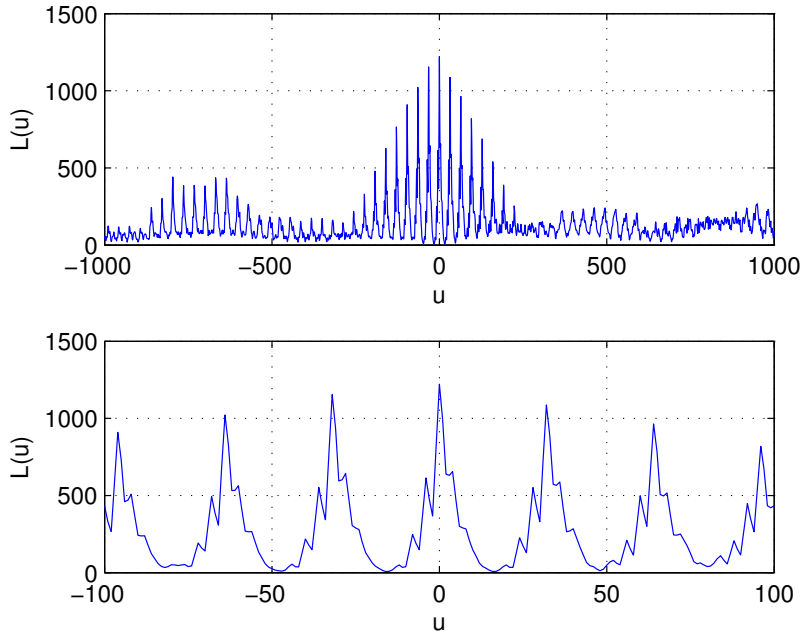

Figure 50: A plot of $L(u)$ for test channel 5 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

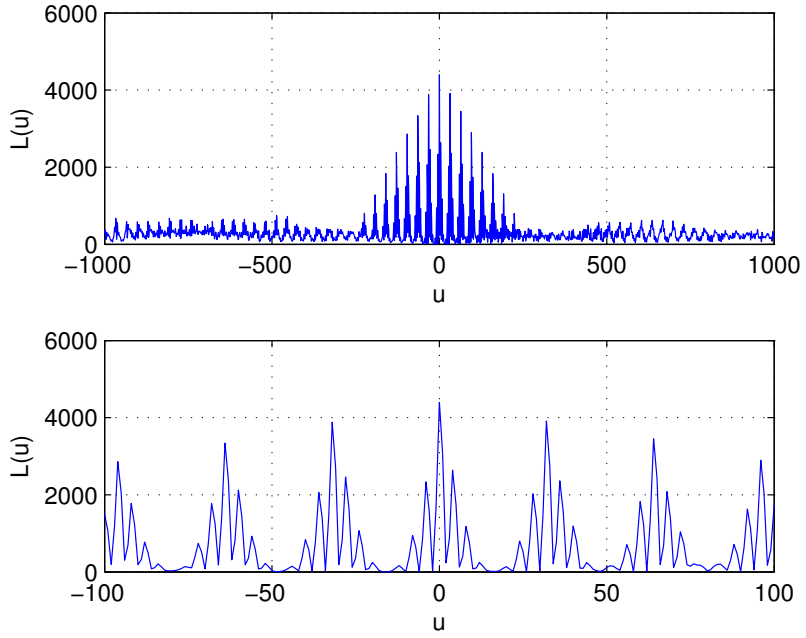

Figure 51: A plot of $L(u)$ for test channel 6 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

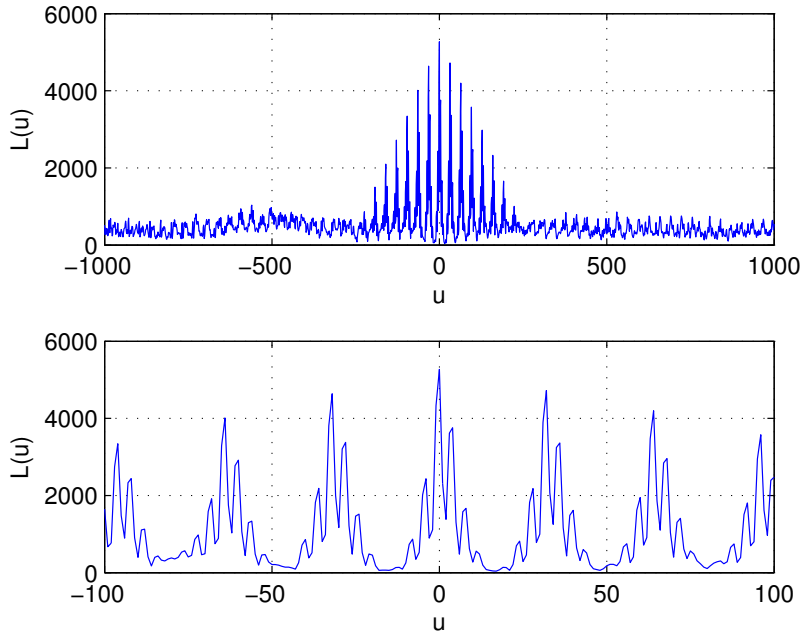

Figure 52: A plot of $L(u)$ for test channel 7 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

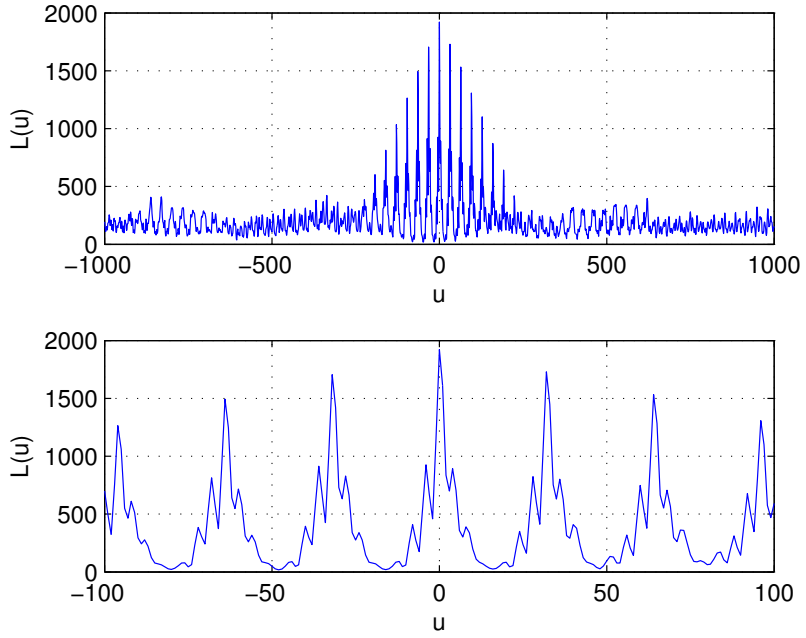

Figure 53: A plot of $L(u)$ for test channel 8 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

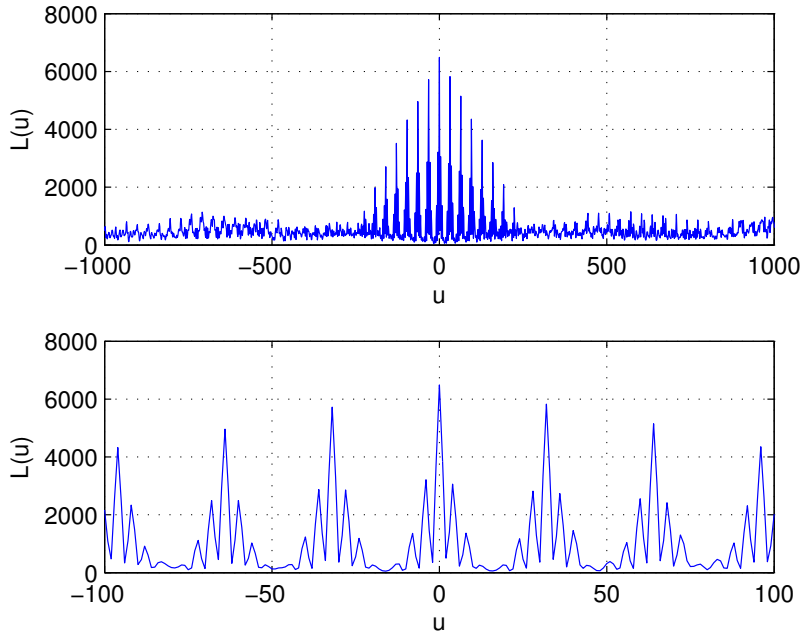

Figure 54: A plot of $L(u)$ for test channel 9 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

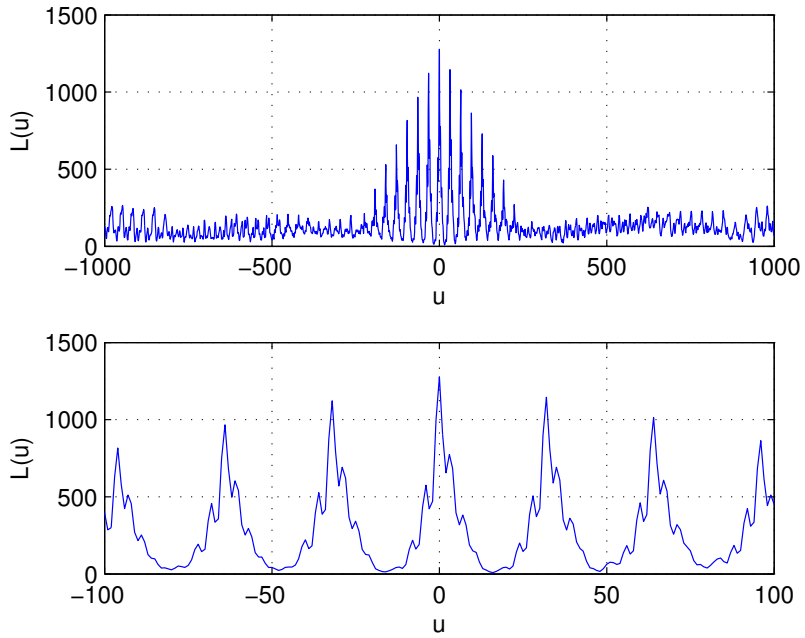

Figure 55: A plot of $L(u)$ for test channel 10 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

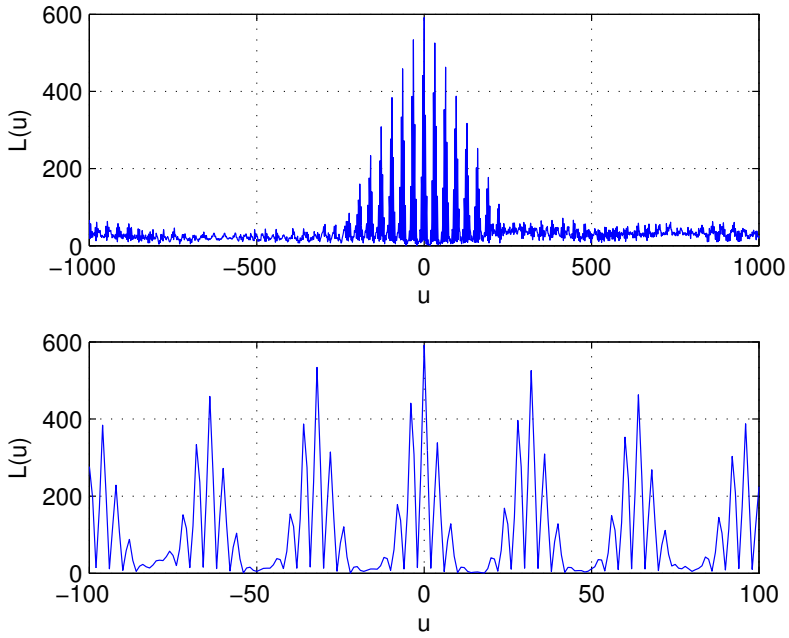

Figure 56: A plot of $L(u)$ for test channel 11 and a 50 kHz frequency offset: (top): $L(u)$ for all SOQPSK-TG samples corresponding the data in Figure 17; (bottom): a zoomed in view.

3 PAQ Frequency Estimators

Most data-aided frequency offset estimators correlate the received signal samples with the known data to create a signal that well-approximates the sinusoid-in-noise problem. The maximum likelihood estimator for the sinusoid-in-noise problem was formulated by Rife and Boorstyn [8] and involves the maximization of the periodogram. The complexity of the maximum-likelihood approach can be reduced by operating on the phase differences associated with the correlation functions. Examples include the Kay [9], Fitz [10], Luise & Reggiannini [11], and Mengali & Morelii [12]. These methods cannot be applied directly to our problem because of the unknown multipath interference. The reason the channel estimate cannot be used is because the channel estimate requires an estimate of the frequency offset. Consequently we want to use a frequency offset estimator that does not depend (at least, strongly) on the channel impulse response. The periodic structure of the preamble allows us to use *modified* versions of these algorithms. The modified algorithms and their simulated performance are presented in this section.

3.1 Preliminaries

The received signal is given by (2) and repeated below for convenience:

$$r(n) = \left[\sum_{k=-N_1}^{N_2} h(k)s(n-k) \right] e^{j\omega_0 n} + w(n) \tag{18}$$

where $h(n)$ is the impulse response of the unknown channel, ω_0 rads/sample is the frequency offset to be estimated, and $w(n)$ is a complex-valued zero-mean Gaussian random process with auto-covariance function

$$\frac{1}{2}\mathrm{E}\left\{w(n)w^*(n-k)\right\} = \sigma_w^2\delta(k). \tag{19}$$

As before, let i be the index in $r(n)$ corresponding to the beginning of the preamble sequence and let L_p be the number of samples in the preamble sequence. The situation is illustrated in Figure 57.

If one mimics the development outlined in [10–12], then the temptation is to multiply $r(n)$ for $n = i, \ldots, i + L_p - 1$ by the conjugate of the preamble sequences $p^*(n)$ for $n = 0, \ldots, L_p - 1$ and explore the properties of the correlation functions. Just to illustrate the problem, we do this now. First form the sequence $z(n) = r(i+n)p^*(n)$. Neglecting noise for the moment, we have

$$z(n) = \left(\left[\sum_{k=-N_1}^{N_2} h(k)s(i+n-k) \right] e^{j\omega_0(i+n)} \right) p^*(n) \tag{20}$$

$$= \left(\left[\sum_{k=-N_1}^{N_2} h(k)p(n-k) \right] e^{j\omega_0(i+n)} \right) p^*(n) \tag{21}$$

$$= \left[\sum_{k=-N_1}^{N_2} h(k)p(n-k)p^*(n) \right] e^{j\omega_0(i+n)} \tag{22}$$

for $n = 0, \ldots, L_p - 1$. The correlation function is

$$R(\delta) = \frac{1}{L_p - \delta} \sum_{n=\delta}^{L_p-1} z(n)z^*(n-\delta) \tag{23}$$

$$= \frac{1}{L_p - \delta} \sum_{n=\delta}^{L_p-1} \left\{ \underbrace{\left[\sum_{k=-N_1}^{N_2} h(k)p(n-k)p^*(n) \right]}_{y(n)} e^{j\omega_0(i+n)} \times \right.$$

$$\left. \underbrace{\left[\sum_{k'=-N_1}^{N_2} h(k')p(n-\delta-k')p^*(n-\delta) \right]}_{y^*(n-\delta)} e^{-j\omega_0(i+n-\delta)} \right\} \tag{24}$$

$$= \frac{1}{L_p - \delta} \sum_{n=\delta}^{L_p-1} y(n)y^*(n-\delta)e^{j\delta\omega_0}. \tag{25}$$

This shows that if we knew $y(n)$ and $y^*(n-\delta)$ we could estimate ω_0 from $R(\delta)$. But we don't, so we can't.

The solution to this problem is to leverage the periodic properties of the iNet preamble. The iNet preamble comprises 8 repetitions of a 16 bit sequence. Consequently, the resulting SOQPSK-TG modulated carrier is also periodic over the time interval corresponding to the occurrence of the preamble. Furthermore the sampled SOQPSK-TG signal is periodic over the interval corresponding to the preamble sequence. All this is difficult to explain with equations, but relatively straightforward to show with a diagram, such as the one in Figure 57. In the figure $s(n)$ are the samples of the complex baseband SOQPSK-TG signal and $p(n')$ for $n' = 0, 1, \ldots, L_p - 1$ are the samples of the complex baseband SOQPSK-TG signal corresponding to the preamble bits. If the preamble sequence starts at index i, then $s(i) = p(0)$, $s(i+1) = p(1)$ and so on to $s(i+L_p-1) = p(L_p-1)$. Now let $q(n)$ be the samples of the complex baseband SOQPSK-tG signal corresponding to the 16-bit sequence that is repeated to form the preamble, and let the length of $q(n)$ be L_q. (At an

equivalent sample rate of 2 samples/bit, $L_p = 256$ and $L_q = 32$.)

For $n = i, i+1, \ldots, i+L_p-1$, we write $r(n) = r(i+\square)$ for $\square = 0, 1, \ldots, L_p-1$ and express it as follows:

$$r(i+\square) = \left[\sum_{k=-N_1}^{N_2} h(k)s(i+\square-k)\right] e^{j\omega_0(i+\square)} + w(i+\square) \tag{26}$$

$$= \left[\sum_{k=-N_1}^{N_2} h(k)p(\square-k)\right] e^{j\omega_0(i+\square)} + w(i+\square). \tag{27}$$

To incorporate the periodic nature of $p(n)$ into the equation, we write

$$\square = \ell L_q + m, \qquad \ell = 0, 1, \ldots, 7, \qquad m = 0, 1, \ldots, L_q - 1. \tag{28}$$

Here we see that ℓ indexes the block and m indexes the sample within the block. Substituting gives

$$r(i+\ell L_q+m) = \left[\sum_{k=-N_1}^{N_2} h(k)p(\ell L_q+m-k)\right] e^{j\omega_0(i+\ell L_q+m)} + w(i+\ell L_q+m) \tag{29}$$

$$= \underbrace{\left[\sum_{k=-N_1}^{N_2} h(k)q(m-k)\right]}_{\alpha(m)} e^{j\omega_0(i+\ell L_q+m)} + w(i+\ell L_q+m) \tag{30}$$

$$= \alpha(m)e^{j\omega_0(i+\ell L_q+m)} + w(i+\ell L_q+m). \tag{31}$$

The last equation shows that the contribution of the channel to $r(i+\ell L_q+m)$ depends only on m, the position inside the $q(\cdot)$ block. This is only true for middle six blocks (i.e., $\ell = 1, \ldots, 6$) because the convolution sum in the first block ($\ell = 0$) includes samples from the data field immediately preceding the first block whereas the last block $\ell = 7$ includes samples from the ASM field immediately following the last block. Eliminating the first and last blocks is sufficient as long as $N_1 < L_q$ and $N_2 < L_q$. If either of these conditions are not true, then (31) is true for fewer blocks.

Assuming $N_1 < L_q$ and $N_2 < L_q$, we are interested in the correlation function involving the middle six blocks:

$$R(\delta) = \frac{1}{6L_q - \delta} \sum_{n=i+L_q+\delta}^{i+7L_q-1} r(n)r^*(n-\delta) \tag{32}$$

Because the samples $r(n)$ are periodic with period L_q, we are interested in evaluating this function

at integer multiples of L_q. For $\delta = L_q$ we have

$$R(L_q) = \frac{1}{5L_q} \sum_{n=i+2L_q}^{i+7L_q-1} r(n) r^*(n - L_q) \tag{33}$$

$$= \frac{1}{5L_q} \sum_{\ell=2}^{6} \sum_{m=0}^{L_q-1} r(i + \ell L_q + m) r^*(i + (\ell-1)L_q + m) \tag{34}$$

$$= \frac{1}{5L_q} \sum_{\ell=2}^{6} \sum_{m=0}^{L_q-1} \left\{ \left[\alpha(m) e^{j\omega_0(i+\ell L_q+m)} + w(i + \ell L_q + m) \right] \right.$$
$$\left. \times \left[\alpha(m) e^{j\omega_0(i+(\ell-1)L_q+m)} + w(i + (\ell-1)L_q + m) \right]^* \right\} \tag{35}$$

$$= \frac{1}{5L_q} \sum_{\ell=2}^{6} \sum_{m=0}^{L_q-1} |\alpha(m)|^2 e^{jL_q\omega_0} + \frac{1}{5L_q} \sum_{\ell=2}^{6} \sum_{m=0}^{L_q-1} \text{noise terms.} \tag{36}$$

The first double summation may be simplified as follows:

$$\frac{1}{5L_q} \sum_{\ell=2}^{6} \sum_{m=0}^{L_q-1} |\alpha(m)|^2 e^{jL_q\omega_0} = \frac{1}{5L_q} \sum_{\ell=2}^{6} \alpha^2 e^{jL_q\omega_0} = \frac{\alpha^2}{L_q} e^{jL_q\omega_0} \tag{37}$$

where

$$\alpha^2 = \sum_{m=0}^{L_q-1} |\alpha(m)|^2. \tag{38}$$

Moving on to the double summation of noise terms, each of the summands is

$$\text{noise terms} = \underbrace{\alpha^*(m) e^{-j\omega_0(i+(\ell-1)L_q+m)} w(i + \ell L_q + m)}_{v_1(i+\ell L_q+m)}$$
$$+ \underbrace{\alpha(m) e^{j\omega_0(i+\ell L_q+m)} w^*(i + (\ell-1)L_q + m)}_{v_2(i+\ell L_q+m)}$$
$$+ w(i + \ell L_q + m) w^*(i + (\ell-1)L_q + m) \tag{39}$$

The first term, $v_1(i + \ell L_q + m)$, is a complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{ |v_1(i + \ell L_q + m)|^2 \right\} = |\alpha(m)|^2 \sigma_w^2. \tag{40}$$

The second term, $v_2(i + \ell L_q + m)$, is also a complex-valued Gaussian random variable with zero

mean and variance

$$\frac{1}{2}\mathrm{E}\left\{|v_2(i+\ell L_q+m)|^2\right\} = |\alpha(m)|^2\sigma_w^2. \tag{41}$$

Furthermore, these two terms are uncorrelated (and therefore independent) by virtue of the independence of the $w(n)$. For the last term, we make the assumption[2] that for sufficiently high signal-to-noise ratios (i.e., sufficiently small σ^2), this product term is much smaller than the first two terms with high probability and may therefore be neglected. Consequently we are left with

$$v(i+\ell L_q+m) = v_1(i+\ell L_q+m) + v_2(i+\ell L_q+m) \tag{42}$$

from which we see that $v(i+\ell L_q+m)$ is a complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{|v(i+\ell L_q+m)|^2\right\} = 2|\alpha(m)|^2\sigma_w^2. \tag{43}$$

Now applying the double summation we have

$$v_{L_q} = \frac{1}{5L_q}\sum_{\ell=2}^{6}\left[\sum_{m=0}^{L_q-1} v(i+\ell L_q+m)\right]. \tag{44}$$

The inner summation produces a complex-valued Gaussian random variable with zero mean and variance $2\alpha^2\sigma^2$. The outer summation and dividing by $5L_q$ produces the random variable v_{L_q}, a complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{|v_{L_q}|^2\right\} = \frac{2\alpha^2}{L_q^2}\sigma^2. \tag{45}$$

In summary, we have

$$R(L_q) = \frac{\alpha^2}{L_q}e^{jL_q\omega_0} + v_{L_q}. \tag{46}$$

Now for the case where $\delta = 2L_q$:

$$R(2L_q) = \frac{1}{4L_q}\sum_{n=i+3L_q}^{i+7L_q-1} r(n)r^*(n-2L_q) \tag{47}$$

$$= \frac{1}{4L_q}\sum_{\ell=3}^{6}\sum_{m=0}^{L_q-1} r(i+\ell L_q+m)r^*(i+(\ell-2)L_q+m) \tag{48}$$

[2]This is a standard assumption in the frequency estimation literature. See [9–12].

$$= \frac{1}{4L_q} \sum_{\ell=3}^{6} \sum_{m=0}^{L_q-1} \left\{ \left[\alpha(m) e^{j\omega_0(i+\ell L_q+m)} + w(i+\ell L_q+m) \right] \right.$$
$$\left. \times \left[\alpha(m) e^{j\omega_0(i+(\ell-2)L_q+m)} + w(i+(\ell-2)L_q+m) \right]^* \right\} \tag{49}$$

$$= \frac{1}{4L_q} \sum_{\ell=3}^{6} \sum_{m=0}^{L_q-1} |\alpha(m)|^2 e^{jL_q\omega_0} + \frac{1}{4L_q} \sum_{\ell=3}^{6} \sum_{m=0}^{L_q-1} \text{noise terms} \tag{50}$$

$$= \frac{\alpha^2}{L_q} e^{j2L_q\omega_0} + v_{2L_q} \tag{51}$$

where α^2 is given by (38) and v_{2L_q} is a complex-valued Gaussian random variable with zero mean and variance given by (45). Continuing, we have

$$R(3L_q) = \frac{\alpha^2}{L_q} e^{j3L_q\omega_0} + v_{3L_q} \tag{52}$$

$$R(4L_q) = \frac{\alpha^2}{L_q} e^{j4L_q\omega_0} + v_{4L_q} \tag{53}$$

$$R(5L_q) = \frac{\alpha^2}{L_q} e^{j5L_q\omega_0} + v_{5L_q} \tag{54}$$

where v_{3L_q}, v_{4L_q}, and v_{5L_q} are complex-valued Gaussian random variables each with the same statistics as v_{L_q}.

3.2 The Perrins Estimator

The Perrins estimator is based on the argument (phase) of $R(L_q)$:

$$\hat{\omega}_0 = \frac{1}{L_q} \arg\left\{ R(L_q) \right\}. \tag{55}$$

Insight into the performance of the estimator is revealed by first writing $R(L_q)$ as

$$R(L_q) = \frac{\alpha^2}{L_q} e^{jL_q\omega_0} + v_{L_q} = \frac{\alpha^2}{L_q} e^{jL_q\omega_0} \left[1 + \tilde{v} \right] \tag{56}$$

where

$$\tilde{v} = \frac{L_q}{\alpha^2} e^{-jL_q\omega_0} v_{L_q} \tag{57}$$

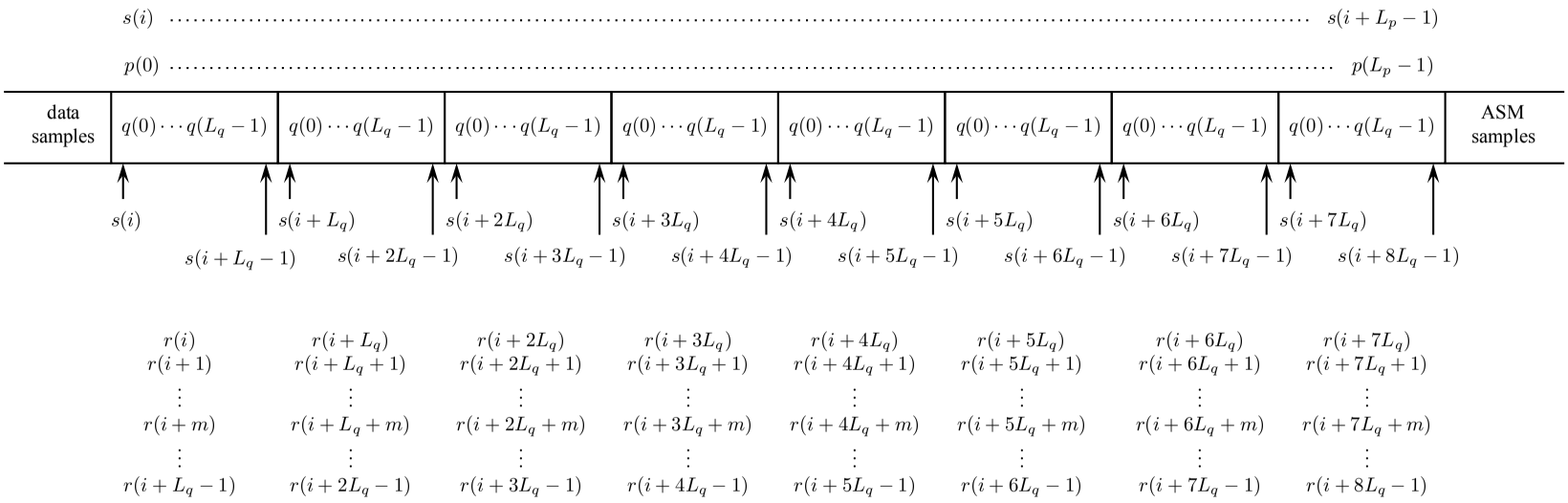

Figure 57: A graphical illustration of the relationship between the indexes of the received samples $r(n)$, the signal samples $s(n)$, the preamble samples $p(n)$ and the short sequences $q(n)$ that constitute the preamble sequence.

is a complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{|\tilde{v}|^2\right\} = \frac{2\sigma_w^2}{\alpha^2}. \tag{58}$$

Writing $\tilde{v} = \tilde{v}_R + j\tilde{v}_I$ we have

$$\begin{aligned}\arg\left\{R(L_q)\right\} = L_q\omega_0 + \arg\left\{1+\tilde{v}\right\} &= L_q\omega_0 + \tan^{-1}\left(\frac{\tilde{v}_I}{1+\tilde{v}_R}\right) \\ &\approx L_q\omega_0 \tan^{-1}\left(\tilde{v}_I\right) \approx L_q\omega_0 + \tilde{v}_I\end{aligned} \tag{59}$$

where the first approximation holds when $\tilde{v}_R \ll 1$ with high probability and the second approximation holds when $\tilde{v}_I \ll 1$ with high probability. This shows that $\hat{\omega}_0$ is approximately a Gaussian random variable with mean ω_0 and variance $2\sigma_w^2/(L_q^2\alpha^2)$.

3.3 The Modified Fitz Estimator

The development of the Fitz estimator [10] begins with N samples of a sinusoid in noise, denoted

$$x_n = e^{j\omega_0 n} + z_n, \qquad n = 0, 1, \ldots, N-1 \tag{60}$$

where z_n is a noise term. The maximum likelihood solution is

$$\hat{\omega}_0 = \underset{\omega_0}{\operatorname{argmax}} \left\{ \left| \sum_{n=0}^{N-1} x_n e^{-j\omega_0 n} \right|^2 \right\}. \tag{61}$$

Expanding the argument and setting the derivative with respect to ω_0 to zero gives

$$\mathrm{Im}\left\{ \sum_{m=1}^{N} m\hat{R}(m) e^{-j\hat{\omega}_0 m} \right\} = 0 \tag{62}$$

where

$$\hat{R}(m) = \sum_{k=m}^{N} x_k x_{k-m}^* \tag{63}$$

is the unnormalized correlation function. Writing $\hat{R}(m) = |\hat{R}(m)|e^{j\arg\{\hat{R}(m)\}}$ and using the approximations

$$\mathrm{Im}\left\{\hat{R}(m)e^{-j\hat{\omega}_0 m}\right\} \approx \left|\hat{R}(m)\right| \sin\left(\arg\left\{\hat{R}(m)\right\} - m\hat{\omega}_0\right) \tag{64}$$

$$\approx \left|\hat{R}(m)\right| \left(\arg\left\{\hat{R}(m)\right\} - m\hat{\omega}_0\right), \tag{65}$$

the condition (62) becomes

$$\sum_{m=1}^{N} m\left(\arg\left\{\hat{R}(m)\right\} - m\hat{\omega}_0\right) = 0. \tag{66}$$

This relationship assumes the availability of the correlation functions for lags $1, 2, \ldots, N$. In our application, not all of these are available. So, we apply the condition (66) using the available correlation functions $R(L_q)$, $R(2L_q)$, $R(3L_q)$, $R(4L_q)$ and $R(5L_q)$. Now, (66) becomes

$$\sum_{m=1}^{5} mL_q\left(\arg\left\{\hat{R}(mL_q)\right\} - mL_q\hat{\omega}_0\right) = 0. \tag{67}$$

Solving for $\hat{\omega}_0$ gives

$$\hat{\omega}_0 = \frac{1}{55L_q}\sum_{m=1}^{5} m\arg\left\{\hat{R}(mL_q)\right\}. \tag{68}$$

3.4 The Modified Luise & Reggiannini Estimator

The development of the L&R estimator [11] begins with N samples of a sinusoid in noise, denoted

$$x_n = e^{j\omega_0 n} + z_n, \qquad n = 0, 1, \ldots, N-1 \tag{69}$$

where z_n is a noise term. The maximum likelihood solution is

$$\hat{\omega}_0 = \underset{\omega_0}{\mathrm{argmax}}\left\{\left|\sum_{n=0}^{N-1} x_n e^{-j\omega_0 n}\right|^2\right\}. \tag{70}$$

Expanding the argument and setting the derivative with respect to ω_0 to zero gives

$$\mathrm{Im}\left\{\sum_{m=1}^{N} m(N-m)R(m)e^{-j\hat{\omega}_0 m}\right\} = 0 \tag{71}$$

where

$$R(m) = \frac{1}{N-m} \sum_{k=m}^{N} x_k x_{k-m}^* \tag{72}$$

is the correlation function. Luise & Reggiannini think of the condition (71) as one that involves a windowed sum of the autocorrelation functions and write

$$\mathrm{Im} \left\{ \sum_{m=1}^{N} w_m R(m) e^{-j\hat{\omega}_0 m} \right\} = 0 \tag{73}$$

where w_m is the m-th window sample. A suboptimal, but simplified, version of (73) is obtained by using a uniform window: $w_m = 1$ for $m = 1, 2, \ldots, N$. Using only the first $M < N$ correlation lags gives

$$\mathrm{Im} \left\{ \sum_{m=1}^{M} R(m) e^{-j\hat{\omega}_0 m} \right\} = 0. \tag{74}$$

For $\hat{\omega}_0 \ll 1$, the approximation

$$e^{-j\hat{\omega}_0 m} = 1 - j\hat{\omega}_0 m - \frac{(\hat{\omega}_0 m)^2}{2!} + j\frac{(\hat{\omega}_0 m)^3}{3!} + \cdots \approx 1 - j\hat{\omega}_0 m \tag{75}$$

is valid. Using $R(m) = R_R(m) + jR_I(m)$, condition (74) may be approximated as

$$0 \approx \mathrm{Im} \left\{ \sum_{m=1}^{M} \Big[R_R(m) + jR_I(m) \Big] \Big[1 - j\hat{\omega}_0 m \Big] \right\} = \sum_{m=1}^{M} \Big[R_I(m) - \hat{\omega}_0 m R_R(m) \Big]. \tag{76}$$

Solving for $\hat{\omega}_0$ gives

$$\hat{\omega}_0 = \frac{\sum_{m=1}^{M} R_I(m)}{\sum_{m=1}^{M} m R_R(m)}. \tag{77}$$

Luise & Reggiannini simply (77) even further as follows.

1. The autocorrelation function may be expressed as

$$R(m) = e^{j\omega_0 m} + v_m \tag{78}$$

where $v_m = v_{m,R} + jv_{m,I}$ is a noise term that is assumed to be small. Using the approxima-

tion (75), we have

$$R(m) \approx 1 + j\omega_0 m + v_{m,R} + jv_{m,I} = \underbrace{1 + v_{m,R}}_{R_R(m)} + j\underbrace{\omega_0 m + v_{m,I}}_{R_I(m)}. \tag{79}$$

2. Using (79) we have, for $v_{m,R} \ll 1$ with high probability

$$\sum_{m=1}^{M} R_R(m) \approx \sum_{m=1}^{M} \Big[1 + v_{m,R}\Big] \approx \sum_{m=1}^{M} 1 = M. \tag{80}$$

3. Using (79) we have, for $v_{m,R} \ll 1$ with high probability

$$\sum_{m=1}^{M} mR_R(m) \approx \sum_{m=1}^{M} m\left[1 + v_{m,R}\right] \approx \sum_{m=1}^{M} m = \frac{M(M+1)}{2}. \tag{81}$$

4. The phase of the sum of correlation functions is

$$\arg\left\{\sum_{m=1}^{M} R(m)\right\} = \tan^{-1}\left\{\frac{\sum_{m=1}^{M} R_I(m)}{\sum_{m=1}^{M} R_R(m)}\right\} \approx \tan^{-1}\left\{\frac{\sum_{m=1}^{M} R_I(m)}{M}\right\} \approx \frac{1}{M}\sum_{m=1}^{M} R_I(m) \tag{82}$$

from which we have

$$\sum_{m=1}^{M} R_I(m) \approx M \arg\left\{\sum_{m=1}^{M} R(m)\right\}. \tag{83}$$

The estimator (77) may be simplified by replacing the numerator with the approximation (83) and the denominator with the approximation (81) to produce the estimator in its familiar form

$$\hat{\omega}_0 = \frac{2}{M+1} \arg\left\{\sum_{m=1}^{M} R(m)\right\}. \tag{84}$$

The L&R estimator (84) assumes the correlation function for the lags $m = 1, 2, \ldots, M$ are available. In our application this is not true. The modification required for our case starts by

rewriting (77) so that the summations only include the available terms:

$$\hat{\omega}_0 = \frac{\sum_{m=1}^{5} R_I(mL_q)}{L_q \sum_{m=1}^{5} mR_R(mL_q)}. \tag{85}$$

Following steps 1–4 as Luise & Reggiannini did, we have the following.

1. The general expression for the correlation functions (46), (51)–(54) may be expressed as

$$R(mL_q) = \frac{\alpha^2}{L_q} e^{j\omega_0 mL_q} + v_{mL_q} \tag{86}$$

for $m = 1, 2, 3, 4, 5$ and where the noise term may be expressed as $v_{mL_q} = v_{mL_q,R} + jv_{mL_q,I}$. Using the approximation (75), we have

$$\begin{aligned} R(m) &\approx \frac{\alpha^2}{L_q}\left[1 + j\omega_0 mL_q\right] + v_{mL_q,R} + jv_{mL_q,I} \\ &= \underbrace{\frac{\alpha^2}{L_q} + v_{mL_q,R}}_{R_R(m)} + j\underbrace{\left(\alpha^2\omega_0 mL_q + v_{mL_q,I}\right)}_{R_I(m)}. \end{aligned} \tag{87}$$

2. Using (87) we have, for $v_{mL_q,R} \ll 1$ with high probability

$$\sum_{m=1}^{5} R_R(mL_q) \approx \sum_{m=1}^{5}\left[\frac{\alpha^2}{L_q} + v_{m,R}\right] \approx \sum_{m=1}^{5}\frac{\alpha^2}{L_q} = \frac{5\alpha^2}{L_q}. \tag{88}$$

3. Using (87) we have, for $v_{mL_q,R} \ll 1$ with high probability

$$\sum_{m=1}^{5} mR_R(mL_q) \approx \sum_{m=1}^{5} m\left[\frac{\alpha^2}{L_q} + v_{m,R}\right] \approx \frac{\alpha^2}{L_q}\sum_{m=1}^{5} m = \frac{15\alpha^2}{L_q}. \tag{89}$$

4. The phase of the sum of correlation functions is

$$\arg\left\{\sum_{m=1}^{5} R(mL_q)\right\} = \tan^{-1}\left\{\frac{\sum_{m=1}^{5} R_I(mL_q)}{\sum_{m=1}^{5} R_R(mL_q)}\right\} \approx \tan^{-1}\left\{\frac{\sum_{m=1}^{5} R_I(mL_q)}{\frac{5\alpha^2}{L_q}}\right\}$$
$$\approx \frac{L_q}{5\alpha^2}\sum_{m=1}^{5} R_I(mL_q) \tag{90}$$

from which we have

$$\sum_{m=1}^{5} R_I(mL_q) \approx \frac{5\alpha^2}{L_q}\arg\left\{\sum_{m=1}^{5} R(mL_q)\right\}. \tag{91}$$

The estimator (85) may be simplified by replacing the numerator with the approximation (91) and the denominator with the approximation (89):

$$\hat{\omega}_0 = \frac{1}{3L_q}\arg\left\{\sum_{m=1}^{5} R(mL_q)\right\}. \tag{92}$$

3.5 The Modified Mengali & Morelli Estimator

The development of the M&M estimator [12] begins with N samples of a sinusoid in noise, denoted

$$x_n = e^{j\omega_0 n} + z_n, \qquad n = 0, 1, \ldots, N-1 \tag{93}$$

where z_n is a proper complex-valued Gaussian random variable with zero mean. The Mengali & Morelli formulate their estimator based on the properties of the correlation function

$$R(m) = \frac{1}{N-m}\sum_{k=m}^{N} x_k x_{k-m}^{*} \tag{94}$$

for $m = 1, 2 \ldots, M$ where $M < N/2$ is a design parameter. The development begins by writing x_n as

$$x_n = e^{j\omega_0 n}\left[1 + \tilde{z}_n\right] \tag{95}$$

where

$$\tilde{z}_n = e^{-j\omega_0 n} z_n \tag{96}$$

is a complex-valued Gaussian random variable with the same statistics as z_n. Substituting, we have for the correlation function

$$R(m) = \frac{1}{N-m} \sum_{n=m}^{M} e^{j\omega_0 n} \left[1 + \tilde{z}_n\right] e^{-j\omega_0 (n-m)} \left[1 + \tilde{z}^*_{n-m}\right] \tag{97}$$

$$= e^{j\omega_0 m} \frac{1}{N-m} \sum_{n=m}^{M} \left[1 + \tilde{z}_n + \tilde{z}^*_{n-m} + \tilde{z}_n \tilde{z}^*_{n-m}\right] \tag{98}$$

$$\approx e^{j\omega_0 m} \frac{1}{N-m} \sum_{n=m}^{M} \left[1 + \tilde{z}_n + \tilde{z}^*_{n-m}\right] \tag{99}$$

$$= e^{j\omega_0 m} \left[1 + \frac{1}{N-m} \sum_{n=m}^{M} \left(\tilde{z}_n + \tilde{z}^*_{n-m}\right)\right] \tag{100}$$

$$= e^{j\omega_0 m} \left[1 + \gamma_m\right] \tag{101}$$

where

$$\gamma_m = \frac{1}{N-m} \sum_{n=m}^{M} \left(\tilde{z}_n + \tilde{z}^*_{n-m}\right) \tag{102}$$

is a sum of complex-valued Gaussian random variables with zero mean. Consequently, γ_m is a complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{ |\gamma_m|^2 \right\} = \sigma^2_\gamma. \tag{103}$$

The step from (98) to (99) follows from the assumption that $\tilde{z}_n$ and $\tilde{z}^*_{n-m}$ are both zero mean with small variances so that the product $\tilde{z}_n \tilde{z}^*_{n-m}$ may be neglected. Writing $\gamma_m = \gamma_{R,m} + j\gamma_{I,m}$ we have

$$\arg\left\{R(m)\right\} = \omega_0 m + \arg\left\{1 + \gamma_m\right\} \tag{104}$$

$$= \omega_0 m + \tan^{-1}\left\{\frac{\gamma_{I,m}}{1 + \gamma_{R,m}}\right\} \tag{105}$$

$$\approx \omega_0 m + \tan^{-1}\left\{\gamma_{I,m}\right\} \tag{106}$$

$$\approx \omega_0 m + \gamma_{I,m}. \tag{107}$$

The challenge with using $\arg\{R(m)\}$ is the phase unwrapping required to produce (107). Mengali & Morelli address the problem by investigating *differences* in phases of correlation functions with consecutive lags. To see this, define

$$\phi(m) = \arg\Big\{R(m)\Big\} - \arg\Big\{R(m-1)\Big\} \tag{108}$$

for $m = 1, 2, \ldots, M$. Substituting the approximation (107) gives

$$\phi(m) = \omega_0 + \gamma_{I,m} - \gamma_{I,m-1} \tag{109}$$

which is linear in ω_0 and may be expressed in vector form as

$$\begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(M) \end{bmatrix} = \omega_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \begin{bmatrix} \gamma_{I,1} - \gamma_{I,0} \\ \gamma_{I,2} - \gamma_{I,1} \\ \vdots \\ \gamma_{I,M} - \gamma_{I,M-1} \end{bmatrix}. \tag{110}$$

This relationship may be expressed as

$$\boldsymbol{\phi} = \omega_0 \mathbf{1} + \boldsymbol{\gamma} \tag{111}$$

where $\boldsymbol{\gamma}$ is a real-valued vector of Gaussian random variables with zero mean and covariance matrix

$$\mathbf{C} = \mathrm{E}\left\{\boldsymbol{\gamma}\boldsymbol{\gamma}^\top\right\} = \sigma_\gamma^2 \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & & \ddots & \\ & & -1 & 2 \end{bmatrix}. \tag{112}$$

Using these vectors and matrices, the likelihood function for ω_0 may be written

$$\Lambda(\omega_0) = (\boldsymbol{\phi} - \omega_0 \mathbf{1})^\top \mathbf{C}^{-1} (\boldsymbol{\phi} - \omega_0 \mathbf{1}) \tag{113}$$

and the corresponding maximum likelihood estimate is

$$\hat{\omega}_0 = \frac{\mathbf{1}^\top \mathbf{C}^{-1} \boldsymbol{\phi}}{\mathbf{1}^\top \mathbf{C}^{-1} \mathbf{1}}. \tag{114}$$

Because $\mathbf{C}$ is tri-diagonal, the general form of its inverse is known. The i, j element of $\mathbf{C}^{-1}$ is [9]

$$\left(\mathbf{C}^{-1}\right)_{i,j} = \frac{1}{\sigma_\gamma^2}\left[\min\{i,j\} - \frac{ij}{M}\right]. \tag{115}$$

This expression may be used to simplify (114).

The estimator (114) assumes the availability of correlation functions with M consecutive lags. Because this is not the case in our application, we must modify the approach. The starting point is the definition of the phase differences (108). Because the only available correlation functions are $R(L_q)$, $R(2L_q)$, $R(3L_q)$, $R(4L_q)$ and $R(5L_q)$, we explore the use of the phase differences

$$\phi(mL_q) = \arg\Big\{R(mL_q)\Big\} - \arg\Big\{R((m-1)L_q)\Big\} \tag{116}$$

for $m = 1, 2, 3, 4, 5$. To proceed, we write (46), (51)–(54) in the general form

$$R(m) = \frac{\alpha^2}{L_q} e^{jmL_q\omega_0} + v_{mL_q} \tag{117}$$

for $m = 1, 2, 3, 4, 5$. Now write

$$R(m) = \frac{\alpha^2}{L_q} e^{jmL_q\omega_0}\left[1 + \tilde{v}_{mL_q}\right] \tag{118}$$

where

$$\tilde{v}_{mL_q} = \frac{L_q}{\alpha^2} e^{-jmL_q\omega_0} v_{mL_q} \tag{119}$$

is a complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{\left|\tilde{v}_{mL_q}\right|^2\right\} = \frac{2\sigma_w^2}{\alpha^2}. \tag{120}$$

Using this expression, we have

$$\arg\Big\{R(m)\Big\} = mL_q\omega_0 + \arg\Big\{1 + \tilde{v}_{mL_q}\Big\} \tag{121}$$

$$= mL_q\omega_0 + \tan^{-1}\left\{\frac{\tilde{v}_{I,mL_q}}{1 + \tilde{v}_{R,mL_q}}\right\} \tag{122}$$

$$\approx mL_q\omega_0 + \tan^{-1}\Big\{\tilde{v}_{I,mL_q}\Big\} \tag{123}$$

$$\approx mL_q\omega_0 + \tilde{v}_{I,mL_q} \tag{124}$$

where the last two approximations hold if $\tilde{v}_{R,mL_q} \ll 1$ and $\tilde{v}_{I,mL_q} \ll 1$ with high probability. Now the phase difference (116) may be expressed as

$$\phi(mL_q) \approx L_q\omega_0 + \tilde{v}_{I,mL_q} - \tilde{v}_{I,(m-1)L_q}. \tag{125}$$

As before, this equation is linear in ω_0 and may be expressed in vector, matrix form as

$$\begin{bmatrix} \phi(L_q) \\ \phi(2L_q) \\ \phi(3L_q) \\ \phi(4L_q) \\ \phi(5L_q) \end{bmatrix} = L_q\omega_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} \tilde{v}_{I,L_q} - \tilde{v}_{I,0} \\ \tilde{v}_{I,2L_q} - \tilde{v}_{I,L_q} \\ \tilde{v}_{I,3L_q} - \tilde{v}_{I,2L_q} \\ \tilde{v}_{I,4L_q} - \tilde{v}_{I,3L_q} \\ \tilde{v}_{I,5L_q} - \tilde{v}_{I,4L_q} \end{bmatrix} \tag{126}$$

which may be expressed as

$$\boldsymbol{\phi} = L_q\omega_0\mathbf{1} + \mathbf{v} \tag{127}$$

where $\mathbf{v}$ is a real-valued Gaussian random vector with zero mean and covariance matrix

$$\mathbf{C} = \frac{2\sigma^2}{\alpha^2} \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}. \tag{128}$$

The log likelihood function is

$$\Lambda(\omega_0) = (\boldsymbol{\phi} - L_q\omega_0\mathbf{1})^\top \mathbf{C}^{-1} (\boldsymbol{\phi} - L_q\omega_0\mathbf{1}) \tag{129}$$

and the maximum likelihood estimate is

$$\hat{\omega}_0 = \frac{1}{L_q} \frac{\mathbf{1}^\top\mathbf{C}^{-1}\boldsymbol{\phi}}{\mathbf{1}^\top\mathbf{C}^{-1}\mathbf{1}}. \tag{130}$$

Using the relationships

$$\mathbf{1}^\top\mathbf{C}^{-1} = \frac{\alpha^2}{2\sigma^2} \begin{bmatrix} \frac{5}{2} & \frac{8}{2} & \frac{9}{2} & \frac{8}{2} & \frac{5}{2} \end{bmatrix} \tag{131}$$

$$\mathbf{1}^\top\mathbf{C}^{-1}\mathbf{1} = \frac{\alpha^2}{2\sigma^2} \frac{35}{2}, \tag{132}$$

The estimator (130) simplifies to

$$\hat{\omega}_0 = \frac{1}{35L_q}\left[5\phi(L_q) + 8\phi(2L_q) + 9\phi(3L_q) + 8\phi(4L_q) + 5\phi(5L_q)\right]. \tag{133}$$

3.6 The Modified Kay Estimator

The Kay estimator [9] begins with N samples of a sinusoid in noise:

$$x_n = e^{j\omega_0 n} + z_n \qquad n = 0, 1, \ldots, N-1 \tag{134}$$

where z_n is a proper complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{|z_n|^2\right\} = \sigma_z^2. \tag{135}$$

Now write

$$x_n = e^{j\omega_0 n}\left[1 + \tilde{z}_n\right] \tag{136}$$

where

$$\tilde{z}_n = e^{-j\omega_0 n} z_n \tag{137}$$

is a proper complex-valued Gaussian random variable with the same statistics as z_n. The phase of x_n may be approximated as

$$\begin{aligned}\arg\{x_n\} = \omega_0 n + \arg\{1 + \tilde{z}_n\} &= \omega_0 n + \tan^{-1}\left\{\frac{\tilde{z}_{I,n}}{1 + \tilde{z}_{R,n}}\right\} \\ &\approx \omega_0 n + \tan^{-1}\{\tilde{z}_{I,n}\} \approx \omega_0 n + \tilde{z}_{I,n}\end{aligned} \tag{138}$$

where the last two approximations hold if $z_{R,n} \ll 1$ and $z_{I,n} \ll 1$ with high probability. Kay points out that $\arg\{x_n\}$ could be used as the basis for an estimator, but the phase unwrapping problem makes this hard. In its place, Kay suggests using phase *differences*

$$\Delta_n = \arg\{x_n\} - \arg\{x_{n-1}\} = \arg\left\{x_n x_{n-1}^*\right\} \tag{139}$$

as this avoids the phase unwrapping problem. Using the approximation (138), the phase difference may be expressed as

$$\Delta_n \approx \omega_0 + \tilde{z}_{I,n} - \tilde{z}_{I,n-1}. \tag{140}$$

Because there are $N-1$ available phase differences, they may be used to formulate an estimate for ω_0. Stacking the $N-1$ phase differences gives the relationship

$$\begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{N-1} \end{bmatrix} = \omega_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \begin{bmatrix} \tilde{z}_{I,1} - \tilde{z}_{I,0} \\ \tilde{z}_{I,2} - \tilde{z}_{I,1} \\ \vdots \\ \tilde{z}_{I,N-1} - \tilde{z}_{I,N-2} \end{bmatrix}. \tag{141}$$

This relationship may be expressed as

$$\mathbf{\Delta} = \omega_0 \mathbf{1} + \mathbf{z} \tag{142}$$

where $\mathbf{z}$ is a vector of real-valued Gaussian random variables with zero mean and covariance matrix

$$\mathbf{C} = \mathrm{E}\left\{\mathbf{z}\mathbf{z}^\top\right\} = \sigma_z^2 \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & & \ddots & \\ & & -1 & 2 \end{bmatrix}. \tag{143}$$

The log likelihood function is

$$\Lambda(\omega_0) = \left(\mathbf{\Delta} - \omega_0 \mathbf{1}\right)^\top \mathbf{C}^{-1} \left(\mathbf{\Delta} - \omega_0 \mathbf{1}\right) \tag{144}$$

from which the maximum likelihood estimate is

$$\hat{\omega}_0 = \frac{\mathbf{1}^\top \mathbf{C}^{-1} \mathbf{\Delta}}{\mathbf{1}^\top \mathbf{C}^{-1} \mathbf{1}}. \tag{145}$$

Because $\mathbf{C}$ is tri-diagonal, the general form of its inverse is known. The i,j element of $\mathbf{C}^{-1}$ is [9]

$$\left(\mathbf{C}^{-1}\right)_{i,j} = \frac{1}{\sigma_\gamma^2}\left[\min\{i,j\} - \frac{ij}{M}\right]. \tag{146}$$

This expression may be used to simplify (145).

This general approach may be applied to our problem. Instead of using the phase of the conjugate-product of adjacent samples, we must use the phase of the conjugate product of samples

spaced L_q samples apart. To see that this is so, we start with (31), repeated here for convenience:

$$r(i+\ell L_q+m)=\alpha(m)e^{j\omega_0(i+\ell L_q+m)}+w(i+\ell L_q+m). \tag{147}$$

This equation is only true for $1\leq\ell\leq 6$ and $0\leq m\leq L_q-1$. Following Kay, this term may be re-written

$$r(i+\ell L_q+m)=\alpha(m)e^{j\omega_0(i+\ell L_q+m)}\left[1+\tilde{w}(i+\ell L_q+m)\right] \tag{148}$$

where

$$\tilde{w}(i+\ell L_q+m)=\frac{1}{\alpha(m)}e^{-j\omega_0(i+\ell L_q+m)}w(i+\ell L_q+m) \tag{149}$$

is a proper complex-valued Gaussian random variable with zero mean and variance

$$\frac{1}{2}\mathrm{E}\left\{\,|\tilde{w}(i+\ell L_q+m)|^2\right\}=\frac{\sigma_w^2}{|\alpha(m)|^2}. \tag{150}$$

Now we have

$$\begin{aligned}\arg\Big\{r(i+\ell L_q+m)\Big\}&=\arg\Big\{\alpha(m)\Big\}+\omega_0(i+\ell L_q+m)+\tan^{-1}\left\{\frac{\tilde{w}_I(i+\ell L_q+m)}{1+\tilde{w}_R(i+\ell L_q+m)}\right\}\\&\approx\arg\Big\{\alpha(m)\Big\}+\omega_0(i+\ell L_q+m)+\tan^{-1}\left\{\tilde{w}_I(i+\ell L_q+m)\right\}\\&\approx\arg\Big\{\alpha(m)\Big\}+\omega_0(i+\ell L_q+m)+\tilde{w}_I(i+\ell L_q+m)\end{aligned} \tag{151}$$

where the last two approximations hold if $\tilde{w}_R(i+\ell L_q+m)\ll 1$ and $\tilde{w}_I(i+\ell L_q+m)\ll 1$ with high probability. Similarly, we have

$$\begin{aligned}r(i+(\ell-1)L_q+m)&=\alpha(m)e^{j\omega_0(i+(\ell-1)L_q+m)}+w(i+(\ell-1)L_q+m) &(152)\\&=\alpha(m)e^{j\omega_0(i+(\ell-1)L_q+m)}\left[1+\tilde{w}(i+(\ell-1)L_q+m)\right] &(153)\end{aligned}$$

so that

$$\arg\Big\{r(i+(\ell-1)L_q+m)\Big\}\approx\arg\Big\{\alpha(m)\Big\}+\omega_0(i+(\ell-1)L_q+m)+\tilde{w}_I(i+(\ell-1)L_q+m). \tag{154}$$

This relationship only works for $\ell=2,3,4,5,6$ and $m=0,1,\ldots,L_q-1$. This allows us to write

$$\Delta_{m,\ell}=\arg\Big\{r(i+\ell L_q+m)r^*(i+(\ell-1)L_q+m)\Big\}$$

$$\begin{aligned}
&\approx \arg\Big\{\alpha(m)\Big\} + \omega_0(i+\ell L_q+m) + \tilde{w}_I(i+\ell L_q+m) \\
-\arg\Big\{\alpha(m)\Big\} &- \omega_0(i+(\ell-1)L_q+m) - \tilde{w}_I(i+(\ell-1)L_q+m) \\
&= L_q\omega_0 + \tilde{w}_I(i+\ell L_q+m) - \tilde{w}_I(i+(\ell-1)L_q+m) \quad (155)
\end{aligned}$$

for $\ell = 2,3,4,5,6$ and $m = 0,1,\ldots,L_q-1$. By properly ordering the phase differences $\Delta_{m,\ell}$ a workable estimator can be derived. Stacking the differences by ℓ then m produces the following set of equations:

$$\begin{bmatrix} \Delta_{0,2} \\ \vdots \\ \Delta_{0,6} \\ \Delta_{1,2} \\ \vdots \\ \Delta_{1,6} \\ \vdots \\ \Delta_{L_q-1,2} \\ \vdots \\ \Delta_{L_q-1,6} \end{bmatrix} = L_q\omega_0 \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \begin{bmatrix} \tilde{w}_I(i+2L_q) - \tilde{w}_I(i+L_q) \\ \vdots \\ \tilde{w}_I(i+6L_q) - \tilde{w}_I(i+5L_q) \\ \tilde{w}_I(i+2L_q+1) - \tilde{w}_I(i+L_q+1) \\ \vdots \\ \tilde{w}_I(i+6L_q+1) - \tilde{w}_I(i+5L_q+1) \\ \vdots \\ \tilde{w}_I(i+3L_q-1) - \tilde{w}_I(i+2L_q-1) \\ \vdots \\ \tilde{w}_I(i+7L_q-1) - \tilde{w}_I(i+6L_q-1) \end{bmatrix} \quad (156)$$

Note that the ordering of the phase differences is not the natural temporal ordering. The ordering is best understood by referring to Figure 57. The ordering starts with the first sample in the 6 middle q-blocks (this is $m = 0$) and lists the phase differences between the first samples in adjacent q-blocks. Next we move the second samples in the 6 middle q-blocks and we list the phase differences between the second samples in adjacent q-blocks. And so on until the last sample in each of the 6 middle q-blocks.

For notational convenience, we express the equations (156) as

$$\begin{bmatrix} \boldsymbol{\Delta}_0 \\ \boldsymbol{\Delta}_1 \\ \vdots \\ \boldsymbol{\Delta}_{L_q-1} \end{bmatrix} = L_q\omega_0 \begin{bmatrix} \mathbf{1}_5 \\ \mathbf{1}_5 \\ \vdots \\ \mathbf{1}_5 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_{L_q-1} \end{bmatrix}. \quad (157)$$

where $\boldsymbol{\Delta}_m$ is a 5×1 segment of the phase differences corresponding to a fixed m, $\mathbf{1}_5$ is the 5×1 vector of ones, and $\mathbf{w}_m$ is the 5×1 vector of noise differences corresponding to a fixed value of

m. The vector $\mathbf{w}_m$ is a real-valued Gaussian random vector with zero mean and covariance matrix

$$\mathrm{E}\left\{\mathbf{w}_m \mathbf{w}_m^\top\right\} = \frac{\sigma_w^2}{|\alpha(m)|^2}\mathbf{C}_5 \tag{158}$$

where

$$\mathbf{C}_5 = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \tag{159}$$

To formulate the maximum likelihood estimate, we will find it more convenient to express (157) as

$$\mathbb{A} = L_q \omega_0 \mathbb{1} + \mathbb{w}. \tag{160}$$

where $\mathbb{w}$ is a real-valued Gaussian random vector with zero mean and covariance matrix

$$\mathbb{C} = \mathrm{E}\left\{\mathbb{w}\mathbb{w}^\top\right\} = \begin{bmatrix} \frac{\sigma_w^2}{|\alpha(0)|^2}\mathbf{C}_5 & & & \\ & \frac{\sigma_w^2}{|\alpha(1)|^2}\mathbf{C}_5 & & \\ & & \ddots & \\ & & & \frac{\sigma_w^2}{|\alpha(L_q-1)|^2}\mathbf{C}_5 \end{bmatrix}. \tag{161}$$

The log likelihood function is

$$\Lambda(\omega_0) = \left(\mathbb{A} - L_q\omega_0\mathbb{1}\right)^\top \mathbb{C}^{-1}\left(\mathbb{A} - L_q\omega_0\mathbb{1}\right). \tag{162}$$

The maximum likelihood estimate is

$$\hat{\omega}_0 = \frac{1}{L_q}\frac{\mathbb{1}^\top\mathbb{C}^{-1}\mathbb{A}}{\mathbb{1}^\top\mathbb{C}^{-1}\mathbb{1}}. \tag{163}$$

Because $\mathbb{C}$ is block diagonal, $\mathbb{C}^{-1}$ may be expressed as

$$\mathbb{C}^{-1} = \begin{bmatrix} \frac{|\alpha(0)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} & & & \\ & \frac{|\alpha(1)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} & & \\ & & \ddots & \\ & & & \frac{|\alpha(L_q-1)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} \end{bmatrix}. \tag{164}$$

Consequently, quadratic term in the numerator of (163) may be written as

$$\mathbb{1}^\top \mathbb{C}^{-1}\mathbb{\Delta} = \begin{bmatrix} \mathbf{1}_5^\top & \mathbf{1}_5^\top & \dots & \mathbf{1}_5^\top \end{bmatrix} \begin{bmatrix} \frac{|\alpha(0)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} & & & \\ & \frac{|\alpha(1)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} & & \\ & & \ddots & \\ & & & \frac{|\alpha(L_q-1)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{\Delta}_0 \\ \mathbf{\Delta}_1 \\ \vdots \\ \mathbf{\Delta}_{L_q-1} \end{bmatrix}$$

$$= \frac{1}{\sigma_w^2}\sum_{m=0}^{L_q-1} |\alpha(m)|^2 \mathbf{1}_5^\top \mathbf{C}_5^{-1}\mathbf{\Delta}_m. \tag{165}$$

Using the fact

$$\mathbf{1}_5^\top \mathbf{C}_5^{-1} = \begin{bmatrix} \frac{5}{2} & \frac{8}{2} & \frac{9}{2} & \frac{8}{2} & \frac{5}{2} \end{bmatrix} \tag{166}$$

(165) may be written as

$$\mathbb{1}^\top \mathbb{C}^{-1}\mathbb{\Delta} = \frac{1}{2\sigma_w^2}\sum_{m=0}^{L_q-1} |\alpha(m)|^2 \left(5\Delta_{m,2} + 8\Delta_{m,3} + 9\Delta_{m,4} + 8\Delta_{m,5} + 5\Delta_{m,6}\right). \tag{167}$$

The denominator term of (165) may be written as

$$\mathbb{1}^\top \mathbb{C}^{-1}\mathbb{1} =$$

$$\begin{bmatrix}\mathbf{1}_5^\top & \mathbf{1}_5^\top & \dots & \mathbf{1}_5^\top\end{bmatrix}\begin{bmatrix}\frac{|\alpha(0)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} & & & \\ & \frac{|\alpha(1)|^2}{\sigma_w^2}\mathbf{C}_5^{-1} & & \\ & & \ddots & \\ & & & \frac{|\alpha(L_q-1)|^2}{\sigma_w^2}\mathbf{C}_5^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{1}_5\\ \mathbf{1}_5\\ \vdots\\ \mathbf{1}_5\end{bmatrix}$$

$$= \frac{1}{\sigma_w^2}\sum_{m=0}^{L_q-1}|\alpha(m)|^2\mathbf{1}_5^\top\mathbf{C}_5^{-1}\mathbf{1}_5. \quad (168)$$

Now using the fact

$$\mathbf{1}_5^\top\mathbf{C}_5^{-1}\mathbf{1}_5 = \frac{35}{2}, \quad (169)$$

(168) may be written

$$\mathbb{1}^\top\mathbb{C}^{-1}\mathbb{1} = \frac{35}{2\sigma_w^2}\sum_{m=0}^{L_q-1}|\alpha(m)|^2 = \frac{35}{2\sigma_w^2}\alpha^2. \quad (170)$$

Putting this all together gives

$$\hat{\omega}_0 = \frac{1}{35L_q}\sum_{m=0}^{L_q-1}\frac{|\alpha(m)|^2}{\alpha^2}\left(5\Delta_{m,2}+8\Delta_{m,3}+9\Delta_{m,4}+8\Delta_{m,5}+5\Delta_{m,6}\right). \quad (171)$$

The challenge with this form of the estimator is that the $\alpha(m)$ are unknown. The $\alpha(m)$ are used to weight each block of phase differences (and this weighting is in proportion to the relative channel output power). A suboptimal approach is to set the weights to be equal:

$$\frac{|\alpha(m)|^2}{\alpha^2} \to \frac{1}{L_q}. \quad (172)$$

The result is

$$\hat{\omega}_0 = \frac{1}{35L_q^2}\sum_{m=0}^{L_q-1}\left(5\Delta_{m,2}+8\Delta_{m,3}+9\Delta_{m,4}+8\Delta_{m,5}+5\Delta_{m,6}\right). \quad (173)$$

3.7 Simulation Results

The performance of the five estimators described in the previous section was simulated over the eleven test channels. Figures 58 – 61 are results for test channel 1; Figures 62 – 65 are the results

for test channel 2; and so on. Each set of plots shows the simulated mean of the estimators for $E_b/N_0 = 10$ dB, 20 dB, and 30 dB, and the estimator error variance. The mean is used to assess the operating range of each estimator. The variance is used to asses how accurate the estimators are.

The key observations regarding the mean performance are the following:

- The modified Fitz and modified M&M estimators have the narrowest estimation range. Is this narrow range sufficient?
  - When we fielded the space-time coded SOQPSK-TG system, we were given a performance specification to accommodate a frequency offset of ± 100 kHz. Assuming this criterion, the modified Fitz and modified M&M estimators will not do.
  - In an email to MDR on 9 May 2013, Mark Geoghegan (Quasonix, LLC) pointed out the following.
    1. Paragraph 5.1.3.1.2.1 of the iNet RAN specification requires the downlink carrier frequency offset error be bounded by ± 7.5 ppm. This corresponds to ± 11.513 kHz at 1525 MHz (the high end of L-band) or ± 38.625 kHz at 5150 MHz (the high end of middle C-band).
    2. Quasonix transmitters (which we are using) typically have a much lower carrier frequency error: ± 2 ppm. This corresponds to ± 3050 kHz at 1525 MHz and ± 10.3 kHz at 5150 MHz.
    3. Paragraph 5.1.3.1.2.5 of the iNet RAN reads

       "The downlink transmission frequency error seen at the Ground Station receiver due to Doppler shall be bounded by ± 2.5 ppm. This corresponds to a frequency error spread of ± 15 kHz at a transmission frequency of 6 GHz. This frequency shift due to Doppler effects is budgeted for the combined total of relative motion between two antennas, either between a stationary antenna and a moving antenna or between two moving antennas."
    4. Assuming an aircraft velocity of Mach 1 (approximately 300 m/s) the worst case frequency shift is ± 1.525 kHz at 1525 MHz and ± 5.15 kHz at 5150 MHz. The PAQ experiments will be conducted using a C-12 aircraft (velocity well less than Mach 1) and with a stationary ground station. Consequently, assuming a Doppler shift of ± 5 kHz should be sufficient for the PAQ experiments.

5. Clearly, the C-band experiments are the limiting case here.
   (a) Using the iNET RAN numbers, the transmitter contributes ± 38.625 kHz of frequency uncertainty and the Doppler contributes ± 12.875 kHz of frequency uncertainty, both at 5150 MHz. The total frequency uncertainty at 5150 MHz is ± 51.5 kHz.
   (b) Using Mark G.'s numbers, the transmitter contributes ± 9.9 kHz of frequency uncertainty and the Doppler contributes ± 5 kHz of frequency uncertainty to give a total of ± 15 kHz of frequency uncertainty. Mark G. recommends designing for twice this, or ± 30 kHz of frequency uncertainty.

   Assuming either the iNET or the Mark G. criterion, all the estimators will do.

- We also observe that at low signal-to-noise ratios, the modified Kay estimator has problems with test channels 1, 2, 3, 5, 7, 8, 10, and 11. Test channels 10 and 11 are particularly troublesome for the Kay estimator.

For the estimator error variance, the following observations are important.

- The modified Kay estimator has problems with test channels 1, 2, 3, 5, 7, 8, 10, and 11. The corresponds to problems observed with the mean performance of the Kay estimator. The performance degradation observed here has its roots in the approximation (172). Plots of $|\alpha(m)|^2/\alpha^2$ for test channels 2, 3, and 4 are shown in Figure 102. If the hypothesis is true, then it must be the case that the approximation (172) is much more accurate for test channel 4 than for test channels 2 and 3.

- Excluding the modified Kay estimator, the ESP estimator has a higher error variance than the modified Fitz, L&R, and M&M estimators.

- The estimators with the lowest error variance are the modified Fitz and M&M estimators, precisely those with the smallest operating range.

- All of the estimators do not reach the Cramér-Rao bound (262). But this is to be expected because the bound (262) assumed the use of $N_{\text{p}} + N_{\text{asm}} - N_1 - N_2$ samples whereas the estimators of Section 3 used $N_{\text{p}} - 2L_q = 6L_q$. The Cramér-Rao bound for estimating the frequency of a sinusoid in noise after observing N samples is roughly proportional to $1/N^3$ [8]. Assuming the estimators of Section 3 are operating at their Cramér-Rao bounds, the difference in signal-to-noise ratio between the Cramér-Rao bound (262) and the performance

of these estimators is roughly

$$\begin{aligned}\Delta\text{SNR} &\sim 10\log_{10}\left(\frac{(N_\text{p}+N_\text{asm}-N_1-N_2)^3}{(6L_q)^3}\right)\\ &= 30\log_{10}\left(\frac{N_\text{p}+N_\text{asm}-N_1-N_2}{6L_q}\right)\\ &= 30\log_{10}\left(\frac{347}{192}\right)\\ &\approx 7.7 \text{ dB}.\end{aligned} \tag{174}$$

This is the difference observed in Figures 65 – 73.

In summary, all the estimators, with the exception of the modified Kay estimator, have very close error variance performance and have operating ranges exceeding the Mark Geoghegan criterion.

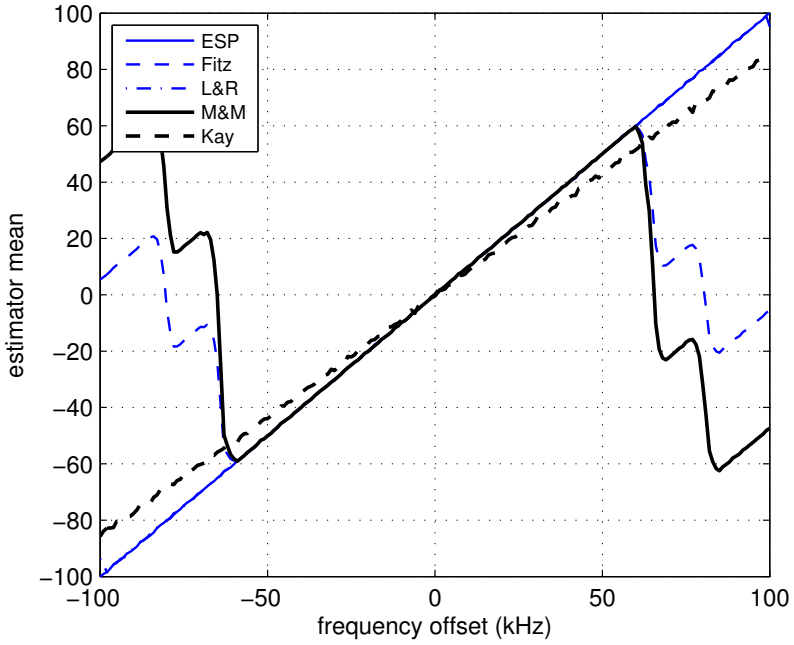

Figure 58: Simulated means of the frequency estimators as a function of frequency offset for test channel 1 with $E_b/N_0 = 10$ dB.

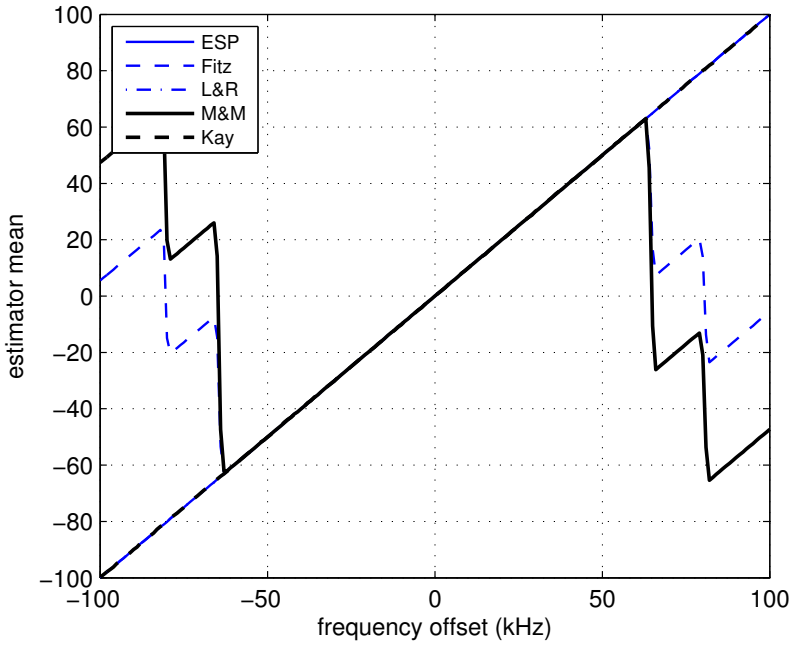

Figure 59: Simulated means of the frequency estimators as a function of frequency offset for test channel 1 with $E_b/N_0 = 20$ dB.

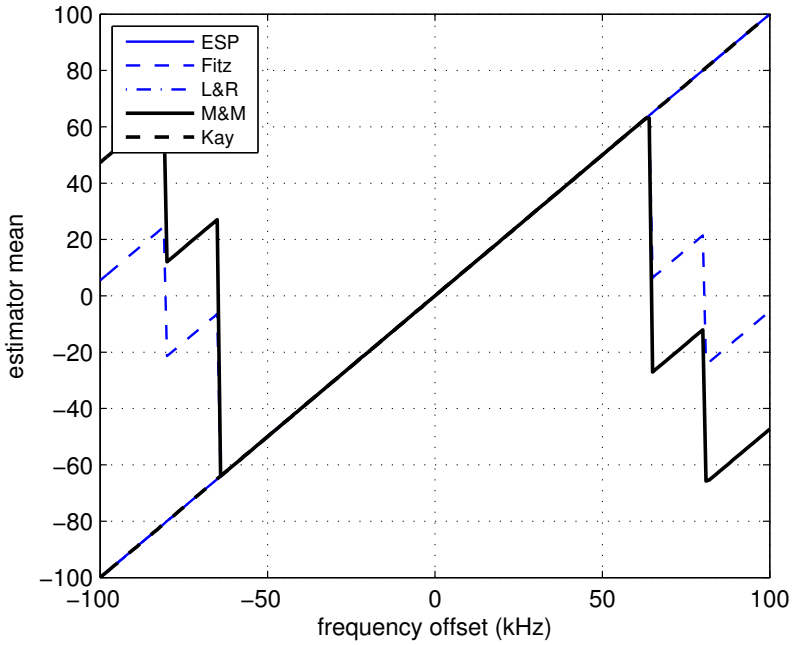

Figure 60: Simulated means of the frequency estimators as a function of frequency offset for test channel 1 with $E_b/N_0 = 30$ dB.

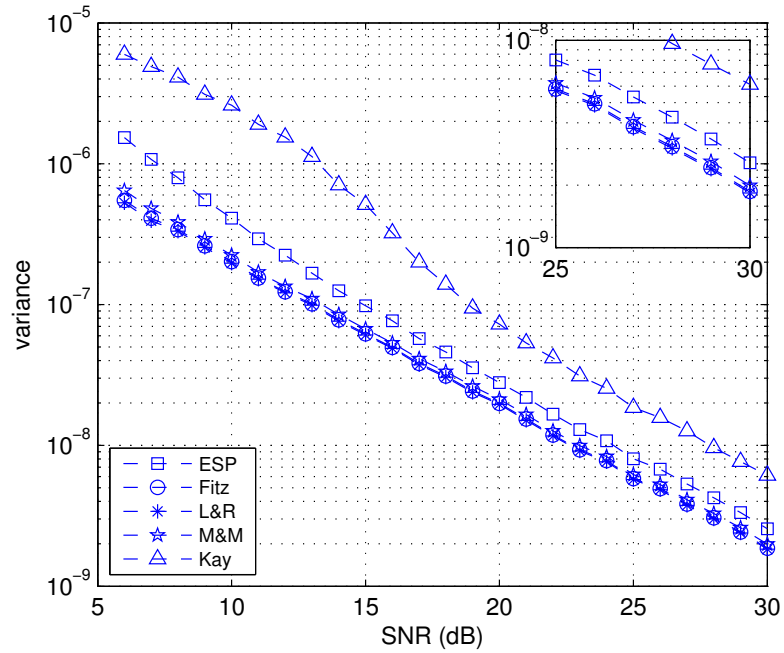

Figure 61: Estimator error variance as a function of E_b/N_0 for test channel 1.

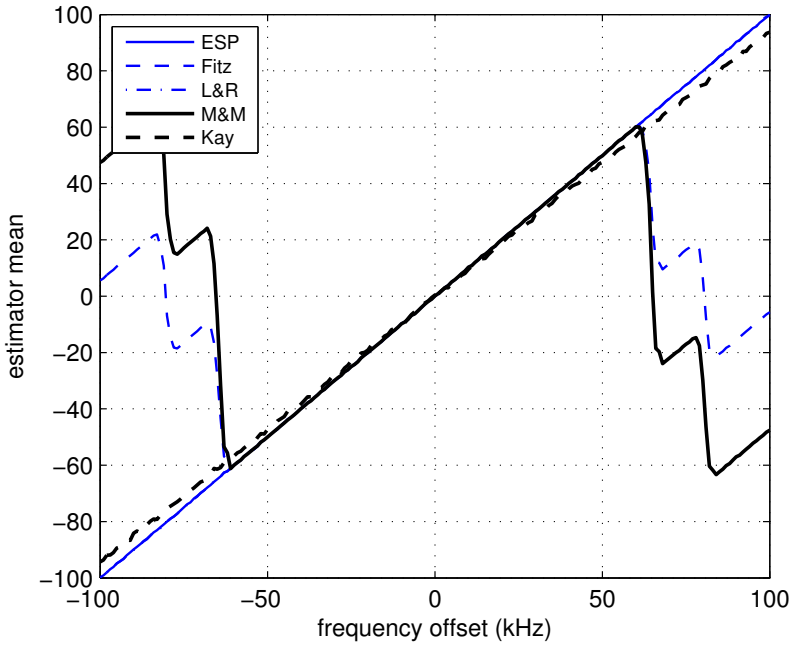

Figure 62: Simulated means of the frequency estimators as a function of frequency offset for test channel 2 with $E_b/N_0 = 10$ dB.

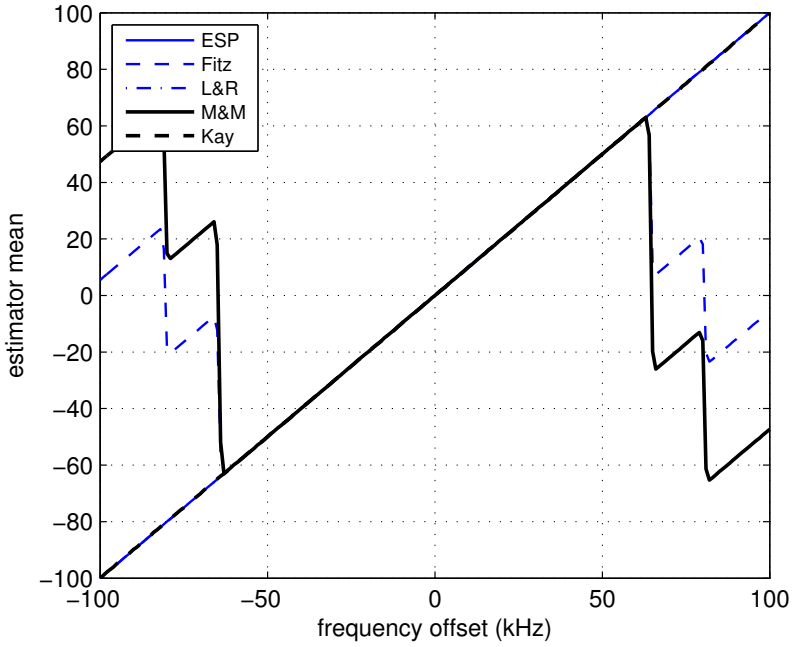

Figure 63: Simulated means of the frequency estimators as a function of frequency offset for test channel 2 with $E_b/N_0 = 20$ dB.

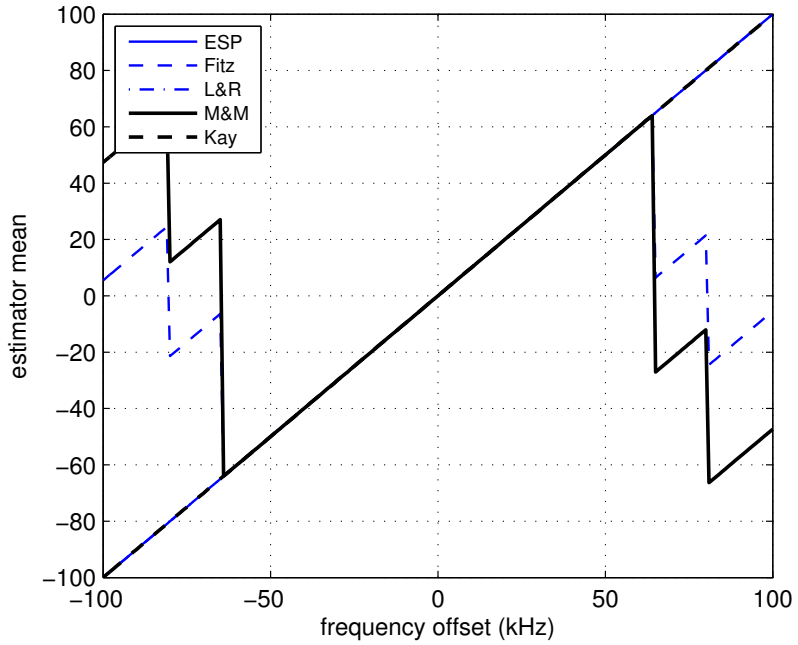

Figure 64: Simulated means of the frequency estimators as a function of frequency offset for test channel 2 with $E_b/N_0 = 30$ dB.

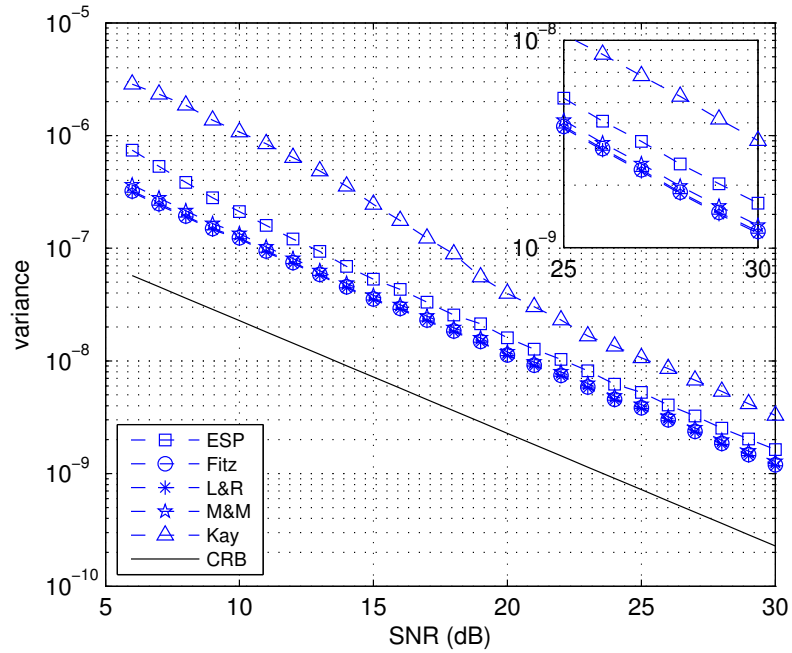

Figure 65: Estimator error variance as a function of E_b/N_0 for test channel 2.

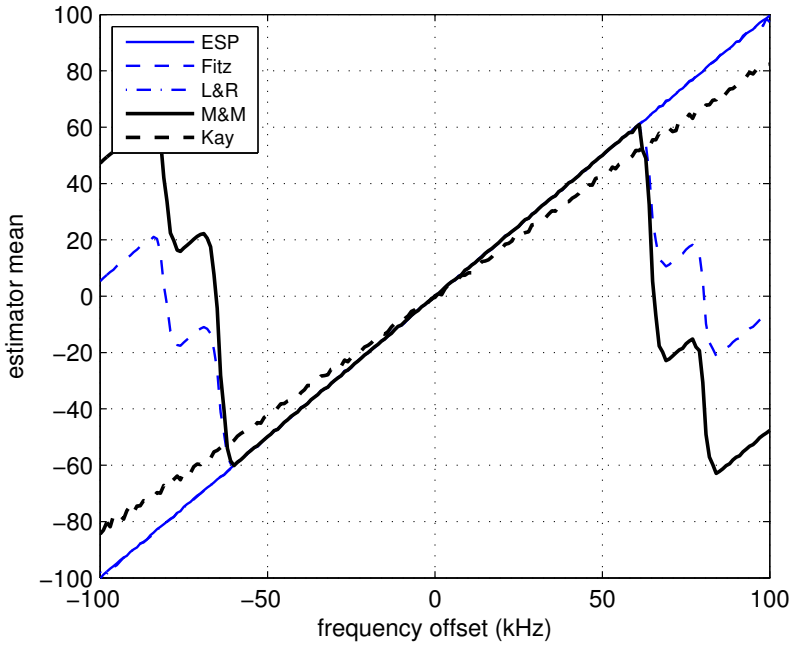

Figure 66: Simulated means of the frequency estimators as a function of frequency offset for test channel 3 with $E_b/N_0 = 10$ dB.

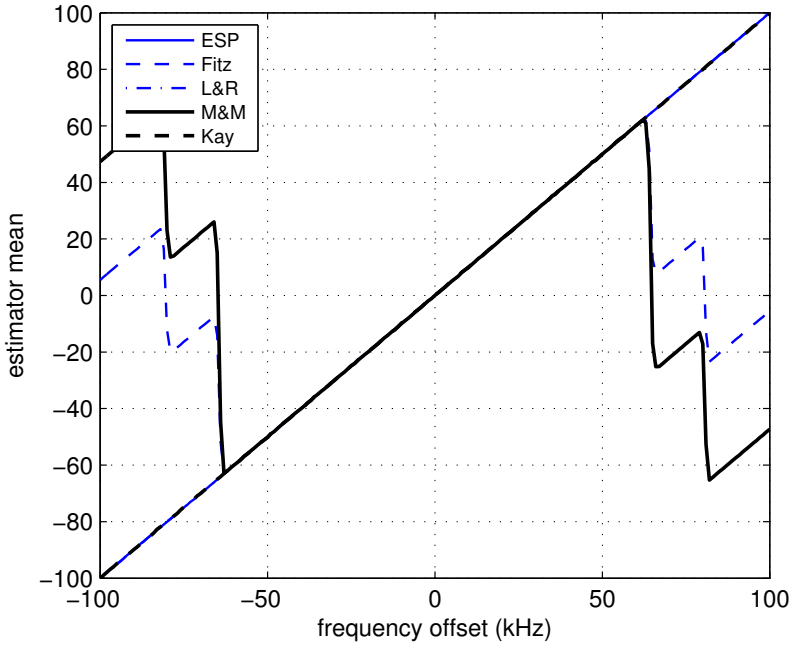

Figure 67: Simulated means of the frequency estimators as a function of frequency offset for test channel 3 with $E_b/N_0 = 20$ dB.

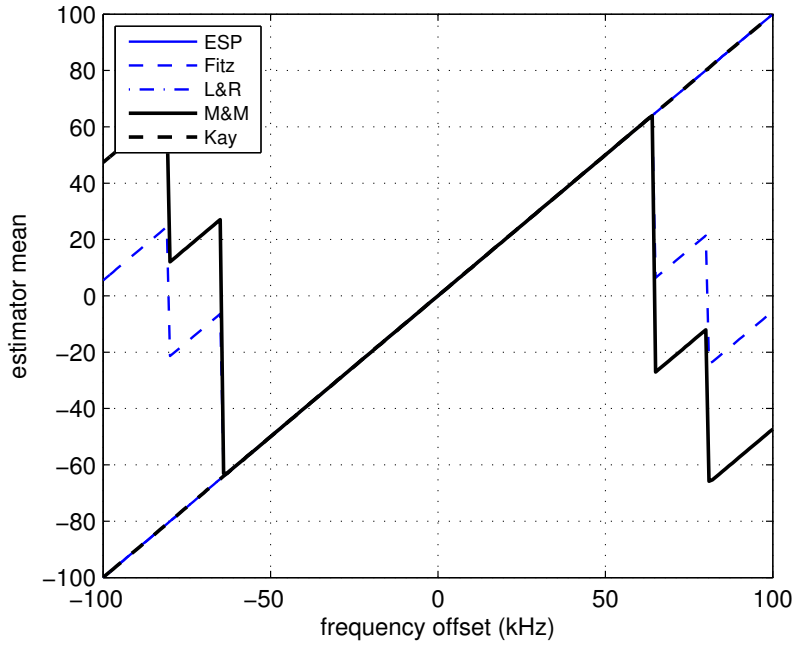

Figure 68: Simulated means of the frequency estimators as a function of frequency offset for test channel 3 with $E_b/N_0 = 30$ dB.

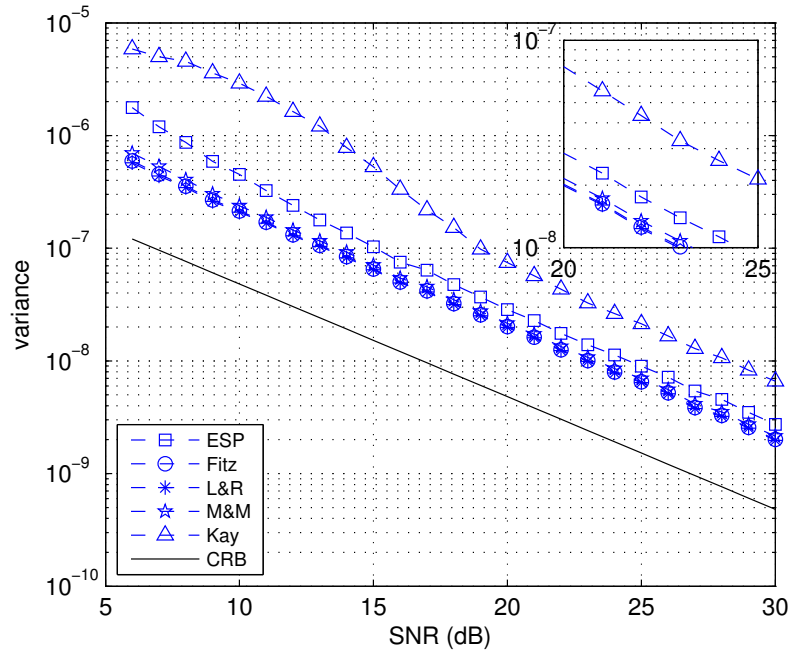

Figure 69: Estimator error variance as a function of E_b/N_0 for test channel 3.

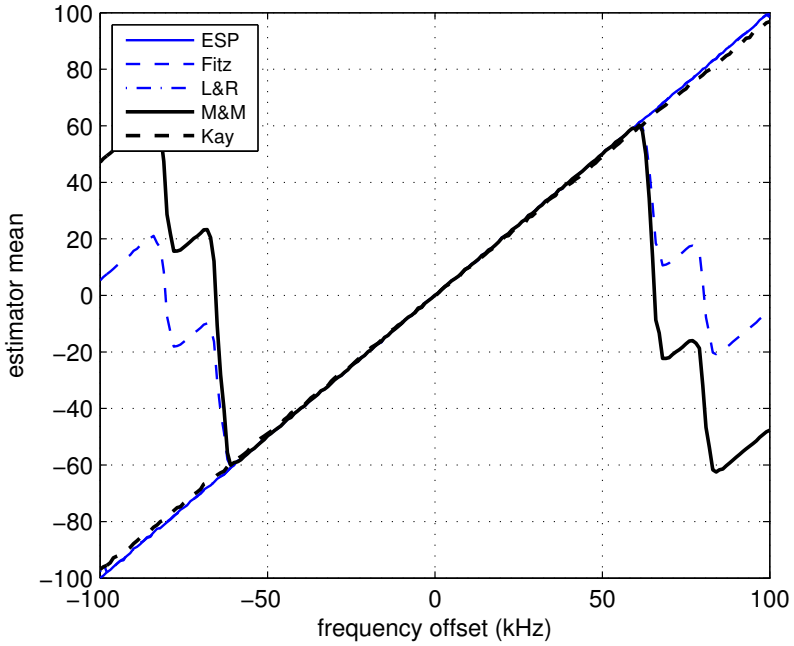

Figure 70: Simulated means of the frequency estimators as a function of frequency offset for test channel 4 with $E_b/N_0 = 10$ dB.

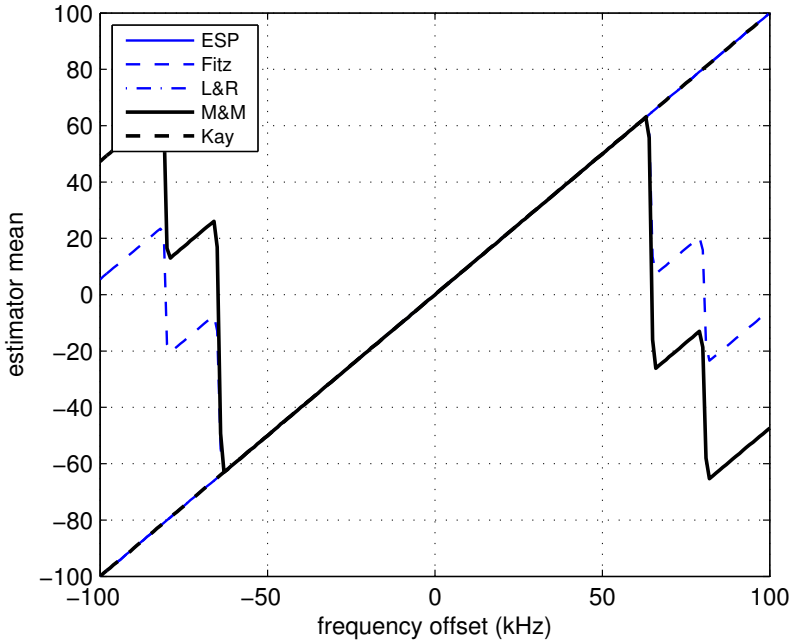

Figure 71: Simulated means of the frequency estimators as a function of frequency offset for example test channel 4 with $E_b/N_0 = 20$ dB.

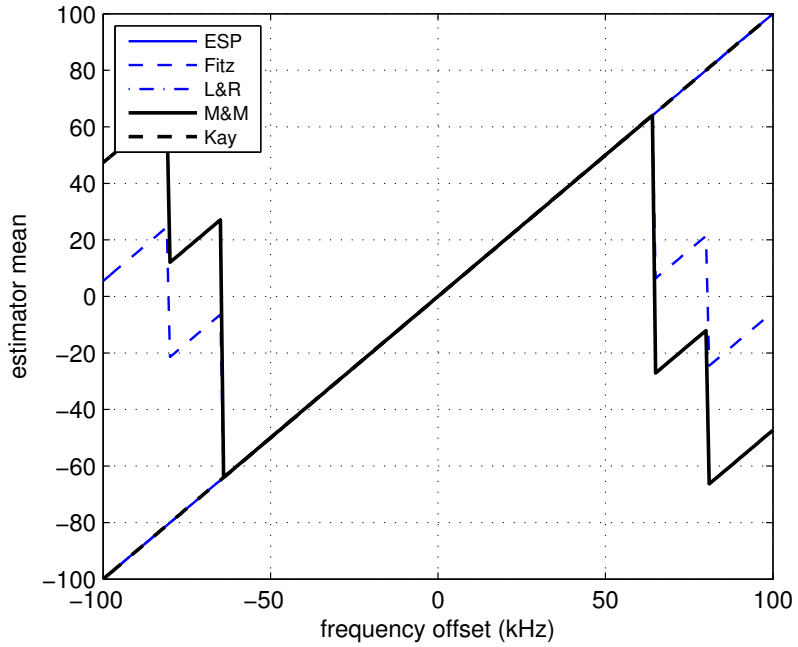

Figure 72: Simulated means of the frequency estimators as a function of frequency offset for example test channel 4 with $E_b/N_0 = 30$ dB.

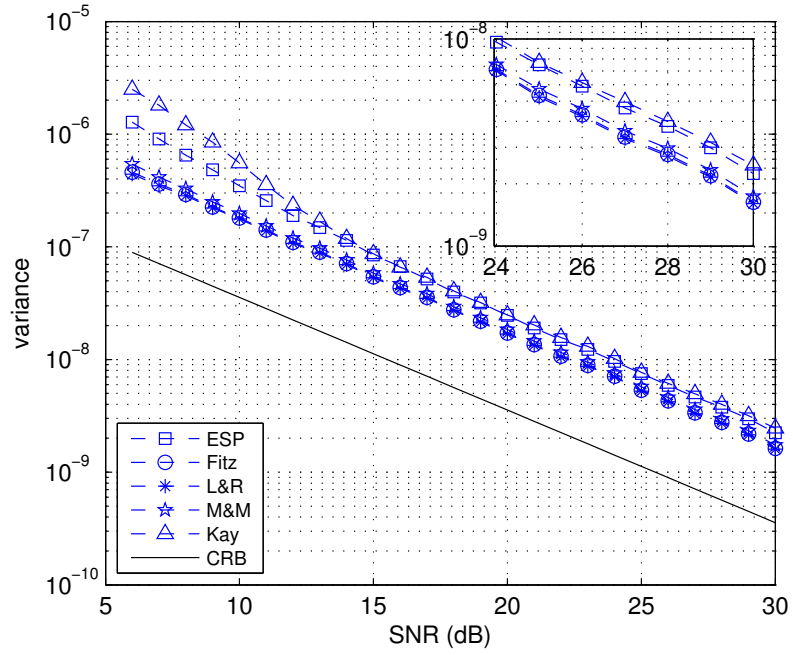

Figure 73: Estimator error variance as a function of E_b/N_0 for test channel 4.

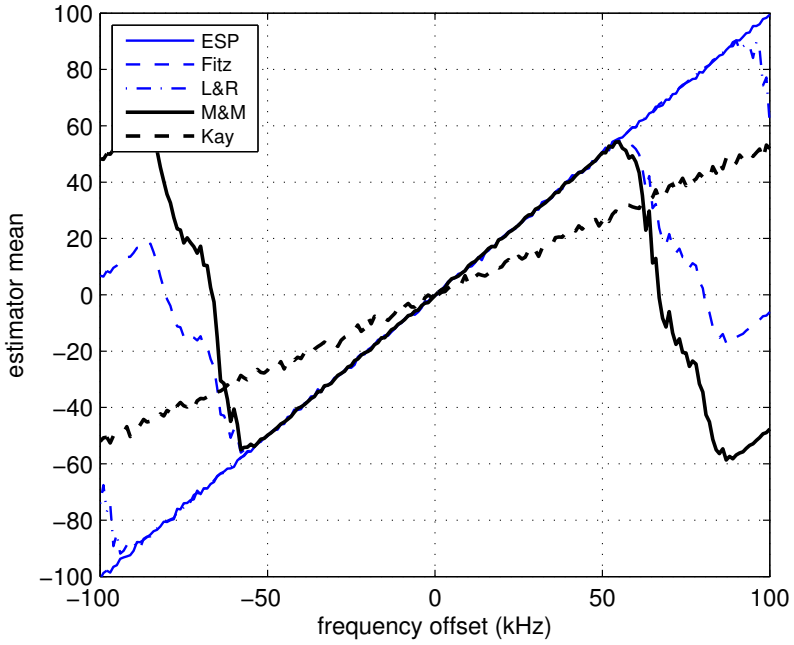

Figure 74: Simulated means of the frequency estimators as a function of frequency offset for test channel 5 with $E_b/N_0 = 10$ dB.

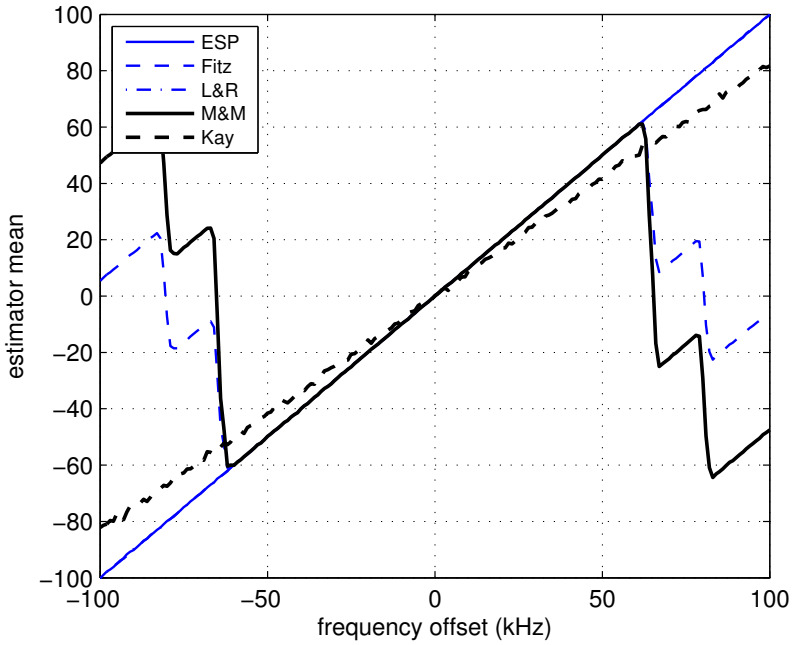

Figure 75: Simulated means of the frequency estimators as a function of frequency offset for example test channel 5 with $E_b/N_0 = 20$ dB.

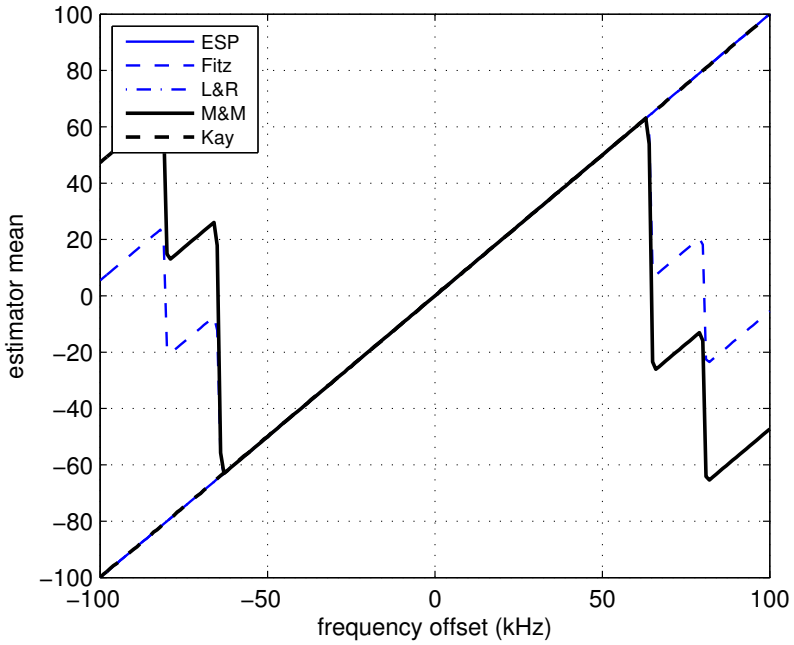

Figure 76: Simulated means of the frequency estimators as a function of frequency offset for example test channel 5 with $E_b/N_0 = 30$ dB.

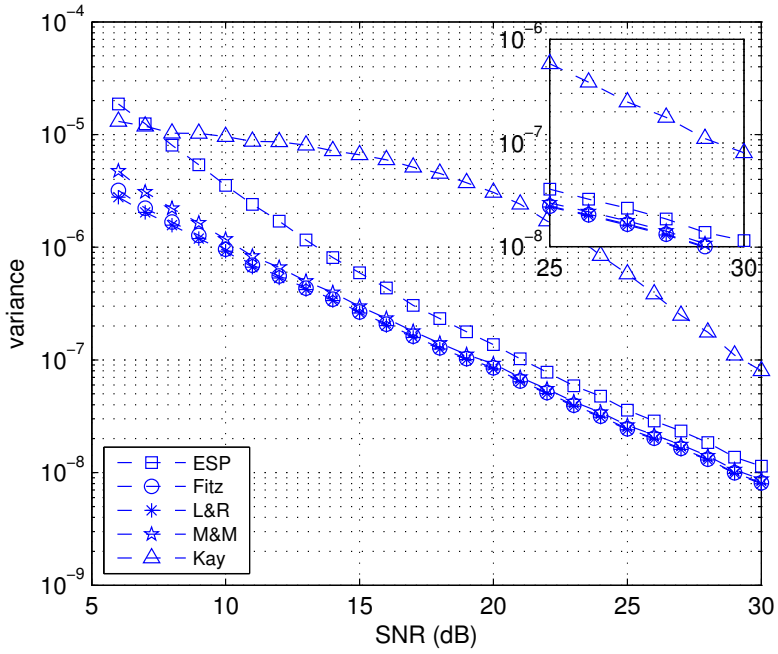

Figure 77: Estimator error variance as a function of E_b/N_0 for test channel 5.

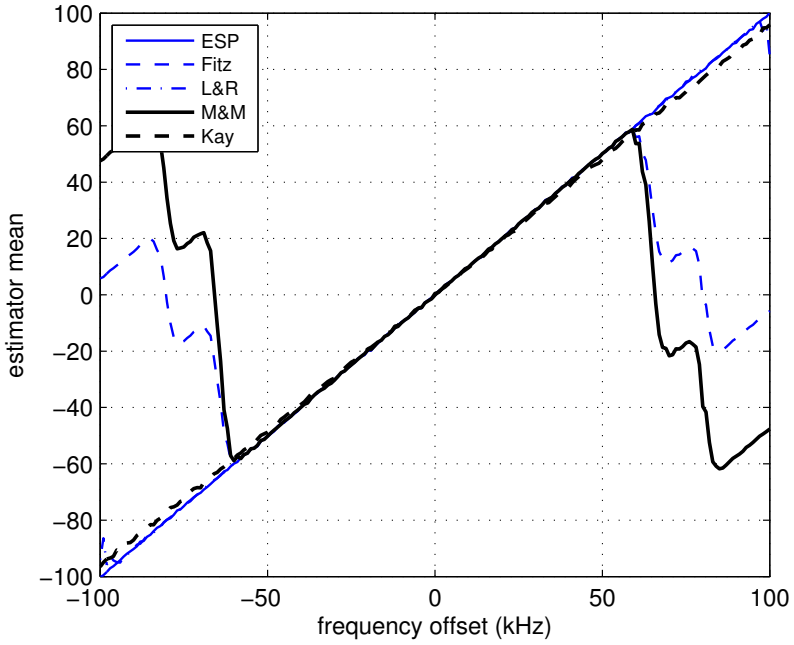

Figure 78: Simulated means of the frequency estimators as a function of frequency offset for test channel 6 with $E_b/N_0 = 10$ dB.

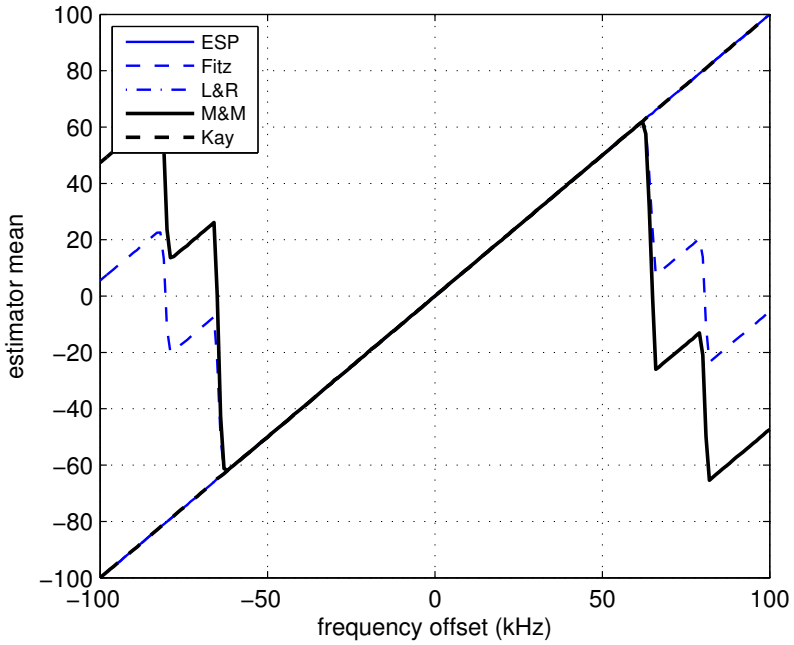

Figure 79: Simulated means of the frequency estimators as a function of frequency offset for example test channel 6 with $E_b/N_0 = 20$ dB.

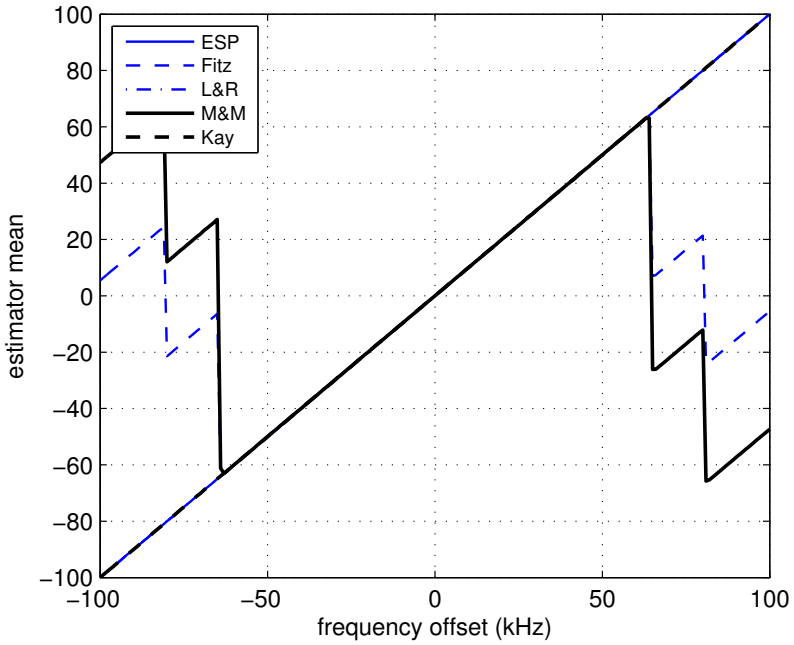

Figure 80: Simulated means of the frequency estimators as a function of frequency offset for example test channel 6 with $E_b/N_0 = 30$ dB.

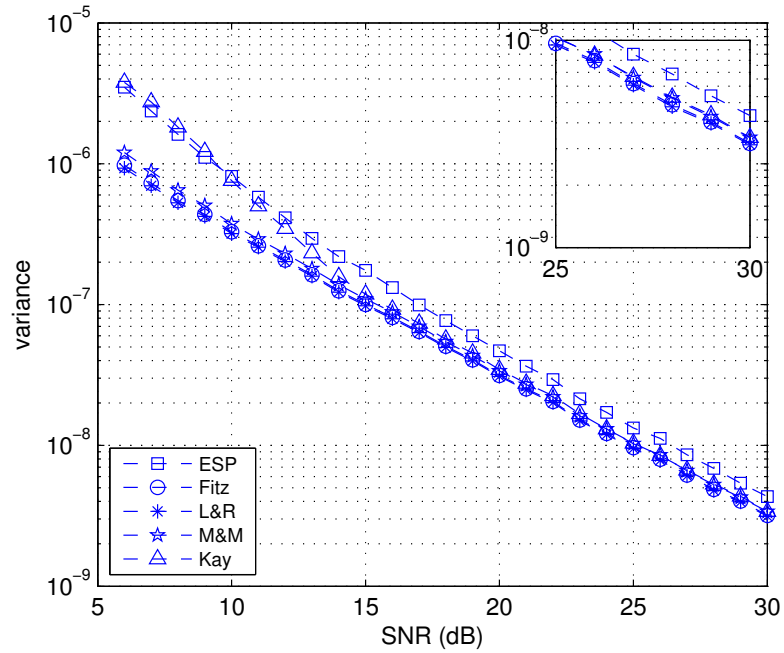

Figure 81: Estimator error variance as a function of E_b/N_0 for test channel 6.

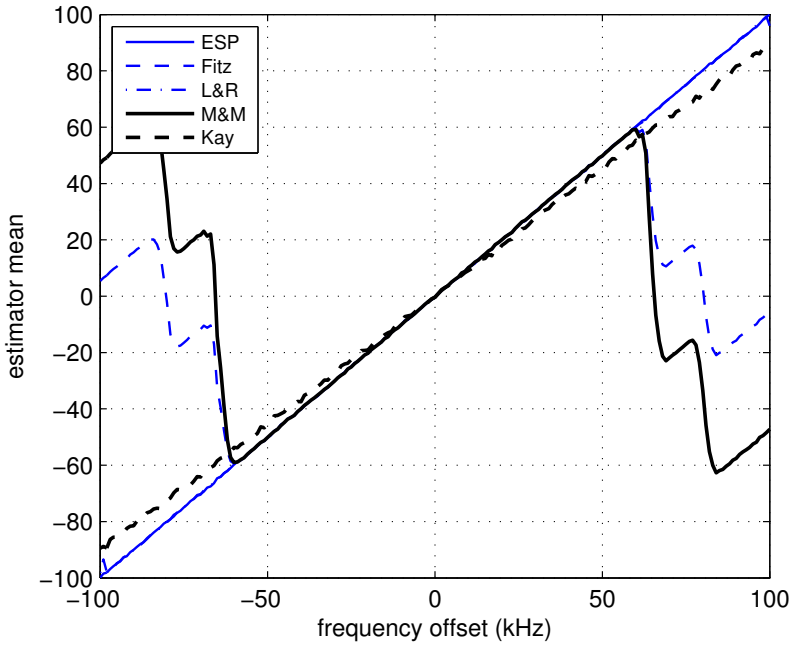

Figure 82: Simulated means of the frequency estimators as a function of frequency offset for test channel 7 with $E_b/N_0 = 10$ dB.

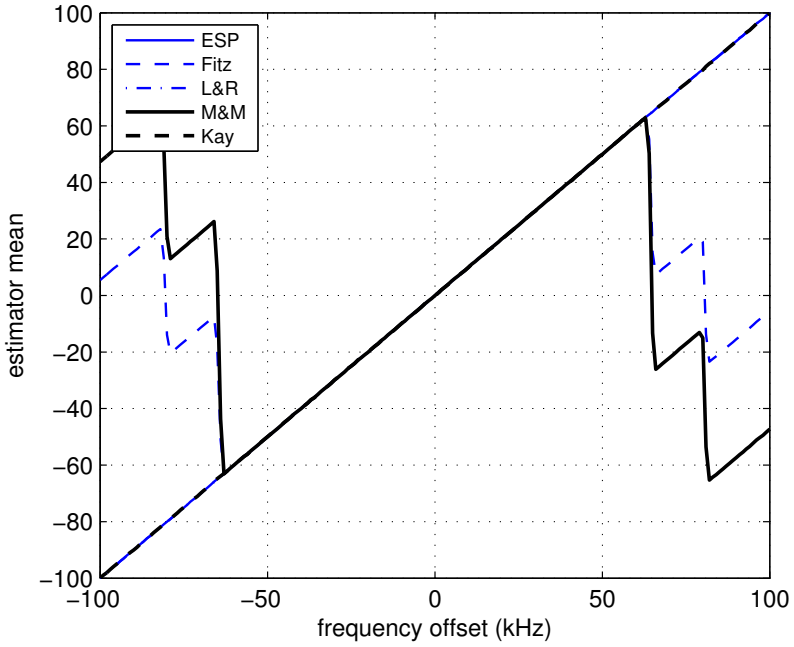

Figure 83: Simulated means of the frequency estimators as a function of frequency offset for example test channel 7 with $E_b/N_0 = 20$ dB.

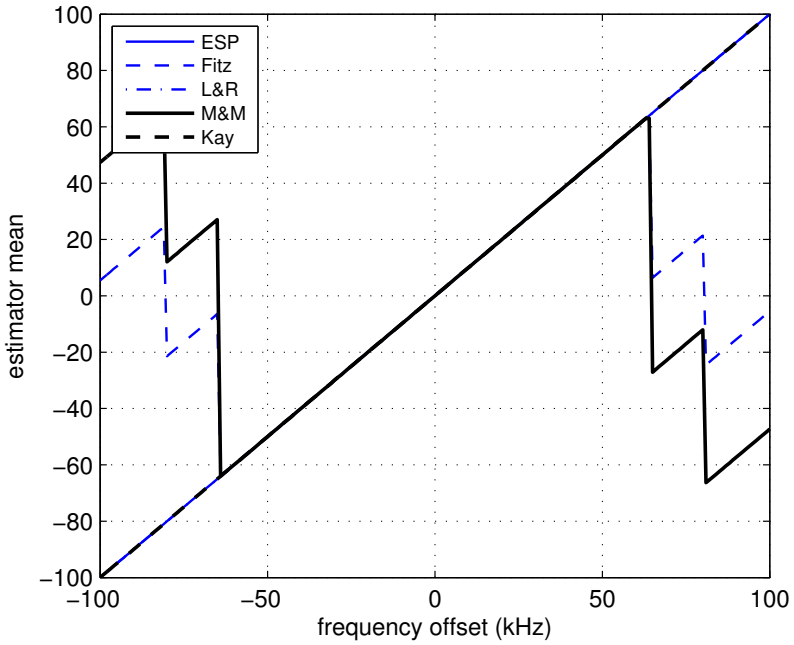

Figure 84: Simulated means of the frequency estimators as a function of frequency offset for example test channel 7 with $E_b/N_0 = 30$ dB.

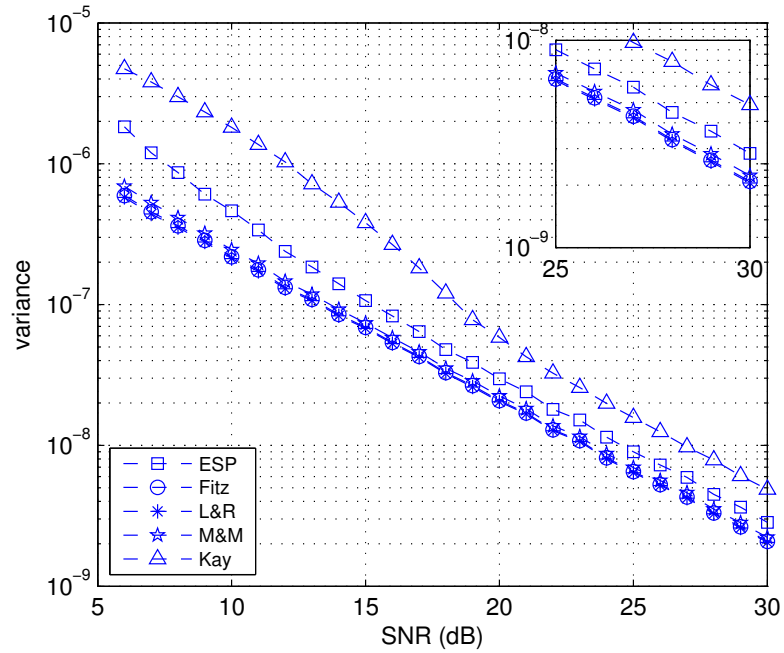

Figure 85: Estimator error variance as a function of E_b/N_0 for test channel 7.

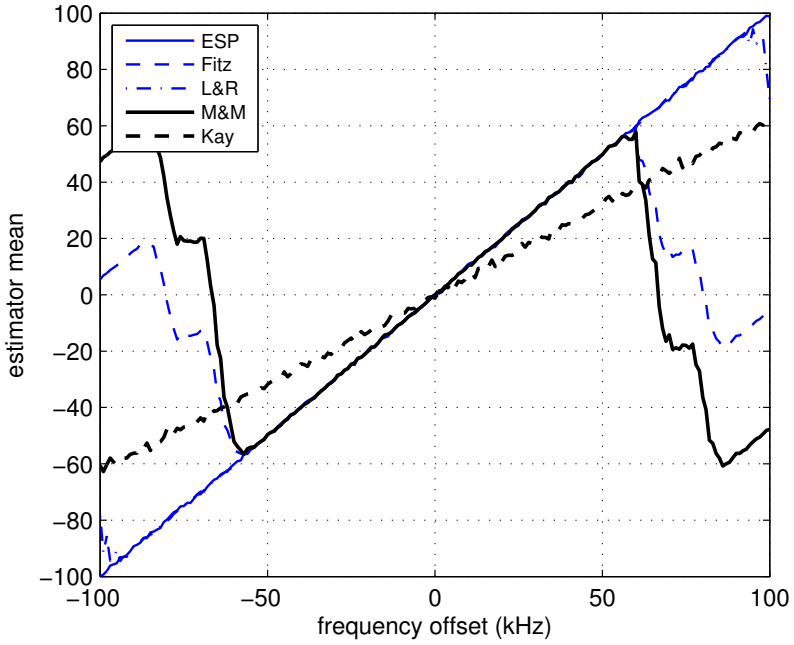

Figure 86: Simulated means of the frequency estimators as a function of frequency offset for test channel 8 with $E_b/N_0 = 10$ dB.

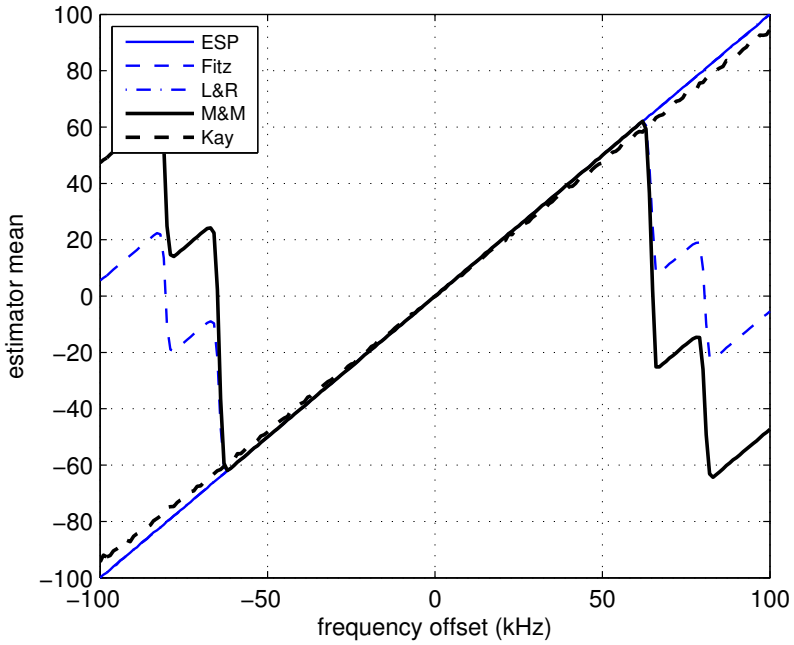

Figure 87: Simulated means of the frequency estimators as a function of frequency offset for example test channel 8 with $E_b/N_0 = 20$ dB.

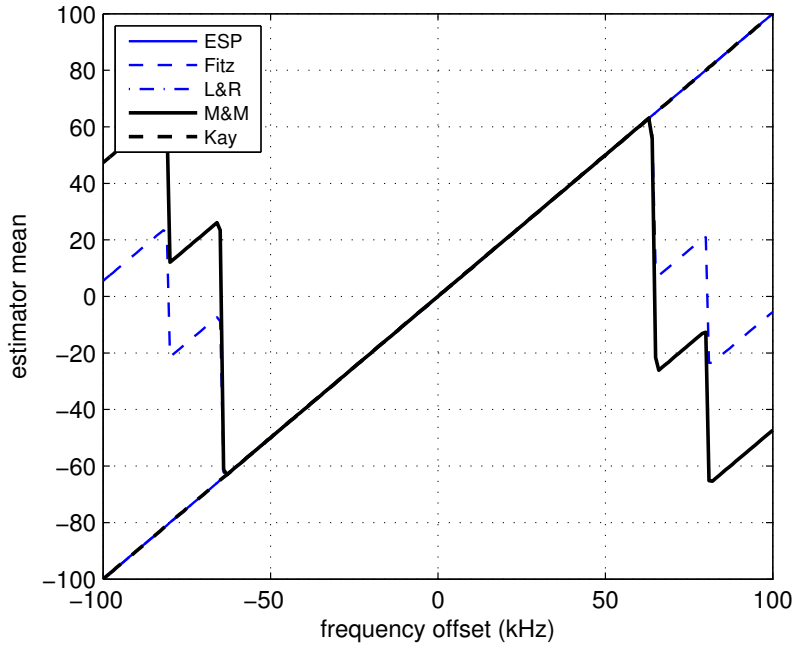

Figure 88: Simulated means of the frequency estimators as a function of frequency offset for example test channel 8 with $E_b/N_0 = 30$ dB.

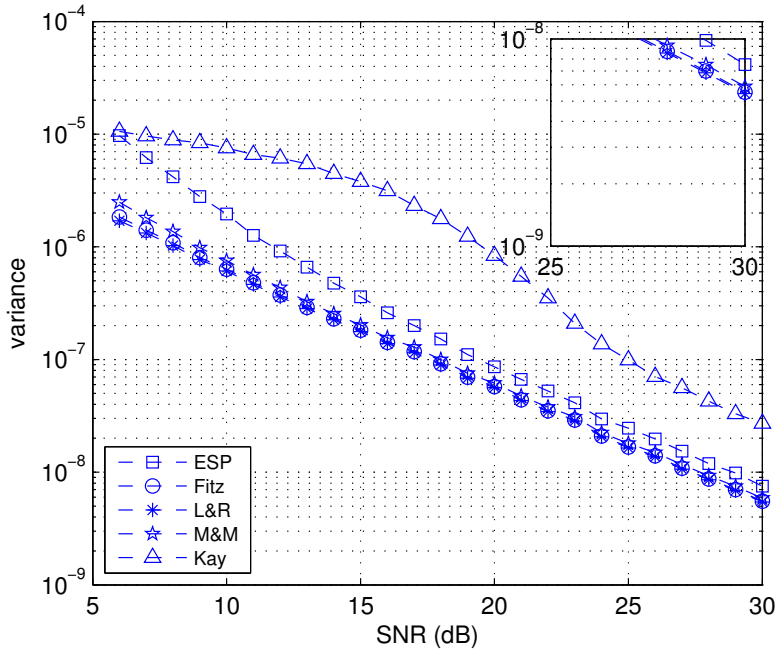

Figure 89: Estimator error variance as a function of E_b/N_0 for test channel 8.

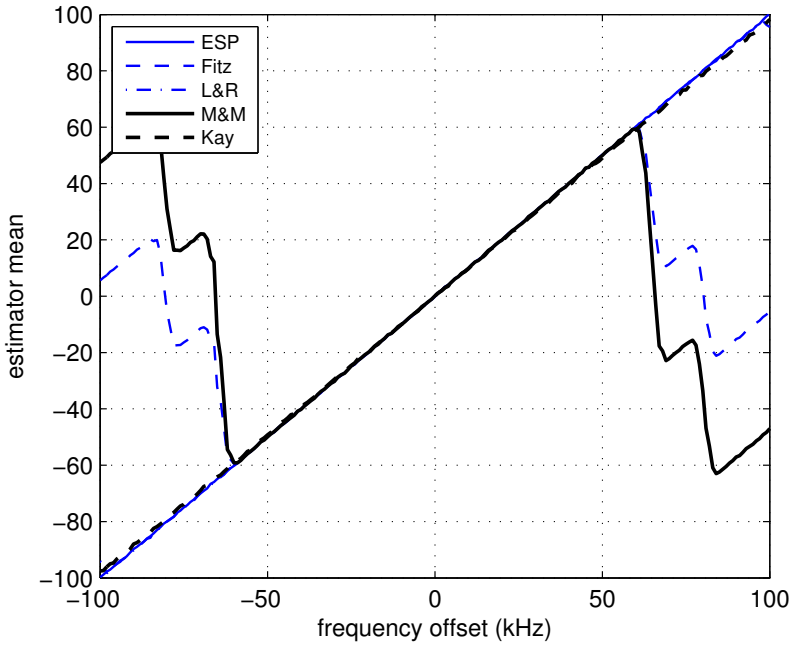

Figure 90: Simulated means of the frequency estimators as a function of frequency offset for test channel 9 with $E_b/N_0 = 10$ dB.

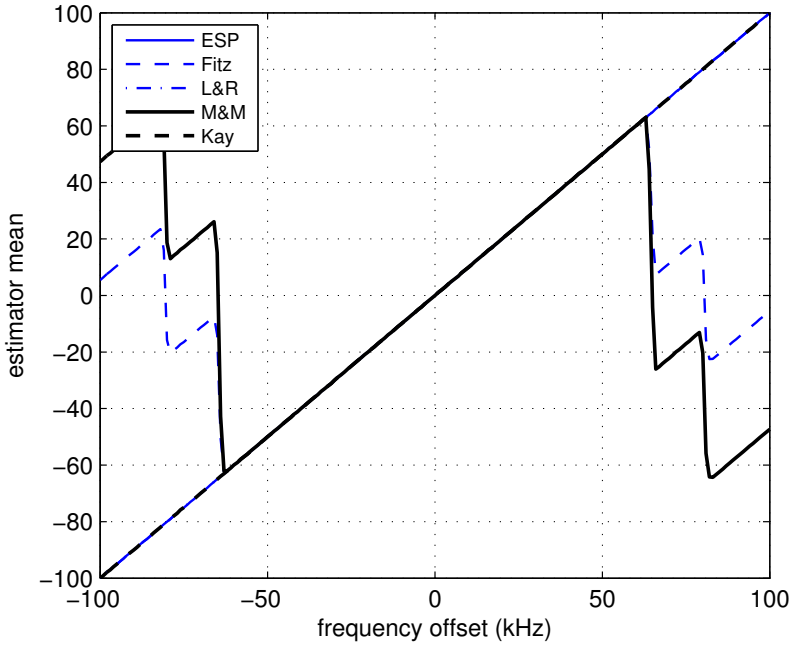

Figure 91: Simulated means of the frequency estimators as a function of frequency offset for example test channel 9 with $E_b/N_0 = 20$ dB.

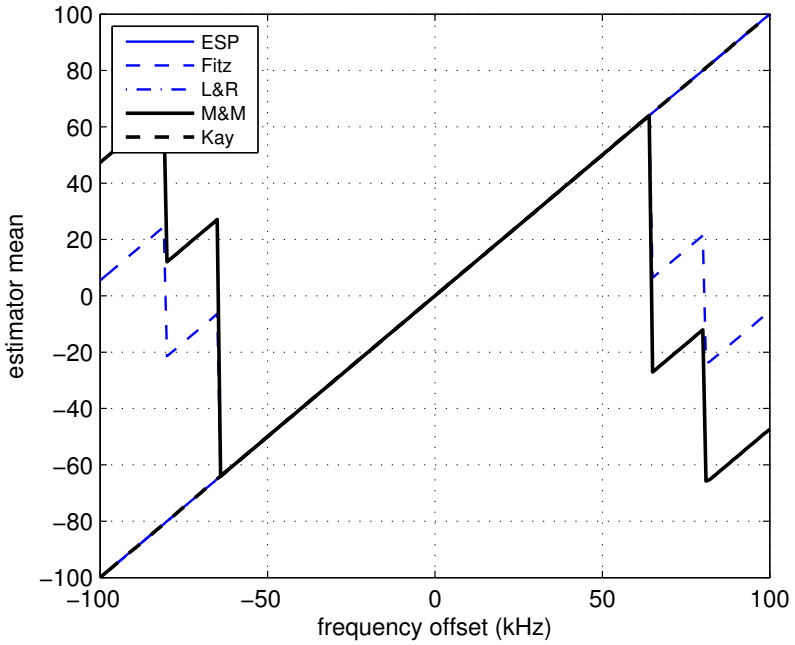

Figure 92: Simulated means of the frequency estimators as a function of frequency offset for example test channel 9 with $E_b/N_0 = 30$ dB.

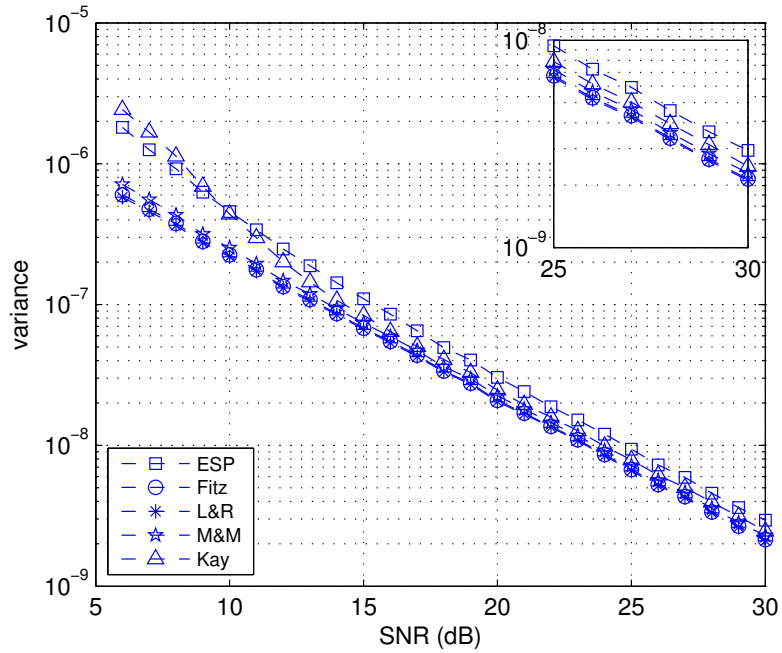

Figure 93: Estimator error variance as a function of E_b/N_0 for test channel 9.

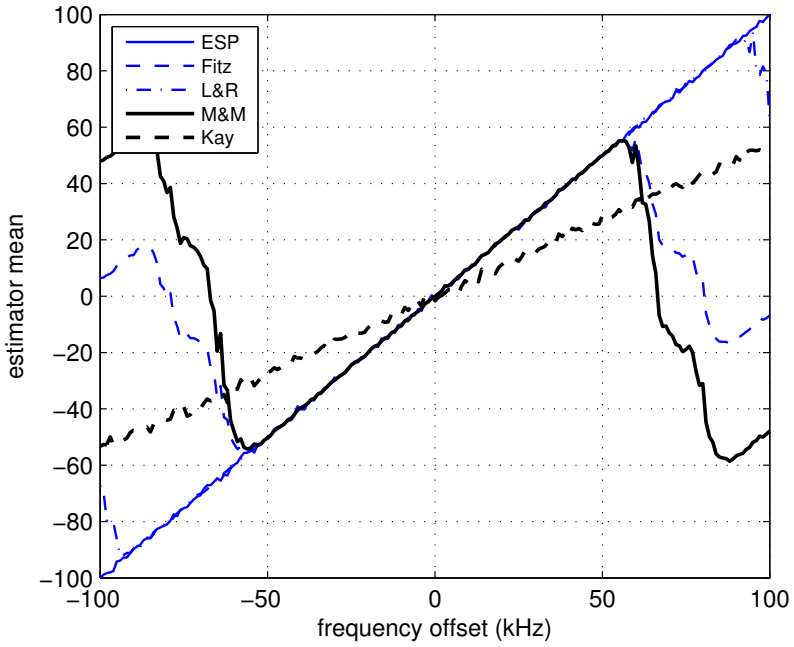

Figure 94: Simulated means of the frequency estimators as a function of frequency offset for test channel 10 with $E_b/N_0 = 10$ dB.

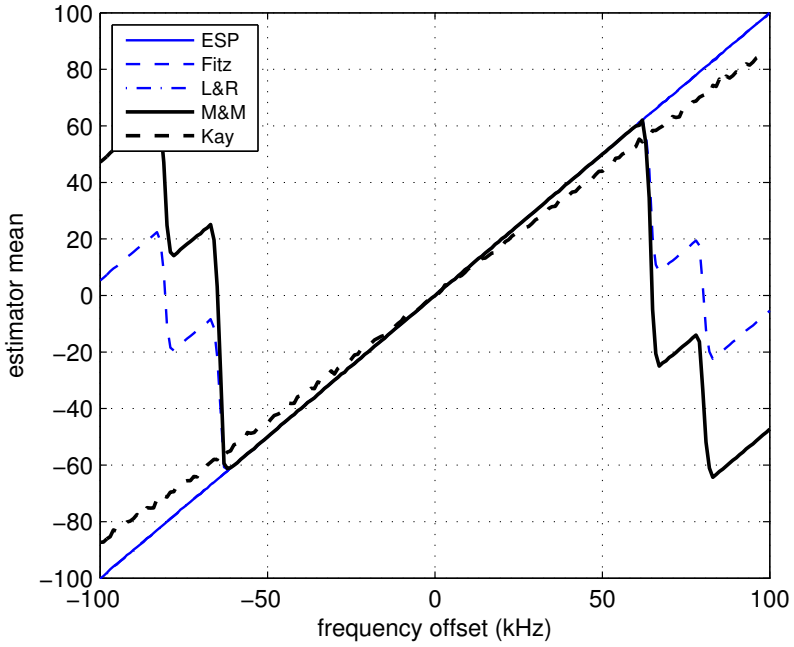

Figure 95: Simulated means of the frequency estimators as a function of frequency offset for example test channel 10 with $E_b/N_0 = 20$ dB.

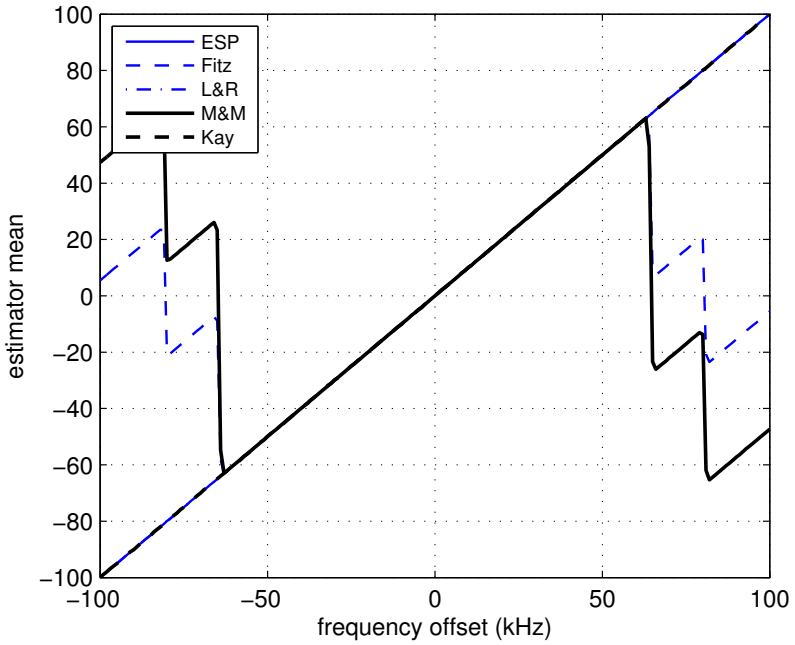

Figure 96: Simulated means of the frequency estimators as a function of frequency offset for example test channel 10 with $E_b/N_0 = 30$ dB.

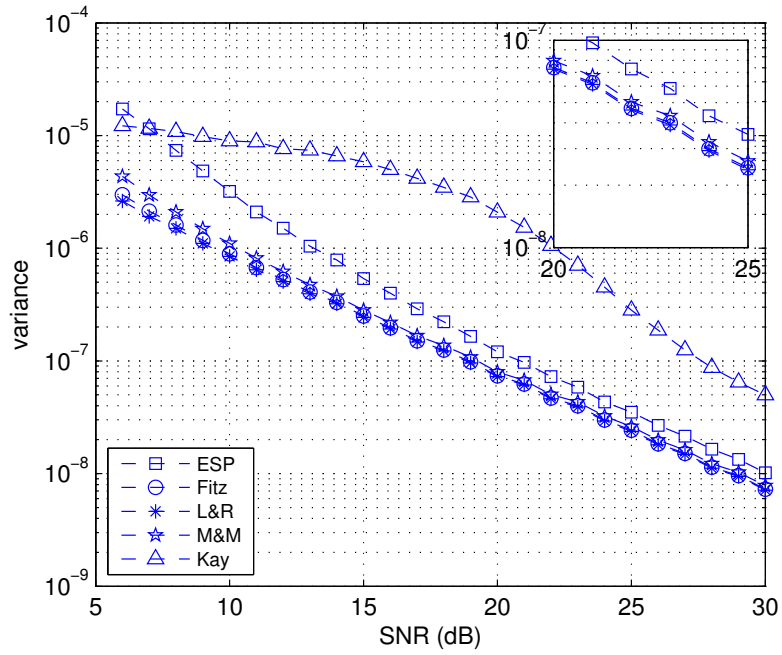

Figure 97: Estimator error variance as a function of E_b/N_0 for test channel 10.

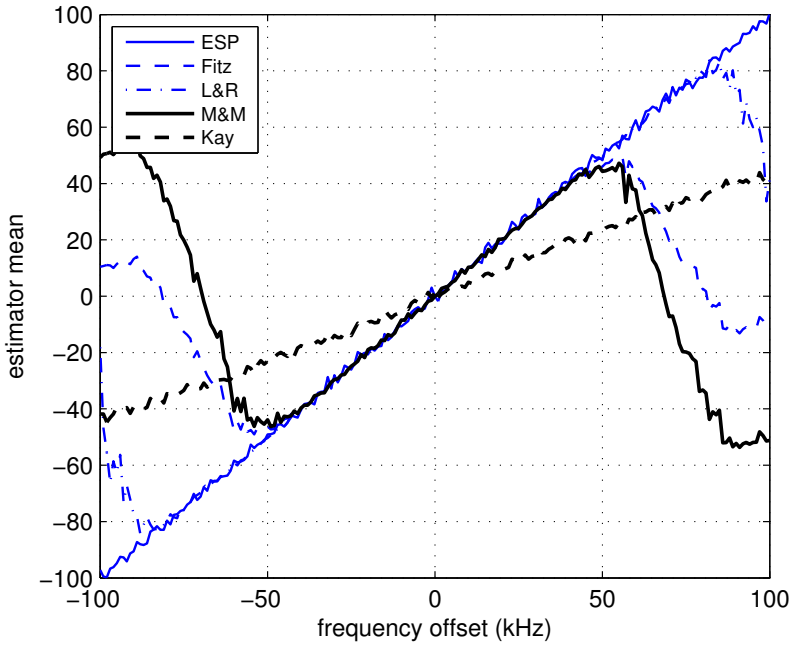

Figure 98: Simulated means of the frequency estimators as a function of frequency offset for test channel 11 with $E_b/N_0 = 10$ dB.

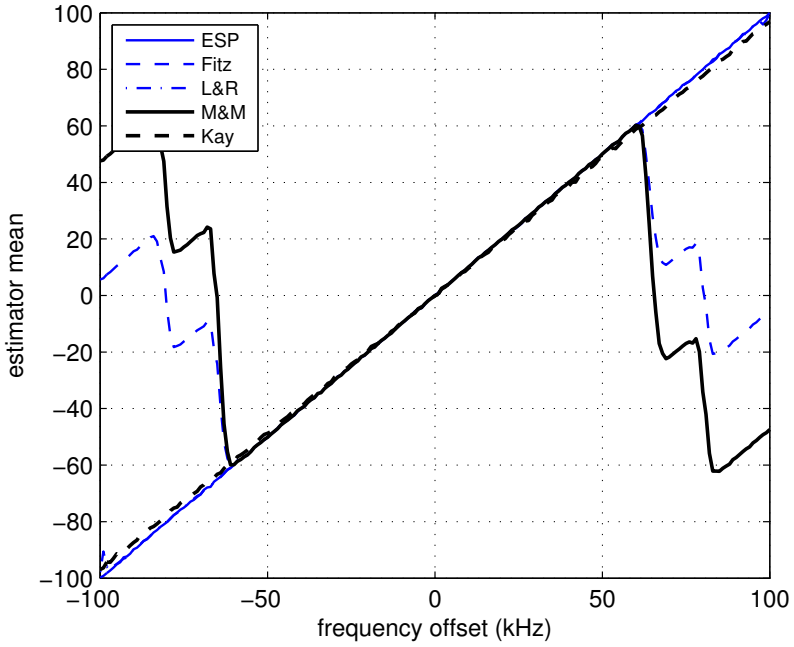

Figure 99: Simulated means of the frequency estimators as a function of frequency offset for example test channel 11 with $E_b/N_0 = 20$ dB.

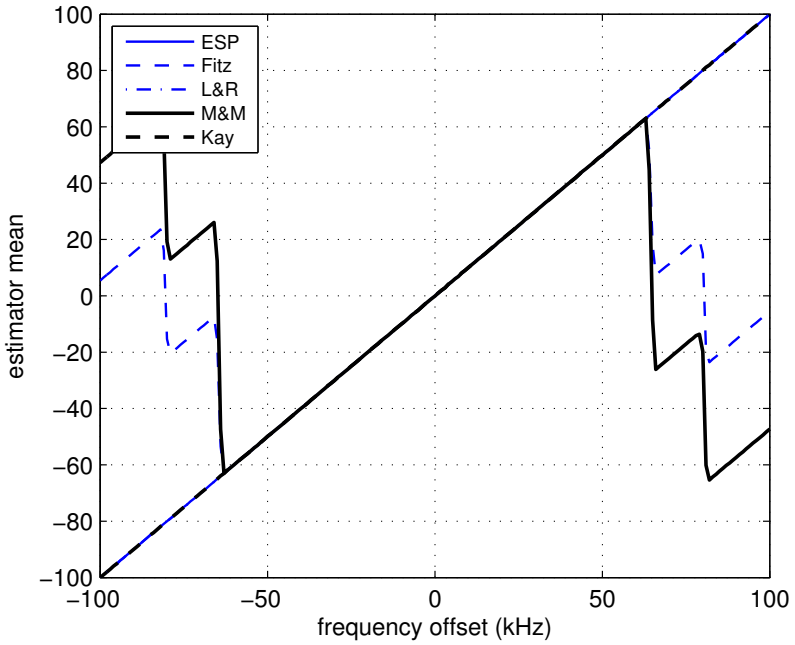

Figure 100: Simulated means of the frequency estimators as a function of frequency offset for example test channel 11 with $E_b/N_0 = 30$ dB.

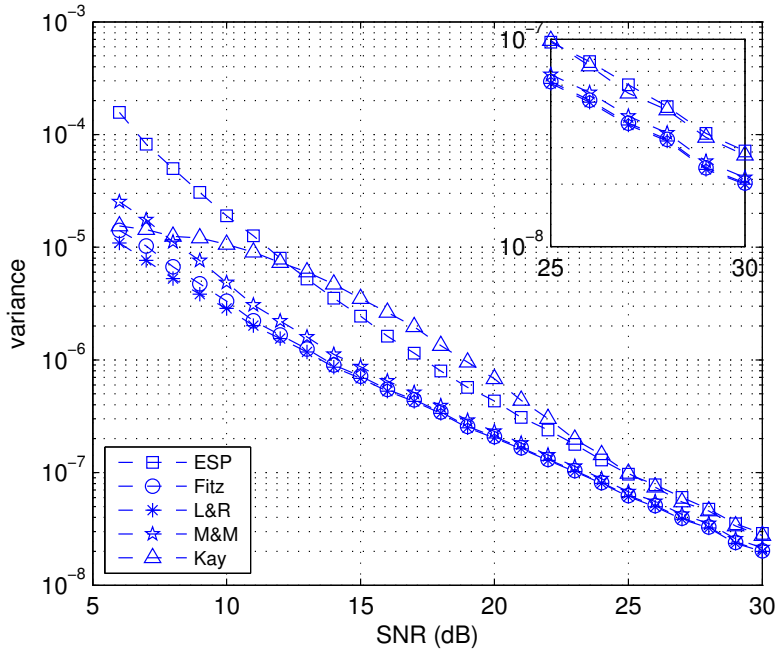

Figure 101: Estimator error variance as a function of E_b/N_0 for test channel 11.

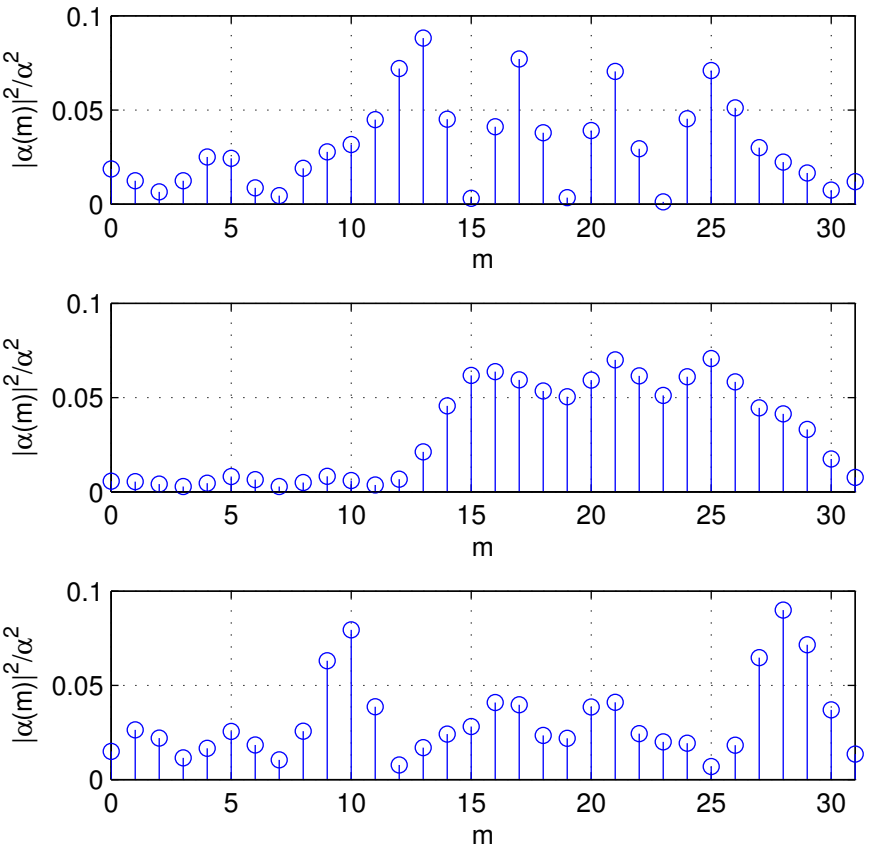

Figure 102: Plots of $|\alpha(m)|^2/\alpha^2$ [see (172)]: (a) for test channel 2; (b) for test channel 3; (c) for test channel 4.

4 Channel Estimator

4.1 Direct Application

Using the relationship (2) and neglecting for the moment the noise, we see that (assuming $N_1 < N_\mathrm{p}$)

$$\begin{aligned} r(i) &= \sum_{k=-N_1}^{N_2} h(k)s(i-k) \\ &= h(-N_1)\underbrace{s(i+N_1)}_{p(N_1)} + \cdots + h(0)\underbrace{s(i)}_{p(0)} + h(1)\underbrace{s(i-1)}_{d_1(i-1)} + \cdots + h(N_2)\underbrace{s(i-N_2)}_{d_1(i-N_2)}. \end{aligned}$$

This shows that $r(i)$ depends on N_2 unknown data samples and therefore cannot be used to estimate the channel. The first sample of $r(\cdot)$ that depends *only* on the preamble samples is (assuming $N_1 + N_2 < N_\mathrm{p}$ and neglecting noise)

$$r(i+N_2) = h(-N_1)\underbrace{s(i+N_2+N_1)}_{p(N_2+N_1)} + \cdots + h(0)\underbrace{s(i+N_2)}_{p(N_2)} + \cdots + h(N_2)\underbrace{s(i)}_{p(0)}. \tag{175}$$

Following the same line of reasoning, the last sample that depends *only* on the preamble samples is (assuming $N_1 + N_2 < N_\mathrm{p}$ and neglecting noise)

$$r(i+N_\mathrm{p}-N_1-1) = h(-N_1)\underbrace{s(i+N_\mathrm{p}-1)}_{p(N_\mathrm{p}-1)} + \cdots + h(N_2)\underbrace{s(i+N_\mathrm{p}-N_1-N_2-1)}_{p(N_\mathrm{p}-N_1-N_2-1)}. \tag{176}$$

Re-writing equations (175) and (176) in matrix vector form, we have

$$\begin{bmatrix} r(i+N_2) \\ \vdots \\ r(i+N_\mathrm{p}-N_1-1) \end{bmatrix} = \begin{bmatrix} p(N_2+N_1) & \cdots & p(0) \\ \vdots & & \vdots \\ p(N_\mathrm{p}-1) & \cdots & p(N_\mathrm{p}-N_1-N_2-1) \end{bmatrix} \begin{bmatrix} h(-N_1) \\ \vdots \\ h(N_2) \end{bmatrix} + \begin{bmatrix} w(i+N_2) \\ \vdots \\ w(i+N_\mathrm{p}-N_1-1) \end{bmatrix}. \tag{177}$$

Equation (177) may be written as

$$\mathbf{r}_1 = \mathbf{P}_\mathrm{i}\mathbf{h} + \mathbf{w}_1 \tag{178}$$

where $\mathbf{r}_1$ and $\mathbf{w}_1$ are $(N_\mathrm{p}-N_1-N_2)\times 1$ vectors formed in the obvious way, $\mathbf{h}$ is the $(N_1+N_2+1)\times 1$ vector formed by the channel coefficients, and $\mathbf{P}_\mathrm{i}$ is the $(N_\mathrm{p}-N_1-N_2)\times(N_1+N_2+1)$ convolution matrix formed by the preamble samples. The subscript "i" is used to convey "iNET Preamble." We anticipate that for practical values of N_1 and N_2, $\mathbf{P}_\mathrm{i}$ has more rows than columns; i.e., $\mathbf{P}_\mathrm{i}$ will be tall and skinny. Consequently, (178) represents an overdetermined system.

At this point, the noise samples are assumed to be uncorrelated zero-mean complex-valued Gaussian random variables. Consequently, the covariance matrix for $\mathbf{w}_1$ is[3]

$$\frac{1}{2}\mathrm{E}\left\{\mathbf{w}\mathbf{w}^\dagger\right\} = \sigma_w^2\mathbf{I}_{L_1} \tag{179}$$

so that the conditional PDF for $\mathbf{r}_1$ is

$$f(\mathbf{r}_1|\mathbf{h}) = \frac{1}{(2\pi\sigma_w^2)^L}\exp\left\{-\frac{1}{2\sigma_w^2}|\mathbf{r}_1-\mathbf{P}_\mathrm{i}\mathbf{h}|^2\right\} \tag{180}$$

where $L = N_\mathrm{p} - N_1 - N_2$.

The log-likelihood function for $\mathbf{h}$ is

$$\Lambda(\mathbf{h}) = -\frac{1}{2\sigma_w^2}|\mathbf{r}_1-\mathbf{P}_\mathrm{i}\mathbf{h}|^2. \tag{181}$$

The maximum likelihood (ML) estimate for $\mathbf{h}$ is the vector that maximizes the log-likelihood function $\Lambda(\mathbf{h})$:

$$\begin{aligned}\hat{\mathbf{h}}_1 &= \underset{\mathbf{h}}{\mathrm{argmax}}\left\{-|\mathbf{r}_1-\mathbf{P}_\mathrm{i}\mathbf{h}|^2\right\}\\ &= \underset{\mathbf{h}}{\mathrm{argmin}}\left\{|\mathbf{r}_1-\mathbf{P}_\mathrm{i}\mathbf{h}|^2\right\}\\ &= \left(\mathbf{P}_\mathrm{i}^\dagger\mathbf{P}_\mathrm{i}\right)^{-1}\mathbf{P}_\mathrm{i}^\dagger\mathbf{r}_1.\end{aligned} \tag{182}$$

The term $\left(\mathbf{P}_\mathrm{i}^\dagger\mathbf{P}_\mathrm{i}\right)^{-1}\mathbf{P}_\mathrm{i}^\dagger$ is the left pseudo-inverse of $\mathbf{P}_\mathrm{i}$.

Examples of the estimator for $N_1 = 12$ and $N_2 = 25$; channels 2, 3, and 4 from our set of representative channels; and $\sigma^2 = 0$ are illustrated in the middle plot of Figures 103 (a)–(c). Here we observe that the behavior of the estimator for channels 2 and 4 is good, but the behavior for channel 3 is not good. I believe this poor behavior is due to the fact that the convolution matrix $\mathbf{P}_\mathrm{i}$ is almost rank-deficient. For these values of N_1 and N_2, the vector $\mathbf{r}_1$ is 219×1 and $\mathbf{P}_\mathrm{i}$ is 219×38.

[3]Here, $\mathbf{v}^\dagger$ is the Hermitian (conjugate transpose) of the vector $\mathbf{v}$. Similarly, $\mathbf{M}^\dagger$ is the Hermitian of the matrix $\mathbf{M}$.

The singular values of $\mathbf{P}_{\mathrm{i}}$ reveal how close to rank-deficient $\mathbf{P}_{\mathrm{i}}$ is. A plot of the $N_1 + N_2 + 1 = 38$ singular values of $\mathbf{P}_{\mathrm{i}}$ is shown in Figure 104. Observe that well over half of the singular values are very very small. The reason for the rank deficiency of $\mathbf{P}_{\mathrm{i}}$ is the structure of the preamble bits. Because the preamble sequence is a repetition of a short bit sequence, the columns of $\mathbf{P}_{\mathrm{i}}$ become linearly dependent for modest values of N_1 and N_2. Channel $h_3(n)$ is much longer than channels $h_2(n)$ and $h_4(n)$. Evidently, it is the case that as more of the available $N_1+N_2+1 = 38$ dimensions are used, the impact of all those small singular values increases.

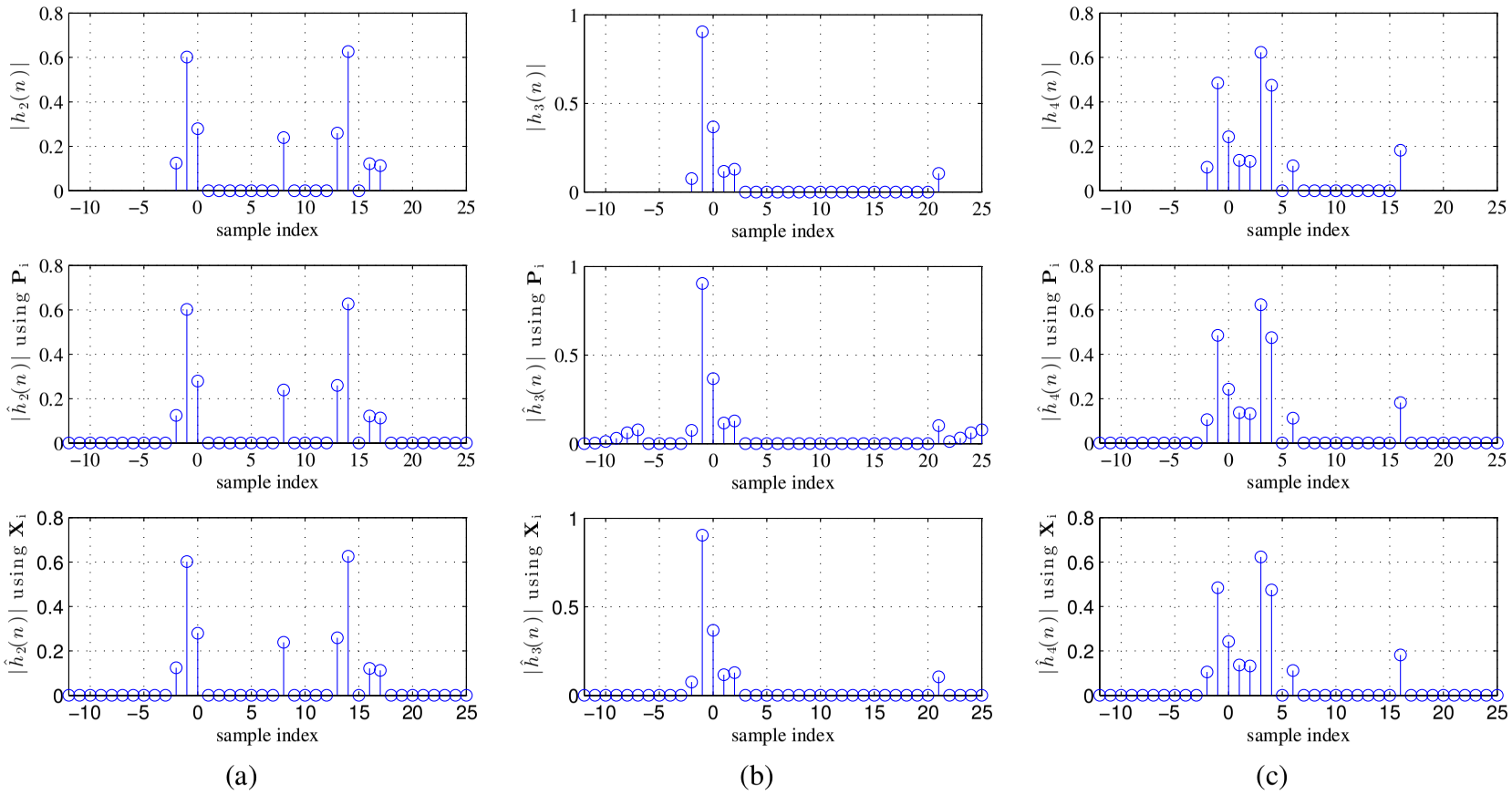

Figure 103: The channel estimates using the iNET preamble. The top plot is the magnitude of the true channel; the middle plot is the estimate (182); the lower plot is the estimate (190). (a) Example using $h_2(n)$; (b) Example using $h_3(n)$; (c) Example using $h_4(n)$.

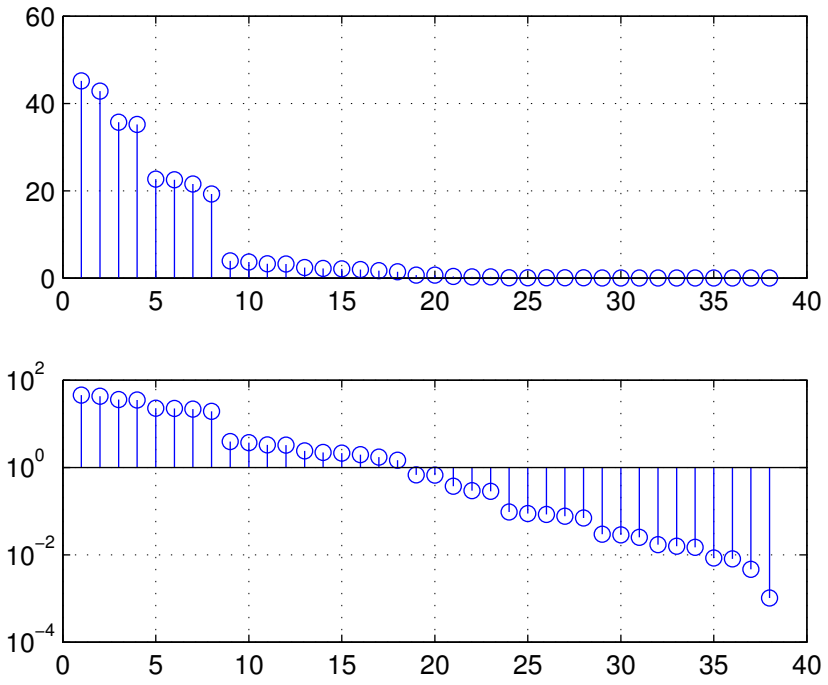

Figure 104: A plot of the singular values of the convolution matrix formed from the SOQPSK-TG samples based on the iNET preamble. Top: the plot using a linear scale; Bottom: the plot using a logarithmic scale.

4.2 Improvements to the Direct Application

One solution to this problem is to use a different preamble sequence: a sequence whose corresponding samples form a convolution matrix closer to full rank. Another possible solution is to include both the preamble and ASM samples in formulating the problem. Both are explored here.

4.2.1 A "Full Rank" Preamble Sequence

Here we form the preamble by appending a 0 to the length-127 PN sequence. The analysis proceeds in the exactly the same way as before, except now the convolution matrix is filled with samples corresponding to the new preamble. This change is documented by using $\mathbf{P}_{\mathrm{p}}$ to denote the convolution matrix filled with samples corresponding to the new preamble. Here the subscript "p" is used to denote "PN sequence." We use $\mathbf{P}_{\mathrm{p}}$ in place of $\mathbf{P}_{\mathrm{i}}$ in Equations (178), (180), (181), and (182) to give

$$\hat{\mathbf{h}}_1 = \left(\mathbf{P}_{\mathrm{p}}^{\dagger}\mathbf{P}_{\mathrm{p}}\right)^{-1}\mathbf{P}_{\mathrm{p}}^{\dagger}\mathbf{r}_1. \tag{183}$$

The results for the estimator (183) are shown in the middle plots of Figures 105 (a)–(c) for the same situation as before; namely, $N_1 = 12$ and $N_2 = 25$; channels 2, 3, and 4 from our set of representative channels; and $\sigma_w^2 = 0$. It is clear that the "full rank" preamble solves the problem estimating channel 3 observed in above. This increases my confidence that the explanation has its origins in the number of small singular values of the convolution matrix $\mathbf{P}_{\mathrm{i}}$. The singular values for $\mathbf{P}_{\mathrm{p}}$ are shown in Figure 106. Here we see that the singular values corresponding to the 0-appended length-127 PN sequence proceed from maximum to minimum much more gradually that the singular values corresponding to the iNET preamble (cf, Figure 104).

The disadvantage with this preamble is that frequency estimation is now very difficult. In fact, with this preamble, the channel and the frequency must be estimated jointly, and this is probably not possible for computational complexity reasons.

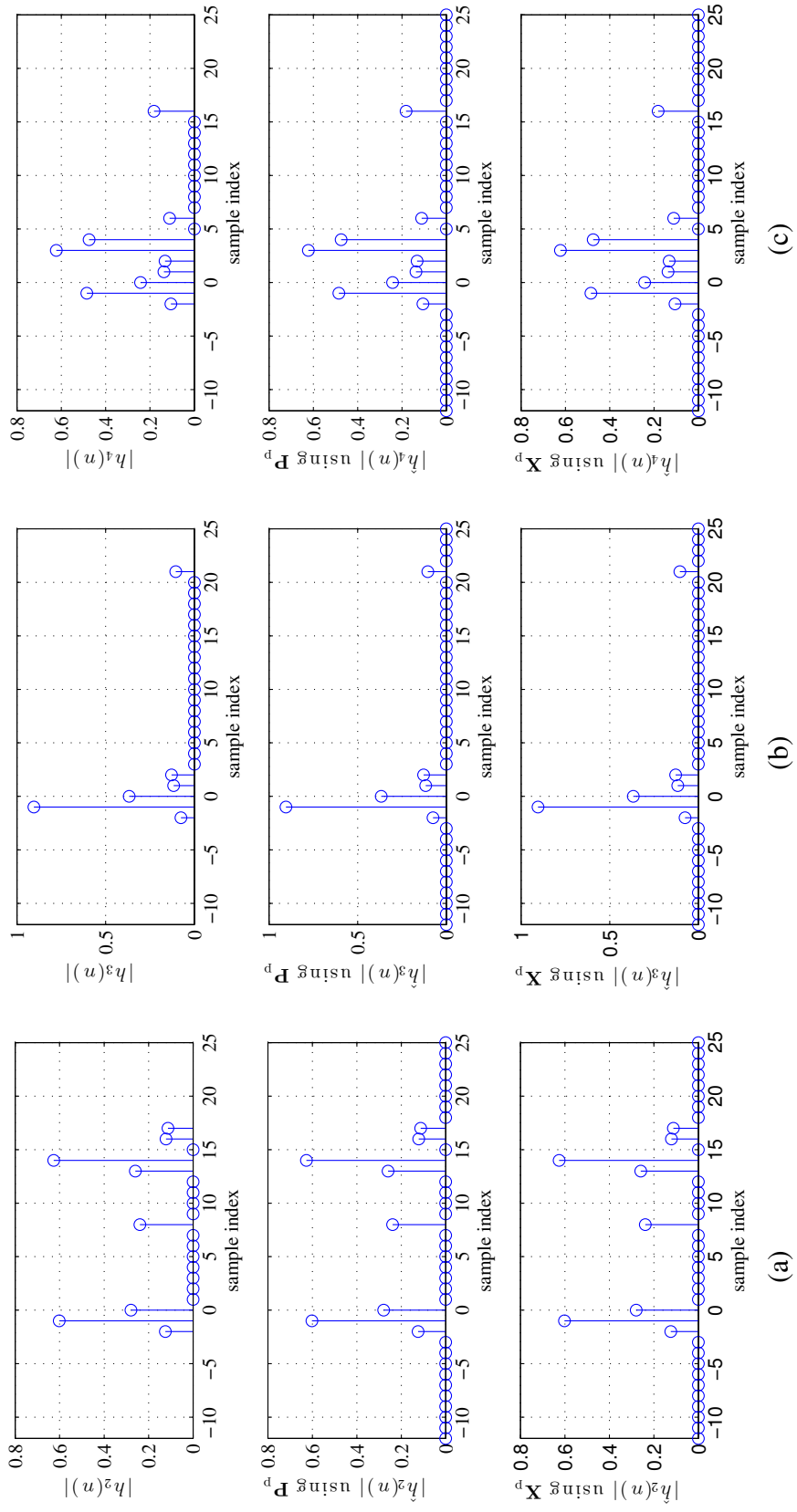

Figure 105: The channel estimates using the "random" preamble created by appending a 0 to the length-127 PN sequence (cf, Figure 103). The top plot is the magnitude of the true channel; the middle plot is the estimate (183); the lower plot is the estimate (190) using $\mathbf{X}_p$ in place of $\mathbf{X}_i$. (a) Example using $h_2(n)$; (b) Example using $h_3(n)$; (c) Example using $h_4(n)$.

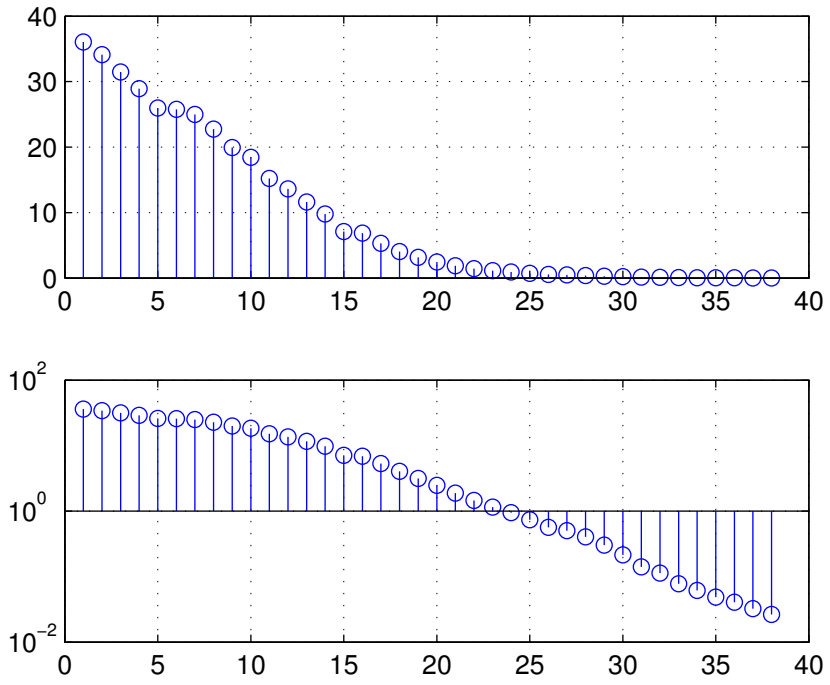

Figure 106: A plot of the singular values of the convolution matrix formed from the SOQPSK-TG samples based on the preamble formed by appending a 0 to the length-127 PN sequence (cf. Figure 104). Top: the plot using a linear scale; Bottom: the plot using a logarithmic scale.

4.2.2 Including the ASM Samples in the Channel Estimator

The appeal here is that the preamble sequence does not have to change (this retains the advantage of a low-complexity frequency offset estimator). Because the ASM bit sequence is not nearly as structured as the preamble bit sequence, there is hope that the resulting convolution matrix will be closer to full rank. For notational convenience, define the following vectors:

$$\mathbf{p} = \begin{bmatrix} p(0) \\ \vdots \\ p(N_{\text{p}} - 1) \end{bmatrix} \qquad \mathbf{a} = \begin{bmatrix} a(0) \\ \vdots \\ a(N_{\text{asm}} - 1) \end{bmatrix}.$$

Now, stack these two vectors to form the $(N_{\text{p}} + N_{\text{asm}}) \times 1$ vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{a} \end{bmatrix}. \tag{184}$$

Proceeding as before, the system of equations based on both the preamble and the ASM samples is

$$\begin{bmatrix} r(i + N_2) \\ \vdots \\ r(i + N_{\text{p}} + N_{\text{asm}} - N_1 - 1) \end{bmatrix} = \begin{bmatrix} x(N_2 + N_1) & \cdots & x(0) \\ \vdots & & \vdots \\ x(N_{\text{p}} + N_{\text{asm}} - 1) & \cdots & x(N_{\text{p}} + N_{\text{asm}} - N_1 - N_2 - 1) \end{bmatrix} \begin{bmatrix} h(-N_1) \\ \vdots \\ h(N_2) \end{bmatrix} + \begin{bmatrix} w(i + N_2) \\ \vdots \\ w(i + N_{\text{p}} + N_{\text{asm}} - N_1 - 1) \end{bmatrix}. \tag{185}$$

Equation (185) may be expressed as

$$\mathbf{r}_2 = \mathbf{X}_{\text{i}}\mathbf{h} + \mathbf{w}_2 \tag{186}$$

where $\mathbf{r}_2$ and $\mathbf{w}_2$ are $(N_{\text{p}} + N_{\text{asm}} - N_1 - N_2) \times 1$ vectors formed in the obvious way, $\mathbf{h}$ is the $(N_1 + N_2 + 1) \times 1$ vector formed by the channel coefficients, and $\mathbf{X}_{\text{i}}$ is the $(N_{\text{p}} + N_{\text{asm}} - N_1 -$

$N_2) \times (N_1 + N_2 + 1)$ convolution matrix formed by the iNET preamble and ASM samples. As before, the subscript "i" is used to denote "iNET preamble" and conveys the fact that the iNET preamble and ASM field were used to generate the samples used to form the convolution matrix. Now, if $\mathbf{P}_\mathrm{i}$ of (178) is tall and skinny, then $\mathbf{X}_\mathrm{i}$ is even more tall and skinny.

As before, we assume the noise samples in $\mathbf{w}_2$ are uncorrelated zero-mean complex-valued Gaussian random variables with covariance matrix

$$\frac{1}{2}\mathrm{E}\left\{\mathbf{w}_2\mathbf{w}_2^\dagger\right\} = \sigma_w^2\mathbf{I}_L \tag{187}$$

so that the conditional PDF of $\mathbf{r}_2$ is

$$f(\mathbf{r}_2|\mathbf{h}) = \frac{1}{(2\pi\sigma_w^2)^L}\exp\left\{-\frac{1}{2\sigma_w^2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2\right\} \tag{188}$$

where $L = N_\mathrm{p} + N_\mathrm{asm} - N_1 - N_2$. The log-likelihood function for $\mathbf{h}$ is

$$\Lambda(\mathbf{h}) = -\frac{1}{2\sigma_w^2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2 \tag{189}$$

and the ML estimate is

$$\begin{aligned}\hat{\mathbf{h}}_2 &= \underset{\mathbf{h}}{\mathrm{argmax}}\left\{-\frac{1}{2\sigma_w^2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2\right\} \\ &= \underset{\mathbf{h}}{\mathrm{argmin}}\left\{\frac{1}{2\sigma_w^2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2\right\} \\ &= \left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\mathbf{r}_2.\end{aligned} \tag{190}$$

Examples of estimator (190) for $N_1 = 12$ and $N_2 = 25$ for channels 2, 3 and 4 from our set of representative channels are shown in bottom plots of Figures 103 (a)–(c). As before, the estimator for the relatively short channels (test channels 2 and 4) work well. Unlike the previous example, the estimator for the relatively long channel (test channel 3) is quite good. I believe this is because the convolution matrix $\mathbf{X}_\mathrm{i}$ is closer to full-rank than the convolution matrix $\mathbf{P}_\mathrm{i}$. To see that this is so, a plot of the singular values of $\mathbf{X}_\mathrm{i}$ is shown in Figure 107. In comparison to the singular values of $\mathbf{P}_\mathrm{i}$, this plot shows that $\mathbf{X}_\mathrm{i}$ is better-behaved than $\mathbf{P}_\mathrm{i}$. Comparing the singular values of $\mathbf{X}_\mathrm{i}$ with those of $\mathbf{P}_\mathrm{p}$ shows that $\mathbf{P}_\mathrm{p}$ is better than $\mathbf{X}_\mathrm{i}$. Thus we have a preliminary ordering for the three

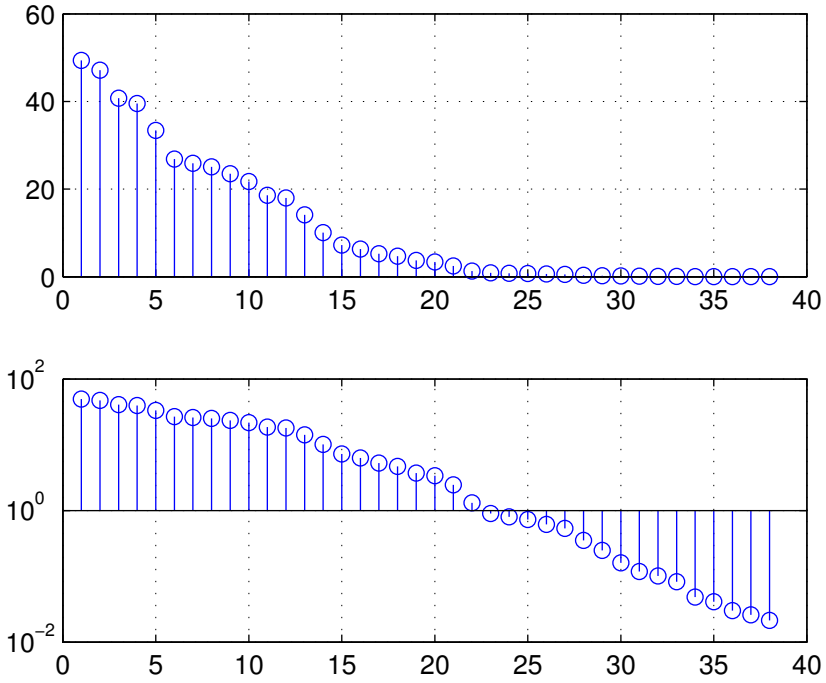

Figure 107: A plot of the singular values of the convolution matrix formed from the SOQPSK-TG samples based on the iNET preamble and the ASM bits. Top: the plot using a linear scale; Bottom: the plot using a logarithmic scale.

different approaches based on how close to singular the convolution matrices are:

$$\text{0-appended length-127 PN sequence} > \text{iNET preamble + ASM} > \text{iNET preamble} \tag{191}$$

For the purposes of completeness, the estimator (190) with[4] $\mathbf{X}_\text{p}$ in place of $\mathbf{X}_\text{i}$ is shown in the bottom plots of Figures 105 (a)–(c). Because the use 0-appended length-127 PN sequence preamble worked quite well, there does not seem to be any improvement in adding the ASM samples.

[4]The relationship of $\mathbf{X}_\text{p}$ to $\mathbf{X}_\text{i}$ is the same as that of $\mathbf{P}_\text{p}$ to $\mathbf{P}_\text{i}$.

4.3 Dealing With an Unknown Frequency Offset

We now explore the behavior of the channel estimator in the presence of a small frequency offset. Here, the received samples $r(n)$ in the previous section are replaced by $r(n)e^{j\Omega_0 n}$ where ω_0 rads/sample is the unknown frequency offset. The signal model (2) becomes[5]

$$r(n) = \left[\sum_{k=-N_1}^{N_2} h(k)s(n-k)\right] e^{j\omega_0 n} + w(n) \tag{192}$$

For the case where the channel estimation is based only on the iNET preamble samples, the relationship (178) becomes

$$\mathbf{r}_1 = \mathbf{\Omega}_0 \mathbf{P}_\mathrm{i} \mathbf{h} + \mathbf{w}_1 \tag{193}$$

where $\mathbf{\Omega}_0$ is a diagonal matrix given by

$$\mathbf{\Omega}_0 = \begin{bmatrix} e^{j\omega_0(i+N_2)} & & \\ & \ddots & \\ & & e^{j\omega_0(i+N_\mathrm{p}-N_1-1)} \end{bmatrix} \tag{194}$$

and $n = i$ is the sample of $r(n)$ corresponding to the start of the preamble sequence. Because the covariance matrix of $\mathbf{w}_1$ is still

$$\frac{1}{2}\left\{\mathbf{w}_1 \mathbf{w}_1^\dagger\right\} = \sigma_w^2 \mathbf{I}_L \tag{195}$$

for $L = N_\mathrm{p} - N_1 - N_2$, the conditional joint PDF of $\mathbf{r}_1$ is

$$f(\mathbf{r}_1|\mathbf{h}, \omega_0) = \frac{1}{(2\pi\sigma_w^2)^L} \exp\left\{-\frac{1}{2\sigma_w^2}\left|\mathbf{r}_1 - \mathbf{\Omega}_0 \mathbf{P}_\mathrm{i} \mathbf{h}\right|^2\right\} \tag{196}$$

The ML estimates for $\mathbf{h}$ and ω_0 are

$$\hat{\mathbf{h}}_1, \hat{\omega}_0 = \underset{\mathbf{h},\omega_0}{\operatorname{argmin}} \left\{\left|\mathbf{r}_1 - \mathbf{\Omega}_0 \mathbf{P}_\mathrm{i} \mathbf{h}\right|^2\right\}. \tag{197}$$

This is a joint estimator. The joint estimate may be computed in two steps. First, the ML estimate for $\mathbf{h}$ is

$$\hat{\mathbf{h}} = \left[\left(\mathbf{\Omega}_0 \mathbf{P}_\mathrm{i}\right)^\dagger \mathbf{\Omega}_0 \mathbf{P}_\mathrm{i}\right]^{-1} \left(\mathbf{\Omega}_0 \mathbf{P}_\mathrm{i}\right)^\dagger \mathbf{r}_1 = \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i}\right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \mathbf{\Omega}_0^\dagger \mathbf{r}_1. \tag{198}$$

[5]The frequency offset is not applied to the noise samples here because the joint PDF describing the vector of noise samples is rotationally invariant.

Next, this expression for $\hat{\mathbf{h}}$ is substituted for $\mathbf{h}$ in (197) to produce the following:

$$\hat{\omega}_0 = \underset{\omega}{\operatorname{argmin}} \left\{ \left| \left[\mathbf{I} - \mathbf{\Omega}_0 \mathbf{P}_\mathrm{i} \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i} \right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \mathbf{\Omega}_0^\dagger \right] \mathbf{r}_1 \right|^2 \right\} \tag{199}$$

$$\hat{\mathbf{h}}_1 = \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i} \right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \hat{\mathbf{\Omega}}_0^\dagger \mathbf{r}_1 \tag{200}$$

where

$$\hat{\mathbf{\Omega}}_0 = \begin{bmatrix} e^{j\hat{\omega}_0(i+N_2)} & & \\ & \ddots & \\ & & e^{j\hat{\omega}_0(i+N_\mathrm{p}-N_1-1)} \end{bmatrix}. \tag{201}$$

The estimator (199) seems hard. For the iNET preamble, the estimator (199) may be replaced by a much simpler algorithm as described in Section 3. The result of the first step is used to derotate the vector of received samples (this is the $\hat{\mathbf{\Omega}}_0^\dagger \mathbf{r}_1$ term) before applying the left pseudo-inverse of $\mathbf{P}_\mathrm{i}$.

The de-rotation frequency is $\hat{\omega}_0$ rads/sample. Because $\hat{\omega}_0 \neq \omega_0$, the de-rotation performed by $\hat{\mathbf{\Omega}}_0^\dagger \mathbf{r}_1$ does not completely remove the frequency offset present in the received samples. To see the impact of this residual frequency offset, we examine the conditional expected value of $\hat{\mathbf{h}}_1$. Using the substitution (193), we have

$$\begin{aligned} \hat{\mathbf{h}}_1 &= \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i} \right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \hat{\mathbf{\Omega}}_0^\dagger \mathbf{r}_1 \\ &= \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i} \right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \hat{\mathbf{\Omega}}_0^\dagger \left(\mathbf{\Omega}_0 \mathbf{P}_\mathrm{i} \mathbf{h} + \mathbf{w}_1 \right) \\ &= \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i} \right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \hat{\mathbf{\Omega}}_0^\dagger \mathbf{\Omega}_0 \mathbf{P}_\mathrm{i} \mathbf{h} + \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i} \right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \hat{\mathbf{\Omega}}_0^\dagger \mathbf{w}_1. \end{aligned} \tag{202}$$

Note that $\hat{\mathbf{\Omega}}_0^\dagger \mathbf{\Omega}_0$ is the diagonal matrix $\mathbf{\Omega}_e$ given by

$$\mathbf{\Omega}_\mathrm{e} = \begin{bmatrix} e^{j\omega_e(i+N_2)} & & \\ & \ddots & \\ & & e^{j\omega_e(i+N_\mathrm{p}-N_1-1)} \end{bmatrix} \tag{203}$$

where

$$\omega_\mathrm{e} = \omega_0 - \hat{\omega}_0. \tag{204}$$

The expected value of the channel estimate, conditioned on ω_e, is

$$\mathrm{E}\left\{ \hat{\mathbf{h}}_1 \middle| \, \omega_e \right\} = \left(\mathbf{P}_\mathrm{i}^\dagger \mathbf{P}_\mathrm{i} \right)^{-1} \mathbf{P}_\mathrm{i}^\dagger \mathbf{\Omega}_\mathrm{e} \mathbf{P}_\mathrm{i} \mathbf{h}. \tag{205}$$

When $\hat{\omega}_0 = \omega_0$, we have $\omega_\mathrm{e} = 0$ and, consequently, $\mathbf{\Omega}_\mathrm{e} = \mathbf{I}$. This gives $\mathbf{h}$ on the right-hand-side of (205) as expected. When $\omega_\mathrm{e} \neq 0$, it is hard to tell from (205) what the impact of the residual frequency offset error is. In the meantime, we can apply a small frequency offset to the running examples and compare the estimate with the estimates summarized above. But we will get to that in a moment.

First, we develop the expressions for the expected value of the estimate for the other two cases we considered.

- For the case where the 0-appended length-127 PN sequence is used for the preamble, the matrix $\mathbf{P}_\mathrm{p}$ replaces $\mathbf{P}_\mathrm{i}$ in Equations (193), (196), (197), (200), (202), and (205). The result is

$$\hat{\mathbf{h}}_1 = \left(\mathbf{P}_\mathrm{p}^\dagger \mathbf{P}_\mathrm{p}\right)^{-1} \mathbf{P}_\mathrm{p}^\dagger \hat{\mathbf{\Omega}}_0^\dagger \mathbf{r}_1 \tag{206}$$

$$\mathrm{E}\left\{ \hat{\mathbf{h}}_1 \middle| \omega_e \right\} = \left(\mathbf{P}_\mathrm{p}^\dagger \mathbf{P}_\mathrm{p}\right)^{-1} \mathbf{P}_\mathrm{p}^\dagger \mathbf{\Omega}_\mathrm{e} \mathbf{P}_\mathrm{p} \mathbf{h} \tag{207}$$

  These equations show that the only difference between the use of the two preambles lies in the properties of the corresponding convolution matrix.

- For the case where the samples corresponding to both the iNET preamble and ASM fields are used for channel estimation, the relationship (186) is replaced by

$$\mathbf{r}_2 = \mathbf{\Omega}_0 \mathbf{X}_\mathrm{i} \mathbf{h} + \mathbf{w}_2 \tag{208}$$

  where this time $\mathbf{\Omega}_0$ is a larger version of (194) and is given by

$$\mathbf{\Omega}_0 = \begin{bmatrix} e^{j\omega_0(i+N_2)} & & \\ & \ddots & \\ & & e^{j\omega_0(i+N_\mathrm{p}+N_\mathrm{asm}-N_1-1)} \end{bmatrix} \tag{209}$$

  where $n = i$ is the sample of $r(n)$ corresponding to the start of the preamble sequence. Proceeding as before, we have

$$\hat{\mathbf{h}}_2 = \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i}\right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \hat{\mathbf{\Omega}}_0^\dagger \mathbf{r}_2 \tag{210}$$

$$\mathrm{E}\left\{ \hat{\mathbf{h}}_2 \middle| \omega_e \right\} = \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i}\right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{\Omega}_\mathrm{e} \mathbf{X}_\mathrm{i} \mathbf{h} \tag{211}$$

where $\mathbf{\Omega}_{\mathrm{e}}$ is the appropriately larger version of (203) given by

$$\mathbf{\Omega}_{\mathrm{e}} = \begin{bmatrix} e^{j\omega_e(i+N_2)} & & \\ & \ddots & \\ & & e^{j\omega_e(i+N_{\mathrm{p}}+N_{\mathrm{asm}}-N_1-1)} \end{bmatrix}. \tag{212}$$

We now examine the performance of the estimators (182), (183), and (190) with $\omega_e = 2\pi \times 10^{-4}$ rads/sample for the same scenario as before: $N_1 = 12$ and $N_2 = 25$; channels 2, 3, and 4 from our set of representative channels; and $\sigma^2 = 0$. The results for the estimators (182) and (190) are shown in Figures 108 (a)–(c). The middle row of Figures 108 (a)–(c) show clearly that the estimator (182) is not adequate. The bottom row of Figures 108 (a)–(c) is better in that it more closely resembles the true channel.

The results for the estimators (183) and (190) are shown in Figures 109 (a)–(c). It is clear that both the estimators benefit from a preamble bit sequence whose samples produce of a "full rank" convolution matrix.

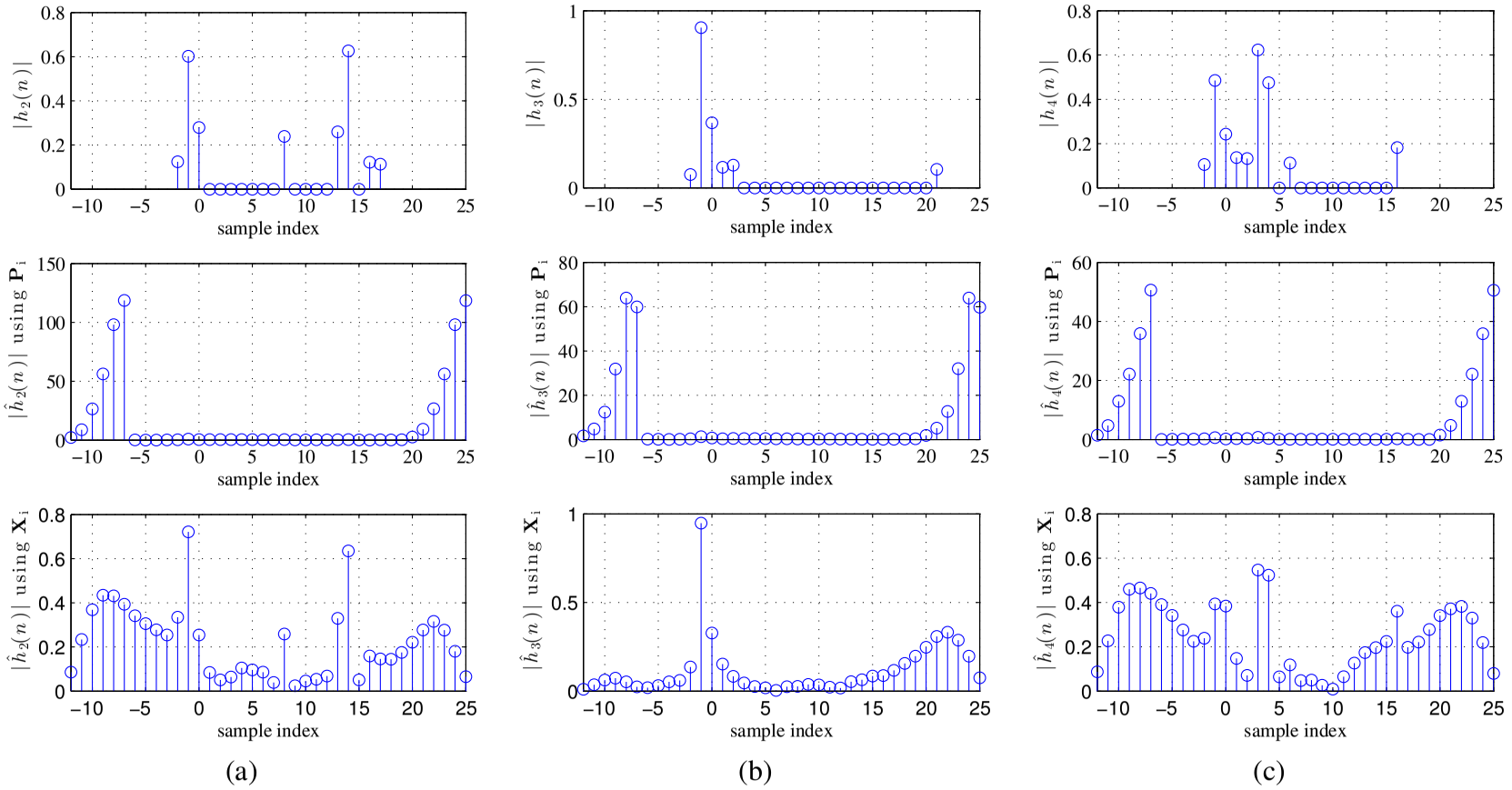

Figure 108: These are the channel estimates using the iNET preamble with a residual frequency offset of $\omega_e = 2\pi \times 10^{-4}$ rads/sample. The top plot is the magnitude of the true channel; the middle plot is the estimate (182); the lower plot is the estimate (190). (a) Example using test channel 2; (b) Example using test channel 3; (c) Example using test channel 4. Compare with Figure 103.

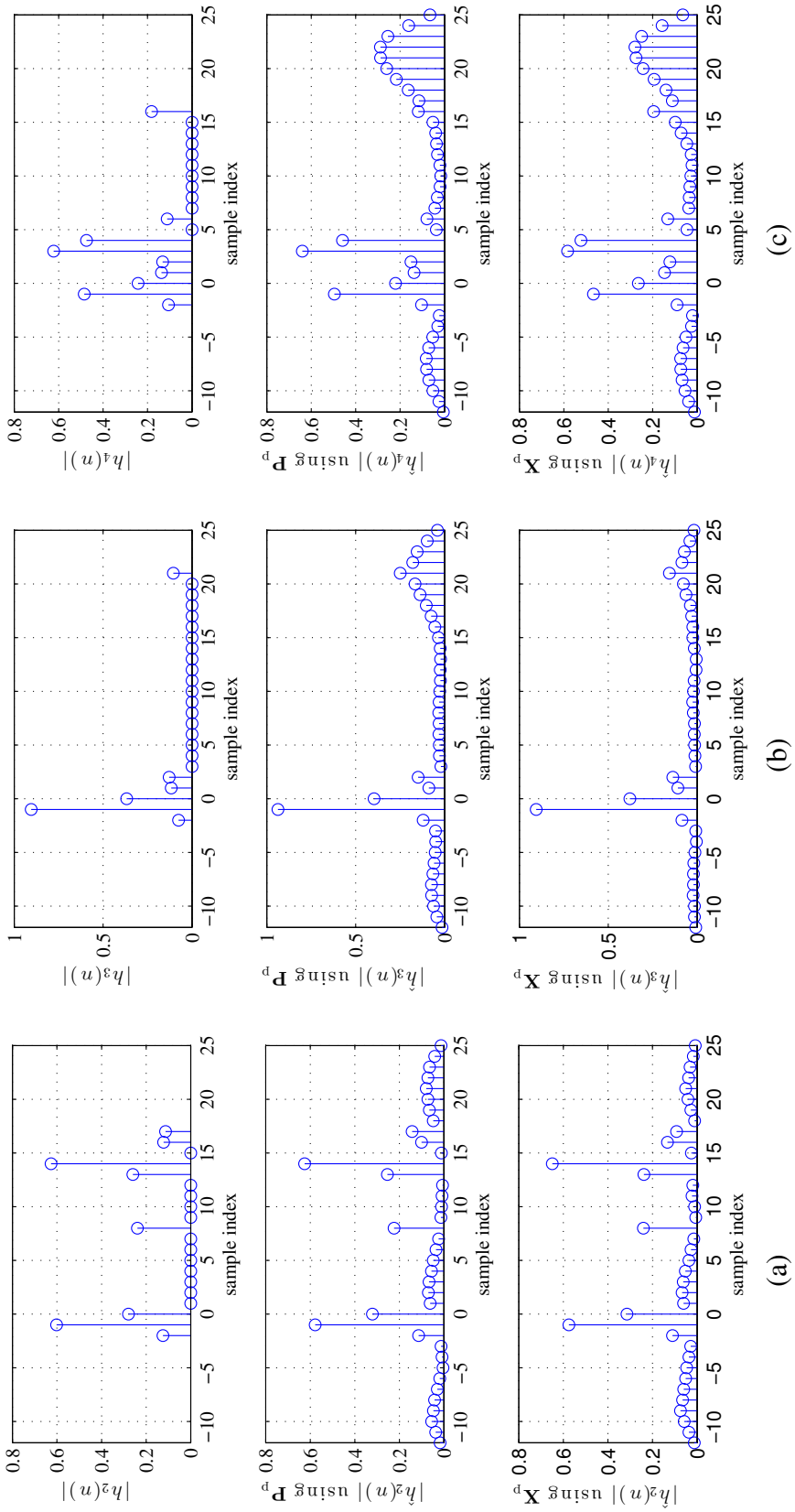

Figure 109: These are the channel estimates using the 0-appended length-127 PN sequence preamble with a residual frequency offset of $\omega_e = 2\pi \times 10^{-4}$ rads/sample. The top plot is the magnitude of the true channel; the middle plot is the estimate (183); the lower plot is the estimate (190) with $\mathbf{X}_p$ in place of $\mathbf{X}_i$. (a) Example using test channel 2; (b) Example using test channel 3; (c) Example using test channel 4. Compare with Figure 103.

5 Theoretical Performance Bounds for the Frequency Offset and Channel Estimators

Here, we compute the Cramér-Rao Bound for a joint estimator of the frequency offset and the $N_1 + N_2 + 1$ complex-valued channel coefficients. We begin with the vector of received samples given by (208) and repeated here for convenience:

$$\mathbf{r}_2 = \mathbf{\Omega}_0 \mathbf{X}_\text{i} \mathbf{h} + \mathbf{w}_2 \tag{213}$$

where $\mathbf{r}_2$ is the $(N_\text{p} + N_\text{asm} - N_1 - N_2) \times 1$ vector formed from the received samples starting at $n = i + N_2$ [$n = i$ is the sample of $r(n)$ corresponding to the start of the preamble sequence], $\mathbf{\Omega}_0$ is the $(N_\text{p} + N_\text{asm} - N_1 - N_2) \times (N_\text{p} + N_\text{asm} - N_1 - N_2)$ diagonal matrix given by

$$\mathbf{\Omega}_0 = \begin{bmatrix} e^{j\omega_0(i+N_2)} & & \\ & \ddots & \\ & & e^{j\omega_0(i+N_\text{p}+N_\text{asm}-N_1-1)} \end{bmatrix}, \tag{214}$$

$\mathbf{X}_\text{i}$ is the $(N_\text{p} + N_\text{asm} - N_1 - N_2) \times (N_1 + N_2 + 1)$ convolution matrix formed from the iNet preamble and ASM samples, $\mathbf{h}$ is the $(N_1 + N_2 + 1) \times 1$ vector representing the channel impulse response, and $\mathbf{w}_2$ is an $(N_\text{p} + N_\text{asm} - N_1 - N_2) \times 1$ complex-valued Gaussian random vector with zero mean and covariance matrix

$$\frac{1}{2}\text{E}\left\{\mathbf{w}_2 \mathbf{w}_2^\dagger\right\} = \sigma_w^2 \mathbf{I}_{N_\text{p}+N_\text{asm}-N_1-N_2}. \tag{215}$$

Because the Cramér-Rao bound is defined only for real-valued parameters, we must re-write (213) in terms of the real and imaginary components of $\mathbf{h}$ and conceptualize the estimator as one that estimates $2(N_1 + N_2 + 1)$ real-valued parameters for the channel in addition to the one real-valued parameter for the frequency offset. To facilitate the new expression, we use the notation

$$\begin{aligned} \mathbf{r}_2 &= \mathbf{r}_R + j\mathbf{r}_I \\ \mathbf{\Omega}_0 &= \mathbf{C} + j\mathbf{S} \\ \mathbf{X}_\text{i} &= \mathbf{X}_R + j\mathbf{X}_I \\ \mathbf{h} &= \mathbf{h}_R + j\mathbf{h}_I \\ \mathbf{w}_2 &= \mathbf{w}_R + j\mathbf{w}_I. \end{aligned} \tag{216}$$

Now, we may write

$$\begin{bmatrix} \mathbf{r}_R \\ \mathbf{r}_I \end{bmatrix} = \begin{bmatrix} \mathbf{C} & -\mathbf{S} \\ \mathbf{S} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X}_R & -\mathbf{X}_I \\ \mathbf{X}_I & \mathbf{X}_R \end{bmatrix} \begin{bmatrix} \mathbf{h}_R \\ \mathbf{h}_I \end{bmatrix} + \begin{bmatrix} \mathbf{w}_R \\ \mathbf{w}_I \end{bmatrix}. \tag{217}$$

For the development below, we use the shorthand notation for (217):

$$\mathbb{r} = \mathbb{Q}\mathbb{X}\mathbb{h} + \mathbb{w}. \tag{218}$$

The vector $\mathbb{w}$ is a real-valued Gaussian random vector with zero mean and covariance matrix

$$\mathrm{E}\left\{\mathbb{w}\mathbb{w}^{\top}\right\} = \sigma_w^2 \mathbf{I}_{2(N_\mathrm{p}+N_\mathrm{asm}-N_1-N_2)}. \tag{219}$$

Let $\boldsymbol{\theta}$ be the $(2(N_1+N_2+1)+1)\times 1$ vector of parameters to be estimated:

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{h}_R \\ \mathbf{h}_I \\ \omega_0 \end{bmatrix} = \begin{bmatrix} \mathbb{h} \\ \omega_0 \end{bmatrix}. \tag{220}$$

The corresponding log-likelihood function is

$$\Lambda(\boldsymbol{\theta}) = -\frac{1}{2\sigma_w^2}\left|\mathbb{r} - \mathbb{Q}\mathbb{X}\mathbb{h}\right|^2 \tag{221}$$

$$= -\frac{1}{2\sigma_w^2}\left[\mathbb{r}^{\top}\mathbb{r} - \mathbb{r}^{\top}\mathbb{Q}\mathbb{X}\mathbb{h} - \mathbb{h}^{\top}\mathbb{X}^{\top}\mathbb{Q}^{\top}\mathbb{r} + \mathbb{h}^{\top}\mathbb{X}^{\top}\mathbb{Q}^{\top}\mathbb{Q}\mathbb{X}\mathbb{h}\right] \tag{222}$$

$$= -\frac{1}{2\sigma_w^2}\left[\mathbb{r}^{\top}\mathbb{r} - \mathbb{r}^{\top}\mathbb{Q}\mathbb{X}\mathbb{h} - \mathbb{h}^{\top}\mathbb{X}^{\top}\mathbb{Q}^{\top}\mathbb{r} + \mathbb{h}^{\top}\mathbb{X}^{\top}\mathbb{X}\mathbb{h}\right] \tag{223}$$

where the step from (222) to (223) follows from the fact that

$$\mathbb{Q}^{\top}\mathbb{Q} = \begin{bmatrix} \mathbf{C} & \mathbf{S} \\ -\mathbf{S} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{C} & -\mathbf{S} \\ \mathbf{S} & \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{N_\mathrm{p}+N_\mathrm{asm}-N_1-N_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N_\mathrm{p}+N_\mathrm{asm}-N_1-N_2} \end{bmatrix}. \tag{224}$$

The Cramér-Rao requires the matrix of all second partial derivatives of $\Lambda(\boldsymbol{\theta})$. Starting with the

first partial derivatives, the vector of first partial derivatives may be expressed as

$$\frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \dfrac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}} \\ \dfrac{\partial \Lambda(\boldsymbol{\theta})}{\partial \omega_0} \end{bmatrix}. \tag{225}$$

Here $\partial \Lambda(\boldsymbol{\theta})/\partial \mathbb{h}$ is a $2(N_1 + N_2 + 1) \times 1$ vector and $\partial \Lambda(\boldsymbol{\theta})/\partial \omega_0$ is a scalar. The matrix of second partial derivatives may be expressed as

$$\begin{aligned} \mathbb{M} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(\frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^{\top} \\ &= \begin{bmatrix} \dfrac{\partial}{\partial \mathbb{h}} \\ \dfrac{\partial}{\partial \omega_0} \end{bmatrix} \begin{bmatrix} \left(\dfrac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}} \right)^{\top} & \dfrac{\partial \Lambda(\boldsymbol{\theta})}{\partial \omega_0} \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{\partial}{\partial \mathbb{h}} \left(\dfrac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}} \right)^{\top} & \dfrac{\partial}{\partial \mathbb{h}} \dfrac{\partial \Lambda(\boldsymbol{\theta})}{\partial \omega_0} \\ \dfrac{\partial}{\partial \omega_0} \left(\dfrac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}} \right)^{\top} & \dfrac{\partial^2 \Lambda(\boldsymbol{\theta})}{\partial \omega_0^2} \end{bmatrix}. \end{aligned} \tag{226}$$

Just to be clear

$$\begin{aligned} \frac{\partial}{\partial \mathbb{h}} \left(\frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}} \right)^{\top} &\text{ is a } 2(N_1 + N_2 + 1) \times 2(N_1 + N_2 + 1) \text{ matrix} \\ \frac{\partial}{\partial \mathbb{h}} \frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \omega_0} &\text{ is a } 2(N_1 + N_2 + 1) \times 1 \text{ vector} \\ \frac{\partial}{\partial \omega_0} \left(\frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}} \right)^{\top} &\text{ is a } 1 \times 2(N_1 + N_2 + 1) \text{ vector} \\ \frac{\partial^2 \Lambda(\boldsymbol{\theta})}{\partial \omega_0^2} &\text{ is a scalar.} \end{aligned}$$

The first of the two first partial derivatives in (225) is

$$\begin{aligned} \frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}} &= -\frac{1}{2\sigma^2} \left[0 - \left(\mathbb{r}^{\top} \mathbb{Q} \mathbb{X} \right)^{\top} - \mathbb{X}^{\top} \mathbb{Q}^{\top} \mathbb{r} + \left(\left(\mathbb{X}^{\top} \mathbb{X} \right)^{\top} + \mathbb{X}^{\top} \mathbb{X} \right) \mathbb{h} \right] \\ &= \frac{1}{\sigma^2} \left[\mathbb{X}^{\top} \mathbb{Q}^{\top} \mathbb{r} - \mathbb{X}^{\top} \mathbb{X} \mathbb{h} \right] \end{aligned} \tag{227}$$

The other first partial derivative in (225) is somewhat more complicated. Because $\mathbb{Q}$ is the only term that is a function of ω_0 we derive the partial derivative of $\mathbb{Q}$ with respect to ω_0:

$$\frac{\partial \mathbb{Q}}{\partial \omega_0} = \begin{bmatrix} \frac{\partial}{\partial \omega_0}\mathbf{C} & -\frac{\partial}{\partial \omega_0}\mathbf{S} \\ \frac{\partial}{\partial \omega_0}\mathbf{S} & \frac{\partial}{\partial \omega_0}\mathbf{C} \end{bmatrix} = \begin{bmatrix} -\mathbf{DS} & -\mathbf{DC} \\ \mathbf{DC} & -\mathbf{DS} \end{bmatrix} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} -\mathbf{S} & -\mathbf{C} \\ \mathbf{C} & -\mathbf{S} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{C} & -\mathbf{S} \\ \mathbf{S} & \mathbf{C} \end{bmatrix} = \mathbb{D}\mathbb{T}\mathbb{Q} \tag{228}$$

where $\mathbf{D}$ is the $(N_\mathrm{p} + N_\mathrm{asm} - N_1 - N_2) \times (N_\mathrm{p} + N_\mathrm{asm} - N_1 - N_2)$ diagonal matrix given by

$$\mathbf{D} = \begin{bmatrix} i + N_2 & & \\ & \ddots & \\ & & i + N_\mathrm{p} + N_\mathrm{asm} - N_1 - 1 \end{bmatrix}. \tag{229}$$

Applying this result to the first partial derivative of $\Lambda(\boldsymbol{\theta})$ given by (223), we have

$$\begin{aligned} \frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \omega_0} &= -\frac{1}{2\sigma_w^2}\left[0 - \mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\mathbb{h} - \mathbb{h}^\top \mathbb{X}^\top \left(\mathbb{D}\mathbb{T}\mathbb{Q}\right)^\top \mathbb{r} + 0\right] \\ &= \frac{1}{2\sigma_w^2}\left[\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\mathbb{h} + \left(\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\mathbb{h}\right)^\top\right] \\ &= \frac{1}{\sigma_w^2}\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\mathbb{h} \end{aligned} \tag{230}$$

where the last line follows because $\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\mathbb{h}$ is a scalar. The four second partial derivatives of (226) are

$$\frac{\partial}{\partial \mathbb{h}}\left(\frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}}\right)^\top = \frac{\partial}{\partial \mathbb{h}}\frac{1}{\sigma_w^2}\left[\mathbb{r}^\top \mathbb{Q}\mathbb{X} - \mathbb{h}^\top \mathbb{X}^\top \mathbb{X}\right] = -\frac{1}{\sigma_w^2}\mathbb{X}^\top \mathbb{X} \tag{231}$$

$$\frac{\partial}{\partial \mathbb{h}}\frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \omega_0} = \frac{\partial}{\partial \mathbb{h}}\frac{1}{\sigma_w^2}\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\mathbb{h} = \frac{1}{\sigma_w^2}\left(\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\right)^\top \tag{232}$$

$$\frac{\partial}{\partial \omega_0}\left(\frac{\partial \Lambda(\boldsymbol{\theta})}{\partial \mathbb{h}}\right)^\top = \frac{\partial}{\partial \omega_0}\frac{1}{\sigma_w^2}\left[\mathbb{r}^\top \mathbb{Q}\mathbb{X} - \mathbb{h}^\top \mathbb{X}^\top \mathbb{X}\right] = \frac{1}{\sigma_w^2}\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X} \tag{233}$$

$$\frac{\partial^2 \Lambda(\boldsymbol{\theta})}{\partial \omega_0^2} = \frac{\partial}{\partial \omega_0}\frac{1}{\sigma_w^2}\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\mathbb{h} = \frac{1}{\sigma_w^2}\mathbb{r}^\top \mathbb{D}\mathbb{T}(\mathbb{D}\mathbb{T}\mathbb{Q})\mathbb{X}\mathbb{h}. \tag{234}$$

The last relationship (234) may be simplified by noting that

$$\begin{aligned}
\mathbb{D}\mathbb{T}\mathbb{D}\mathbb{T} &= \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{D} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D} \end{bmatrix} \\
&= \begin{bmatrix} -\mathbf{D}^2 & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}^2 \end{bmatrix} \\
&= -\mathbb{D}^2. \qquad (235)
\end{aligned}$$

Now, the matrix of second partial derivatives may be expressed as

$$\mathbb{M} = \frac{1}{\sigma_w^2} \begin{bmatrix} -\mathbb{X}^\top \mathbb{X} & \left(\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\right)^\top \\ \mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X} & -\mathbb{r}^\top \mathbb{D}^2\mathbb{Q}\mathbb{X}\mathbb{h} \end{bmatrix}. \qquad (236)$$

The Fisher information matrix $\mathbf{J}$ is

$$\mathbf{J} = -\mathrm{E}\left\{\mathbb{M}\right\} \qquad (237)$$

Using the substitution (218) for $\mathbb{r}$ and noting that

$$\mathrm{E}\left\{\mathbb{r}\right\} = \mathrm{E}\left\{\mathbb{Q}\mathbb{X}\mathbb{h} + \mathbb{w}\right\} = \mathbb{Q}\mathbb{X}\mathbb{h} + \cancelto{0}{\mathrm{E}\left\{\mathbb{w}\right\}} = \mathbb{Q}\mathbb{X}\mathbb{h}, \qquad (238)$$

we compute the expectations for each term in $\mathbb{M}$ as follows: we have

$$-\mathrm{E}\left\{-\mathbb{X}^\top \mathbb{X}\right\} = \mathbb{X}^\top \mathbb{X} \qquad (239)$$

$$-\mathrm{E}\left\{\left(\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\right)^\top\right\} = -\left(\mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\right)^\top \mathrm{E}\left\{\mathbb{r}\right\} = -\left(\mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\right)^\top \mathbb{Q}\mathbb{X}\mathbb{h} = \mathbb{X}^\top \mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h} \qquad (240)$$

$$-\mathrm{E}\left\{\mathbb{r}^\top \mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\right\} = \left(\mathbb{X}^\top \mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}\right)^\top \qquad (241)$$

$$-\mathrm{E}\left\{-\mathbb{r}^\top \mathbb{D}^2\mathbb{Q}\mathbb{X}\mathbb{h}\right\} = \mathrm{E}\left\{\mathbb{r}^\top\right\} \mathbb{D}^2\mathbb{Q}\mathbb{X}\mathbb{h} = \left(\mathbb{Q}\mathbb{X}\mathbb{h}\right)^\top \mathbb{D}^2\mathbb{Q}\mathbb{X}\mathbb{h} = \mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{D}\mathbb{X}\mathbb{h}. \qquad (242)$$

The last step in (240) requires some explanation. Starting with step immediately after the expecta-

tion we have

$$\begin{aligned}
\left(\mathbb{D}\mathbb{T}\mathbb{Q}\mathbb{X}\right)^{\top}\mathbb{Q}\mathbb{X}\mathbb{h} &= \mathbb{X}^{\top}\mathbb{Q}^{\top}\mathbb{T}^{\top}\mathbb{D}^{\top}\mathbb{Q}\mathbb{X}\mathbb{h} && \text{(transpose of matrix product)} && (243)\\
&= -\mathbb{X}^{\top}\mathbb{Q}^{\top}\mathbb{T}\mathbb{D}\mathbb{Q}\mathbb{X}\mathbb{h} && (\mathbb{T}^{\top} = -\mathbb{T}, \mathbb{D}^{\top} = \mathbb{D}) && (244)\\
&= -\mathbb{X}^{\top}\mathbb{Q}^{\top}\mathbb{T}\mathbb{Q}\mathbb{D}\mathbb{X}\mathbb{h} && (\mathbb{D} \text{ and } \mathbb{Q} \text{ are both diagonal}) && (245)\\
&= -\mathbb{X}^{\top}\mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h} && (\mathbb{Q}^{\top}\mathbb{T}\mathbb{Q} = \mathbb{T}). && (246)
\end{aligned}$$

The last step in (242) follows the properties of the term $\mathbb{Q}^{\top}\mathbb{D}^{\top}\mathbb{D}\mathbb{Q}$. First, we have

$$\begin{aligned}
\mathbb{Q}^{\top}\mathbb{D}^{\top} &= \begin{bmatrix}\mathbf{C} & \mathbf{S}\\ -\mathbf{S} & \mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\end{bmatrix} && (247)\\
&= \begin{bmatrix}\mathbf{CD} & \mathbf{SD}\\ -\mathbf{SD} & \mathbf{CD}\end{bmatrix} && (248)\\
&= \begin{bmatrix}\mathbf{DC} & \mathbf{DS}\\ -\mathbf{DS} & \mathbf{DC}\end{bmatrix} && (249)\\
&= \begin{bmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{C} & \mathbf{S}\\ -\mathbf{S} & \mathbf{C}\end{bmatrix} && (250)\\
&= \mathbb{D}^{\top}\mathbb{Q}^{\top} && (251)
\end{aligned}$$

where (249) follows from (248) because $\mathbf{C}$, $\mathbf{S}$, and $\mathbf{D}$ are diagonal. Similarly, $\mathbb{D}\mathbb{Q} = \mathbb{Q}\mathbb{D}$ because both matrices are diagonal. Now we have

$$\mathbb{Q}^{\top}\mathbb{D}^{\top}\mathbb{D}\mathbb{Q} = \mathbb{D}^{\top}\mathbb{Q}^{\top}\mathbb{Q}\mathbb{D} = \mathbb{D}^{\top}\mathbb{D} \tag{252}$$

where the last equality follows from (224). The Fisher information matrix is thus

$$\mathbf{J} = \frac{1}{\sigma_w^2}\begin{bmatrix}\mathbb{X}^{\top}\mathbb{X} & \mathbb{X}^{\top}\mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}\\ \left(\mathbb{X}^{\top}\mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}\right)^{\top} & \mathbb{h}^{\top}\mathbb{X}^{\top}\mathbb{D}^{\top}\mathbb{D}\mathbb{X}\mathbb{h}\end{bmatrix} \tag{253}$$

The estimator error variances are given by the diagonal entries of $\mathbf{J}^{-1}$. For our purposes, we are interested in the composite channel estimate error variance:

$$\sigma_{\mathrm{h}}^2 = \sum_{n=-N_1}^{N_2} \mathrm{E}\left\{\left|h(n) - \hat{h}(n)\right|^2\right\} \tag{254}$$

$$= \sum_{n=-N_1}^{N_2} \mathrm{E}\left\{ \left[h_R(n) - \hat{h}_R(n)\right]^2 + \left[h_I(n) - \hat{h}_I(n)\right]^2 \right\} \tag{255}$$

where $\hat{h}(n)$ is the n-th element of $\hat{\mathbf{h}}$. Because of the block structure of $\mathbf{J}$, $\mathbf{J}^{-1}$ also has a block structure. Using the matrix inversion lemma, we have

$$\sigma_{\mathrm{h}}^2 \geq \sigma_w^2 \times \mathrm{Tr}\left(\mathbb{X}^\top \mathbb{X} - \frac{\mathbb{X}^\top \mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h} \left(\mathbb{X}^\top \mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}\right)^\top}{\mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{D}\mathbb{X}\mathbb{h}} \right)^{-1} \tag{256}$$

$$= \sigma_w^2 \times \mathrm{Tr}\left(\mathbb{X}^\top \left[\mathbf{I} - \frac{\mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}\mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{T}^\top}{\mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{D}\mathbb{X}\mathbb{h}} \right] \mathbb{X} \right)^{-1} \tag{257}$$

$$= \sigma_w^2 \times \mathrm{Tr}\left(\mathbb{X}^\top \mathbb{T} \left[\mathbf{I} - \frac{\mathbb{D}\mathbb{X}\mathbb{h}\mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top}{\mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{D}\mathbb{X}\mathbb{h}} \right] \mathbb{T}^\top \mathbb{X} \right)^{-1} \tag{258}$$

where the last line follows from the fact that $\mathbb{T}\mathbb{T}^\top = \mathbf{I}$. The frequency estimator error variance

$$\sigma_{\omega_0}^2 = \mathrm{E}\left\{ (\omega_0 - \hat{\omega}_0)^2 \right\}, \tag{259}$$

is lower bounded by the bottom-right element of $\mathbf{J}^{-1}$ which, using the matrix inversion lemma, may be expressed as

$$\sigma_{\omega_0}^2 \geq \frac{\sigma_w^2}{\mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{D}\mathbb{X}\mathbb{h} - \left(\mathbb{X}^\top \mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}\right)^\top \left(\mathbb{X}^\top \mathbb{X}\right)^{-1} \mathbb{X}^\top \mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}} \tag{260}$$

$$= \frac{\sigma_w^2}{\mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{D}\mathbb{X}\mathbb{h} - \mathbb{h}^\top \mathbb{X}^\top \mathbb{D}^\top \mathbb{T}^\top \mathbb{X} \left(\mathbb{X}^\top \mathbb{X}\right)^{-1} \mathbb{X}^\top \mathbb{T}\mathbb{D}\mathbb{X}\mathbb{h}} \tag{261}$$

$$= \frac{\sigma_w^2}{\left(\mathbb{D}\mathbb{X}\mathbb{h}\right)^\top \left(\mathbf{I} - \mathbb{T}^\top \mathbb{X} \left(\mathbb{X}^\top \mathbb{X}\right)^{-1} \mathbb{X}^\top \mathbb{T} \right) \mathbb{D}\mathbb{X}\mathbb{h}}. \tag{262}$$

6 Simulation Results for the Frequency Offset and Channel Estimators

Here, the performance of the Perrins frequency offset estimator (55) and the "iNET Preamble + ASM" channel estimator (190) is assessed using computer simulations for the eleven test channels. The mean squared error performance of the tandem estimators is compared to the Cramér-Rao bounds developed in the previous section. Whereas the mean-squared error performance of the Perrins frequency estimator was presented in Section 3, the impact of frequency estimator errors on channel estimation has not been assessed. The gap is closed in this section.

The simulations follow the processing outlined in the lower portion of Figure 3. Using the known start of the preamble, the frequency of each received packet is estimated using (55). The data are then de-rotated by the frequency estimate. Next, the de-rotated data corresponding to the preamble and ASM fields are used to estimate the channel using the estimator (190) with $N_1 = 12$ and $N_2 = 25$.

The mean-squared error for the frequency offset is estimated using

$$\mathrm{E}\left\{\left(\omega_0 - \hat{\omega}_0\right)^2\right\} \approx \frac{1}{N}\sum_{i=0}^{N-1}\left(\omega_0 - \hat{\omega}_0(i)\right)^2 \tag{263}$$

where N is the number of iterations and $\hat{\omega}_0(i)$ is the frequency offset estimate for the i-th iteration. The mean squared error for the frequency offset versus E_b/N_0 is compared to the Cramér-Rao bound (262) for each test channel in the plots below.

The mean-squared error for the channel estimate is estimated using

$$\sum_{n=-N_1}^{N_2}\mathrm{E}\left\{\left|h(n) - \hat{h}(n)\right|^2\right\} \approx \frac{1}{N}\sum_{i=0}^{N-1}\sum_{n=-N_1}^{N_2}\left|h(n) - \hat{h}(n;i)\right|^2 \tag{264}$$

where $\hat{h}(n;i)$ is the estimate of the channel impulse response for the i-th iteration. The mean squared error for the channel versus E_b/N_0 is compared to the Cramér-Rao bound (258) for each test channel in the plots below.

The simulation results for the eleven test channels are plotted in Figures 110 – 120. In each case, the channel estimator works as best it can, in the sense that the mean-squared error performance meets its lower bound (258). The impact the channel estimation errors on equalized bit error rate performance is assessed in the next section.

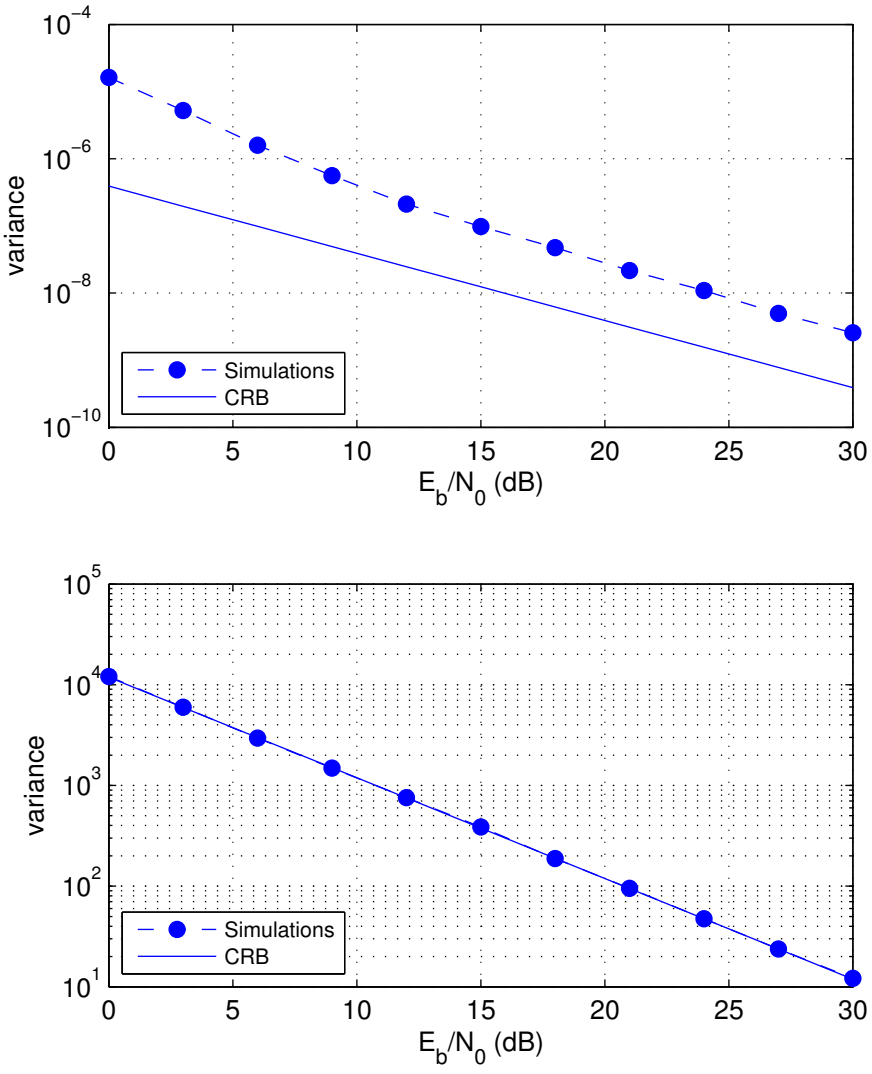

Figure 110: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 1.

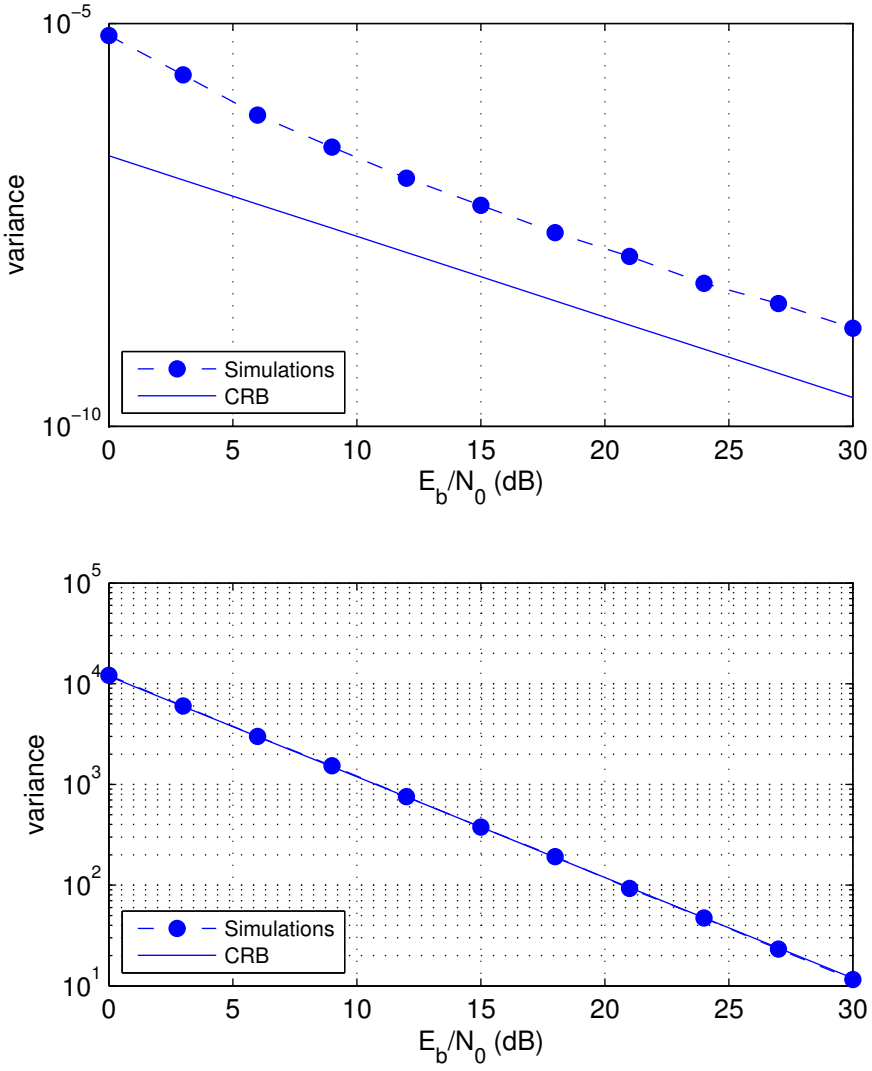

Figure 111: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 2.

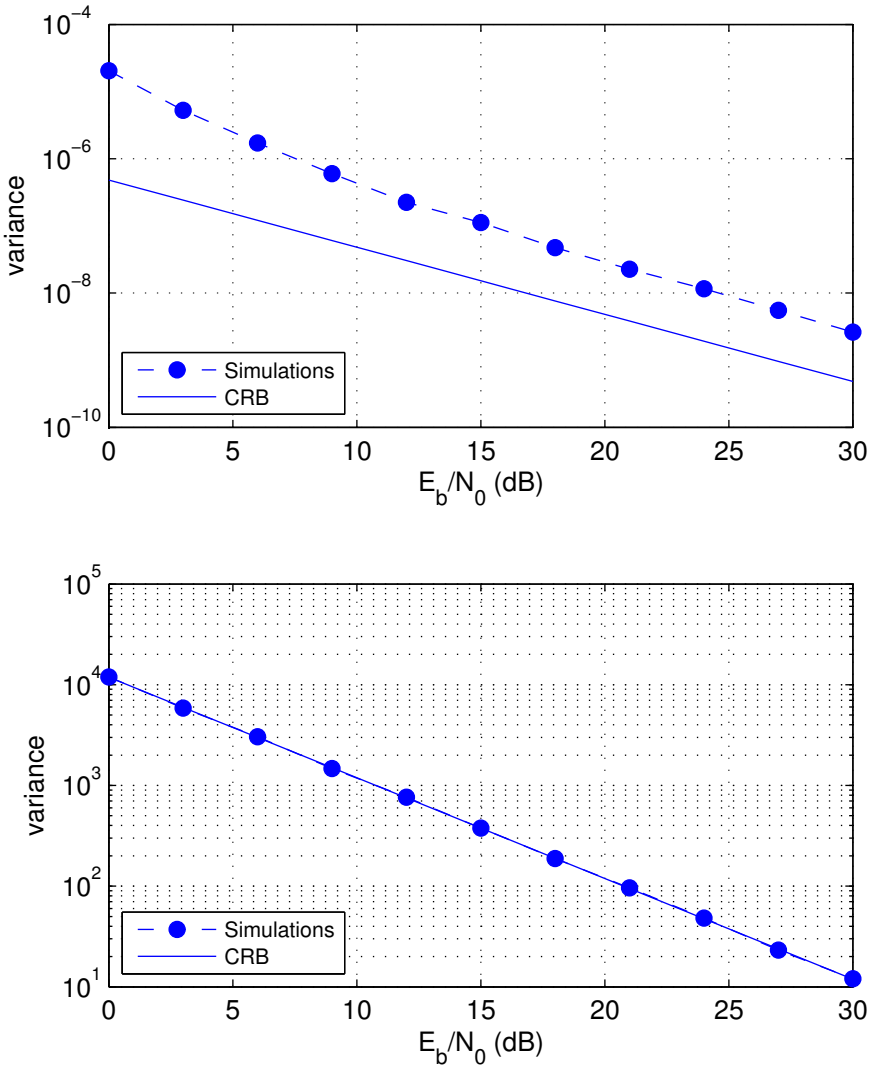

Figure 112: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 3.

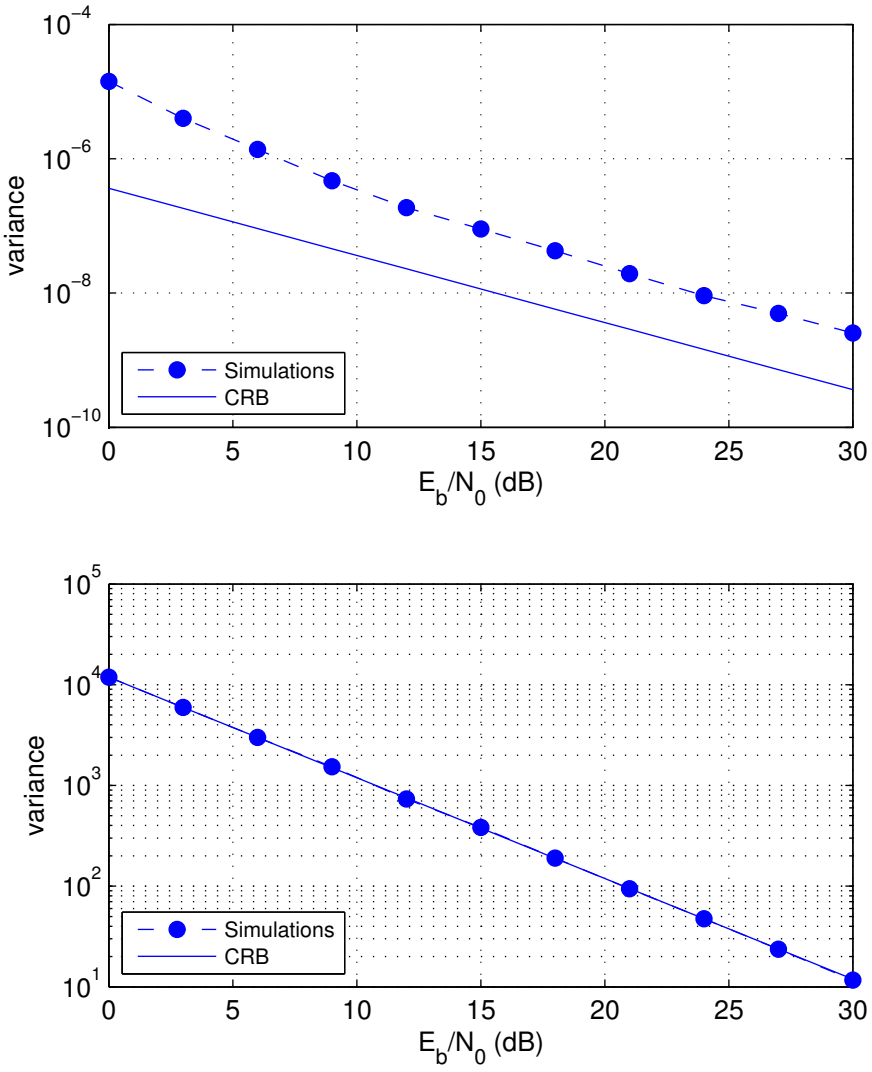

Figure 113: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 4.

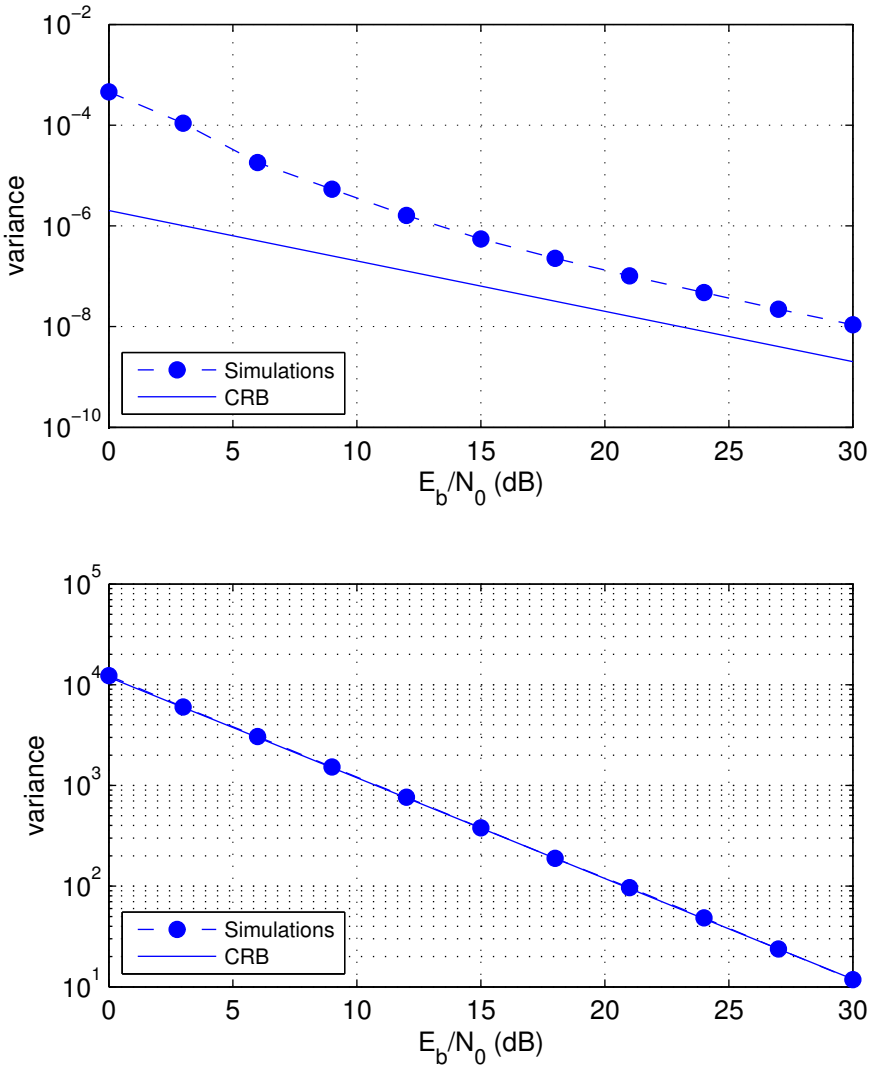

Figure 114: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 5.

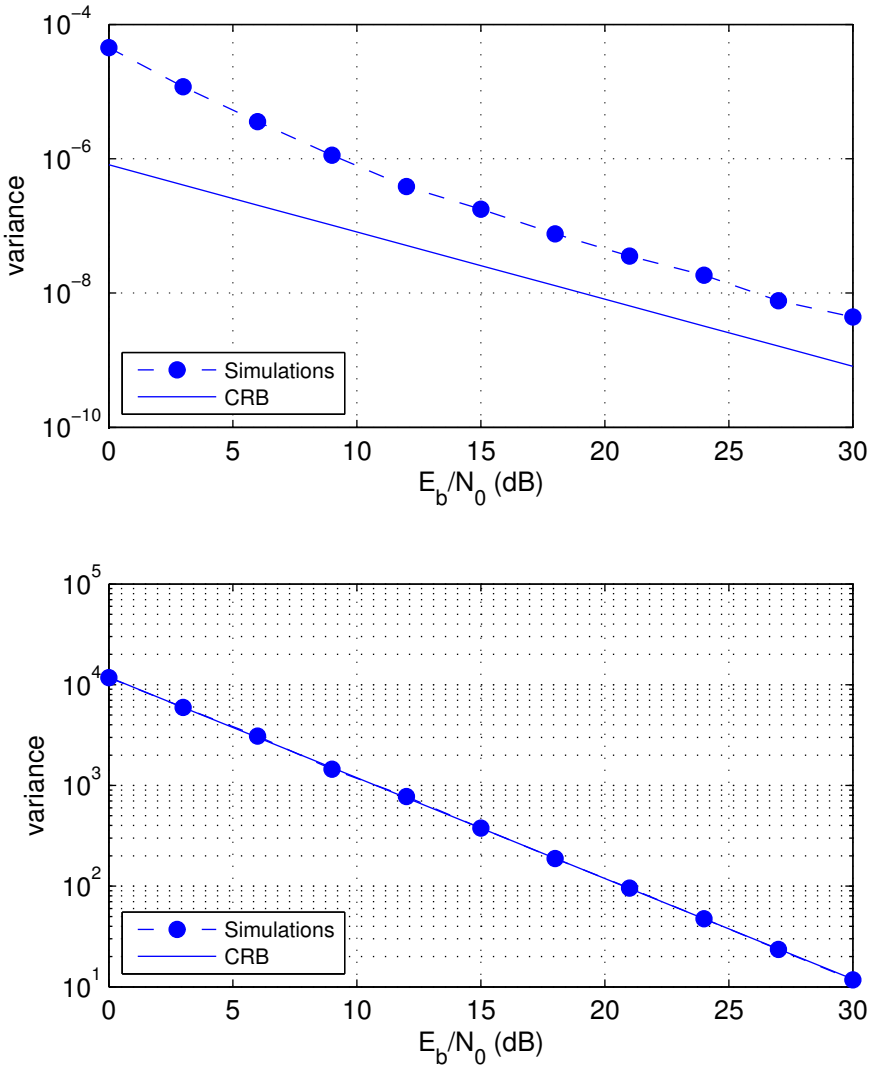

Figure 115: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 6.

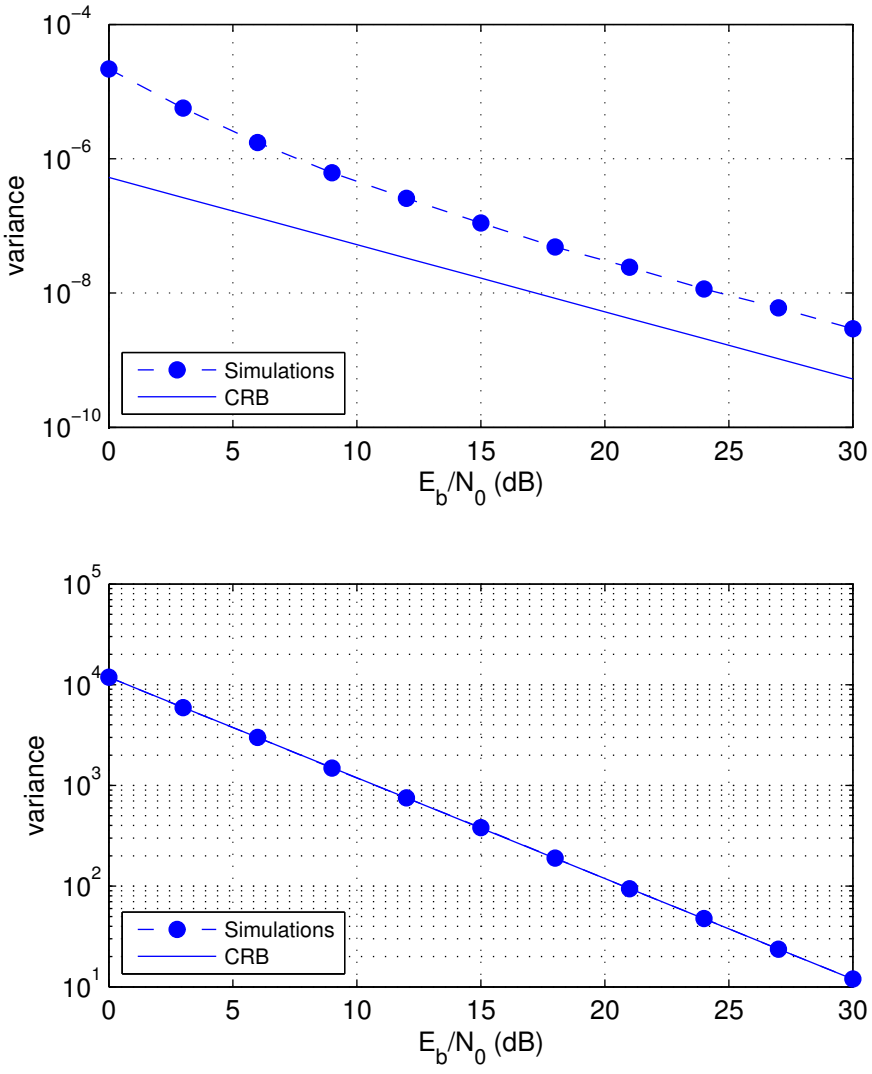

Figure 116: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 7.

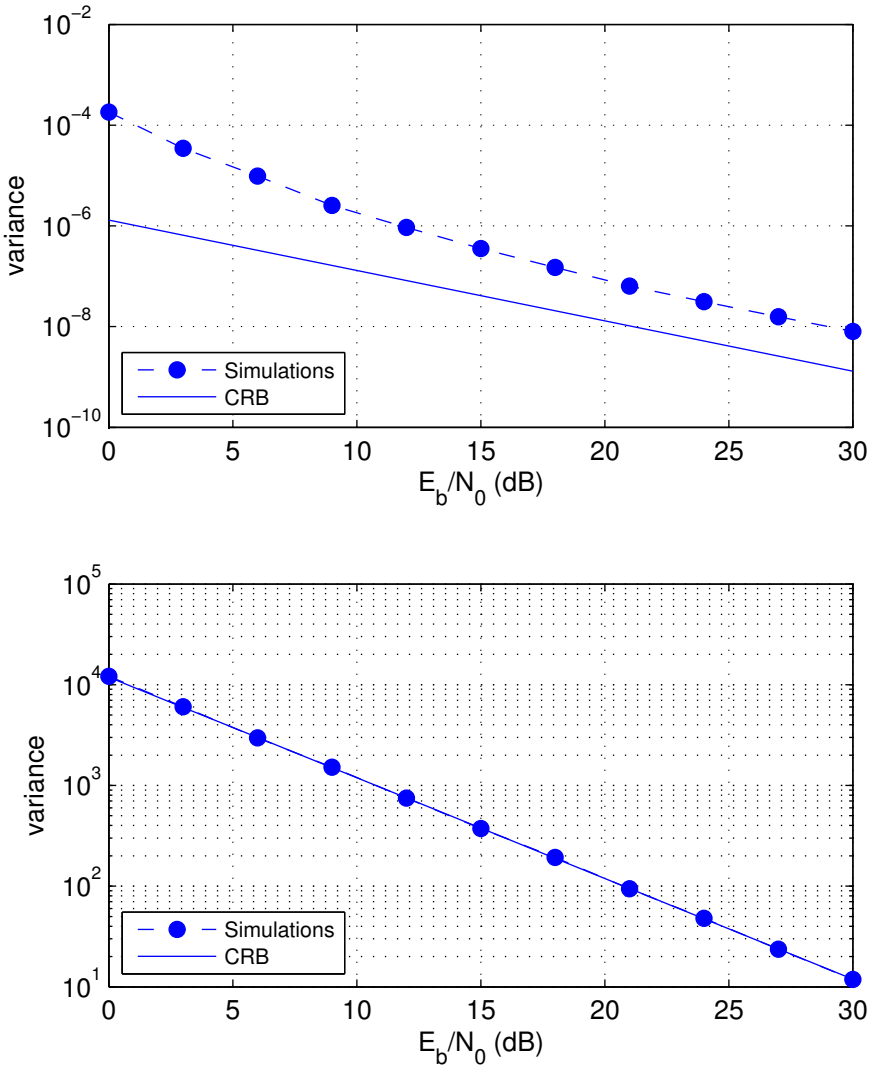

Figure 117: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 8.

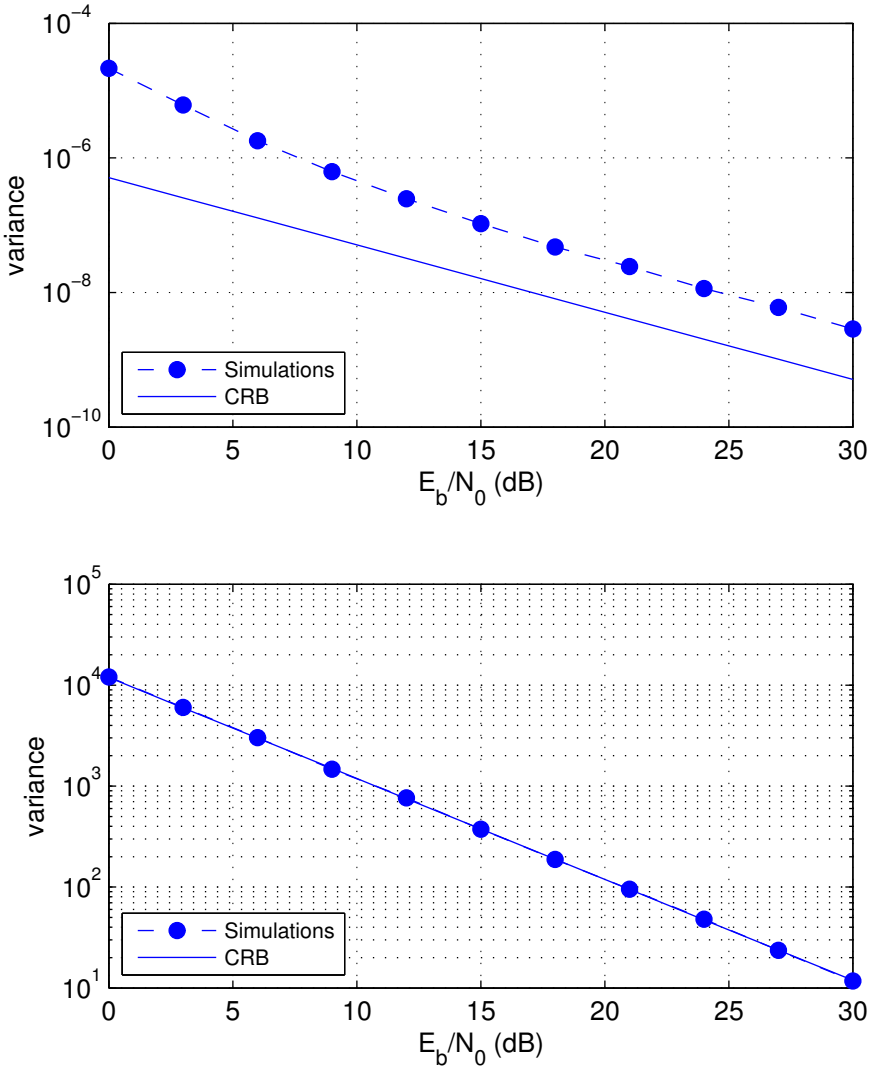

Figure 118: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 9.

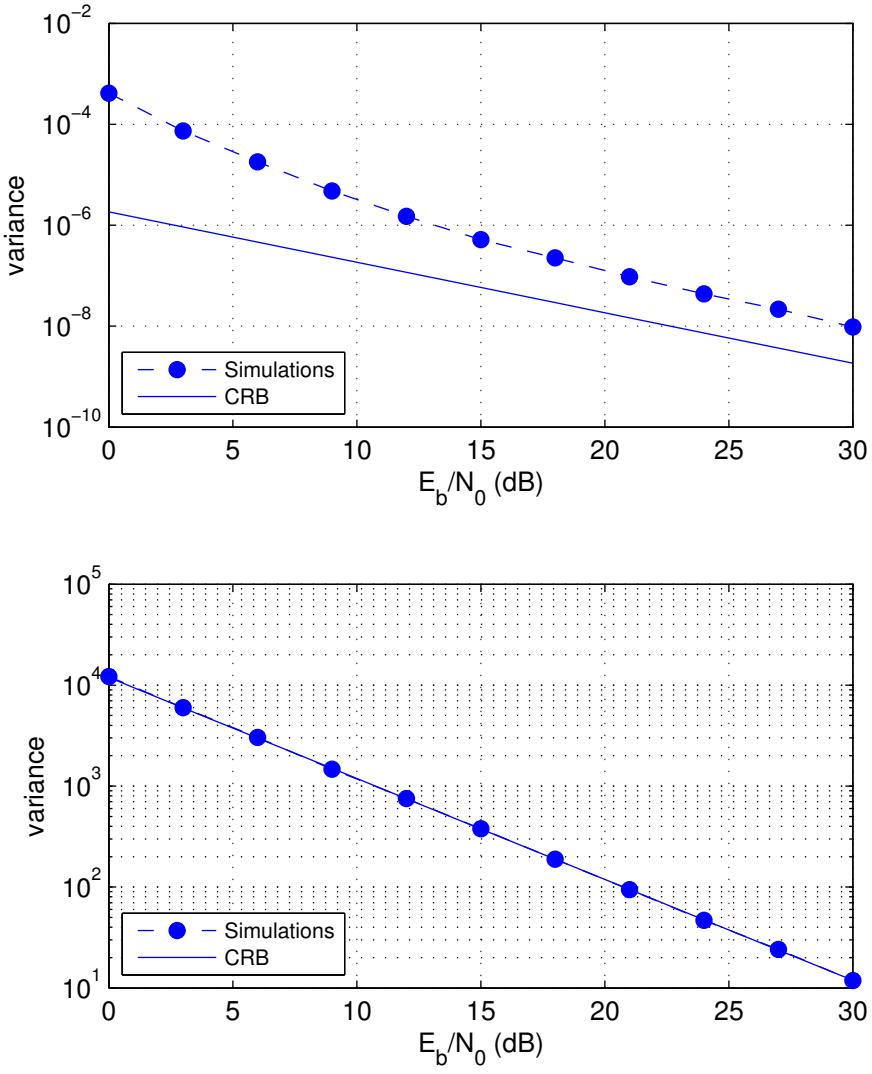

Figure 119: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 10.

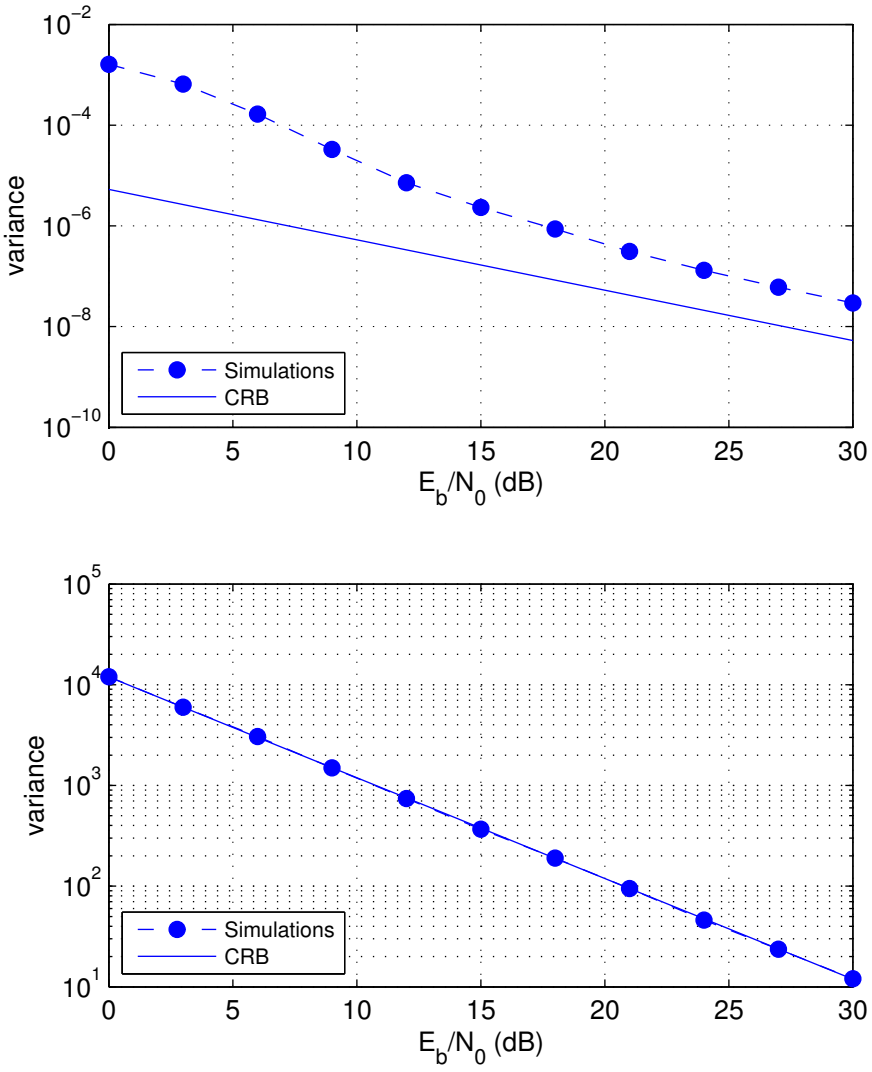

Figure 120: Simulated performance of the frequency estimator (55) [top] and the channel estimator (190) [bottom] for test channel 11.

7 Signal-to-Noise Ratio Estimator

7.1 Introduction

The optimum coefficients for the MMSE equalizer, described in Section 8.3 below, requires knowledge of the signal-to-noise ratio σ_s^2/σ_w^2. The numerator is the signal power

$$\sigma_s^2 = \frac{1}{2}\mathrm{E}\left\{|s(t)|^2\right\}. \tag{265}$$

The denominator is the variance of the additive noise, see (2) and (3). As explained below, the applications of interest assume σ_s^2 is known (in the case of computer simulations) or set (approximately) by an automatic gain control (in practice). For this reason, the focus will be on the estimators for σ_w^2.

7.2 Noise Variance Estimator

Because the received samples are corrupted by the multipath channel, the channel impulse response must be taken into account in the development. To this end, we generalize the channel impulse response estimator developed in Section 4.2.2 to create a joint estimator. The joint estimator is derived from the log-likelihood function for the channel impulse response and noise variance. The channel impulse response estimator is identical to the one derived in Section 4.2.2 and given by (190). The channel impulse response estimate is used as part of the noise variance estimator. We show that the ML noise variance estimator is biased. An unbiased estimator for the noise variance is derived and analyzed.

The starting point in the development is vector of received samples corresponding to the preamble and ASM fields. This is because this formulation gives the best estimator for the channel impulse response. The vector, denoted $\mathbf{r}_2$ in (186), is

$$\mathbf{r}_2 = \mathbf{X}_{\mathrm{i}}\mathbf{h} + \mathbf{w}_2 \tag{266}$$

where $\mathbf{r}_2$ and $\mathbf{w}_2$ are the $(N_{\mathrm{p}} + N_{\mathrm{asm}} - N_1 - N_2) \times 1$ vectors

$$\mathbf{r}_2 = \begin{bmatrix} r(i+N_2) \\ \cdots \\ r(i+N_{\mathrm{p}}+N_{\mathrm{asm}}-N_1-1) \end{bmatrix} \quad \mathbf{w}_2 = \begin{bmatrix} w(i+N_2) \\ \cdots \\ w(i+N_{\mathrm{p}}+N_{\mathrm{asm}}-N_1-1) \end{bmatrix}, \tag{267}$$

$\mathbf{X}_\mathrm{i}$ is the $(N_\mathrm{p} + N_\mathrm{asm} - N_1 - N_2) \times (N_1 + N_2 + 1)$ convolution matrix formed from the samples corresponding to the transmitted preamble and ASM fields [see (186)], and $\mathbf{h}$ is the $(N_1+N_2+1)\times 1$ vector formed by the channel coefficients,

$$\mathbf{h} = \begin{bmatrix} h(-N_1) \\ \vdots \\ h(N_2) \end{bmatrix}. \tag{268}$$

The noise samples in $\mathbf{w}_2$ are uncorrelated zero-mean complex-valued Gaussian random variables with covariance matrix

$$\frac{1}{2}\mathrm{E}\left\{\mathbf{w}_2\mathbf{w}_2^\dagger\right\} = \sigma_w^2\mathbf{I}_{L_2} \tag{269}$$

so that the conditional PDF of $\mathbf{r}_2$ is

$$f(\mathbf{r}_2|\mathbf{h},\sigma_w^2) = \frac{1}{(2\pi\sigma_w^2)^{L_2}}\exp\left\{-\frac{1}{2\sigma_w^2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2\right\} \tag{270}$$

where $L_2 = N_\mathrm{p} + N_\mathrm{asm} - N_1 - N_2$. The log-likelihood function for $\mathbf{h}$ and σ_w^2 is

$$\Lambda(\mathbf{h},\sigma_w^2) = -L_2\log\left(2\pi\sigma_w^2\right) - \frac{1}{2\sigma_w^2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2. \tag{271}$$

The maximum likelihood estimates for the channel impulse response $\mathbf{h}$ and the noise variance σ_w^2 are the solutions to the equations

$$\frac{\partial}{\partial\mathbf{h}^\dagger}\Lambda(\mathbf{h},\sigma_w^2) = 0 \tag{272}$$

$$\frac{\partial}{\partial\sigma_w^2}\Lambda(\mathbf{h},\sigma_w^2) = 0. \tag{273}$$

First, the left-hand-side of (272) evaluates to

$$\frac{\partial}{\partial\mathbf{h}^\dagger}\Lambda(\mathbf{h},\sigma_w^2) = -\frac{\partial}{\partial\mathbf{h}^\dagger}\frac{1}{2\sigma_w^2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2 = \frac{1}{2\sigma_w^2}\mathbf{X}_\mathrm{i}^\dagger\left[\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right]. \tag{274}$$

Setting this equation equal to zero and solving for $\mathbf{h}$ gives

$$\hat{\mathbf{h}} = \left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\mathbf{r}_2. \tag{275}$$

This is identical to the estimator (190), which is to be expected because (190) does not depend on

σ_w^2. Moving to (273), the left-hand-side evaluates to

$$\frac{\partial}{\partial \sigma_w^2}\Lambda(\mathbf{h}, \sigma_w^2) = -\frac{L_2}{\sigma_w^2} + \frac{1}{2\sigma_w^4}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2. \tag{276}$$

Setting this equal to zero and solving for σ_w^2 gives

$$\tilde{\sigma}_w^2 = \frac{1}{2L_2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\mathbf{h}\right|^2. \tag{277}$$

Finally, substituting (275) for $\mathbf{h}$ produces

$$\begin{aligned}\tilde{\sigma}_w^2 &= \frac{1}{2L_2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\hat{\mathbf{h}}\right|^2 \\ &= \frac{1}{2L_2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\mathbf{r}_2\right|^2 \\ &= \frac{1}{2L_2}\left|\left(\mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\right)\mathbf{r}_2\right|^2. \end{aligned} \tag{278}$$

We now evaluate the performance of the estimator (278). The performance of this estimator is measured by its mean and variance. The performance analysis will leverage the following properties:

1. The matrix $\mathbf{M} = \mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger$ is idempotent:

$$\begin{aligned}\mathbf{MM} &= \left[\mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\right]\left[\mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\right] \\ &= \mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger + \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger \\ &= \mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger + \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger \\ &= \mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger \\ &= \mathbf{M}. \end{aligned}$$

2. The rank of an idempotent matrix is equal to its trace [13]:

$$\mathrm{Rank}\left\{\mathbf{M}\right\} = \mathrm{Tr}\left\{\mathbf{M}\right\}. \tag{279}$$

3. The matrix $\mathbf{M} = \mathbf{I} - \mathbf{X}_\mathrm{i} \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger$ is symmetric:

$$\begin{aligned} \mathbf{M}^\dagger = \left(\mathbf{I} - \mathbf{X}_\mathrm{i} \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \right)^\dagger &= \mathbf{I} - \mathbf{X}_\mathrm{i} \left[\left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \right]^\dagger \mathbf{X}_\mathrm{i}^\dagger \\ &= \mathbf{I} - \mathbf{X}_\mathrm{i} \left[\left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^\dagger \right]^{-1} \mathbf{X}_\mathrm{i}^\dagger = \mathbf{M}. \end{aligned}$$

4. The channel estimation error vector $\mathbf{e} = \mathbf{h} - \hat{\mathbf{h}}$ is

$$\mathbf{h} - \hat{\mathbf{h}} = - \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{w}_2. \tag{280}$$

Based on the statistical properties of $\mathbf{w}_2$ [see (269)], the channel estimation error vector is a Gaussian random vector with

$$\mathrm{E}\left\{ \mathbf{e} \right\} = \mathbf{0} \tag{281}$$

$$\frac{1}{2} \mathrm{E}\left\{ \mathbf{e}\mathbf{e}^\dagger \right\} = \sigma_w^2 \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \tag{282}$$

To see that this is so, express $\hat{\mathbf{h}}$ in terms of $\mathbf{X}_\mathrm{i}$ and $\mathbf{r}_2$, and use (266) for $\mathbf{r}_2$:

$$\begin{aligned} \mathbf{e} &= \mathbf{h} - \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{r}_2 \\ &= \mathbf{h} - \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \left(\mathbf{X}_\mathrm{i} \mathbf{h} + \mathbf{w}_2 \right) \\ &= \mathbf{h} - \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \mathbf{h} - \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{w}_2 \\ &= \mathbf{h} - \mathbf{h} - \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{w}_2 \\ &= - \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{w}_2 \end{aligned}$$

where $\mathbf{w}_2$ is the vector of complex-valued Gaussian random variables with zero mean and covariance (269). This gives

$$\begin{aligned} \mathrm{E}\left\{ \mathbf{e} \right\} &= \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathrm{E}\left\{ \mathbf{w}_2 \right\} = \mathbf{0} \\ \frac{1}{2} \mathrm{E}\left\{ \mathbf{e}\mathbf{e}^\dagger \right\} &= \frac{1}{2} \mathrm{E}\left\{ \left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{w}_2 \left[\left(\mathbf{X}_\mathrm{i}^\dagger \mathbf{X}_\mathrm{i} \right)^{-1} \mathbf{X}_\mathrm{i}^\dagger \mathbf{w}_2 \right]^\dagger \right\} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\mathrm{E}\left\{\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\mathbf{w}_2\mathbf{w}_2^\dagger\mathbf{X}_\mathrm{i}\left[\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\right]^\dagger\right\} \\
&= \left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\left[\frac{1}{2}\mathrm{E}\left\{\mathbf{w}_2\mathbf{w}_2^\dagger\right\}\right]\mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1} \\
&= \left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\left[\sigma_w^2\mathbf{I}\right]\mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1} \\
&= \sigma_w^2\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1} \\
&= \sigma_w^2\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}.
\end{aligned}
$$

Now for the analysis. The estimator (278) is a scaled version of the squared norm of the vector $\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\hat{\mathbf{h}}$. This vector may be expressed as

$$
\begin{aligned}
\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\hat{\mathbf{h}} &= \mathbf{X}_\mathrm{i}\mathbf{h} + \mathbf{w}_2 - \mathbf{X}_\mathrm{i}\hat{\mathbf{h}} \\
&= \mathbf{w}_2 + \mathbf{X}_\mathrm{i}\left(\mathbf{h} - \hat{\mathbf{h}}\right) \\
&= \mathbf{w}_2 - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\mathbf{w}_2 \qquad \text{[see (280)]} \\
&= \left[\mathbf{I} - \mathbf{X}_\mathrm{i}\left(\mathbf{X}_\mathrm{i}^\dagger\mathbf{X}_\mathrm{i}\right)^{-1}\mathbf{X}_\mathrm{i}^\dagger\right]\mathbf{w}_2 \\
&= \mathbf{M}\mathbf{w}_2 \qquad (283)
\end{aligned}
$$

The estimator may now be expressed in terms of the noise vector $\mathbf{w}_2$:

$$
\begin{aligned}
\tilde{\sigma}_w^2 &= \frac{1}{2L_2}\left|\mathbf{r}_2 - \mathbf{X}_\mathrm{i}\hat{\mathbf{h}}\right|^2 \\
&= \frac{1}{2L_2}\left|\mathbf{M}\mathbf{w}_2\right|^2 \qquad \text{[relationship (283)]} \\
&= \frac{1}{2L_2}\mathbf{w}_2^\dagger\mathbf{M}^\dagger\mathbf{M}\mathbf{w}_2 \qquad \text{[definition of squared norm]} \\
&= \frac{1}{2L_2}\mathbf{w}_2^\dagger\mathbf{M}\mathbf{M}\mathbf{w}_2 \qquad \text{[M is symmetric]} \\
&= \frac{1}{2L_2}\mathbf{w}_2^\dagger\mathbf{M}\mathbf{w}_2 \qquad \text{[M is idempotent]} \qquad (284)
\end{aligned}
$$

Because $\mathbf{M}$ is idempotent, the random variable $(1/\sigma_w^2)\mathbf{w}_2^\dagger\mathbf{M}\mathbf{w}_2$ is a chi-square distributed random variable with $2 \times \mathrm{Rank}\{\mathbf{M}\}$ degrees of freedom. This is usually denoted $(1/\sigma_w^2)\mathbf{w}_2^\dagger\mathbf{M}\mathbf{w}_2 \sim \chi_{2\rho}^2$

where $\rho = \text{Rank}\{\mathbf{M}\}$. Consequently,

$$\text{E}\left\{\frac{1}{\sigma_w^2}\mathbf{w}_2^\dagger \mathbf{M}\mathbf{w}_2\right\} = 2\rho \tag{285}$$

$$\text{VAR}\left\{\frac{1}{\sigma_w^2}\mathbf{w}_2^\dagger \mathbf{M}\mathbf{w}_2\right\} = 4\rho. \tag{286}$$

Because

$$\tilde{\sigma}_w^2 = \frac{\sigma_w^2}{2L_2} \times \frac{1}{\sigma_w^2}\mathbf{w}_2^\dagger \mathbf{M}\mathbf{w}_2, \tag{287}$$

we have

$$\text{E}\left\{\tilde{\sigma}_w^2\right\} = \frac{\sigma_w^2}{2L_2} \times 2\rho = \frac{\rho}{L_2}\sigma_w^2 \tag{288}$$

$$\text{VAR}\left\{\tilde{\sigma}_w^2\right\} = \left(\frac{\sigma_w^2}{2L_2}\right)^2 \times 4\rho = \frac{\rho}{L_2^2}\sigma_w^4. \tag{289}$$

This shows that $\tilde{\sigma}_w^2$ is a *biased* estimator if $\rho \neq L_2$. This fact motivates the use of the unbiased estimator

$$\hat{\sigma}_w^2 = \frac{1}{2\rho}\left|\mathbf{r}_2 - \mathbf{X}_\text{i}\hat{\mathbf{h}}\right|^2 = \frac{1}{2\rho}\left|\mathbf{M}\mathbf{r}_2\right|^2. \tag{290}$$

Following the same steps culminating in (284), we have

$$\hat{\sigma}_w^2 = \frac{1}{2\rho}\mathbf{w}_2^\dagger \mathbf{M}\mathbf{w}_2 = \frac{\sigma_w^2}{2\rho} \times \frac{1}{\sigma_w^2}\mathbf{w}_2^\dagger \mathbf{M}\mathbf{w}_2 \tag{291}$$

so that

$$\text{E}\left\{\hat{\sigma}_w^2\right\} = \frac{\sigma_w^2}{2\rho} \times 2\rho = \sigma_w^2 \tag{292}$$

$$\text{VAR}\left\{\hat{\sigma}_w^2\right\} = \left(\frac{\sigma_w^2}{2\rho}\right)^2 \times 4\rho = \frac{1}{\rho}\sigma_w^4 \tag{293}$$

$$= \frac{\sigma_w^4}{\text{Trace}\{\mathbf{M}\}}. \qquad \text{[see (279)]} \tag{294}$$

Some important observations for the estimator (290) are the following:

1. The first of the two forms of estimator in (290) requires that $\mathbf{X}_\text{i}$ and ρ be known. Because these quantities are not a function of the received data, they can be precomputed and stored. Note that $\mathbf{X}_\text{i}$ is also required for the channel impulse response estimator (275), so the storage requirement here is not an additional requirement. The second of the two forms requires

the $L_2 \times L_2$ matrix $\mathbf{M}$. As with $\mathbf{X}_\text{i}$, $\mathbf{M}$ is not a function of the received data, so it can be precomputed and stored. This represents an additional storage requirement because the noise variance estimator is the only estimator that requires this matrix. In either case, ρ is needed. The quantity ρ is a real-valued scalar, so storage requirements are minimal.

2. The computational complexities of the two forms (290) are slightly different.

   - For the first of the two forms: $\mathbf{X}_\text{i}\hat{\mathbf{h}}$ requires $L_2(N_1 + N_2 + 1)$ complex-valued multiplications and $L_2(N_1 + N_2)$ complex-valued additions; $\mathbf{r}_2 - \mathbf{X}_\text{i}\hat{\mathbf{h}}$ requires L_2 complex-valued additions; $\left|\mathbf{r}_2 - \mathbf{X}_\text{i}\hat{\mathbf{h}}\right|^2$ requires L_2 complex-valued multiplications and $L_2 - 1$ complex-valued additions. In total, this form requires $L_2(N_1+N_2+2)$ complex-valued multiplications and $L_2(N_1 + N_2 + 2) - 1$ complex-valued additions.
   - For the second of the two forms: $\mathbf{M}\mathbf{r}_2$ requires L_2^2 complex-valued multiplications and $L_2(L_2 - 1)$ complex-valued additions; $\mathbf{r}_2^\dagger(\mathbf{M}\mathbf{r}_2)$ requires L_2 complex-valued multiplications and $L_2 - 1$ complex-valued additions. In total, this form requires $(L_2 + 1)L_2$ complex-valued multiplications and $(L_2+1)(L_2-1)$ complex-valued complex-valued additions.

   These results are summarized in Table 2. It is expected that in normal circumstances, $L_2 = (N_\text{p} + N_\text{asm} - N_1 - N_2) > (N_1 + N_2 + 1)$. Consequently, the first form of the estimator requires fewer arithmetic operations.

3. The variance of the biased estimator (278) is lower (better!) than the variance of the unbiased estimator (290). This follows from fact that $L_2 > \rho$. For example, in the simulations that follow, the matrix $\mathbf{M}$ is formed from $\mathbf{X}_\text{i}$ created at 2 samples/bit, and set $N_1 = 12$ and $N_2 = 25$. This gives $L_2 = N_\text{p} + N_\text{asm} - N_1 - N_2 = 2 \times 128 + 2 \times 64 - 12 - 25 = 347$, and $\rho = \text{Trace}\{\mathbf{M}\} = 309$.

The performance of the unbiased estimator (290) was evaluated in simulation. The simulations were performed as follows:

1. Samples of the SOQPSK-TG signal corresponding to 200 data bits, the preamble bits, and the ASM bits were generated at $N = 2$ samples/bit. The SOQPSK-TG signal was passed through a filter whose impulse response was set to a test channel. Noise samples, with a known variance, were added to the test channel filter output.

Table 2: Comparison of the Computational Complexity for the Two Forms of the Noise Variance Estimator (290).

Form of (290)	# complex-valued multiplications	# complex-valued additions
First form of (290)	$L_2(N_1 + N_2 + 2)$	$L_2(N_1 + N_2)$
Second form of (290)	$L_2(L_2 + 1)$	$(L_2 + 1)(L_2 - 1)$

2. The start of the preamble was assumed known. The channel estimate was computed using (275) for $N_1 = 12$ and $N_2 = 25$. The estimator (290) was applied to produce an estimate of the noise variance.

3. The previous two steps were repeated 1000 times. The sample mean and variance of the estimator were computed from the 1000 estimates.

The results are plotted in Figure 121. In the top figure, the solid line is a plot of mean predicted by (292) whereas the circular markers are the results of the simulations. In the lower figure, the solid line is a plot of the estimator error variance predicted by (294) whereas the circular markers are the results of the simulations.

A plot of the same results, but in terms of E_b/N_0 is shown in Figure 122. In the top figure, the solid line is a plot of mean predicted by (292) whereas the circular markers are the results of the simulations. In the lower figure, the solid line is a plot of the estimator error variance predicted by (294) whereas the circular markers are the results of the simulations. Note that this is the same data as that in Figure 121, the data are plotted against a different abscissa. The relationship between the noise variance σ_w^2 and E_b/N_0 is given by (307) below for $N = 2$ samples/bit.

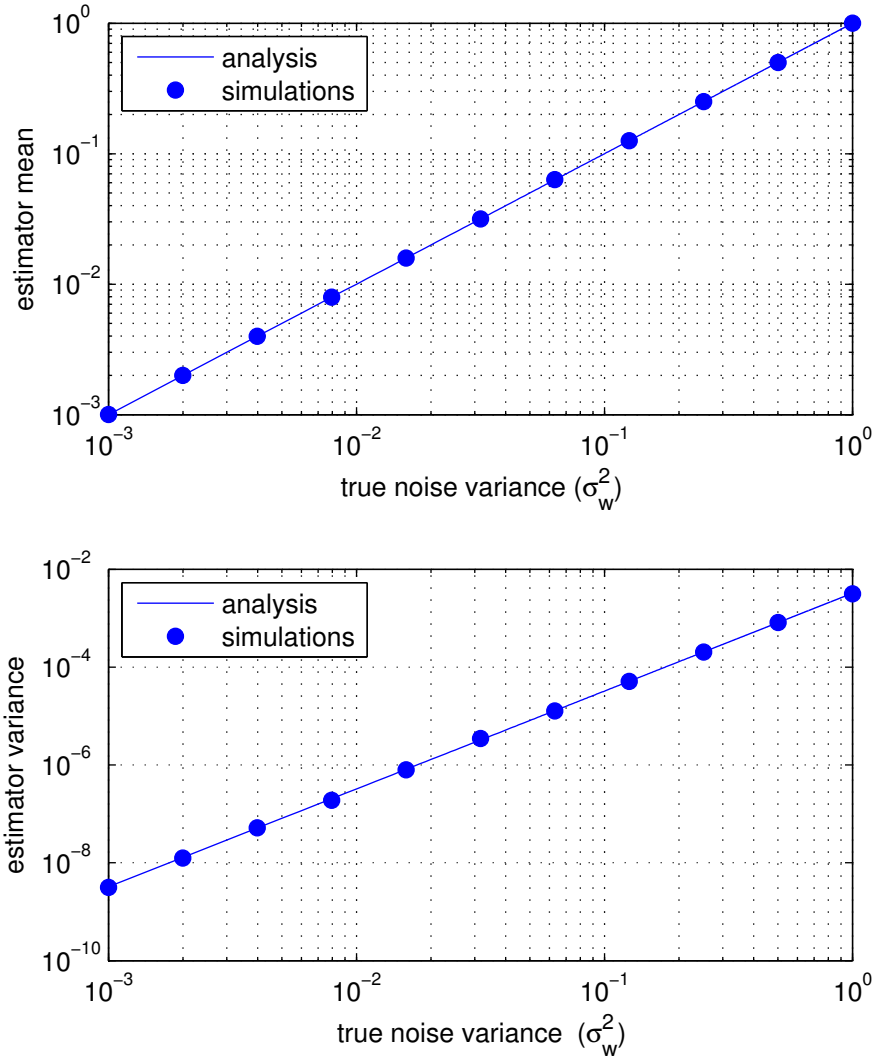

Figure 121: The performance of the noise variance estimator (290) for SOQPSK-TG samples (at 2 samples/bit) over test channel 1 as a function of the true variance: (top) the estimator mean; (bottom) the estimator variance.

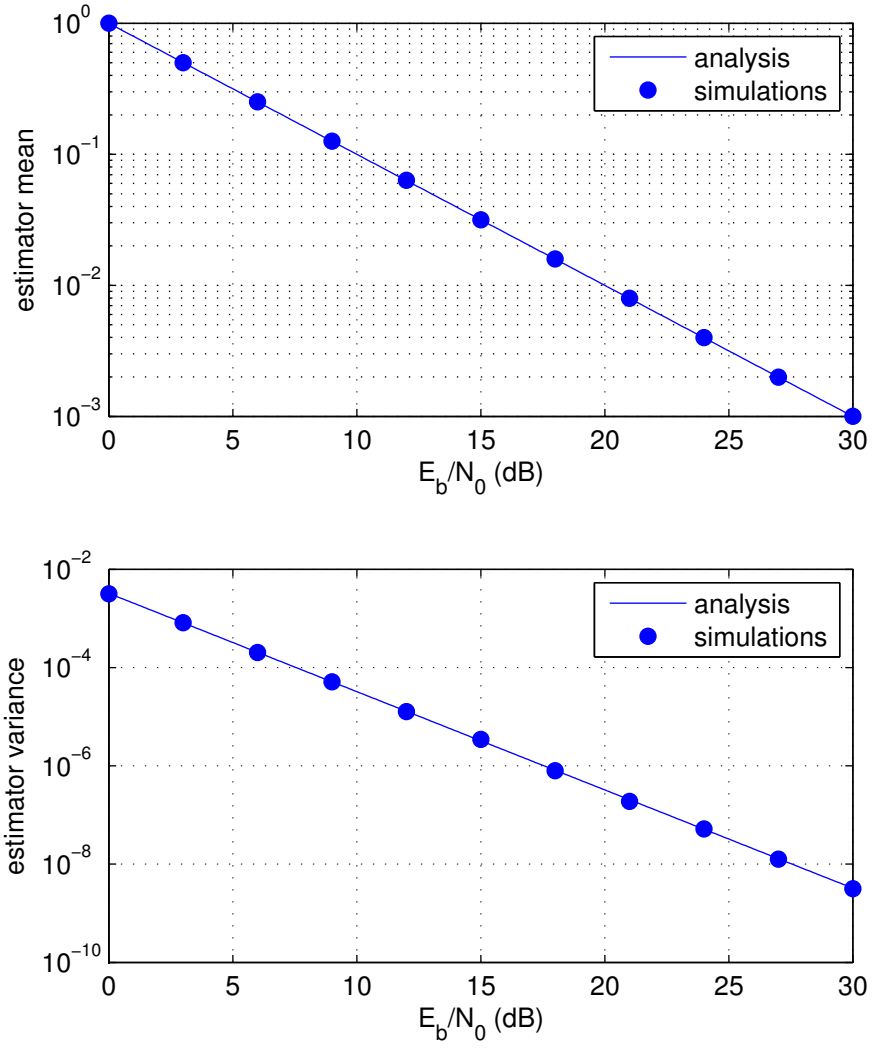

Figure 122: The performance of the noise variance estimator (290) for SOQPSK-TG samples (at 2 samples/bit) over test channel 1 as a function of E_b/N_0, cf. Figure 121: (top) the estimator mean; (bottom) the estimator variance.

7.3 Signal-to-Noise Ratio Estimator in the Field and in Simulation

In fielded systems, automatic gain control (AGC) is used to scale the received signal $r(t)$ to a fixed level. This is equivalent to setting the received power to a known reference level R. Assuming the signal and noise are uncorrelated, we have

$$R = \sigma_s^2 + \sigma_w^2. \tag{295}$$

The signal-to-noise ratio may be expressed in terms of the reference level as

$$\frac{\sigma_s^2}{\sigma_w^2} = \frac{R}{\sigma_w^2} - 1. \tag{296}$$

Thus, given the known reference level R, only an estimate of σ_w^2 is required to estimate the signal-to-noise ratio. The estimator (290) may be used here:

$$\left(\frac{\sigma_s^2}{\sigma_w^2}\right)_{\text{est}} = \frac{R}{\hat{\sigma}_w^2} - 1. \tag{297}$$

The performance of (297) was evaluated in simulation. The simulation parameters are identical to those used to create Figures 121 and 122. The simulation results are shown in Figure 123. Simulation results are presented for two cases, the AWGN case and the case where test channel 1 is in play. The SNR estimator is unbiased for the AWGN case, but displays a bias for the test channel 1 case. The bias derives from the way the AGC deals with the distortion caused by the channel. This leaves us with one of two possibilities: 1) derive an algorithm for backing out the impact of the channel on the AGC using the channel estimate; or 2) ensure that the MMSE equalizer performance is robust to small errors in the signal-to-noise ratio.

Note, the relationship between the signal-to-noise ratio σ_s^2/σ_w^2 and E_b/N_0 is derived below and given by (304).

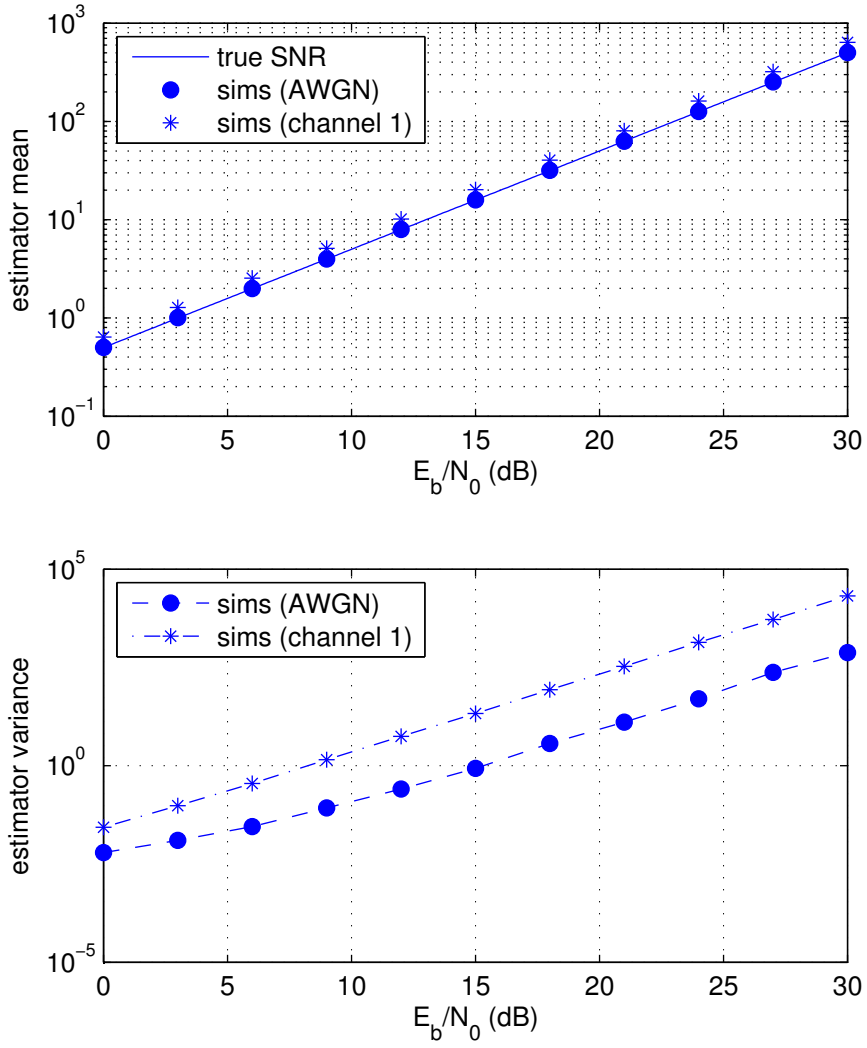

Figure 123: Performance of the signal-to-noise estimator (297): (top) estimator mean (bottom) estimator variance.

In computer simulations, $|s(nT)|^2$ is fixed at a known value. This also fixes σ_s^2 at a known value. Consequently, the desired signal-to-noise ratio is only a function of σ_w^2. For example, in this report, the computer simulations set $|s(nT)|^2 = 1$. This gives $\sigma_s^2 = 1/2$. The signal-to-noise ratio is thus

$$\frac{\sigma_s^2}{\sigma_w^2} = \frac{1}{2\sigma_w^2}. \tag{298}$$

The noise variance estimate (290) may be used on the right-hand-side to produce the the estimate of the signal-to-noise ratio:

$$\left(\frac{\sigma_s^2}{\sigma_w^2}\right)_{\text{est}} = \frac{1}{2\hat{\sigma}_w^2}. \tag{299}$$

The performance of (299) was evaluated in simulation. The simulation parameters are identical to those used to create Figures 121 and 122. The simulation results are shown in Figure 124 for the case using test channel 1. The close agreement between the estimator mean and the true SNR shows that, unlike the case where the AGC is used to set the signal level, the estimator (299) benefits from knowing σ_s^2.

Note, the relationship between the signal-to-noise ratio σ_s^2/σ_w^2 and E_b/N_0 is derived below and given by (304).

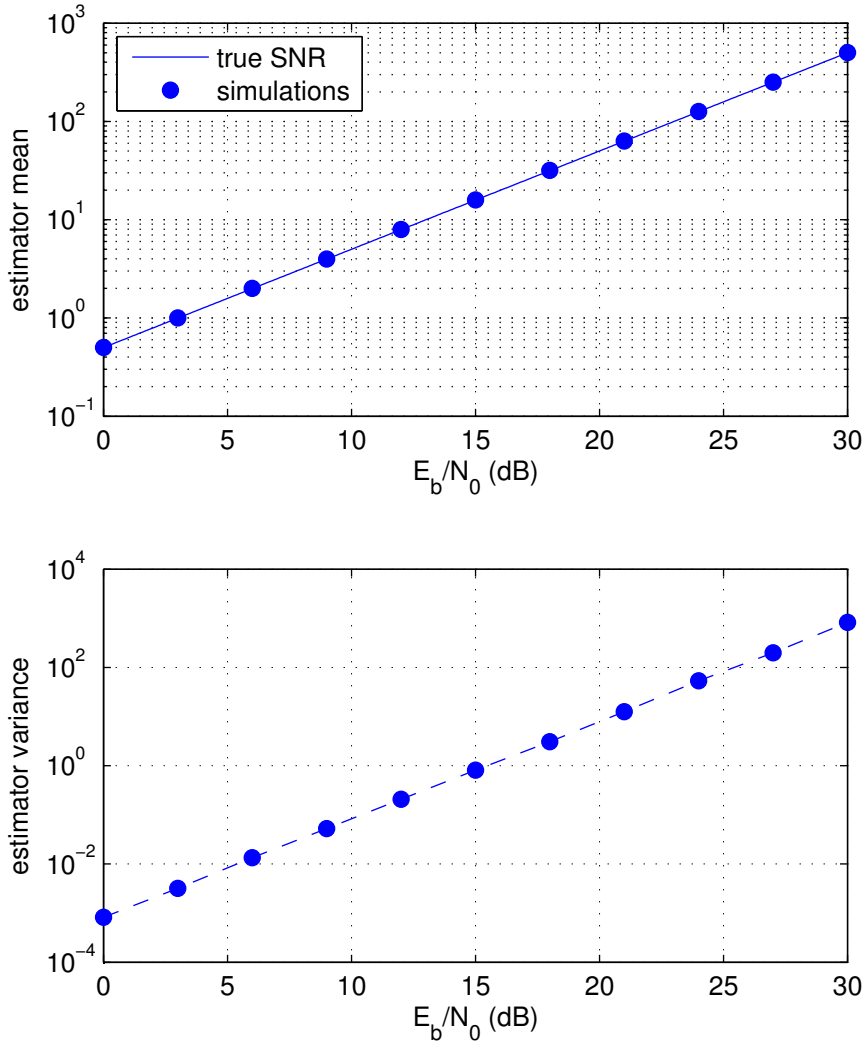

Figure 124: Performance of the signal-to-noise estimator (299): (top) estimator mean (bottom) estimator variance.

In simulations, system performance is usually parameterized by the ratio of bit energy to noise power spectral density level: E_b/N_0. As such, it becomes necessary to express σ_s^2 in terms of E_b and σ_w^2 in terms of N_0. Because computer simulations necessarily operate on samples of the desired signals, we must relate σ_s^2 and σ_w^2 in terms of samples of $s(t)$ and $w_2(t)$. We assume T-spaced samples of these signals are available.

We begin with the relationship between σ_s^2 and E_b. First, note that

$$\sigma_s^2 = \frac{1}{2}\mathrm{E}\left\{|s(t)|^2\right\} = \frac{1}{2}\mathrm{E}\left\{|s(nT)|^2\right\} = \frac{1}{2}|s(nT)|^2 \tag{300}$$

where the last step is true because SOQPSK-TG is a CPM, which means $|s(t)|^2$ is a constant. The bit energy is given by

$$E_b = \frac{1}{2}\int_0^{T_b} |s(t)|^2\, dt \approx \frac{1}{2}T\sum_{n=0}^{N-1}|s(nT)|^2 = \frac{T}{2}\sum_{n=0}^{N-1} 2\sigma_s^2 = TN\sigma_s^2 \tag{301}$$

where $N = T_b/T$ is the number of samples/bit.

Moving to σ_w^2 we begin with the expression of $w(t)$ in terms of its in-phase and quadrature components:

$$w(t) = w_I(t) + jw_Q(t). \tag{302}$$

The power spectral densities of the inphase and quadrature components are shown in Figure 125 [14]. Samples of $w_I(t)$ and $w_Q(t)$ are obtained by filtering $w_I(t)$ and $w_Q(t)$ by a low-pass anti-aliasing filter and setting $t = nT$. Assuming an ideal low-pass filter for the anti-aliasing filter, this process is represented by setting $B = 1/(2T)$ in Figure 125. The variance σ_w^2 is the variance of the filtered version of $w_I(t)$ or the variance of the filtered version of $w_Q(t)$. The variance of the filtered version of $w_I(t)$ is

$$\sigma_w^2 = \int_{-\infty}^{\infty} P_I(f)df = \int_{-1/2T}^{1/2T} N_0 df = \frac{N_0}{T}. \tag{303}$$

The ratio E_b/N_0 is obtained from (301) and (303):

$$\frac{E_b}{N_0} = \frac{TN\sigma_s^2}{T\sigma_w^2} = N\frac{\sigma_s^2}{\sigma_w^2}. \tag{304}$$

This relates E_b/N_0 so the signal-to-noise ratio σ_s^2/σ_w^2. [This is the relationship used in (332) in Section 8.3.] In the computer simulations performed as part of this work, $|s(nT)|^2 = 1$ which

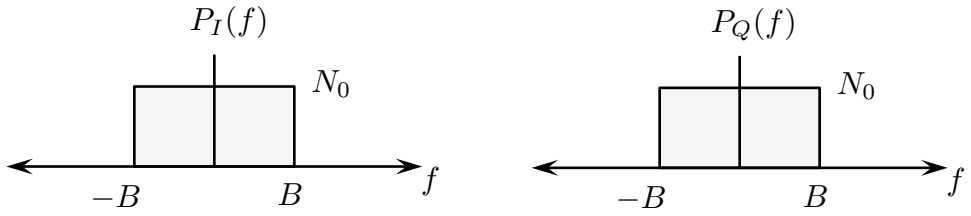

Figure 125: The power spectral densities of the inphase and quadrature components of $w(t)$. Adopted from [14].

gives $\sigma_s^2 = 1/2$. Consequently, the desired signal-to-noise ratio is

$$\frac{E_b}{N_0} = \frac{N}{2\sigma_w^2}. \tag{305}$$

Because most of the signal processing occurs at $N = 2$ samples/bit, the relationship for our simulations becomes

$$\frac{E_b}{N_0} = \frac{1}{\sigma_w^2}. \tag{306}$$

On a related note, (305) can be used to express the variance of the additive noise in terms of E_b/N_0:

$$\sigma_w^2 = \frac{N}{2\dfrac{E_b}{N_0}}. \tag{307}$$

This means for $\sigma_a^2 = 1/2$, (307) is used to scale the additive noise samples to simulate the desired the performance of the system for a given E_b/N_0. Because the modified receiver produces samples of the received signal at an equivalent sample rate of 2 samples/bit, the most usual case for the simulations in this report is $N = 2$ samples/bit.

8 Equalization

8.1 Overview

Equalization, as a technique to overcome inter-symbol interference (ISI) due to multipath propagation (or other causes), has been well-studied (see, for example, Chapters 9 and 10 of [14]). In most contexts, equalization is applied to linear modulations (e.g., QPSK) and operates on the symbol estimates. A notable exception is the *fractionally-spaced equalizer* that operates at an equivalent sample rate in excess of one sample/symbol [14, Section 9.4-4]. An equalizer is a filter. The optimum filter coefficients are a function of the channel impulse response and the statistics of the signal.

Equalization techniques can be categorized as *data-aided* or *blind*. Data-aided equalizers depend on the presence of a known data sequence. The known data sequence can be used in one of two ways. First, the known data sequence may be used to estimate the channel impulse response. This estimate, together with the signal statistics, is used to compute the optimum equalizer filter coefficients. The second possible use of the known data sequence is as training data for an adaptive filter version of the equalizer. Here the training data is used to compute the error signal that drives the filter coefficient update equation, such as LMS or RLS [15]. Usually, the training data is sufficiently long to allow the filter update algorithm to converge. After convergence, the adaptive filter continues to update (or adapt) to small changes in the channel impulse response by using an error signal derived from the symbol decisions.

The second broad category is blind equalizers. Here, the equalization algorithm has no knowledge of the data symbols (hence the term "blind") and uses some statistical property of the signal or symbols as the object function. In (almost?) all cases, blind equalizers are adaptive equalizers.

Because SOQPSK-TG is a nonlinear modulation, the notion of equalizing the data symbols is not straight-forward. Like the fractionally-spaced equalizer structure, we apply an equalizer filter to the *samples* of the received signal in this project. The basic idea is outlined in Figure 126. Here, the derotated data samples are given by [cf. (2)]

$$r(n) = \left[\sum_{k=-N_1}^{N_2} h(k) s(n-k) \right] e^{j(\omega_0 - \hat{\omega}_0)n} + w(n) \tag{308}$$

where $h(n)$ is the channel impulse response, ω_0 rads/sample is the frequency offset and $\hat{\omega}_0$ is its estimate, and $w(n)$ is a complex-valued zero-mean Gaussian random process with auto-covariance

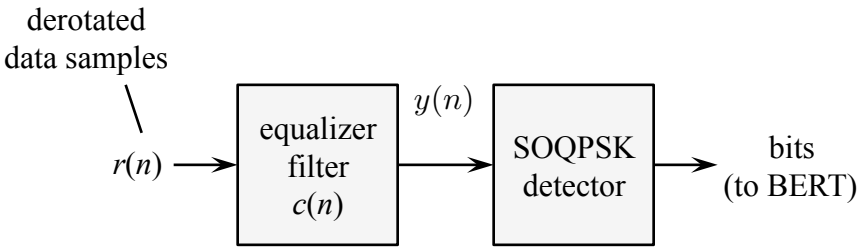

Figure 126: The basic outline of the fractionally-spaced equalizer.

function

$$\frac{1}{2}\mathrm{E}\left\{w(n)w^*(n-k)\right\} = \sigma_w^2\delta(k). \tag{309}$$

The key feature here that the equalizer operates on the samples of the received signal instead of the symbol estimates *prior* to the detection algorithm.[6] Three equalization techniques fit this structure:

1. The zero-forcing (ZF) equalizer.

2. The minimum mean-squared error (MMSE) equalizer.

3. The constant-modulus algorithm (CMA) equalizer.

The ZF and MMSE equalizers are data-aided equalizers and have both fixed-coefficient filter and adaptive filter realizations. The CMA, on the other hand, is blind and only has an adaptive realization. It is the blind, adaptive CMA algorithm that has received most of the attention in aeronautical telemetry applications [16–19].

In the next two sections, we investigate the performance of the ZF and MMSE equalizers, operating at 2 samples/bit, over the eleven test channels. Following that, we investigate a modified version of the CMA algorithm, called CMA+AMA over the same channels. We conclude with longitudinal comparisons and thoughts on some other equalization algorithms.

8.2 The Zero-Forcing Equalizer

The zero-forcing equalizer is a filter whose impulse response attempts to "invert" the channel. That is $c(n)$ is chosen so that

$$h(n) * c(n) \approx \delta(n - n_0). \tag{310}$$

[6]The detection algorithms are outlined in the Appendix.

The approximation is used because the "inverse" of an finite impulse response (FIR) system is an infinite impulse response (IIR) system [15]. Restricting $c(n)$ to be FIR for computational complexity reasons, means that the desired goal can only be approximately achieved. The delay n_0 follows from possibility that non-causal channels (see Figure 2) are best served by non-causal equalizer filters.

The zero-forcing equalizer is rarely used in practice because it does not take into consideration additive noise. When "inverting" a channel with one or more deep spectral nulls, the noise power in the spectral region corresponding to the null is multiplied by a large amount. This "noise amplification" effect reduces overall performance. Consequently, zero-forcing equalizers are usually only found in applications where the channels to not have deep nulls and signal-to-noise ratio is always high. The zero-forcing equalizer is included here primarily for theoretical reasons: candidate equalizers should outperform the zero-forcing equalizer.

We assume $c(n)$ has support on $-L_1 \leq n \leq L_2$ and that $h(n)$ has support on $-N_1 \leq n \leq N_2$ (see Figure 2). Using the least-squares error criterion, the optimum ZF filter coefficients are

$$\mathbf{c} = \left(\mathbf{H}^{\dagger}\mathbf{H}\right)^{-1}\mathbf{H}^{\dagger}\mathbf{u}_{n_0} \tag{311}$$

where $\mathbf{c}$ is an $(L_1 + L_2 + 1) \times 1$ vector of ZF equalizer filter coefficients and $\mathbf{u}_{n_0}$ is the $(N_1 + N_2 + L_1 + L_2 + 1) \times 1$ vector representing the desired composite impulse response: that is, a vector comprising all-zeros with a one in position n_0. These vectors are

$$\mathbf{c} = \begin{bmatrix} c(-L_1) \\ \vdots \\ c(0) \\ \vdots \\ c(L_2) \end{bmatrix} \qquad \mathbf{u}_{n_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n_0 - 1 \text{ zeros} \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N_1 + N_2 + L_1 + L_2 - n_0 + 1 \text{ zeros} \end{matrix} \tag{312}$$

and $\mathbf{H}$ is the $(N_1 + N_2 + L_1 + L_2 + 1) \times (L_1 + L_2 + 1)$ convolution matrix formed by the channel

impulse response:

$$\mathbf{H} = \begin{bmatrix} h(-N_1) & & & & \\ h(-N_1+1) & h(-N_1) & & & \\ \vdots & \vdots & \ddots & & \\ h(N_2) & h(N_2-1) & & h(-N_1) & \\ & h(N_2) & & h(-N_1+1) & \\ & & & \vdots & \\ & & & h(N_2) & \end{bmatrix}. \tag{313}$$

The simulated bit error rate (BER) performance of the ZF equalizer was performed using the following parameters.

- The SOQPSK-TG signal and channel were sampled at a rate equivalent to 2 samples/bit.
- The ZF equalizer length was set to approximately 4 times the channel length by using $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$.
- The delay parameter n_0 of (310) – (312) was set to $n_0 = N_1 + L_1 + 1$. This assignment places the one in $\mathbf{u}_{n_0}$ at the zero-delay tap $n = 0$.
- The "ideal" simulations are based on a detector endowed with perfect knowledge of the frequency offset [$\omega_0 - \hat{\omega}_0 = 0$ in (308)] and the channel. As such the ideal detectors described in the Appendix are used: the ideal detector of Figure 157 (a) is used for the symbol-by-symbol (SxS) detector and the ideal detector of Figure 161 using the trellis of Figure 160 with partial path metric updates (396) 4-state trellis detector.
- The "with estimator" simulations are based on estimates of the frequency offset and channel. The frequency offset estimator (55) is used to produce the frequency offset estimate which is used to de-rotate the preamble, ASM, and data samples. Consequently, $\omega_0 - \hat{\omega}_0 \neq 0$ in (308). The de-rotated preamble and ASM are used to estimate the channel impulse response using (190) assuming $N_1 = 12$ and $N_2 = 25$. The channel impulse response estimates $\hat{h}(n)$ for $-N_1 \leq n \leq N_2$ are used in place of $h(n)$ in forming the matrix $\mathbf{H}$ defined by (313). The phase-tracking-enabled detectors described in the Appendix are used here: the detector shown in Figure 157 (b) is used for the symbol-by-symbol (SxS) detector and the detector shown in Figure 161 using the trellis of Figure 160 with partial path metric updates (414) – (416) is used for the 4-state trellis detector.

The simulation results for test channels 1 – 11 are shown in Figures 127 – 132, respectively. In each plot, the clear markers mark the BER performance of the ideal case whereas the filled markers mark the BER performance of the case where estimates replace the ideal values. The circles mark the performance of the SxS detector and the squares mark the performance of the 4-state trellis detector. Two important conclusions are illustrated by the simulation results

1. The performance differences between the SxS and 4-state trellis detectors for both the ideal and non-ideal cases are in the 0 to 1 dB range. This is consistent with the AWGN results reported in [20, 21].

2. The penalty for using the estimates is 0 to 2 dB for all test channels except test channels 2 and 4. For channels 2 and 4, the penalty is worse. This is because the performance of the estimators in these channels is not as good.

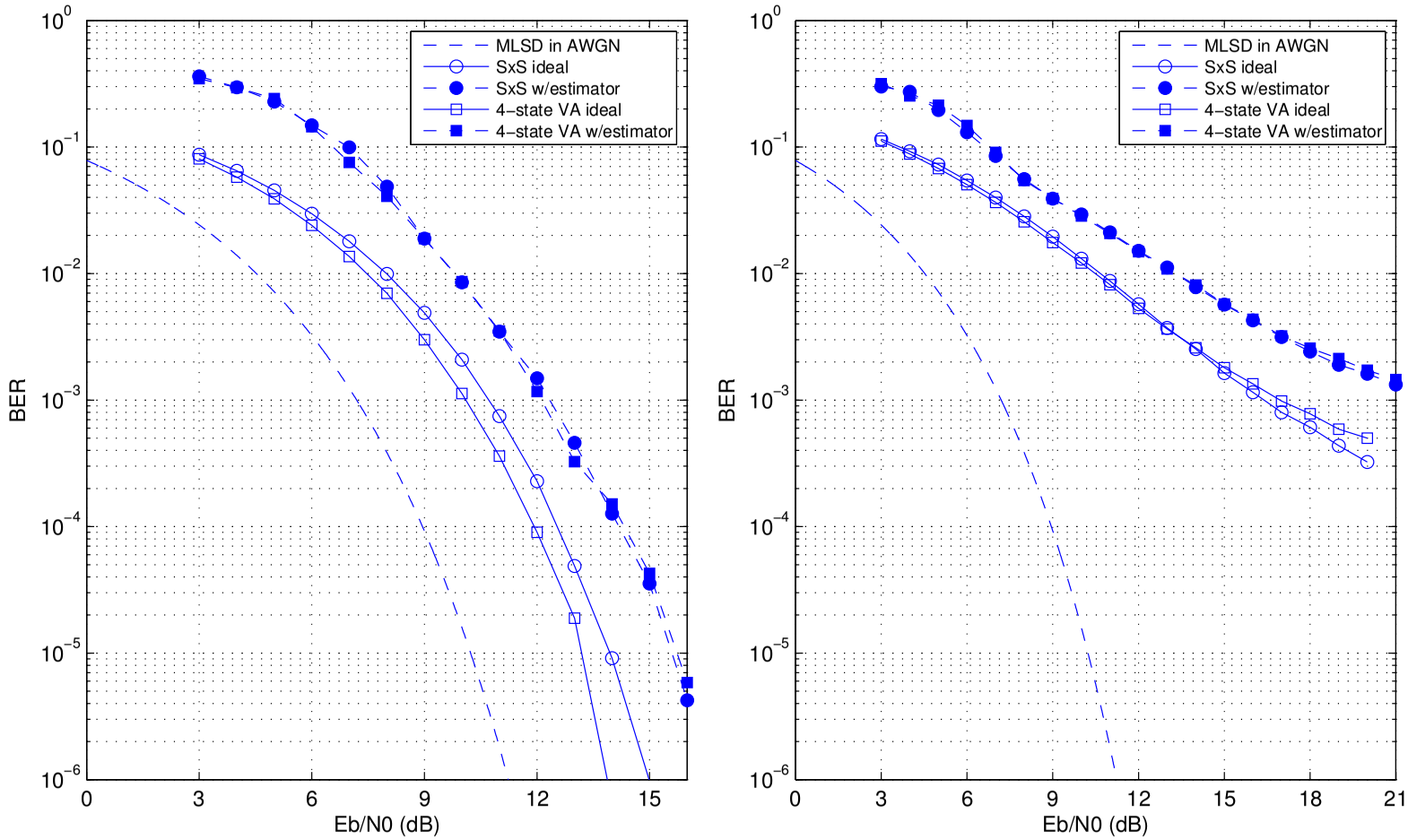

Figure 127: Simulated bit error rate (BER) performance of the ZF equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 1 (left) and test channel 2 (right).

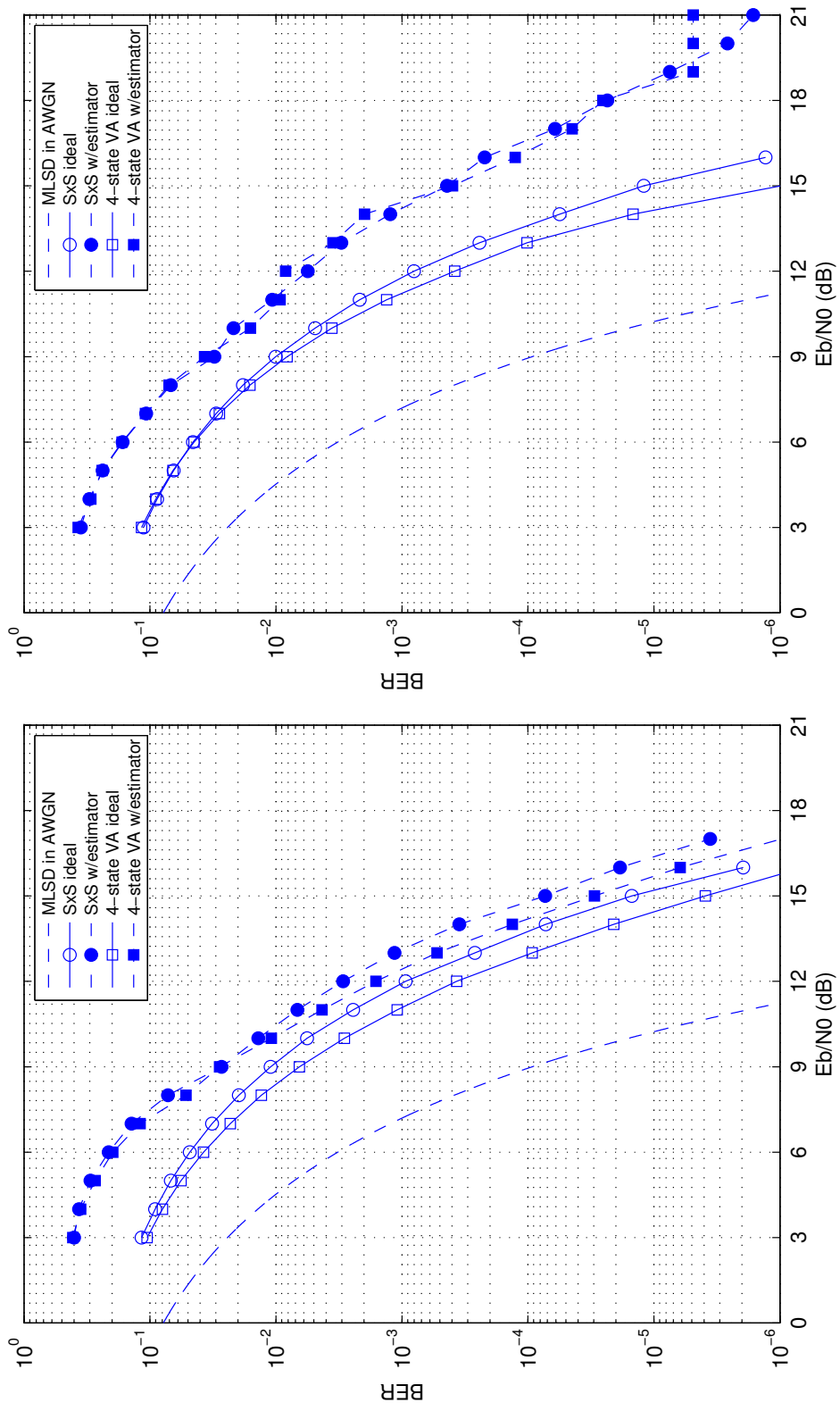

Figure 128: Simulated bit error rate (BER) performance of the ZF equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 3 (left) and test channel 4 (right).

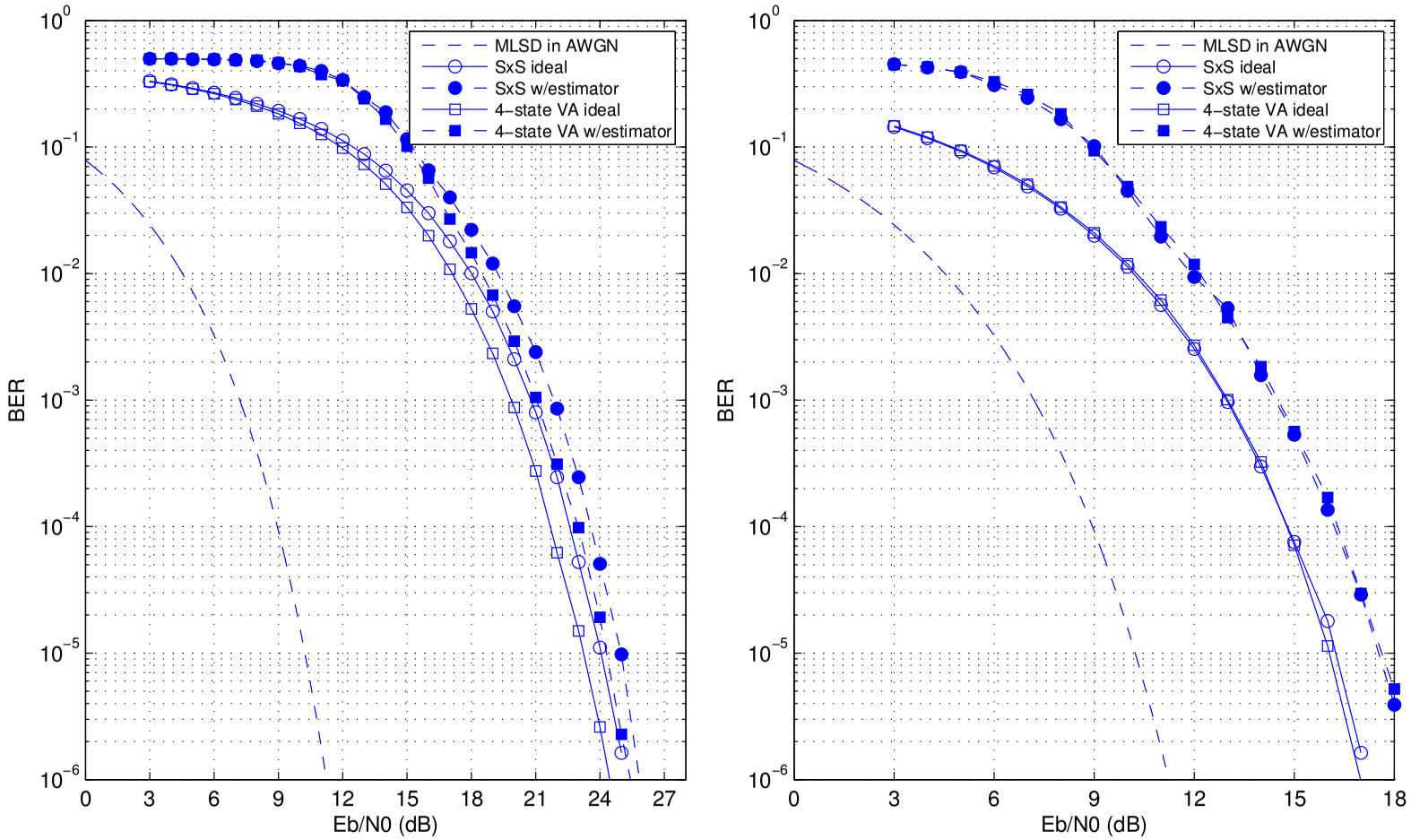

Figure 129: Simulated bit error rate (BER) performance of the ZF equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 5 (left) and test channel 6 (right).

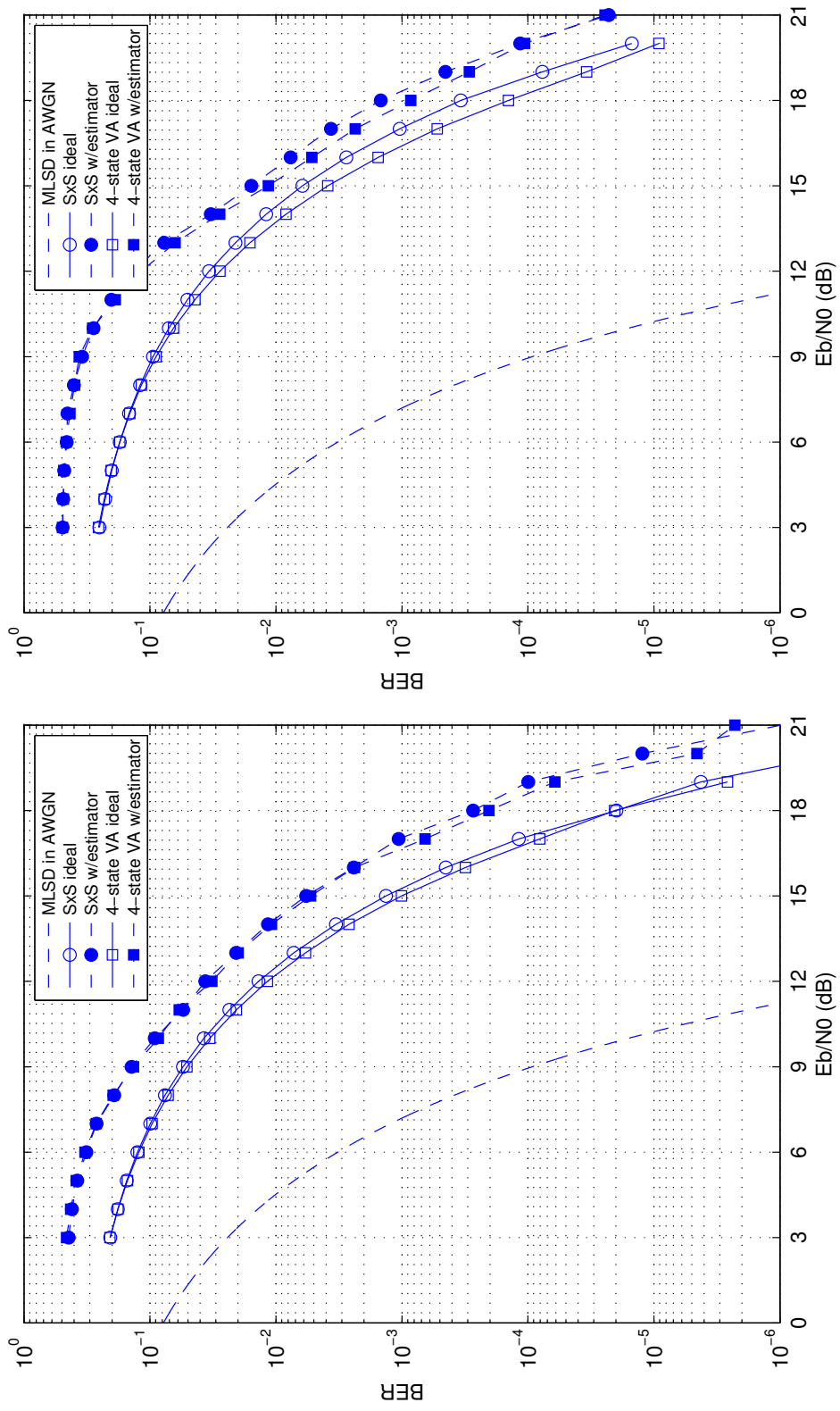

Figure 130: Simulated bit error rate (BER) performance of the ZF equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 7 (left) and test channel 8 (right).

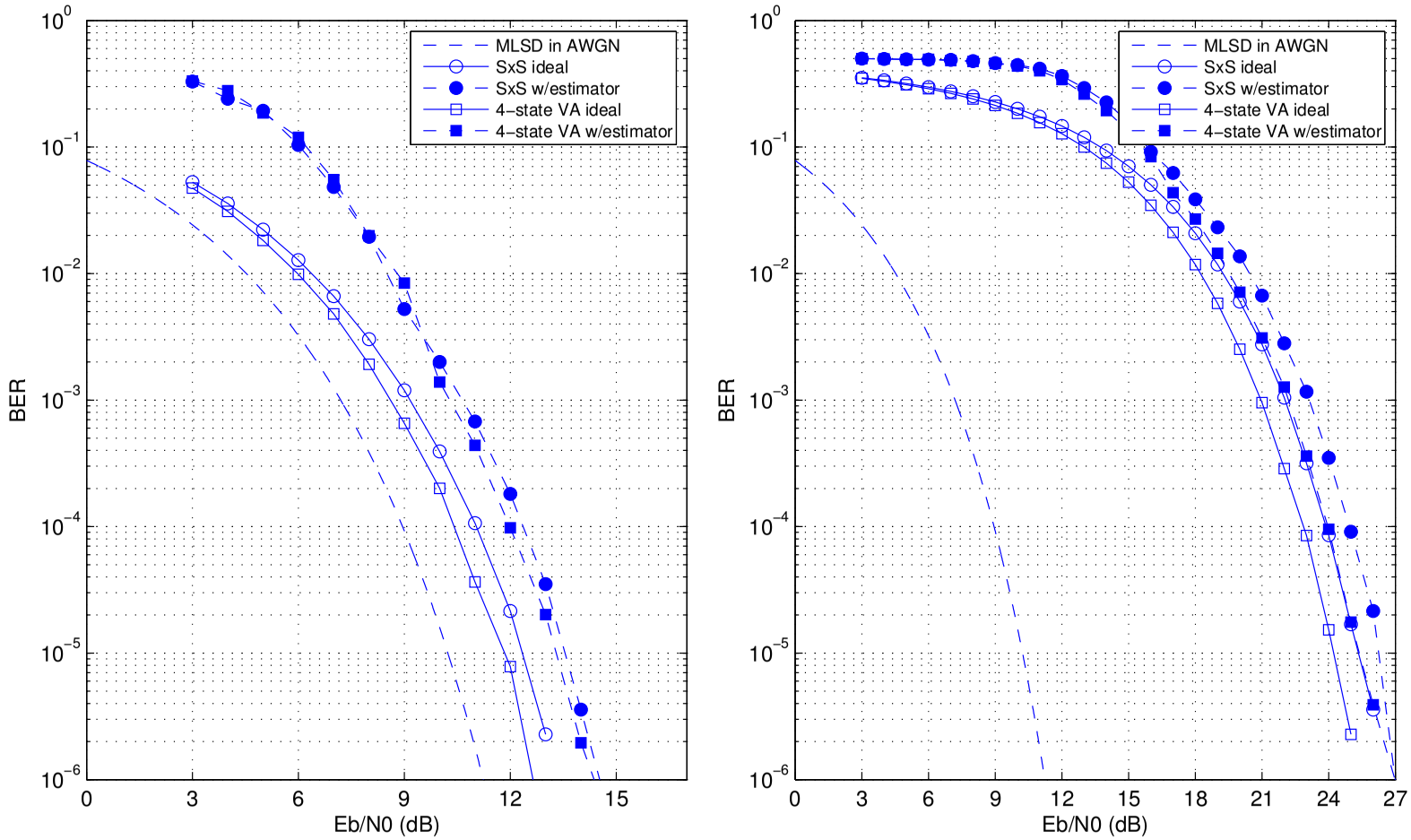

Figure 131: Simulated bit error rate (BER) performance of the ZF equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 9 (left) and test channel 10 (right).

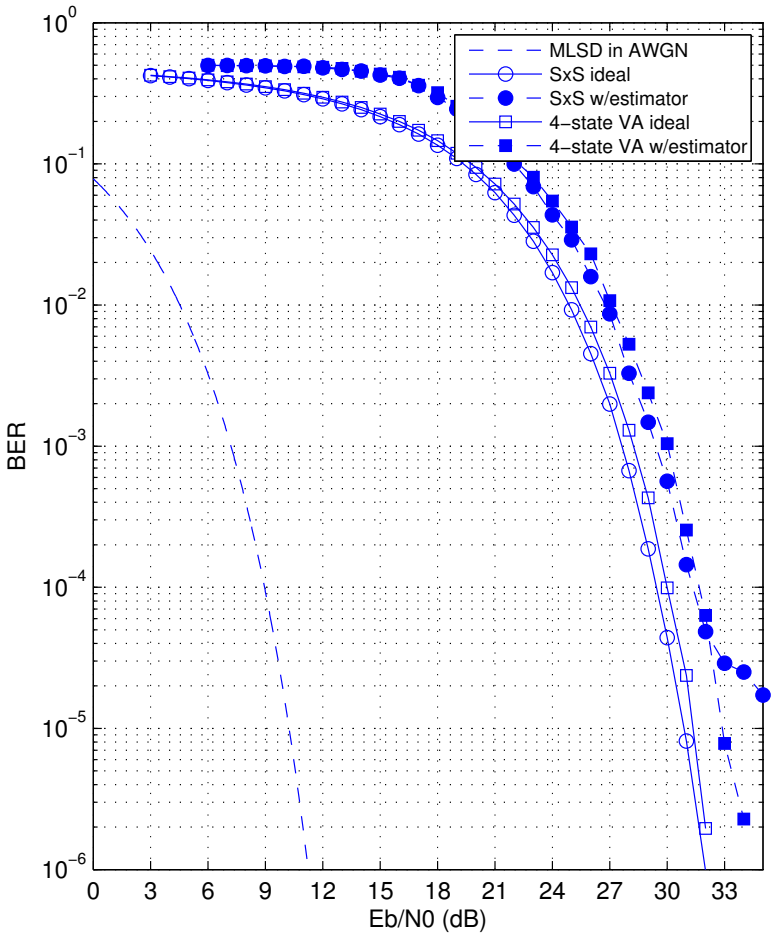

Figure 132: Simulated bit error rate (BER) performance of the ZF equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 11.

8.3 The Minimum Mean-Squared Error (MMSE) Equalizer

The general framework for the MMSE equalizer is the same as that for the ZF equalizer and is illustrated in Figure 126. As before, the equalizer filter coefficients $c(n)$ are assumed to have support on $-L_1 \leq n \leq L_2$ and the channel $h(n)$ is assumed to have support on $-N_1 \leq n \leq N_2$. The output of the equalizer filter is

$$y(n) = r(n) * c(n) = \sum_{k=-L_1}^{L_2} c(k)r(n-k). \tag{314}$$

The MMSE equalizer filter coefficients are selected to minimize the mean squared error

$$\mathcal{E}(n) = \mathrm{E}\left\{ |s(n) - y(n)|^2 \right\}. \tag{315}$$

Clearly, the solution is a function of the second order statistics of $s(n)$, the channel, and the second order statistics of the noise [see (309)]. The challenge with equalizing SOQPSK-TG *samples* is that the underlying continuous-time waveform is not wide-sense stationary [14]. This fact carries over to the samples so that the autocorrelation function of $s(n)$ is of the form

$$R_s(k,\ell) = \frac{1}{2}\mathrm{E}\left\{ s(k)s^*(\ell) \right\}, \tag{316}$$

that is, the autocorrelation function is a function of both sample indexes, not the difference between them. Consequently, the equalizer filter coefficients are a function of the sample index n. It is hard to see how this solution has any practical utility, especially in the presence of a real-time performance requirement. In the end, the designer is left with suboptimal approaches of reduced computational complexity whose accompanying performance penalty is acceptable.

The simplest suboptimal approach is to assume the signal samples are wide-sense stationary. Here, the autocorrelation function is of the form

$$R_s(k-\ell) = \frac{1}{2}\mathrm{E}\left\{ s(k)s^*(\ell) \right\}, \tag{317}$$

that is, the autocorrelation function depends on the difference of the sample time indexes. The wide-sense stationary assumption for $s(n)$ greatly simplifies the solution. Because the optimum equalizer coefficients no longer depend on the samples index n, the mean squared error is of the form

$$\mathcal{E} = \mathrm{E}\left\{ \left| x(n) - y(n) \right|^2 \right\}, \tag{318}$$

that is, the mean squared error does not depend on the sample index n.

The vector of filter coefficients that minimizes the mean squared error is given by

$$\mathbf{c} = \left[\mathbf{G}\mathbf{R}_{s,1}\mathbf{G}^{\dagger} + \mathbf{R}_w\right]^{-1}\mathbf{R}_{s,2}\mathbf{g}^{\dagger} \tag{319}$$

where $\mathbf{c}$ is the $(L_1 + L_2 + 1) \times 1$ vector of equalizer filter coefficients, $\mathbf{G}$ is the $(L_1 + L_2 + 1) \times (N_1 + N_2 + L_1 + L_2 + 1)$ matrix

$$\mathbf{G} = \begin{bmatrix} h(N_2) & \cdots & h(-N_1) & & \\ & h(N_2) & \cdots & h(-N_1) & \\ & & \ddots & & \\ & & & h(N_2) & \cdots \quad h(-N_1) \end{bmatrix}; \tag{320}$$

$\mathbf{R}_{s,1}$ is the $(N_1 + N_2 + L_1 + L_2 + 1) \times (N_1 + N_2 + L_1 + L_2 + 1)$ matrix

$$\mathbf{R}_{s,1} = \begin{bmatrix} R_s(0) & R_s(-1) & \cdots & R_s(-L_1 - L_2 - N_1 - N_2) \\ R_s(1) & R_s(0) & \cdots & R_s(-L_1 - L_2 - N_1 - N_2 + 1) \\ \vdots & & & \vdots \\ R_s(L_1 + L_2 + N_1 + N_2) & R_s(L_1 + L_2 + N_1 + N_2 - 1) & \cdots & R_s(0) \end{bmatrix}; \tag{321}$$

$\mathbf{R}_w$ is the $(L_1 + L_2 + 1) \times (L_1 + L_2 + 1)$ noise autocorrelation matrix given by

$$\mathbf{R}_w = \begin{bmatrix} R_w(0) & \cdots & R_w(-L_1 - L_2) \\ \vdots & & \vdots \\ R_w(L_1 + L_2) & \cdots & R_w(0) \end{bmatrix}; \tag{322}$$

$\mathbf{R}_{s,2}$ is the $(L_1 + L_2 + 1) \times (L_1 + L_2 + 1)$ matrix given by

$$\mathbf{R}_{s,2} = \begin{bmatrix} R_s(0) & R_s(-1) & \cdots & R_s(-L_1 - L_2) \\ R_s(1) & R_s(0) & \cdots & R_s(-L_1 - L_2 + 1) \\ \vdots & & & \vdots \\ R_s(L_1 + L_2) & R_s(L_1 + L_2 - 1) & \cdots & R_s(0) \end{bmatrix}; \tag{323}$$

and $\mathbf{g}$ is the $1 \times (L_1 + L_2 + 1)$ vector given by

$$\mathbf{g} = \begin{bmatrix} h(L_1) & \cdots & h(-L_2) \end{bmatrix} \tag{324}$$

where it is understood that $h(n) = 0$ for $n < -N_1$ or $n > N_2$. (How many zeros need to be prepended and appended depends on the relationship between L_1 and N_2 and the relationship between L_2 and N_1.)

The question is now, what function should be used for the autocorrelation function $R_s(k)$? Two approximations are investigated here. The first is an empirically-derived autocorrelation function. The empirical autocorrelation function is obtained by generating a large number of samples $s(n)$ and using the standard estimation technique assuming wide sense stationarity. Given L samples of $x(n)$ for $n = 0, 1, \ldots, L-1$, this empirical autocorrelation function is

$$R_{\mathrm{e}}(k) = \frac{1}{2(L-k)} \sum_{n=k}^{L-1} s(n)s^*(n-k), \qquad 0 \le k < L-1 \tag{325}$$

together with

$$R_{\mathrm{e}}(k) = R_{\mathrm{e}}^*(-k), \qquad -L < k < 0. \tag{326}$$

A plot of $R_{\mathrm{e}}(k)$ corresponding to $L = 2 \times 10^6$ samples of SOQPSK-TG sampled at 2 samples/bit is shown in Figure 133 for the first 100 lags (i.e., $-100 \le k \le 100$). The top plot shows the real part of $R_{\mathrm{e}}(k)$ and the lower plot shows the imaginary part of $R_{\mathrm{e}}(k)$. The only significant values are those for $-5 \le k \le 5$ and indicated by markers on the plot. Consequently, in the simulation results presented below, we assume $R_{\mathrm{e}}(k) = 0$ for $|k| > 5$.

The second approximation is to assume the data are uncorrelated. This generates a correlation function of the form

$$R_{\mathrm{i}}(k) = \sigma_s^2 \delta(k) \tag{327}$$

where

$$\sigma_s^2 = \frac{1}{2} \mathrm{E}\left\{ |s(t)|^2 \right\}. \tag{328}$$

Here, the corresponding correlation matrices $\mathbf{R}_{s,1}$ and $\mathbf{R}_{s,2}$ become

$$\mathbf{R}_{s,1} = \sigma_s^2 \mathbf{I}_{N_1+N_2+L_1+L_2+1} \qquad \mathbf{R}_{s,1} = \sigma_s^2 \mathbf{I}_{L_1+L_2+1}. \tag{329}$$

The performance of these approximations was evaluated in simulation. To do so, we use test channel 4, illustrated in Figure 7 and assume perfect estimates for the frequency offset [$\omega_0 - \hat{\omega}_0 = 0$ in (308)] and perfect knowledge of the channel. The equalizer output is applied to the symbol-by-symbol detector for SOQPSK-TG shown in Figure 157 (a) in the Appendix. The equalizer coefficients were computed using (319) for $L_1 = 5 \times N_1$ and $L_2 = 5 \times N_2$ and using $R_{\mathrm{e}}(k)$ and $R_{\mathrm{i}}(k)$.

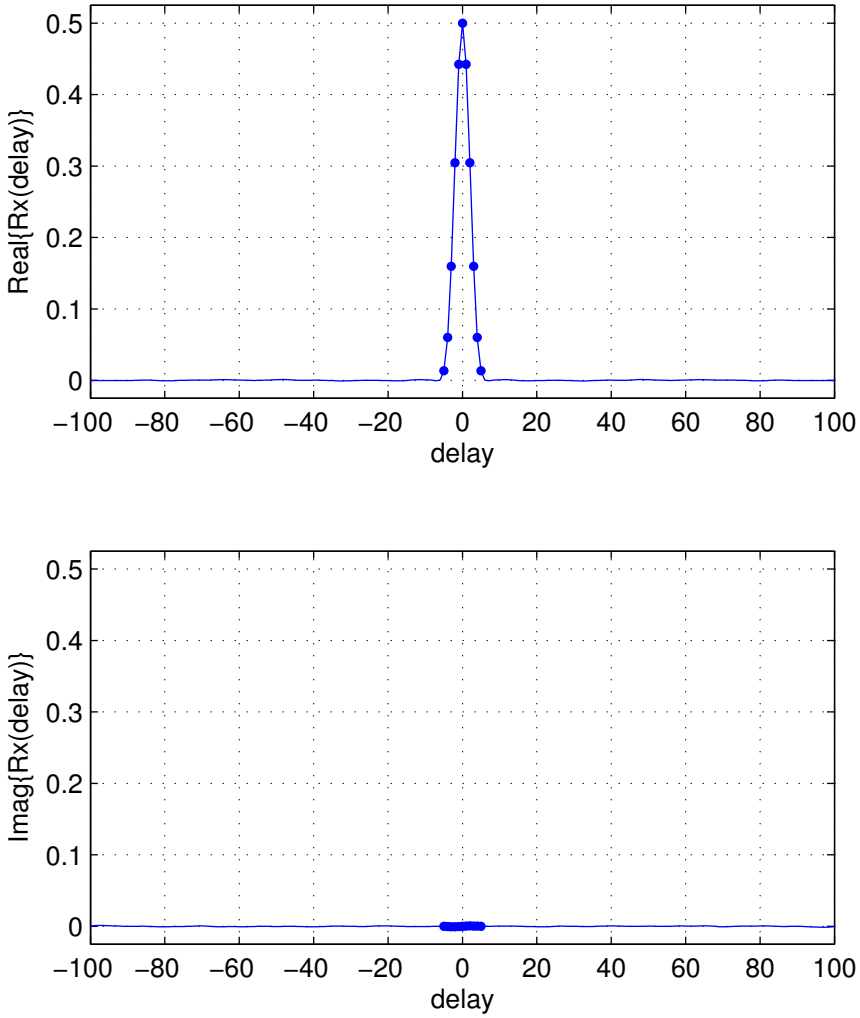

Figure 133: A plot of the empirical autocorrelation function for SOQPSK-TG: (top) the real part of $R_{\mathrm{e}}(k)$; (bottom) the imaginary part of $R_{\mathrm{e}}(k)$. The sample rate for the SOQPSK-TG samples is at 2 samples/bit. Markers indicate the values for $-5 \leq k \leq 5$.

The effectiveness of using $R_{\mathrm{e}}(k)$ and $R_{\mathrm{i}}(k)$ in computing the equalizer coefficients was measured by the simulated bit error rate. The simulated bit error rate is plotted in Figure 134, where we observe that there is essentially no difference in the performance between the two. Consequently, we prefer $R_{\mathrm{i}}(k)$ over $R_{\mathrm{e}}(k)$ in what follows because this choice simplifies the computations of the equalizer filter coefficients. In this case, the optimum equalizer filter coefficients are given by

$$\mathbf{c} = \left[\mathbf{G}\mathbf{G}^{\dagger} + \frac{\sigma_w^2}{\sigma_s^2}\mathbf{I}_{L_1+L_2+1}\right]^{-1}\mathbf{g}^{\dagger} \tag{330}$$

where $\mathbf{I}_{L_1+L_2+1}$ is the $(L_1 + L_2 + 1) \times (L_1 + L_2 + 1)$ identity matrix. For computer simulations, we desire to express the signal-to-noise ratio σ_s^2/σ_w^2 in terms of E_b/N_0. Using the relationship [see (304) in Section 7]

$$\frac{\sigma_s^2}{\sigma_w^2} = \frac{1}{N}\frac{E_b}{N_0}, \tag{331}$$

where N is the number of samples/bit, the vector of MMSE equalizer filter coefficients may be expressed as

$$\mathbf{c} = \left[\mathbf{G}\mathbf{G}^{\dagger} + \frac{N}{(E_b/N_0)}\mathbf{I}_{L_1+L_2+1}\right]^{-1}\mathbf{g}^{\dagger} \tag{332}$$

The simulated bit error rate (BER) of the MMSE equalizer using (330) was performed using the following parameters.

- The SOQPSK-TG signal and channel were sampled at a rate equivalent to 2 samples/bit.
- The MMSE equalizer length was set to approximately 4 times the channel length by using $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$.
- The "ideal" simulations are based on a detector endowed with perfect knowledge of the frequency offset [$\omega_0 - \hat{\omega}_0 = 0$ in (308)], the signal-to-noise ratio E_b/N_0, and the channel. As such the ideal detectors described in the Appendix are used: the ideal detector of Figure 157 (a) is used for the symbol-by-symbol (SxS) detector and the ideal detector of Figure 161 using the trellis of Figure 160 with partial path metric updates (396) 4-state trellis detector.
- The "with estimator" simulations are based on estimates of the frequency offset signal-to-noise ratio, and channel. The frequency offset estimator (55) is used to produce the frequency offset estimate which is used to de-rotate the preamble, ASM, and data samples. Consequently, $\omega_0 - \hat{\omega}_0 \neq 0$ in (308). The de-rotated preamble and ASM are used to estimate the noise variance and the channel impulse response. The noise variance estimate $\hat{\sigma}_w^2$ was

calculated using (290). From the noise variance, the signal-to-noise ratio estimate is [see (298)]

$$\left(\frac{\sigma_s^2}{\sigma_w^2}\right)_{\text{est}} = \frac{1}{2\hat{\sigma}_w^2}. \tag{333}$$

The channel impulse response estimate was calculated using (190) assuming $N_1 = 12$ and $N_2 = 25$. The channel impulse response estimates $\hat{h}(n)$ for $-N_1 \leq n \leq N_2$ are used in place of $h(n)$ in $\mathbf{G}$ and $\mathbf{g}$ given by (320) and (324), respectively. The phase-tracking-enabled detectors described in the Appendix are used here: the detector shown in Figure 157 (b) is used for the symbol-by-symbol (SxS) detector and the detector shown in Figure 161 using the trellis of Figure 160 with partial path metric updates (414) – (416) is used for the 4-state trellis detector.

The simulation results for test channels 1 – 11 are shown in Figures 135 – 140, respectively. In each plot, the clear markers mark the BER performance of the ideal case whereas the filled markers mark the BER performance of the case where estimates replace the ideal values. The circles mark the performance of the SxS detector and the squares mark the performance of the 4-state trellis detector. Three important conclusions are illustrated by the simulation results.

1. The performance differences between the SxS and 4-state trellis detectors for both the ideal and non-ideal cases are in the 0 to 1 dB range. This is consistent with the AWGN results reported in [20, 21].

2. The penalty for using the estimates is 0 to 2 dB for all test channels except test channels 2 and 4. For channels 2 and 4, the penalty is worse. This is because the performance of the estimators in these channels is not as good.

3. The results are very consistent with the ZF equalizer performance summarized in Figures 127–132. The difference here is that the MMSE equalizer is about 2 dB better than the ZF equalizer, but this difference diminishes as E_b/N_0 increases.

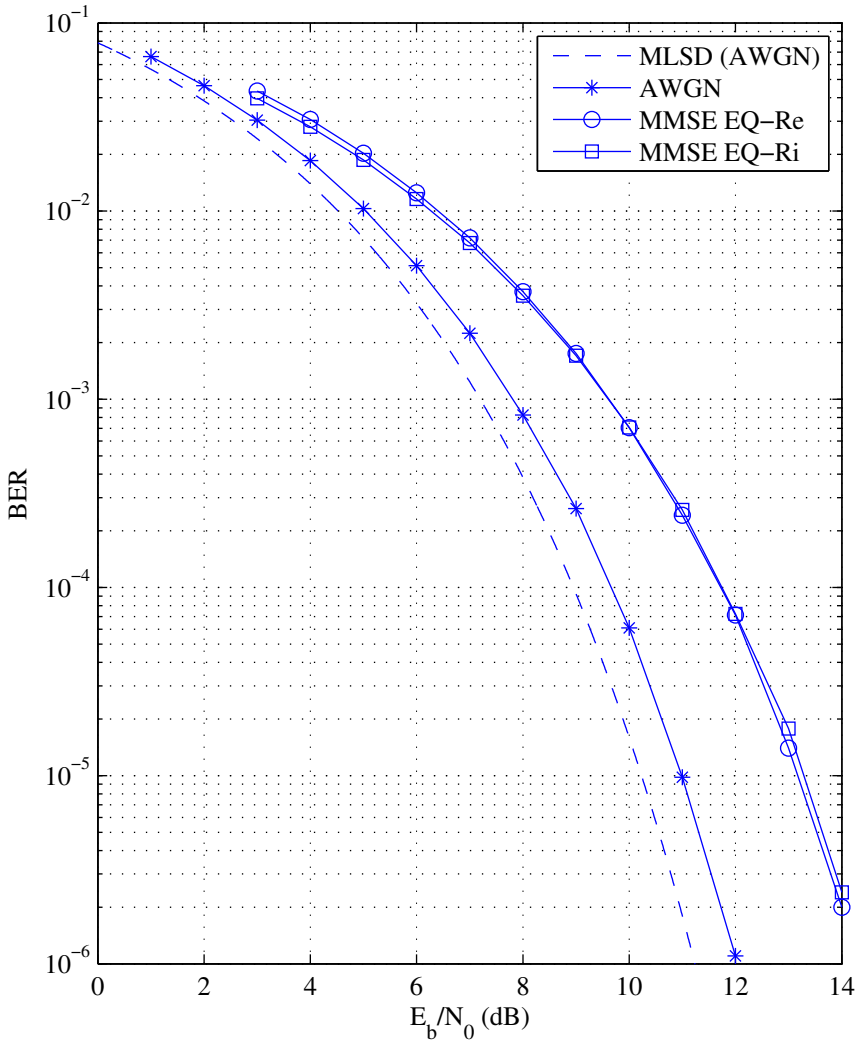

Figure 134: Simulated bit error rate results for the system shown in Figure 126 using the MMSE coefficients (319) and test channel 4. Also included for reference is the performance of the symbol-by-symbol detector in the AWGN channel and the optimum maximum likelihood sequence detector (MLSD) in the AWGN channel (see [21]).

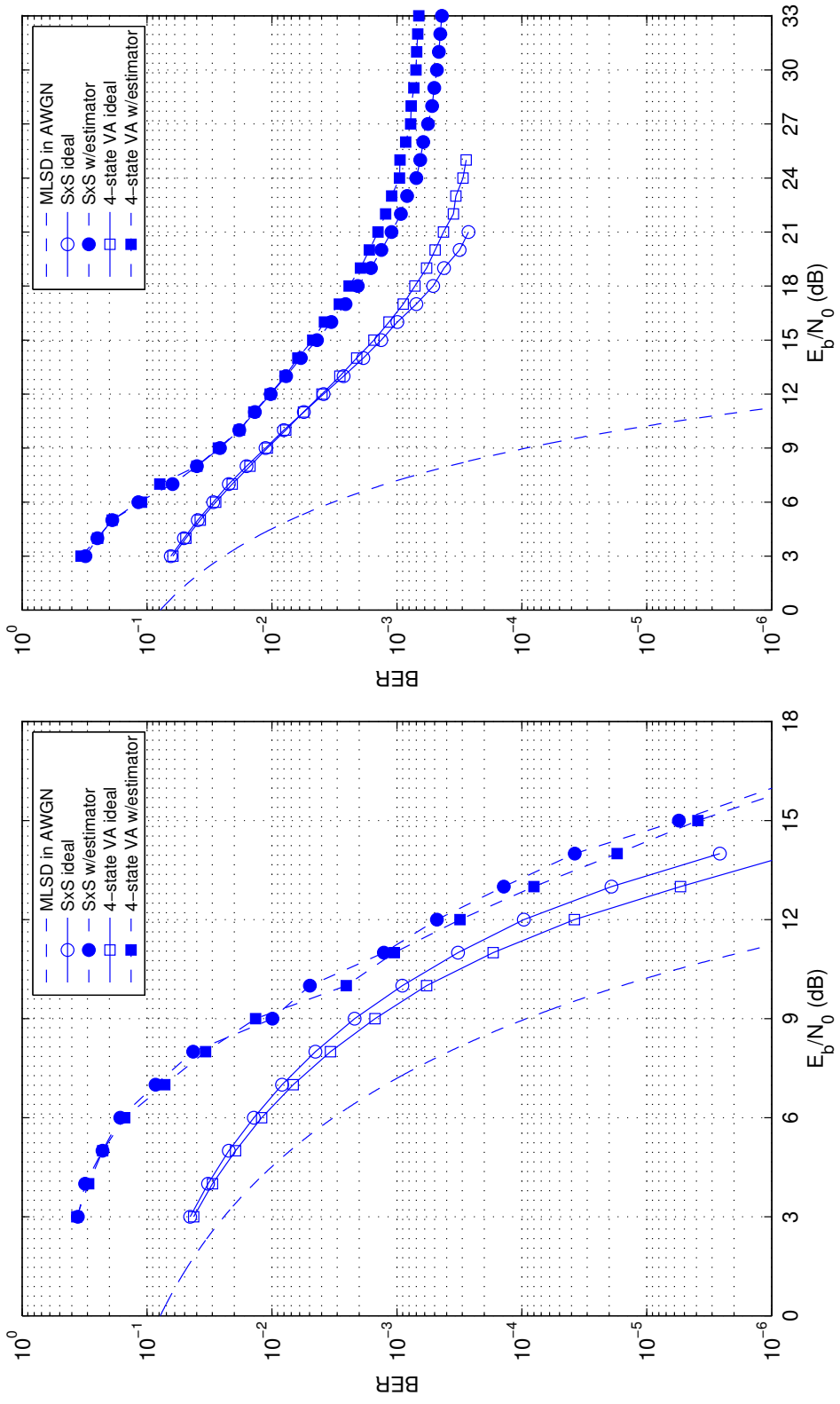

Figure 135: Simulated bit error rate (BER) performance of the MMSE equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 1 (left) and test channel 2 (right).

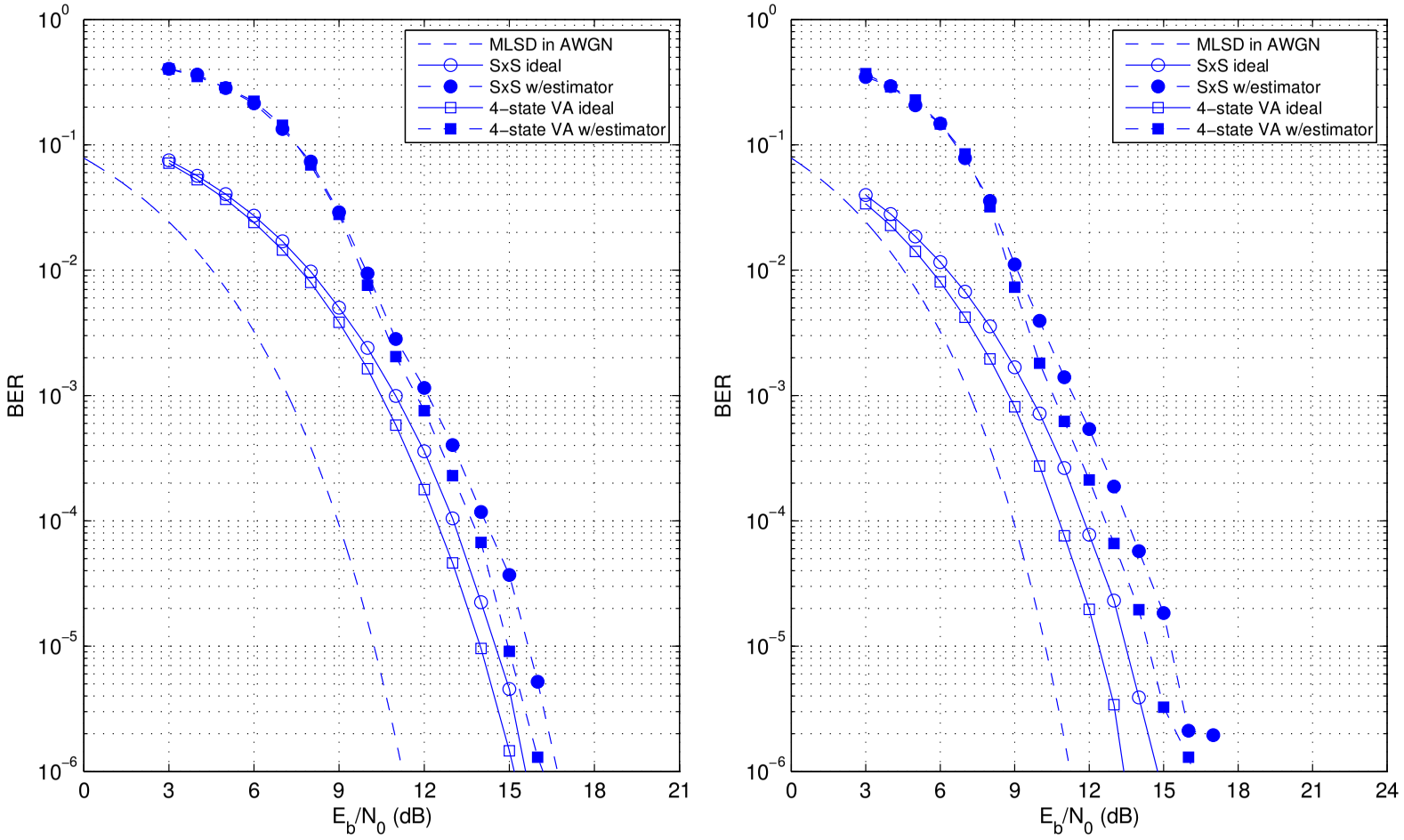

Figure 136: Simulated bit error rate (BER) performance of the MMSE equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 3 (left) and test channel 4 (right).

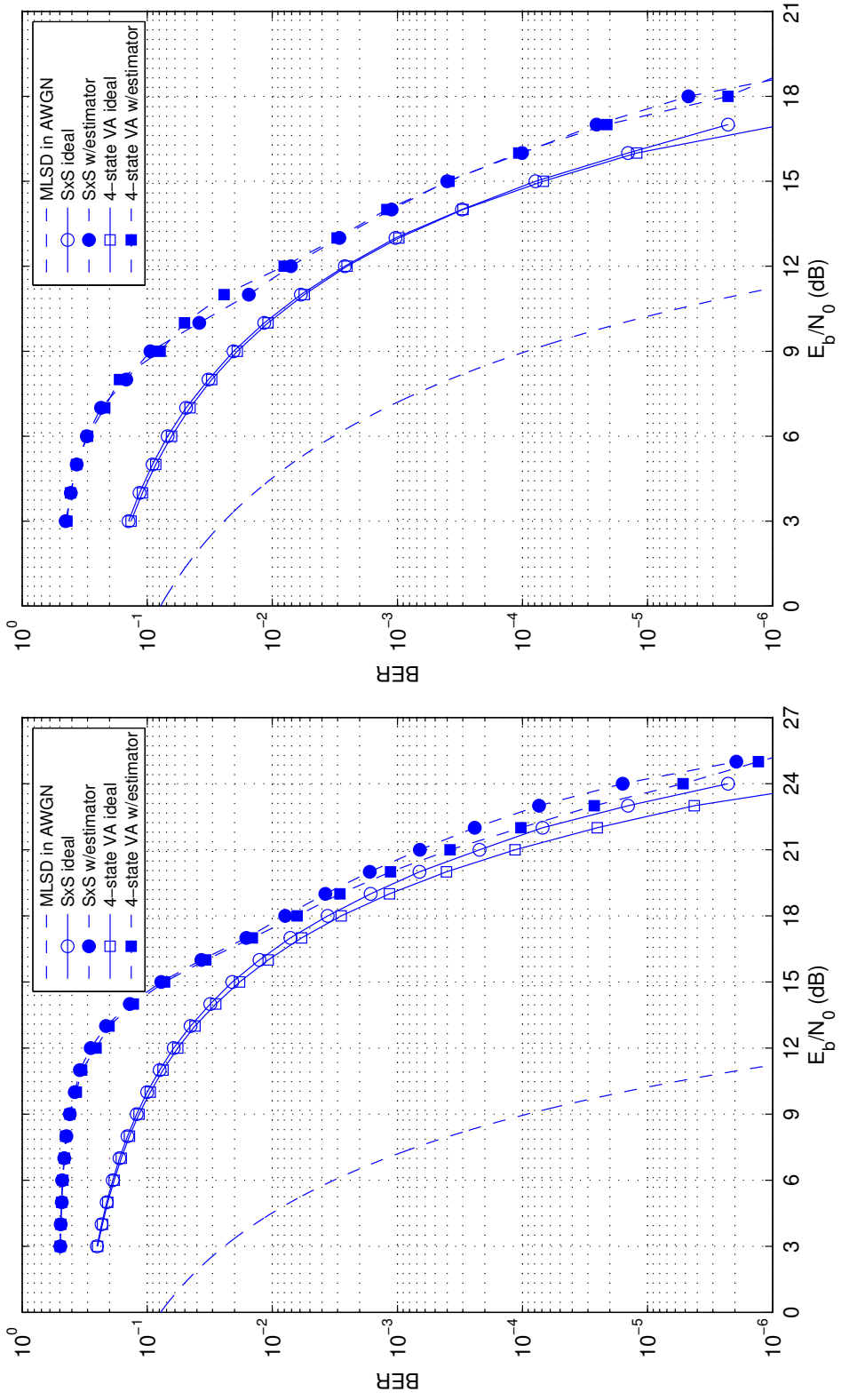

Figure 137: Simulated bit error rate (BER) performance of the MMSE equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 5 (left) and test channel 6 (right).

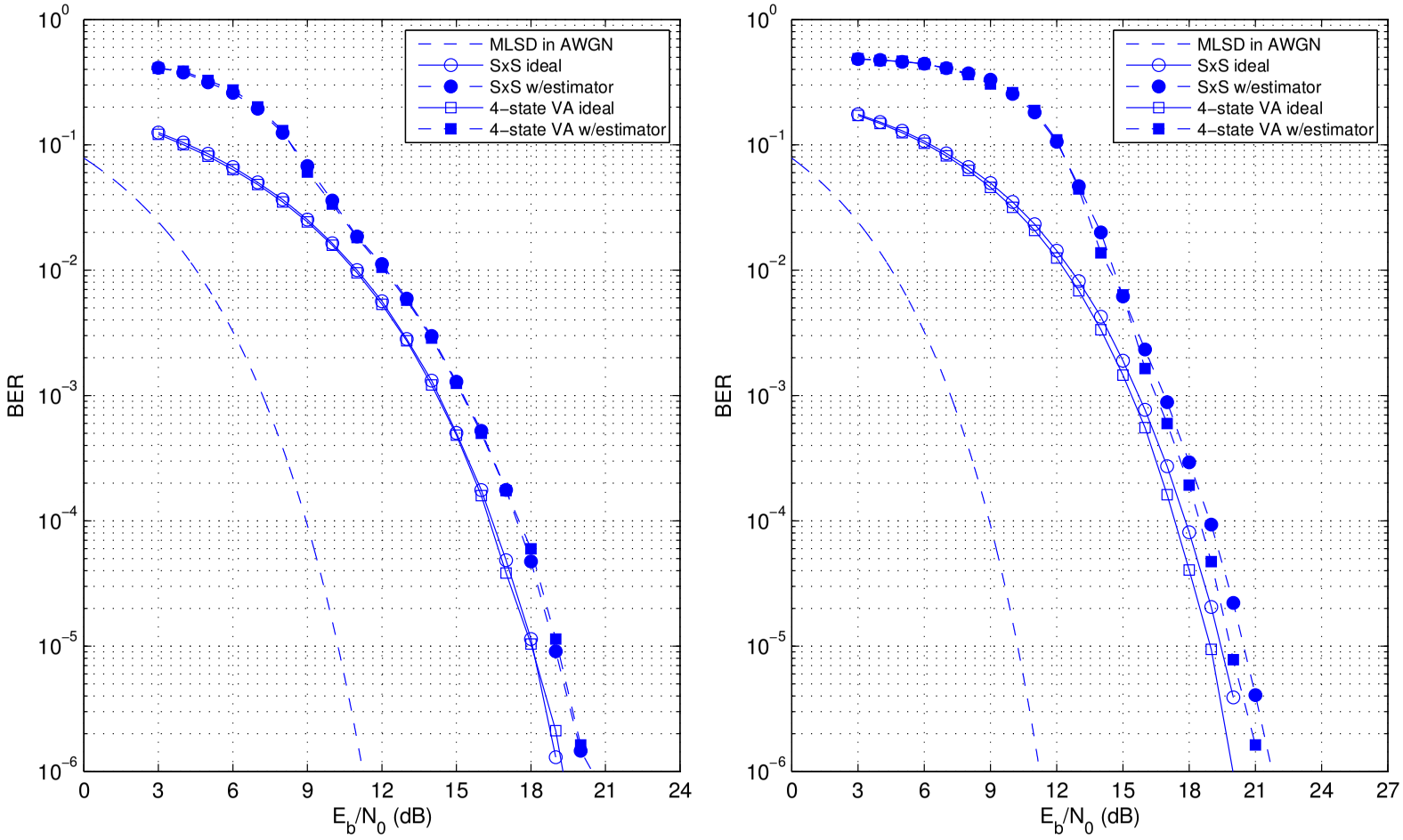

Figure 138: Simulated bit error rate (BER) performance of the MMSE equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 7 (left) and test channel 8 (right).

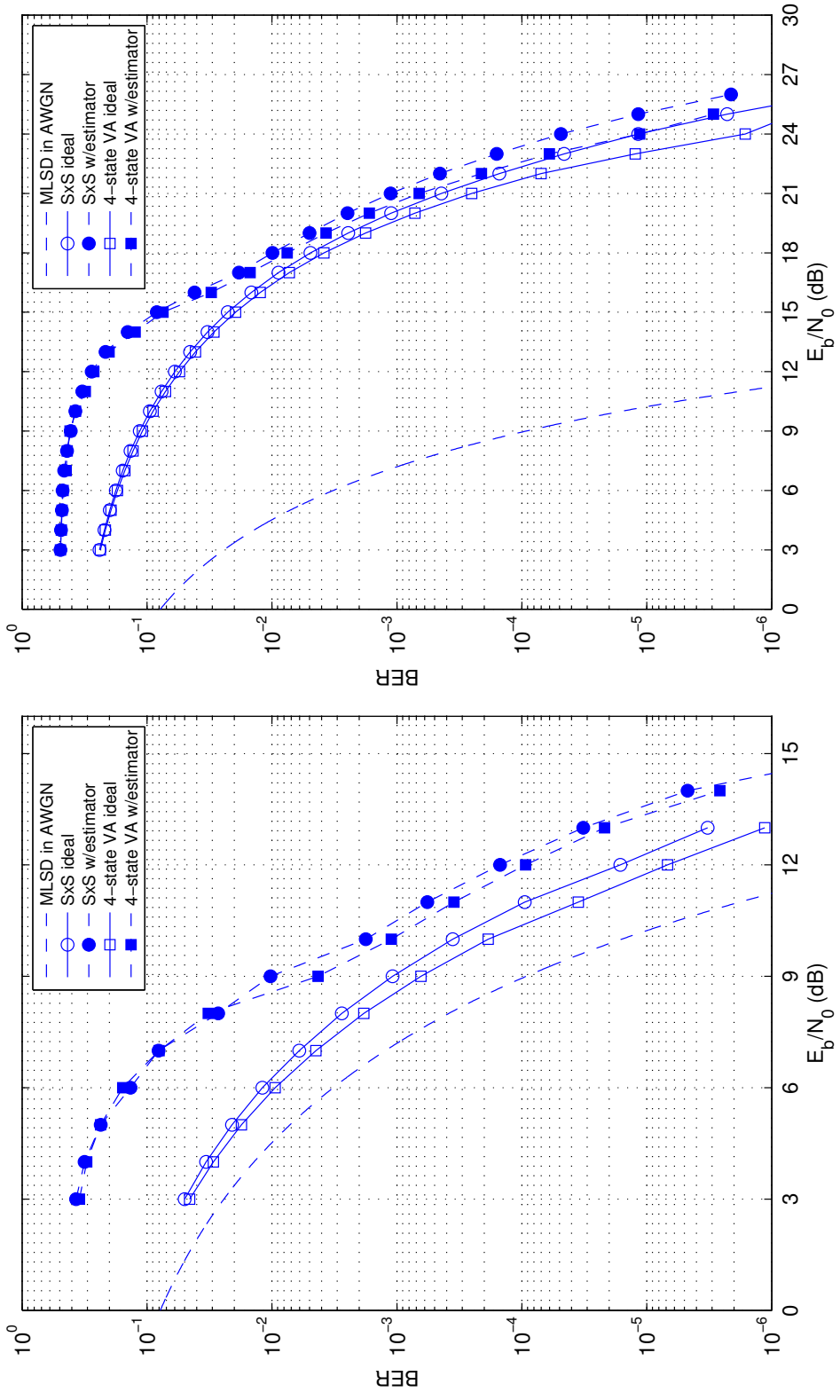

Figure 139: Simulated bit error rate (BER) performance of the MMSE equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 9 (left) and test channel 10 (right).

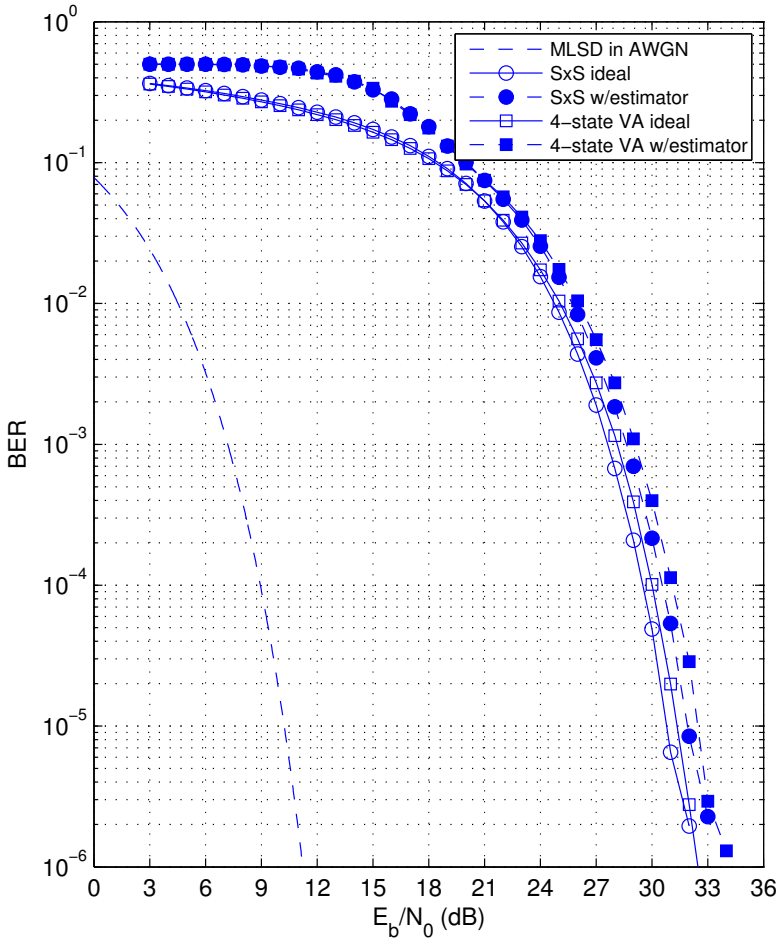

Figure 140: Simulated bit error rate (BER) performance of the MMSE equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 11.

8.4 CMA + AMA Equalizer

8.4.1 Background/Overview

The constant modulus algorithm (CMA) is the most popular adaptive blind equalizer used today, because of its relative simplicity and its good global convergence properties [22]. It is very effective in equalizing signals which are of constant modulus. However, for non-constant modulus signals, CMA equalization can suffer from a considerable amount of residual ISI, and produce non-minimal convergence. In [23], Barbarossa and Scaglione described an alphabet-matched algorithm (AMA), which is able to provide better equalization of non-constant modulus signals. The AMA cost function tries to force the equalizer output to belong to a signal constellation of interest, and it is shown in that this method is able to lower the residual ISI and improve the convergence rate over that of CMA alone. But, AMA requires a good initialization. In previous work [23], the initialization was performed using the the CMA equalizer.

In previous work [24], Beasley and Cole-Rhodes developed and evaluated the performance of a blind adaptive channel equalization scheme which is based on a single cost function created by combining the CMA and AMA algorithms. This combined cost function takes into account both the amplitude and the phase of the equalizer output, which allows for more efficient equalization of signal constellation types such as QPSK and QAM, and after final equalization the only phase ambiguity in the output is some multiple of $\pi/2$. Because SOQPSK-TG is a CPM (see the Appendix), the SOQPSK-TG samples possess the constant modulus property (i.e., the samples of the complex-valued baseband equivalent signal lie on a circle in the complex plane). Consequently we expect that SOQPSK-TG will be well-suited for the CMA equalizer. On the other hand, the AMA equalizer is expected to be effective on the recovered quaternary data symbols (associated with the OQPSK interpretation of SOQPSK-TG — see the Appendix).

8.4.2 Description

The CMA+AMA equalization algorithm investigated in this project is outlined by the block diagram in Figure 141. The equalizer is integrated into a detector based on the OQPSK interpretation of SOQPSK-TG described in the Appendix. This version assumes perfect frequency offset synchronization (cf., Figure 142).

The derotated samples $r(n)$ corresponding to the $N_b = 6144$ data bits in the b-th packet (or "block")[7] are filtered by the equalizer filter with impulse response $c_b(n)$ for $-L_1 \leq n \leq L_2$. The

[7]Here, we use a simplified notation to make the description easier to read. The samples of b-th block are $r(0), r(1), \ldots$. The samples of the $b+1$-st block are also $r(0), r(1), \ldots$. Thus, the index n is a relative position

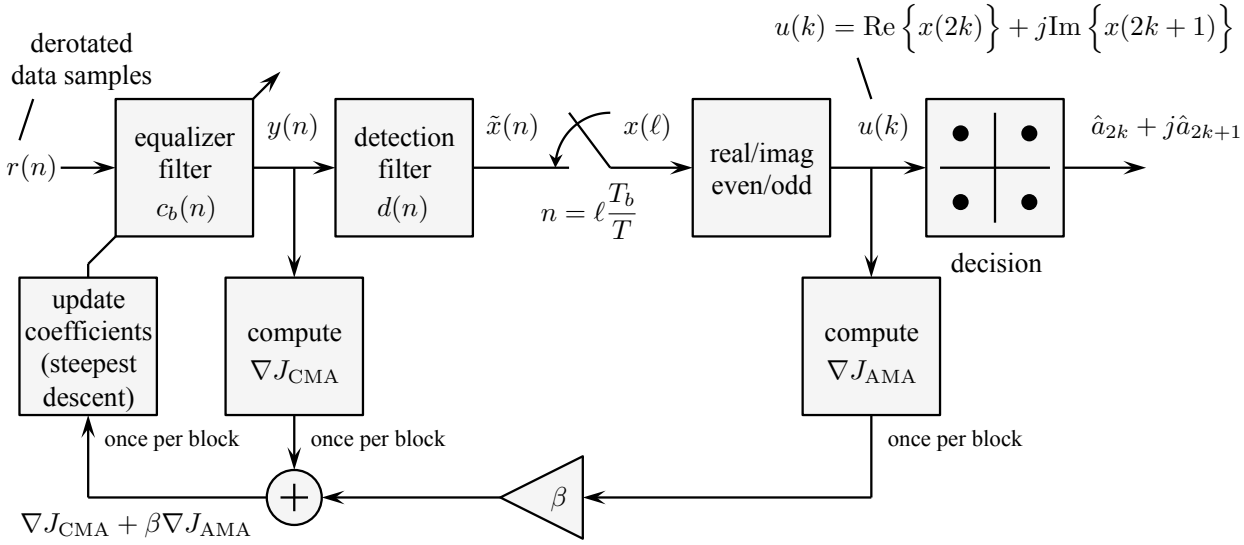

Figure 141: A block diagram of the adaptive CMA+AMA equalizer.

subscript b is the "block index," that is, these are the equalizer filter coefficients used during the "block" b. The equalizer filter coefficients are updated once per block as described below. The equalizer filter output is

$$y(n) = \sum_{m_1=-L_1}^{L_2} c_b(m_1) r(n - m_1). \tag{334}$$

The equalizer filter outputs are filtered by the detection filter with impulse response $d(n)$ for $-L_d \leq n \leq L_d$. The detection filter is described in the Appendix. The detection filter output is

$$\tilde{x}(n) = \sum_{m_2=-L_d}^{L_d} d(m_2) y(n - m_2). \tag{335}$$

The detection filter output is down-sampled by $N = T_b/T$ ($N = 2$ in our implementation) to produce

$$x(\ell) = \tilde{x}(N\ell). \tag{336}$$

The decision variable $u(k)$ is formed by pairing the real part of even-indexed samples with the imaginary part of odd-indexed samples. Using the notation

$$x(\ell) = x_R(\ell) + j x_I(\ell), \tag{337}$$

index: it gives the position of the sample in the b-th block. Because there is no need to relate the samples *within* a block to the block index itself, we use the simpler notation.

the decision variable may be expressed as

$$u(k) = x_R(2k) + jx_I(2k+1). \tag{338}$$

Just to be clear, the relationship between the decision variables and the downsampled detection filter outputs is

$$\begin{aligned} u(0) &= x_R(0) + jx_I(1) \\ u(1) &= x_R(2) + jx_I(3) \\ u(2) &= x_R(4) + jx_I(5) \\ &\vdots \\ u(N_s-1) &= x_R(2N_s-2) + jx_I(2N_s-1) \end{aligned} \tag{339}$$

where $N_s = {}^{6144}/_2 = 3072$ is the number of quaternary data symbols in a packet and $N_b = 6144$ is the number of data bits in a packet.

Finally, the decisions are made in the obvious way:

$$\hat{a}_{2k} = \text{sign}\left\{x_R(2k)\right\} \qquad \hat{a}_{2k+1} = \text{sign}\left\{x_I(2k+1)\right\}. \tag{340}$$

As mentioned above, this equalizer is based on an adaptive filter. The adaptation minimizes the cost function

$$J = J_{\text{CMA}} + \beta J_{\text{AMA}} \tag{341}$$

for a weighting constant $\beta \geq 0$ where J_{CMA} is the CMA cost function and J_{AMA} is the AMA cost function.

The CMA cost function is

$$J_{\text{CMA}}\left(y(n)\right) = \text{E}\left\{\left(|y(n)|^2 - R_2\right)^2\right\} \tag{342}$$

where

$$R_2 = \frac{\text{E}\left\{|s(n)|^4\right\}}{\text{E}\left\{|s(n)|^2\right\}}. \tag{343}$$

This cost function attempts to restore the shape of the signal by taking into account the squared error between the modulus of the equalizer output and the closest point on a circle with radius R_2. CMA does not take the signal phase into consideration.

The AMA cost function is

$$J_{\text{AMA}}\left(u(k)\right) = \text{E}\left\{1 - \sum_{a\in\mathcal{C}} \exp\left[-\frac{1}{2\sigma^2}\left|u(k)-a\right|^2\right]\right\} \tag{344}$$

where $\mathcal{C}$ is the set of constellation points

$$\mathcal{C} = \Big\{ -A - jA,\ -A + jA,\ +A - jA,\ +A + jA \Big\}. \tag{345}$$

The AMA cost function attempts to restore the shape of the constellation by considering the distance between the equalizer output symbol and each of the known constellation symbols, and it assigns an appropriate penalty based on this distance. When an equalized symbol is sufficiently close to a constellation point the cost function is minimized. The parameter σ is used to control the width of the nulls placed around each constellation point. Its value is chosen so that these nulls do not overlap for adjacent constellation points.

The combined cost function (341) is motivated by the factors outlined in the introduction. The minimizing J_{CMA} produces a signal whose modulus corresponds to a circle of radius R_2 in the I/Q plane. But this occurs with arbitrary phase rotation because J_{CMA} is rotationally invariant. The role of the AMA cost function is to force the phase of the equalizer output to produce decision variables corresponding to the constellation points. This is accomplished with a phase ambiguity associated with the constellation: $^{pi}/_{2}$ with no timing synchronization and π with timing synchronization [25].

The adaptation is applied on a block-by-block basis. Here "block" refers to the samples corresponding to an iNET packet (see Figure 1). The adaptation uses the steepest decent algorithm to find the filter coefficients that minimize the cost function (341). The adaptation is of the form

$$\mathbf{c}_{b+1} = \mathbf{c}_b - \mu\nabla J \tag{346}$$

where

$$\mathbf{c}_{b+1} = \begin{bmatrix} c_{b+1}(-L_1) \\ \vdots \\ c_{b+1}(0) \\ \vdots \\ c_{b+1}(L_2) \end{bmatrix} \qquad \mathbf{c}_{b} = \begin{bmatrix} c_{b}(-L_1) \\ \vdots \\ c_{b}(0) \\ \vdots \\ c_{b}(L_2) \end{bmatrix} \tag{347}$$

and

$$\nabla J = \begin{bmatrix} \dfrac{\partial J}{\partial c_b^*(-L_1)} \\ \vdots \\ \dfrac{\partial J}{\partial c_b^*(0)} \\ \vdots \\ \dfrac{\partial J}{\partial c_b^*(L_2)} \end{bmatrix}. \tag{348}$$

The $(L_1 + L_2 + 1) \times 1$ gradient vector ∇J "points" in the direction of increasing J. The update (346) adjusts the filter coefficients in the $(L_1 + L + 2 + 1) \times 1$ filter coefficient vector $\mathbf{c}_b$ in the direction of *decreasing* J (hence, the negative sign). For this reason, the iterative approach to function minimization described by (346) is called *steepest descent* [15].

Because $J = J_{\text{CMA}} + \beta J_{\text{AMA}}$, $\nabla J = \nabla J_{\text{CMA}} + \beta \nabla J_{\text{AMA}}$. Starting with ∇J_{CMA}, we have

$$\begin{aligned} \frac{\partial J_{\text{CMA}}}{\partial c_b^*(i)} &= \frac{\partial}{\partial c_b^*(i)} \text{E}\left\{ \Big[y(n)y^*(n) - R_2 \Big]^2 \right\} \\ &= \text{E}\left\{ 2 \Big[y(n)y^*(n) - R_2 \Big] y(n) \frac{\partial}{\partial c_b^*(i)} y^*(n) \right\} \\ &= \text{E}\left\{ 2 \Big[y(n)y^*(n) - R_2 \Big] y(n) \frac{\partial}{\partial c_b^*(i)} \sum_{m_1=-L_1}^{L_2} c_b^*(m_1) r^*(n - m_1) \right\} \\ &= \text{E}\left\{ 2 \Big[y(n)y^*(n) - R_2 \Big] y(n) r^*(n - i) \right\} \end{aligned} \tag{349}$$

for $-L_1 \leq i \leq L_2$. Putting this into vector form we have

$$\nabla J_{\text{CMA}} = \text{E}\left\{ 2 \Big[y(n)y^*(n) - R_2 \Big] y(n) \mathbf{r}^*(n) \right\} \tag{350}$$

where

$$\mathbf{r}^*(n) = \begin{bmatrix} r^*(n + L_1) \\ \vdots \\ r^*(n) \\ \vdots \\ r^*(n - L_2) \end{bmatrix}. \tag{351}$$

For the case where the sample rate is $N = 2$ samples/bit, (350) and (351) are defined for $0 \leq n \leq 2N_b - 1$ where $N_b = 6144$ is the number of data bits in a packet. Consequently, the expectation in

(350) may be approximated by the sample mean so that

$$\nabla J_{\text{CMA}} \approx \frac{2}{2N_b} \sum_{n=0}^{2N_b-1} \Big[y(n)y^*(n) - R_2 \Big] \, y(n)\mathbf{r}^*(n). \tag{352}$$

Now moving to ∇J_{AMA}, the presence of the detection filter and the downsample operation make this one a little more difficult to calculate. Starting as before,

$$\begin{aligned} \frac{\partial J_{\text{AMA}}}{\partial c_b^*(i)} &= \frac{\partial}{\partial c_b^*(i)} \text{E} \left\{ 1 - \sum_{a \in \mathcal{C}} \exp \left[-\frac{1}{2\sigma^2} \left| u(k) - a \right|^2 \right] \right\} \\ &= \text{E} \left\{ \sum_{a \in \mathcal{C}} \exp \left[-\frac{1}{2\sigma^2} \left| u(k) - a \right|^2 \right] \frac{1}{2\sigma^2} \Big[u(k) - a \Big] \frac{\partial}{\partial c_b^*(i)} u^*(k) \right\}. \end{aligned} \tag{353}$$

Because $u(k) = x_R(2k) + jx_I(2k+1)$, the partial derivative in (353) requires

$$\frac{\partial}{\partial c_b^*(i)} \text{Re} \Big\{ x^*(2k) \Big\} \qquad \text{and} \qquad \frac{\partial}{\partial c_b^*(i)} \text{Im} \Big\{ x^*(2k+1) \Big\} \tag{354}$$

The first partial derivative in (354) is

$$\begin{aligned} \frac{\partial}{\partial c_b^*(i)} \text{Re} \Big\{ x^*(2k) \Big\} &= \text{Re} \left\{ \frac{\partial}{\partial c_b^*(i)} x^*(2k) \right\} \\ &= \text{Re} \left\{ \frac{\partial}{\partial c_b^*(i)} \tilde{x}^*(4k) \right\} \\ &= \text{Re} \left\{ \frac{\partial}{\partial c_b^*(i)} \sum_{m_2=-L_d}^{L_d} d(m_2) y^*(4k - m_2) \right\} \\ &= \text{Re} \left\{ \frac{\partial}{\partial c_b^*(i)} \sum_{m_2=-L_d}^{L_d} d(m_2) \sum_{m_1=-L_1}^{L_2} c_b^*(m_1) r^*(4k - m_2 - m_1) \right\} \\ &= \text{Re} \left\{ \sum_{m_2=-L_d}^{L_d} d(m_2) r^*(4k - m_2 - i) \right\} \\ &= \sum_{m_2=-L_d}^{L_d} d(m_2) r_R(4k - m_2 - i) \end{aligned} \tag{355}$$

for $-L_1 \leq i \leq L_2$ and where we have used the notation $r(n) = r_R(n) + jr_I(n)$. The vector of

partial derivatives may be expressed as

$$\begin{bmatrix} \dfrac{\partial x_R(2k)}{\partial c_b^*(-L_1)} \\ \vdots \\ \dfrac{\partial x_R(2k)}{\partial c_b^*(0)} \\ \vdots \\ \dfrac{\partial x_R(2k)}{\partial c_b^*(L_2)} \end{bmatrix} = \begin{bmatrix} d(-L_d) & \cdots & d(L_d) & & & & \\ & \ddots & & & & & \\ & & d(-L_d) & \cdots & d(L_d) & & \\ & & & \ddots & & & \\ & & & & d(-L_d) & \cdots & d(L_d) \end{bmatrix} \begin{bmatrix} r_R(4k+L_d+L_1) \\ \vdots \\ r_R(4k) \\ \vdots \\ r_R(4k-L_d-L_2) \end{bmatrix}. \tag{356}$$

This equation may be expressed as

$$\nabla x_R(2k) = \mathbf{D}\mathbf{r}_R(4k) \tag{357}$$

where $\mathbf{D}$ is the $(L_1+L_2+1) \times (2L_d+L_1+L_2)$ matrix defined in (356) and $\mathbf{r}_R(4k)$ is the $(2L_d+L_1+L_2) \times 1$ vector defined in (356).

The second partial derivative in (354) is

$$\begin{aligned} \frac{\partial}{\partial c_b^*(i)} \text{Im}\left\{x^*(2k+1)\right\} &= \text{Im}\left\{\frac{\partial}{\partial c_b^*(i)} x^*(2k+1)\right\} \\ &= \text{Im}\left\{\frac{\partial}{\partial c_b^*(i)} \tilde{x}^*(4k+2)\right\} \\ &= \text{Im}\left\{\frac{\partial}{\partial c_b^*(i)} \sum_{m_2=-L_d}^{L_d} d(m_2) y^*(4k+2-m_2)\right\} \\ &= \text{Im}\left\{\frac{\partial}{\partial c_b^*(i)} \sum_{m_2=-L_d}^{L_d} d(m_2) \sum_{m_1=-L_1}^{L_2} c_b^*(m_1) r^*(4k+2-m_2-m_1)\right\} \\ &= \text{Im}\left\{\sum_{m_2=-L_d}^{L_d} d(m_2) r^*(4k+2-m_2-i)\right\} \\ &= -\sum_{m_2=-L_d}^{L_d} d(m_2) r_I(4k+2-m_2-i) \end{aligned} \tag{358}$$

for $-L_1 \le i \le L_2$ and where again we have used the notation $r(n) = r_R(n) + jr_I(n)$. The vector

of partial derivatives may be expressed as

$$\begin{bmatrix} \dfrac{\partial x_I(2k+1)}{\partial c_b^*(-L_1)} \\ \vdots \\ \dfrac{\partial x_I(2k+1)}{\partial c_b^*(0)} \\ \vdots \\ \dfrac{\partial x_I(2k+1)}{\partial c_b^*(L_2)} \end{bmatrix} = - \begin{bmatrix} d(-L_d) & \cdots & d(L_d) & & & & \\ & \ddots & & & & & \\ & & d(-L_d) & \cdots & d(L_d) & & \\ & & & \ddots & & & \\ & & & & d(-L_d) & \cdots & d(L_d) \end{bmatrix} \begin{bmatrix} r_I(4k+2+L_d+L_1) \\ \vdots \\ r_I(4k+2) \\ \vdots \\ r_I(4k+2-L_d-L_2) \end{bmatrix}. \tag{359}$$

This equation may be expressed as

$$\nabla x_I(2k+1) = -\mathbf{D}\mathbf{r}_I(4k+2) \tag{360}$$

where $\mathbf{D}$ is the $(L_1+L_2+1) \times (2L_d+L_1+L_2)$ matrix defined in (359) and $\mathbf{r}_I(4k+2)$ is the $(2L_d+L_1+L_2) \times 1$ vector defined in (359).

Substituting the matrix-vector forms (356) and (359) into (353) and "vectorizing" gives

$$\nabla J_{\text{AMA}} = \mathrm{E}\left\{ \sum_{a \in \mathcal{C}} \exp\left[-\frac{1}{2\sigma^2} \left| u(k) - a \right|^2 \right] \frac{1}{2\sigma^2} \Big[u(k) - a \Big] \mathbf{D} \Big(\mathbf{r}_R(4k) - j\mathbf{r}_I(4k+2) \Big) \right\}. \tag{361}$$

Because this equation is defined for $0 \leq k \leq N_s - 1$, the sample mean may be used to approximate the expectation:

$$\nabla J_{\text{AMA}} \approx \frac{1}{N_s} \frac{1}{2\sigma^2} \sum_{k=0}^{N_s-1} \sum_{a \in \mathcal{C}} \exp\left[-\frac{1}{2\sigma^2} \left| u(k) - a \right|^2 \right] \Big[u(k) - a \Big] \mathbf{D} \Big(\mathbf{r}_R(4k) - j\mathbf{r}_I(4k+2) \Big) \tag{362}$$

In summary, the equalizer filter coefficients update once per block following the steepest descent update equation

$$\mathbf{c}_{b+1} = \mathbf{c}_b - \mu \Big(\nabla J_{\text{CMA}} + \beta \nabla J_{\text{AMA}} \Big) \tag{363}$$

where ∇J_{CMA} is given by (352), ∇J_{AMA} is given by (362), and $\mu > 0$ is the data-adaptive step size.

The system outlined above corresponds to the block diagram shown in Figure 141 and assumes perfect carrier frequency synchronization. For the case where the carrier frequency offset is not perfectly known, a small residual frequency offset imparts a linearly increasing (or decreasing) phase shift on the data. It is possible to "track the phase shift out" by incorporating a first order

phase lock loop (PLL) into the equalizer structure. This system is shown in Figure 142. Here, the only change to the equalization algorithms is that the constellation point estimates $u(k)$ that form the input to the AMA filter are derived from the rotated downsampled detection filter outputs. That is, $x_r(\ell)$ replaces $x(\ell)$ in (338).

8.4.3 Discussion and Simulation Results

The motivation for the combined CMA+AMA equalizer outlined in Figures 141 and 142 is the performance of the CMA equalizer operating alone. The CMA cost function (342) is rotationally invariant; no phase information is used by the cost function in directing the adaptation. Consequently, the equalizer may converge with an arbitrary phase shift applied to the signal. This is especially true when the adaptive filter coefficients are initialized to all zeros or "center tap" initialized. While it is true that a PLL track this arbitrary phase shift out, the acquisition and lock time required by the PLL is undesirable.

The equalizer based on the AMA cost function is designed to address this issue. Because the AMA cost function forces the equalizer place the quaternary decision variables $u(k)$ in one of four locations centered in the quadrants, the arbitrary phase shift is removed to within a phase ambiguity defined by the rotational symmetry of the constellation [25].

To help matters along, we initialized the equalizer filter using the MMSE filter coefficients described in Section 8.3. The initial motivations were to solve the phase ambiguity problem and shorten the convergence time of the adaptive filter. Numerous experiments showed that when the equalizer filter is initialized with the MMSE equalizer coefficients derived from the channel estimates, enabling the AMA equalizer either made no difference or made matters worse! Consequently, the best simulation results are those using the MMSE-initialized equalizer with $\beta = 0$.

More work needs to be done to determine what conditions, if any, favor the inclusion of the AMA cost function in the adaptive equalizer. It could be that as the CMA equalizer adapts, the updated equalizer filter coefficients impose a phase shift on the data that the simple PLL of Figure 142 is unable to track. This undesirable "phase performance" may result from, say, phase noise that is an unavoidable part of physical receivers.

Another important issue associated with the AMA algorithm is scaling. Because SOQPSK-TG is only approximately an offset QPSK, the AMA error is multiple orders of magnitude larger than the CMA error and this magnitude difference carries over to the gradients. Consequently, if β is not carefully chosen, the AMA error swamps any contribution from the CMA error. The difficulty is that the scale factor appears to depend on the channel and signal-to-noise ratio.

The simulated bit error rate (BER) of the CMA equalizer outlined in Figure 142 was performed

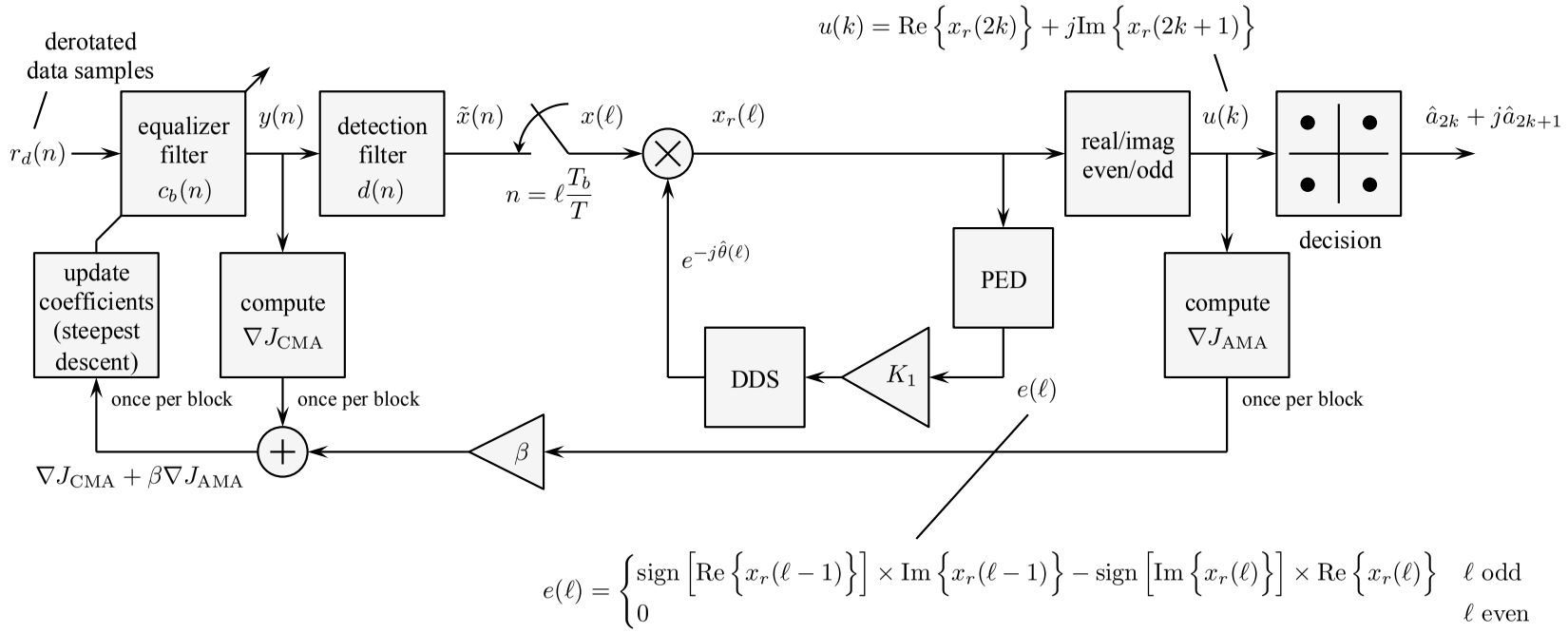

Figure 142: A block diagram of the adaptive CMA+AMA equalizer with a first order PLL for tracking the a linear phase increment due to an uncompensated frequency offset.

using the following parameters.

- The SOQPSK-TG signal and channel were sampled at a rate equivalent to 2 samples/bit.
- The equalizer length was set to approximately 4 times the channel length by using $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$.
- The AMA component to the error gradient was disabled. This was accomplished by setting $\beta = 0$: see (341) and (363). Had the AMA algorithm been enabled (i.e., by using $\beta > 0$), we would have used $A = 2$ in (345).
- The step size was set to $\mu = 5 \times 10^{-3}$.
- The equalizer filter was initialized with the MMSE equalizer filter coefficients given by (330). The MMSE equalizer filter coefficients, in turn, was based on data de-rotated by the frequency offset estimate (55), the channel estimate (275) using $N_1 = 12$ and $N_2 = 25$, and the signal to noise ratio (333) based on the noise variance estimate (290).
- The "ideal case" was not simulated. This is because estimation errors only affect the initialization of the adaptive filter. It was hard to see how a convergence behavior comparison involving an initialization based on the true channel and frequency offset was of practical use: the adaptive filter converges to the same filter in either the ideal or estimated cases.
- Before counting errors, the equalizer filter was allowed to converge. The convergence time was set to 100 blocks (that is, 100 iNET packets).

The simulation results for test channels 1 – 11 are shown in Figures 143 – 148, respectively. The authors' experience with initialization showed that for some test channels (test channels 2, 5, 7, 8, 10, and 11) convergence was achieved in as little as 35 blocks where as for other test channels (test channels 1, 3, 4, 6, and 9) 100 blocks were needed for convergence. Clearly, the number of blocks required for optimization needs to be examined more closely to determine if any generally applicable optimization is possible. In addition, it may be the case that a periodic MMSE-based re-initialization is required. The best period for this initialization is entirely unknown and is probably best examined once the real-time system is up and running.

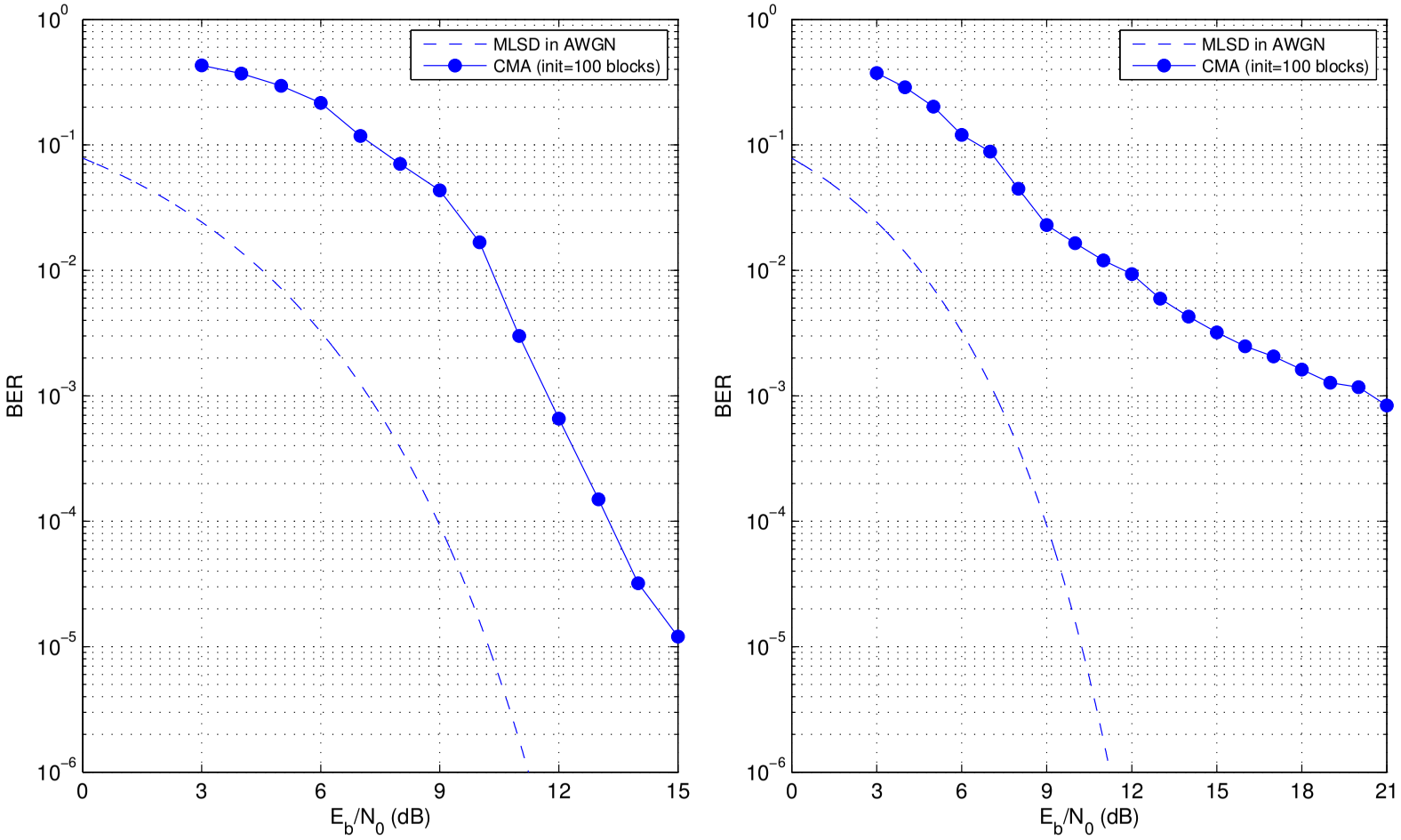

Figure 143: Simulated bit error rate (BER) performance of the CMA equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 1 (left) and test channel 2 (right).

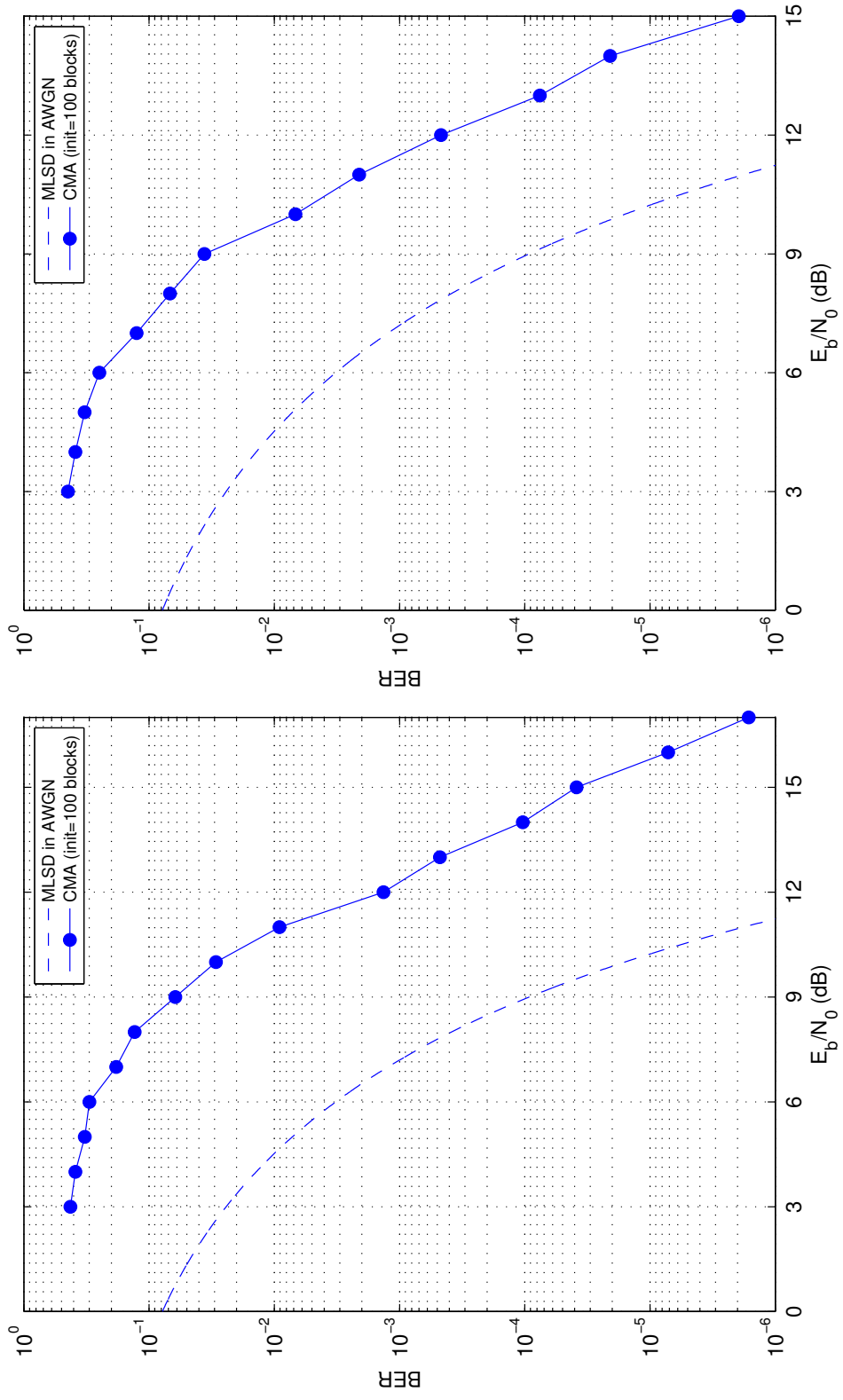

Figure 144: Simulated bit error rate (BER) performance of the CMA equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 3 (left) and test channel 4 (right).

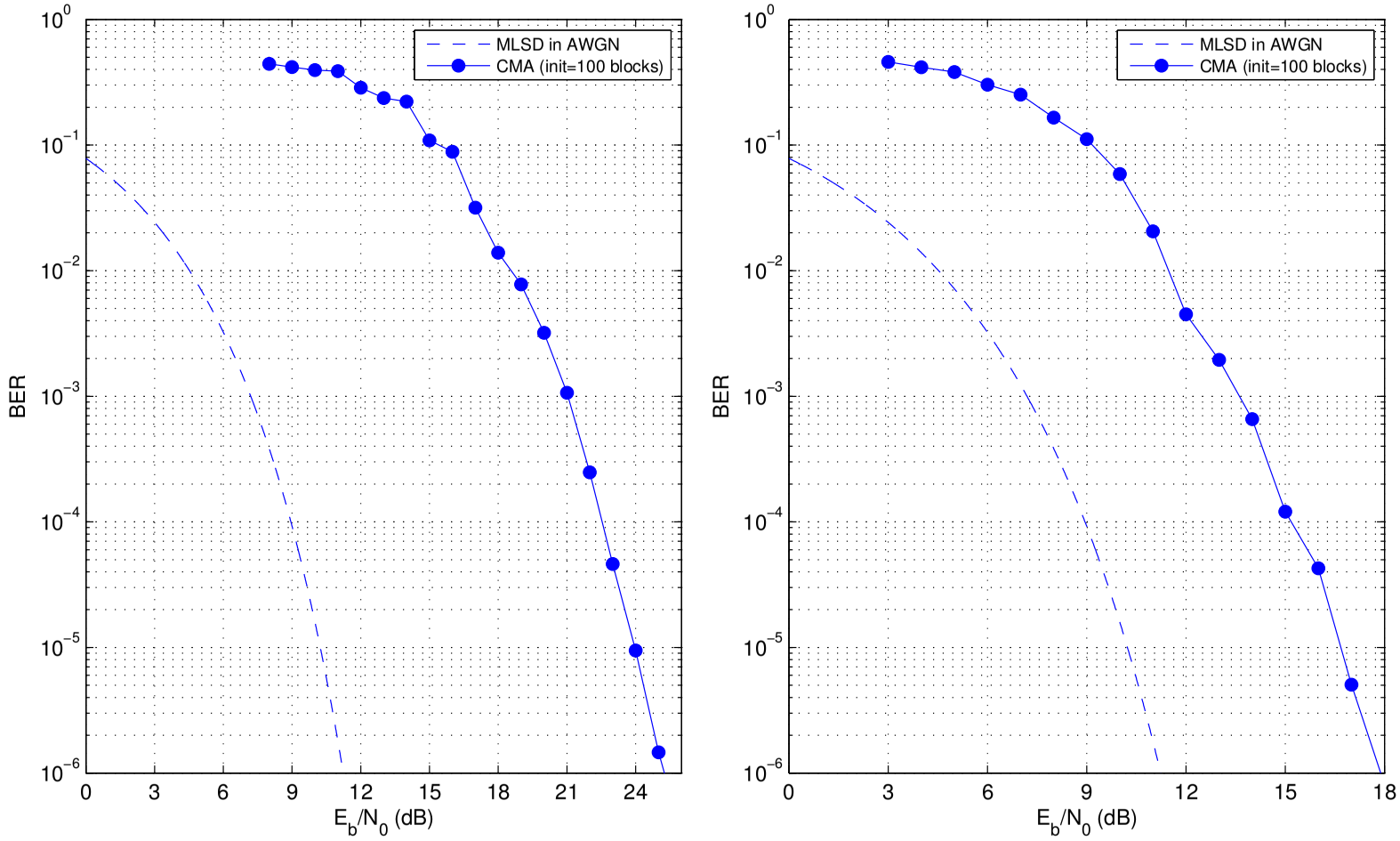

Figure 145: Simulated bit error rate (BER) performance of the CMA equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 5 (left) and test channel 6 (right).

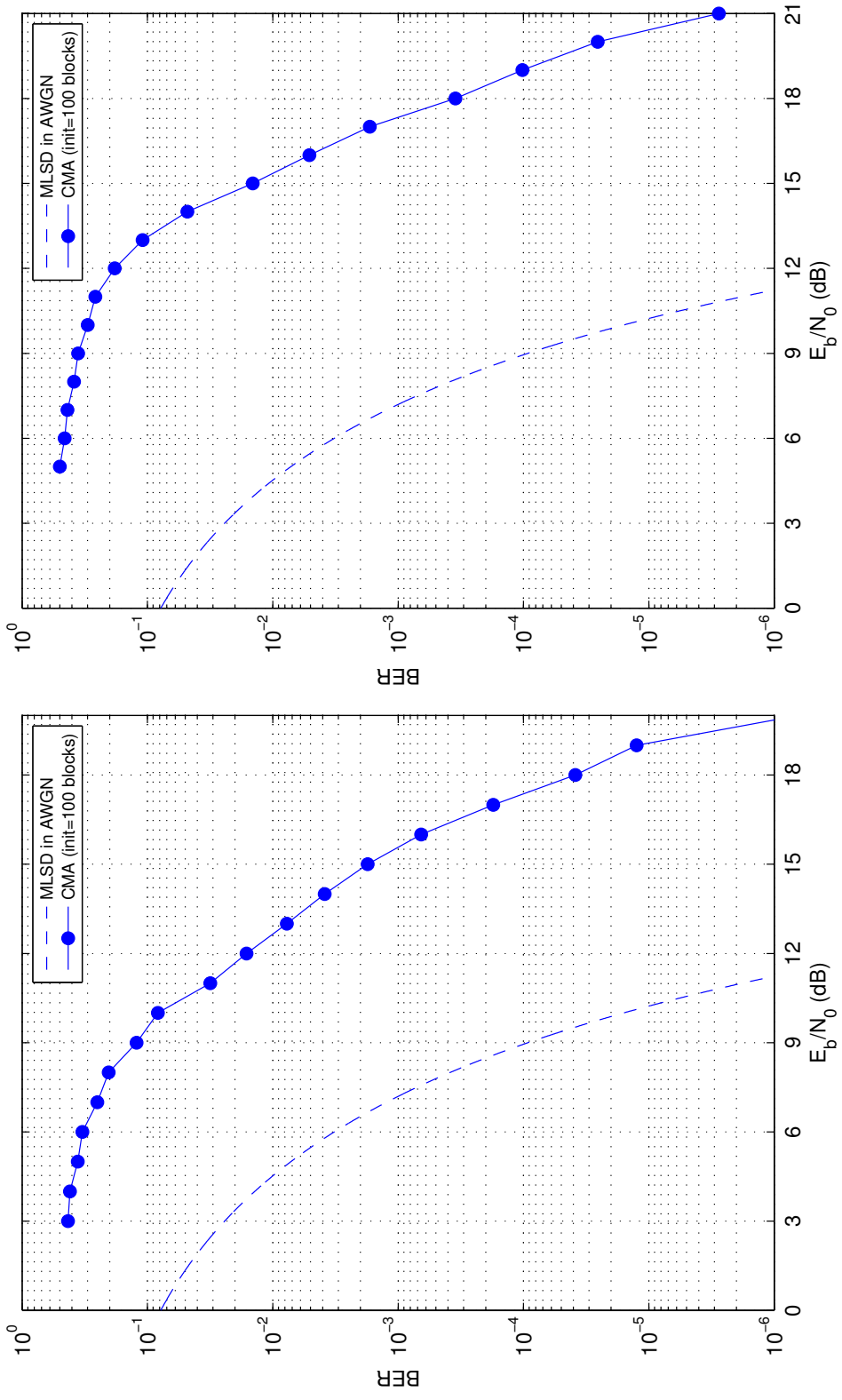

Figure 146: Simulated bit error rate (BER) performance of the CMA equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 7 (left) and test channel 8 (right).

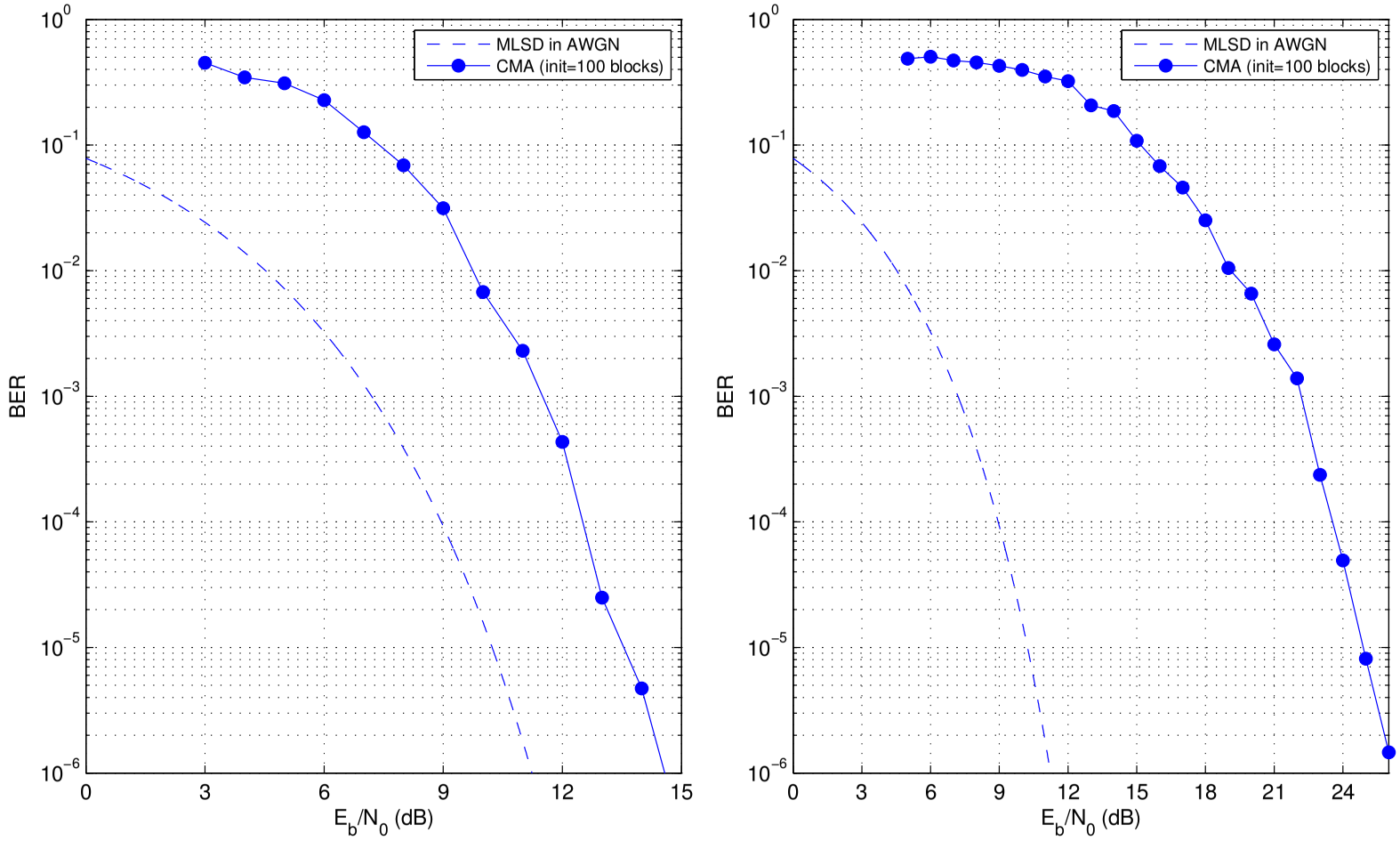

Figure 147: Simulated bit error rate (BER) performance of the CMA equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 9 (left) and test channel 10 (right).

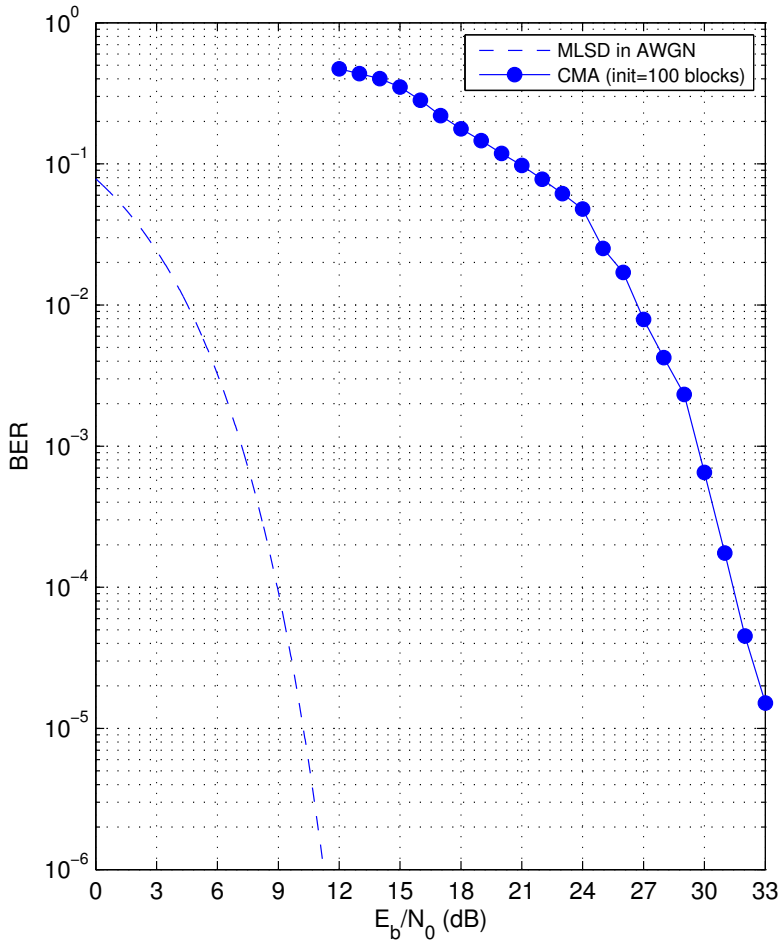

Figure 148: Simulated bit error rate (BER) performance of the CMA equalizer with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 11.

8.5 Longitudinal Comparisons

Here we compare the simulated BER performance for the ZF, MMSE, and CMA equalizers using estimates for the frequency offset, channel impulse response, and (in the case of the MMSE and CMA equalizers) the noise variance. The results presented here are those of Sections 8.2, 8.3, and 8.4 plotted on the same set of axes. The comparisons are plotted in Figures 149 – 154. For the most part, the MMSE and CMA equalizers have similar performance. The CMA equalizer has a slight advantage for high E_b/N_0 on test channels 1, 2, 3, 6, 8, and 11. The CMA equalizer seems to have problems for low E_b/N_0 on test channels 1, 3, and 9. With the exception of low E_b/N_0 on test channels 1, 3, and 9, the BER performance of both the MMSE and the CMA equalizers is superior to that of the ZF equalizer for the values of E_b/N_0 used in the simulations.

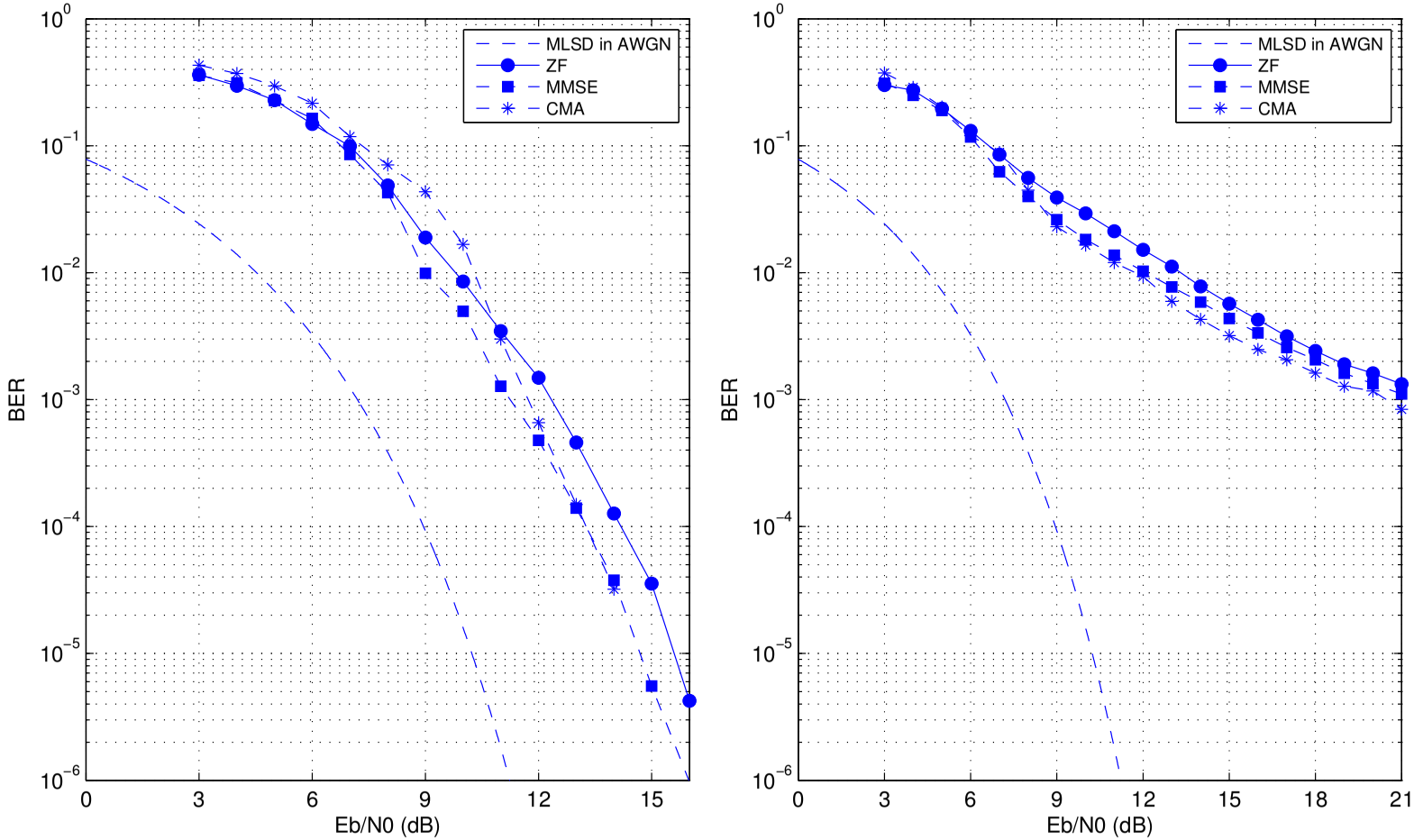

Figure 149: Simulated bit error rate (BER) performance of the ZF, MMSE, and CMA equalizers with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 1 (left) and test channel 2 (right).

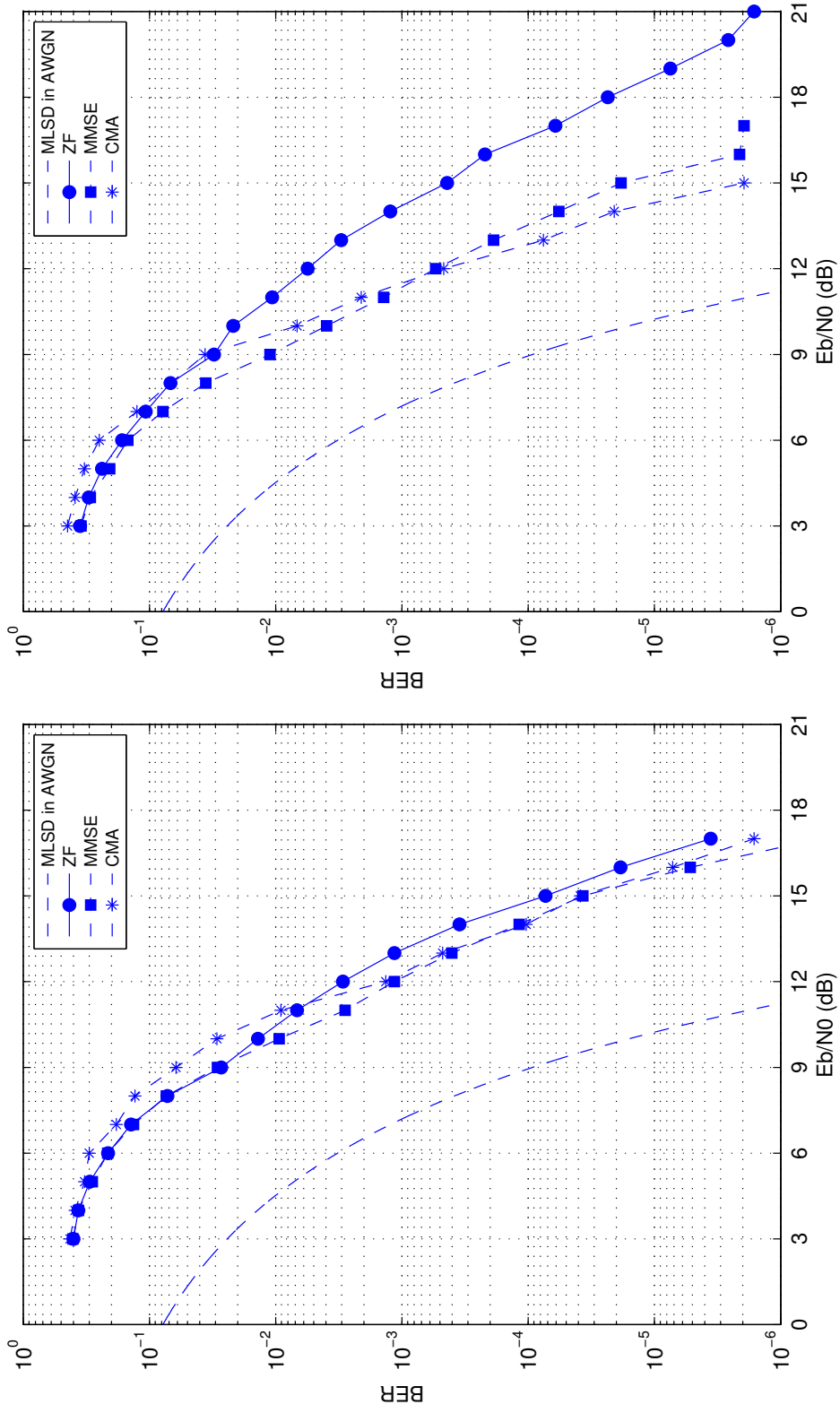

Figure 150: Simulated bit error rate (BER) performance of the ZF, MMSE, and CMA equalizers with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 3 (left) and test channel 4 (right).

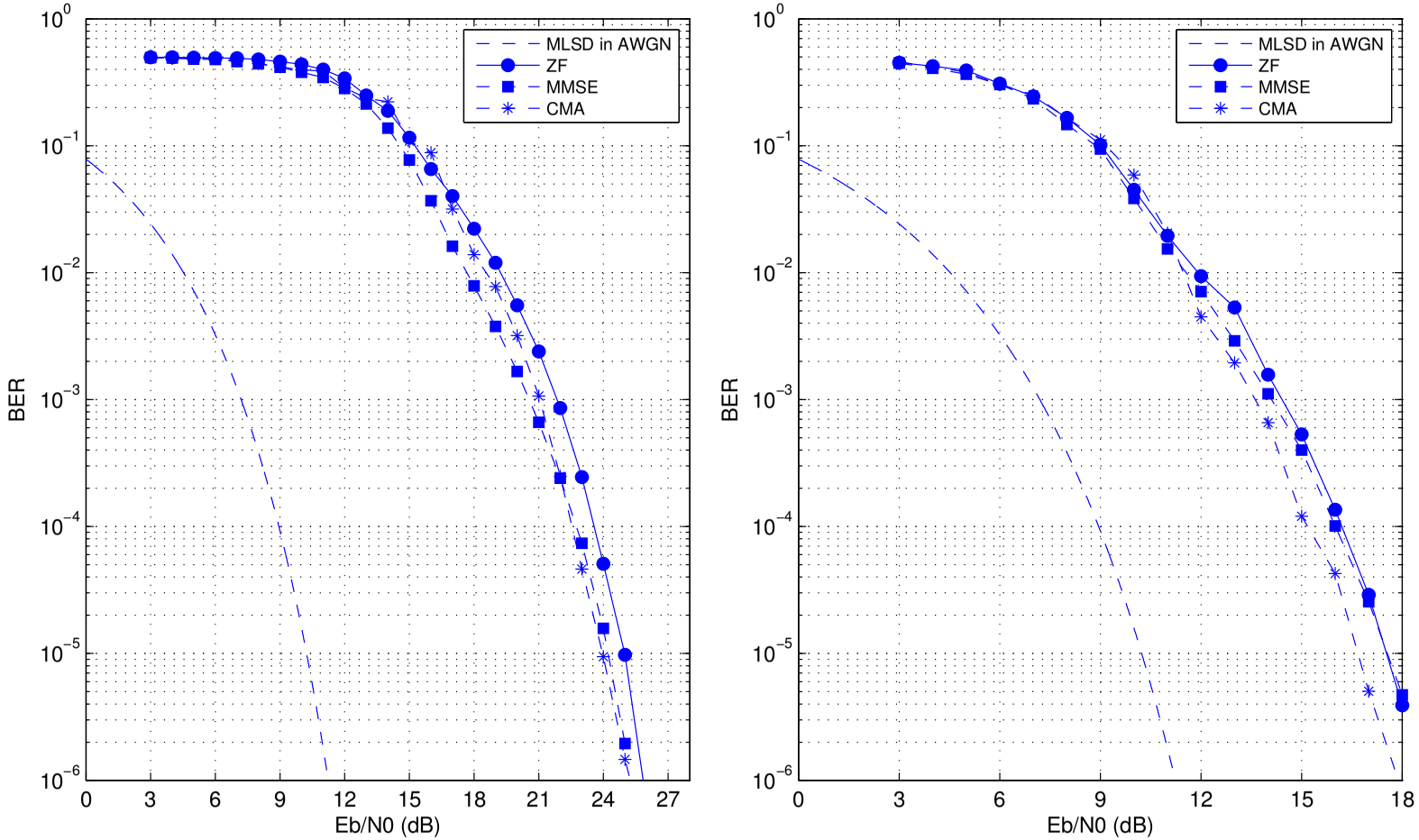

Figure 151: Simulated bit error rate (BER) performance of the ZF, MMSE, and CMA equalizers with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 5 (left) and test channel 6 (right).

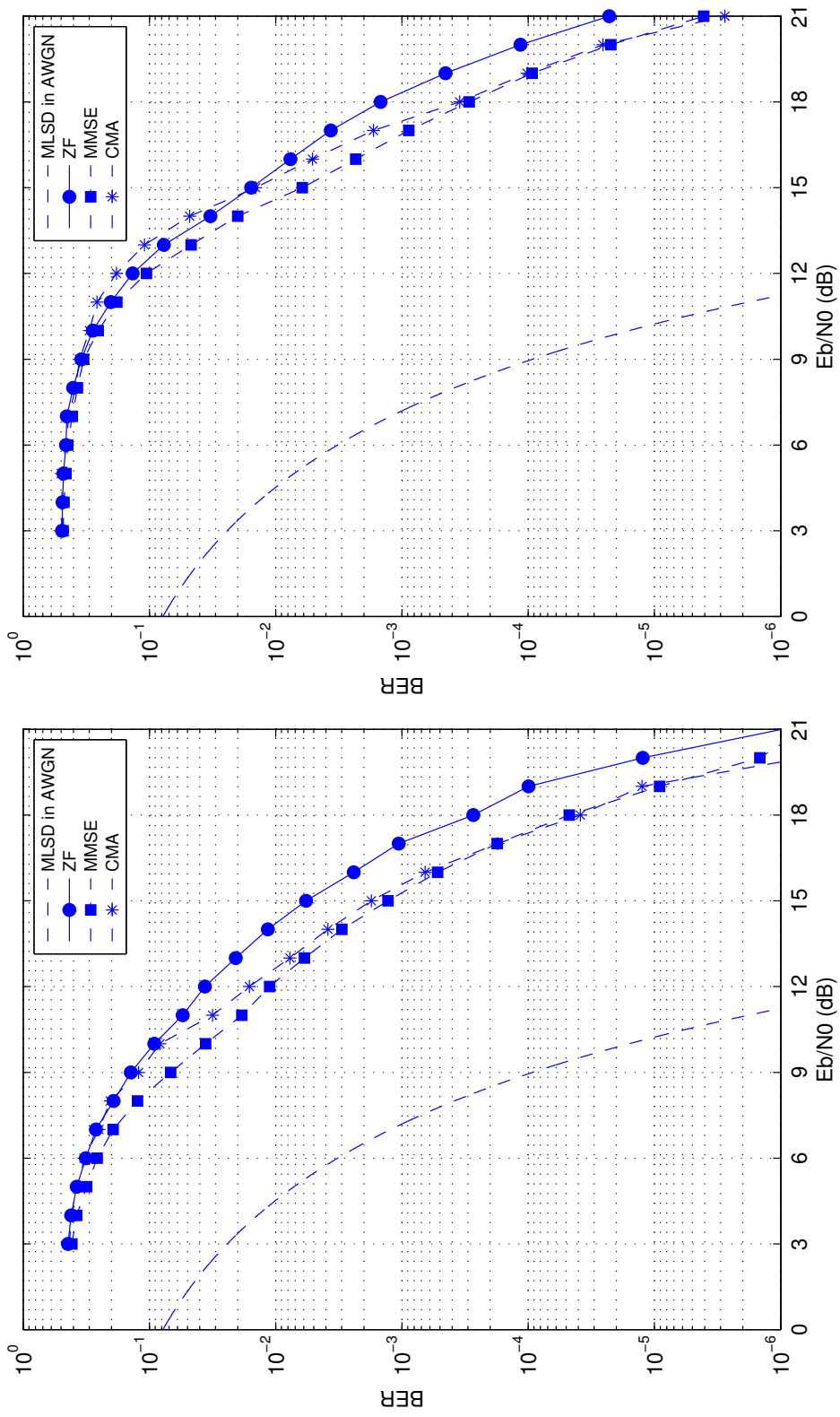

Figure 152: Simulated bit error rate (BER) performance of the ZF, MMSE, and CMA equalizers with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 7 (left) and test channel 8 (right).

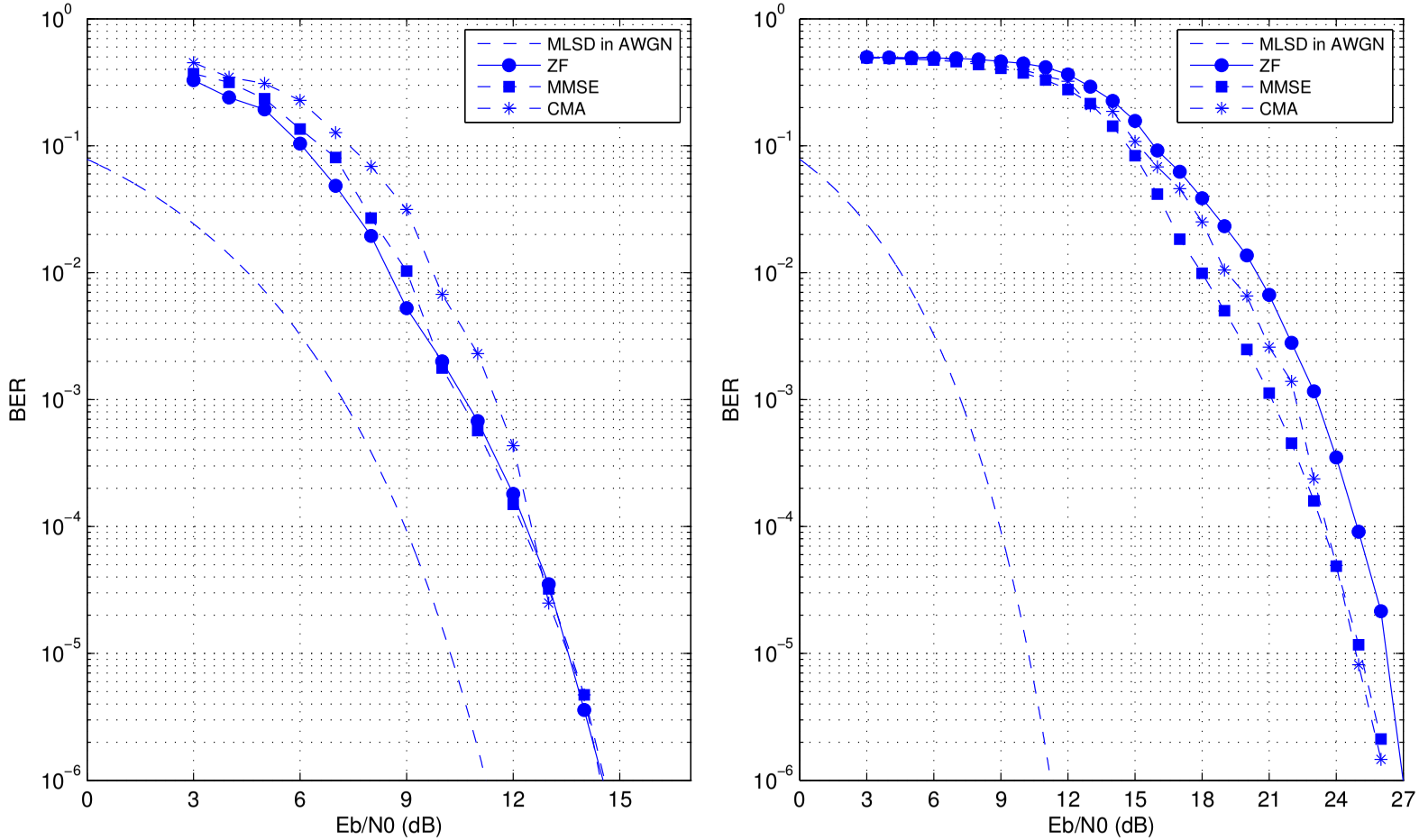

Figure 153: Simulated bit error rate (BER) performance of the ZF, MMSE, and CMA equalizers with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 9 (left) and test channel 10 (right).

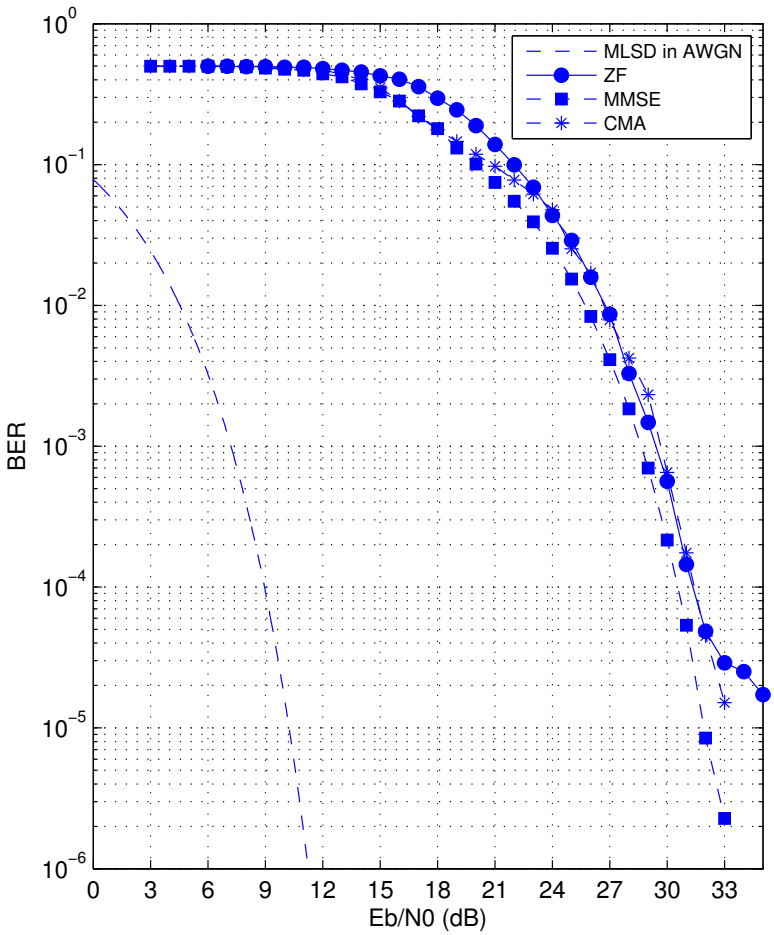

Figure 154: Simulated bit error rate (BER) performance of the ZF, MMSE, and CMA equalizers with $L_1 = 4 \times N_1$ and $L_2 = 4 \times N_2$ for test channel 11.

9 Thoughts on Other Equalization Techniques

9.1 Maximum Likelihood Sequence Estimation

The optimum "equalization" technique in the maximum likelihood sense is the maximum likelihood sequence estimator (MLSE). The theory and development of the MLSE is detailed in Chapter 9 of [14]. The fundamental idea is that an ISI channel induces memory on the received waveform. This memory can be modeled as a finite state machine. The number of states in the finite state machine are determined by the modulation alphabet and the channel length. The Viterbi algorithm may be used to identify the best path through the trellis defined by the finite state machine.

In this application, the memory induced by the channel is *additional* memory because the SOQPSK-TG waveform possesses its own memory. For example, SOQPSK-TG as a modulation is defined by a 512-state trellis. A length-L_h channel introduces an additional 2^{L_h-1} states. Consequently, the super-trellis formed by integrating the modulation states with the channel states contains $512 \times 2^{L_h-1}$ states. Given the fact that 512 states is too complex for real-time implementations, the MLSE operating on the $512 \times 2^{L_h-1}$-state super trellis does not seem feasible.

The 4-state approximation for SOQPSK-TG could be used here. (See Section A.3 in the Appendix.) In this case, the super trellis formed by the modulation and the length-L_h channel contains $4 \times 2^{L_h-1}$ states. This might seem a feasible approach if L_h is not too large. In the simulations presented above, $L_h \approx (N_1 + N_2 + 1)/2 = 19$. (The division by 2 follows from the fact that the channel was defined at 2 samples/bit whereas the trellis detector operates at 1 sample/bit.) In this case, the MLSE equalizer must operate on a trellis with $4 \times 2^{19} = 2{,}097{,}152$ states. This is still a prohibitively high number.

The limitations imposed by the computational burdens are not unique to this application. There has been some nice work on complexity reducing techniques. Some examples include the following:

- Combined decision feedback processing and maximum likelihood sequence estimation [26–30]. Here a truncated channel is used to reduce the size of the super trellis and decision feedback is used to cancel the residual ISI for the tentative (hypothesized) decisions.

- Reduced state sequence estimation [31, 32]. Here, the number of states in the super trellis is reduced in a way that minimizes the loss in performance. Interestingly, Eyuboğlu and Quereshi combine *both* reduced state sequence estimation *and* decision feedback in [33].

- Channel shortening, see [34] and the citations therein. Here, the channel impulse response

is shortened following a specified criterion, such as minimum mean-squared error or the "maximum shortening signal-to-noise ratio." The shortened channel is then used to form the super trellis. (The resulting super trellis has few states than the super trellis formed from the unshortened channel.)

How effective these techniques may be for SOQPSK-TG in the aeronautical telemetry channels remains an open question.

9.2 Decision Feedback Equalization

The decision feedback equalizer (DFE) is a non-linear equalizer well suited to channels with deep spectral nulls [14]. The adaptive DFE for offset QPSK was developed by Bello and Pahlavan [35] and later by Tu [36] in a matrix-vector formulation. These ideas were applied to the OQPSK interpretation of SOQPSK-TG (see the Appendix) by Rice and Satorius [37, 38]. Due to the strong ISI inherent in the SOQPSK-TG waveform (especially when interpreted as an OQPSK), the adaptation step size during training had to be small. Consequently, convergence was slow and an extremely long training sequence was required. Furthermore, the bit error rate performance was not measurable superior to that of the simple CMA equalizer. A non-adaptive formulation was examined by Rice, Narumanchi, and Saquib [39]. Here, the signal processing was quite complex and the bit error rate performance was not sufficiently better than the simple MMSE equalizer to justify added the computational complexity.

Clearly some more work needs to be done to determine the feasibility of applying decision feedback equalization to SOQPSK-TG in the aeronautical telemetry channels. Some open questions are

1. Is the OQPSK interpretation of SOQPSK-TG the best interpretation for DFE?

2. How effective is DFE when applied to *samples* of SOQPSK-TG in a decision feedback architecture? What is the best approach to reconstructing samples of the SOQPSK-TG waveform for the ISI cancelation through feedback?

3. How sensitive to channel estimation errors is the DFE operating on SOQPSK-TG?

The answers to these questions will be examined during Phase 2.

9.3 Adaptive Versions of ZF and MMSE equalizers

The ZF and MMSE equalizers described in Sections 8.2 and 8.3, respectively, have adaptive filter versions. Adaptive equalization is described in Chapter 10 of [14] for linear modulations (e.g., QPSK). The application to SOQPSK-TG is relatively straight-forward using the OQPSK interpretation of SOQPSK-TG outlined in the Appendix. There is an advantage and a disadvantage with adaptive equalization:

- The advantage is that an adaptive equalizer might be better able to deal with channel estimation errors. The ZF and MMSE equalizers presented in Sections 8.2 and 8.3, respectively, use filter coefficients computed from functions of the channel estimate. Channel estimation errors produce suboptimum equalizer filter coefficients as demonstrated by the differences between the "ideal" and "estimator" BER simulations summarized in Figures 127–132 for the ZF equalizer and Figures 135–140 for the MMSE equalizer. An adaptive equalizer may be able to adjust the equalizer filter coefficients to compensate for the channel estimation errors. An additional benefit might be the the ability to make adjustments to the equalizer filter coefficients in response to changes in the channel impulse response between occurrences of the preamble and ASM fields.

- The disadvantage of adaptive equalizers is convergence time – the number of samples (or data bits) required for the adaptive equalizer filter to converge to a usable filter from some initial setting. This problem is mostly (completely?) mitigated here by using, as the initialization, the optimum filter coefficients computed from the channel estimate.

Because SOQPSK-TG is only *approximately* an OQPSK, the ability of the SxS detector to produce a useful error signal for the adaptations remains an open question. The CMA+AMA equalizer results show promise. Motivated by this promise, the performance of the adaptive equalizer will be examined in the real-time system developed in Phase 2.

9.4 Frequency Domain Equalization (FDE)

Frequency domain equalization is popular equalization technique that leverages the well-known computational advantages of the FFT to perform filtering of long data sequences by long filters in discrete frequency domain. The FDE filter is most commonly a discrete frequency domain version of the MMSE equalizer. (The ZF equalizer could also work here.) Because multiplication in the discrete frequency domain corresponds to circular convolution in the discrete time domain, a cyclic prefix is required to account for the length of the channel. Because there is no provision for a

cyclic prefix in the iNET packet format, it might appear, at first glance, that FDE is not feasible for iNET packets. Coon *et al*, showed that unique word (i.e. preamble) based single carrier systems without a cyclic prefix can still be equalized using FDE as long as the preamble possesses a certain structure [40]. It turns out the iNET preamble has this structure! Consequently, we will investigate the performance and feasibility of the FDE version of the MMSE equalizer in Phase 2.

A SOQPSK-TG Detection

Since the 1970s, pulse code modulation/frequency modulation (PCM/FM) has been the dominant modulation used for test and evaluation on government test ranges in the USA, Europe, and Asia. PCM/FM is binary continuous phase modulation (CPM) with a digital modulation index $h = 0.7$ and a frequency pulse which is a rectangular pulse with a duration of one bit time that has been low-pass filtered. In the USA, the main spectral allocations for aeronautical telemetry are L-band (1435 - 1535 MHz), lower S-band (2200 - 2290 MHz), and upper S- band (2310 - 2390 MHz). Increasing data rate requirements along with an ever increasing number of test flights put tremendous pressure on these spectral allocations in the 1980s and 1990s. The situation was further exacerbated in 1997 when the lower portion of upper S-band from 2310 to 2360 MHz was reallocated in two separate auctions.[8] In response to this situation, the Telemetry Group of the Range Commanders Council adopted a more bandwidth efficient modulation as part of its Interrange Instrumentation Group (IRIG) standard, IRIG-106 [41], in 2000. This modulation, known as FQPSK-B, was a proprietary version of FQPSK described in [42]. Efforts to reduce some aspects of the implementation complexity resulted in a non-proprietary version, known as FQPSK-JR [43] which was adopted as part of IRIG-106 in 2004. Also in 2004, a version of SOQPSK [44], known as SOQPSK-TG, was adopted as a license-free, fully interoperable alternative in the IRIG-106 standard.

A.1 SOQPSK-TG: A Mathematical Description

SOQPSK-TG is defined as a continuous phase modulation (CPM) [45] of the form

$$s_{\mathrm{S}}(t, \boldsymbol{\alpha}) = \sqrt{\frac{2E_b}{T_b}} \exp\left[j\left(\phi(t, \boldsymbol{\alpha}) + \phi_0\right)\right]. \tag{364}$$

The phase is

$$\begin{aligned}\phi(t, \boldsymbol{\alpha}) &= 2\pi h \int_{-\infty}^{t} \sum_{n=-\infty}^{\infty} \alpha_n g(\tau - nT_b) d\tau \\ &= 2\pi h \sum_{n=-\infty}^{\infty} \alpha_n q(t - nT_b).\end{aligned} \tag{365}$$

[8] 2320 - 2345 MHz was reallocated for digital audio radio in one auction while 2305 - 2320 MHz and 2345 - 2360 MHz were allocated to wireless communications services in the other auction.

where $g(t)$ is the frequency pulse; $q(t)$ is the phase pulse, whose relationship to the frequency pulse is

$$q(t) = \int_{-\infty}^{t} g(\tau) d\tau; \tag{366}$$

ϕ_0 is an arbitrary phase which, without loss of generality, can be set to 0; T_b is the bit interval (or reciprocal of the bit rate); $h = 1/2$ is the modulation index; and $\alpha_n \in \{-1, 0, 1\}$ are the ternary symbols which are related to the binary input symbols $a_n \in \{-1, 1\}$ by [46]

$$\alpha_n = (-1)^{n+1} \frac{a_{n-1}\left(a_n - a_{n-2}\right)}{2}. \tag{367}$$

The binary input symbols $a_n \in \{-1, +1\}$ are related to the binary input bits $b_n \in \{0, 1\}$ by

$$a_n = \begin{cases} -1 & b_n = 0 \\ +1 & b_n = 1. \end{cases} \tag{368}$$

The frequency pulse for SOQPSK-TG is a spectral raised cosine windowed by a temporal raised cosine [44]:

$$g(t) = C \frac{\cos\left(\frac{\pi \rho B t}{2T_b}\right)}{1 - 4\left(\frac{\rho B t}{2T_b}\right)^2} \times \frac{\sin\left(\frac{\pi B t}{2T_b}\right)}{\left(\frac{\pi B t}{2T_b}\right)} \times w(t) \tag{369}$$

for

$$w(t) = \begin{cases} 1 & 0 \le \left|\frac{t}{2T_b}\right| < T_1 \\ \frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi}{T_2}\left(\frac{t}{2T_b} - T_1\right)\right) & T_1 \le \left|\frac{t}{2T_b}\right| \le T_1 + T_2 \\ 0 & T_1 + T_2 < \left|\frac{t}{2T_b}\right| \end{cases}. \tag{370}$$

For SOQPSK-TG, the parameters are[9] $\rho = 0.7$, $B = 1.25$, $T_1 = 1.5$, and $T_2 = 0.5$. The constant C is chosen to make $q_S(t) = 1/2$ for $t \ge 2(T_1 + T_2)T_b$. The frequency pulse and corresponding phase pulse for this case are shown in Figure 155. Observe that these values of ρ, B, T_1 and T_2 make SOQPSK-TG a partial response CPM spanning $L = 8$ bit intervals.

[9] In the original publication [44], two versions of SOQPSK were described: SOQPSK-A defined by $\rho = 1$, $B = 1.35$, $T_1 = 1.4$, and $T_2 = 0.6$ and SOQPSK-B defined by $\rho = 0.5$, $B = 1.45$, $T_1 = 2.8$, and $T_2 = 1.2$. SOQPSK-A has a slightly narrower bandwidth (measured at the -60 dB level) and slightly worse detection efficiency than SOQPSK-B. The Telemetry Group of the Range Commanders Council adopted the compromise waveform, designated SOQPSK-TG in 2004.

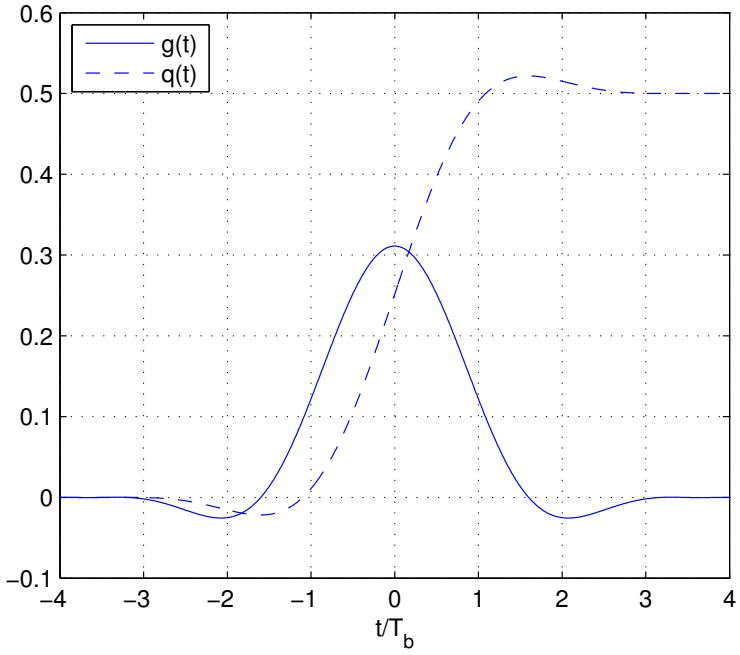

Figure 155: The frequency pulse $g(t)$ and the phase pulse $q(t)$ for SOQPSK-TG.

Because of the memory inherent in the modulated waveform, the maximum likelihood (ML) detector is a maximum likelihood sequence detector (MLSD) based on a trellis [45]. The number of states in the trellis is a function of the size of the symbol alphabet, the modulation index, and the length of the frequency pulse [45]. The ML detector for SOQPSK-TG comprises 512 states. The analysis of ML detection of SOQPSK was carried out by Geoghegan [47], [48], [49] following the standard union bound technique based on pairwise error probabilities [45]. The binary-to-ternary mapping (367) contributes an extra step to the analysis. Let

$$\mathbf{a} = \ldots a_{k-3}, a_{k-2}, a_{k-1}, a_k, a_{k+1}, a_{k+2}, a_{k+3}, \ldots \tag{371}$$

represent a generic binary symbol sequence with $a_k \in \{-1, +1\}$. The minimum distance error event occurs between the waveforms corresponding to two binary symbol sequences whose difference satisfies [50]

$$\mathbf{a}_1 - \mathbf{a}_2 = \pm [\ldots, 0, 0, 0, 2, 0, 0, 0, \ldots] \tag{372}$$

where the difference (or erroneous symbol) occurs at index k. There are 64 such sequence pairs. As it turns out, there are two ways a pair of binary sequences can produce (372) Sequence pairs for which $a_{k-1} = a_{k+1}$ are characterized by waveforms separated by a normalized squared Euclidean

distance of 1.60. There are 32 such sequences. Sequence pairs for which $a_{k-1} = -a_{k+1}$ are characterized by waveforms separated by a normalized squared Euclidean distance of 2.58. There are 32 such sequences. The probability of error is well approximated by [50]

$$P_b \approx \frac{1 \times 32}{64} Q\left(\sqrt{1.60 \frac{E_b}{N_0}}\right) + \frac{1 \times 32}{64} Q\left(\sqrt{2.58 \frac{E_b}{N_0}}\right). \tag{373}$$

The curve labeled "MLSD in AWGN" in the bit error rate plots above are plots of (373).

Because a 512-state trellis is too complex for real-time operation at the high end of aeronautical telemetry data rates, reduced-complexity versions of the detector are used. The two most common, the symbol-by-symbol (SxS) detector a 4-state trellis detector are described in Sections A.2 and A.3 below.

A.2 The Symbol-by-Symbol (SxS) Detector

The symbol-by-symbol (SxS) SOQPSK-TG detector is the simplest and most widely deployed detector. This simple form is the basis for the "interoperability" with FQPSK (-B and -JR versions). The motivation for the SxS detector is illustrated in Figure 156. The left-hand plot shows eye diagrams formed from the inphase and quadrature components (real and imaginary parts of the complex-valued baseband equivalent signal, respectively) of SOQPSK-TG corresponding to a random data bit pattern. The key feature here is the position of the maximum average eye openings. The maximum average eye opening for the inphase component corresponds to even-integer multiples of the bit time T_b whereas the maximum average eye opening for the quadrature component corresponds to odd-integer multiples of T_b. This characteristic is identical to traditional offset QPSK (OQPSK) [25], except for the presence of noticeable intersymbol interference (ISI). Sampling the inphase and quadrature eye diagrams at their corresponding maximum average eye openings produces the scatter plot shown on the right-hand side of Figure 156. The precoding defined by (367) allows "symbol-by-symbol" (to use the OQPSK terminology) detection of the data bits from each pair of samples.

The SxS detector, shown in Figure 157, is based on an approximate offset QPSK (OQPSK) interpretation of SOQPSK-TG. With reference to the basic system illustrated in Figure 126, the detector shown in Figure 157 (a), which assumes ideal timing and phase synchronization, operates on the equalizer output $y(n)$. The equalizer output is processed by a detection filter with impulse response $d(n)$. (The detection filter replaces the matched filter in standard OQPSK detectors because a true matched filter does not exist for SOQPSK-TG.) The detection filter output

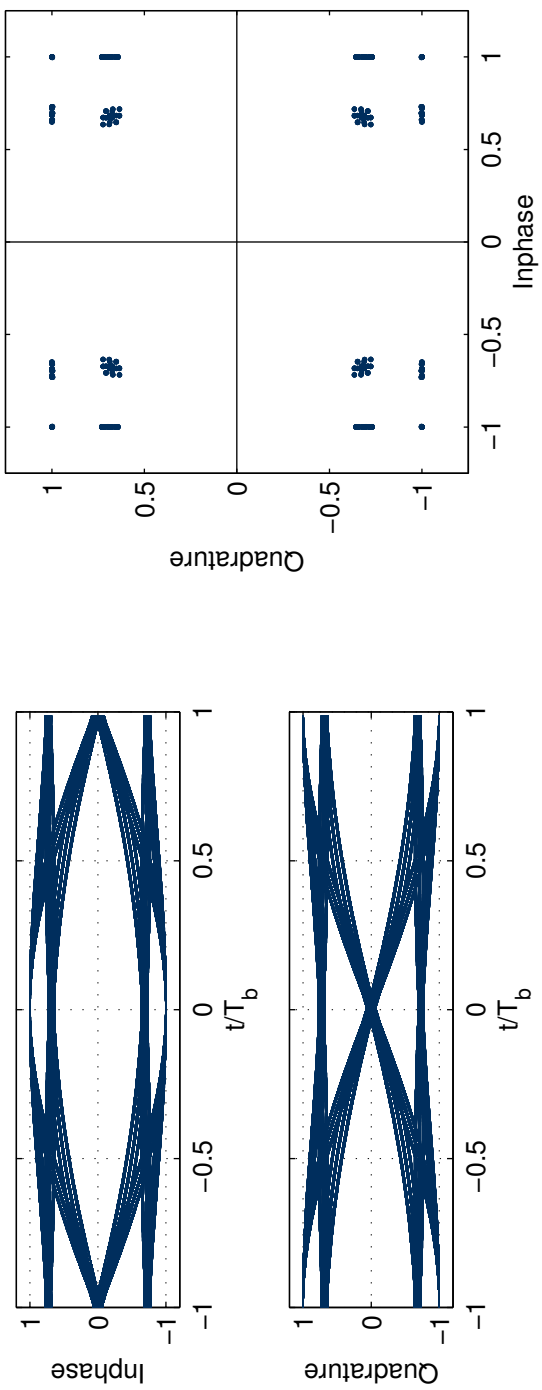

Figure 156: The SOQPSK-TG waveform: (left) eye diagrams formed from the inphase and quadrature components of an SQPSK-TG waveform corresponding to random data bits; (right) a scatter diagram formed from samples of the inphase eye diagram sampled at even-integer multiples of T_b paired with samples of the quadrature eye diagram sampled at odd-integer multiples of T_b.

$\tilde{x}(n)$ is downsampled by T_b/T (this ratio is 2 in our implementation) to produce the sequence $x(k)$, operating at 1 sample/bit. That is

$$x(k) = \tilde{x}\left(\frac{T_b}{T}k\right). \tag{374}$$

The decision variable $u(k)$ is extracted from the real and imaginary parts of $x(k)$ as follows:

$$u(k) = \begin{cases} \text{Re}\left\{x(k)\right\} & k \text{ even} \\ \text{Im}\left\{x(k)\right\} & k \text{ odd} \end{cases} \tag{375}$$

where a 0-based indexing is assumed (i.e, the downsampled detection filter output is $x(0)$, $x(1)$, ...). The sign of the decision variable is used to produce an estimate of the antipodal representation of the k-th bit $\hat{a}_k$.

The detector of Figure 157 (a) assumes perfect timing synchronization and perfect carrier phase synchronization. In the context of data-aided equalization, the equalizer accounts for timing offset in computing the optimum equalizer coefficients (that is, the timing offset is "part of the channel"). Consequently, equalizer output is time synchronized (at least, as best as can be achieved in the given environment), and, when the frequency offset is perfectly known, phase synchronized.

For the case where the frequency offset is not perfectly known (as in the case when a frequency estimator is used), the equalizer output is still time synchronized, but not phase synchronized. This is because any residual frequency offset imparts a linearly increasing (or decreasing) phase shift on the data samples at the equalizer output. This same phase shift applies to the detection filter output as well. To compensate for this unknown and linearly increasing (decreasing) phase offset, a simple first order phase lock loop (PLL) must be applied. The SxS detector with a PLL is illustrated in Figure 157 (b). Here, a PLL is used to estimate the residual phase shift. The residual phase shift estimate is used to derotate the downsampled detection filter output samples as shown. These rotated detection filter outputs, denoted $x_r(k)$ in the figure, are used in place of the $x(k)$ of Figure 157 (a) to produce the decision variables.

In steady-state operation ("in lock") the PLL produces a phase estimate that drives the error signal to zero. The error signal $e(k)$ is proportional to the residual phase shift and is computed the PED (for "phase error detector") block using the standard OQPSK phase error detector [25]:

$$e(k) = \begin{cases} \text{sign}\left[\text{Re}\left\{x_r(k-1)\right\}\right] \times \text{Im}\left\{x_r(k-1)\right\} - \text{sign}\left[\text{Im}\left\{x_r(k)\right\}\right] \times \text{Re}\left\{x_r(k)\right\} & k \text{ odd} \\ 0 & k \text{ even} \end{cases} \tag{376}$$

The filtered error signal is converted to a pair of quadrature sinusoids, represented by the complex exponential, using a standard DDS architecture [25].

The bit error rate performance of the SxS detector is a function of the detection filter $d(n)$. The simple integrate-and-dump filter (well approximated by a detection filter comprising $2T_b/T$ ones). was used in assessing the interoperability of FQPSK (-B and -JR) with SOQPSK-TG.[10] In [47], Geoghegan applied a stochastic steepest descent algorithm [51] to identify the detection filter that minimized the bit error rate. Although the bit error rate performance of this numerically optimized filter were presented in [47], the actual filter response $d(n)$ was not published (one assumes for proprietary reasons). The procedure was reproduced by Perrins and plotted in Figure 3 of [21]. The result, reproduced here, is illustrated in Figure 158 for a sample rate equivalent to 2 samples/bit. The bit error rate performance using the integrate-and-dump detection and numerically optimized detection filters are presented in Figure 10 of [21]. These results show that the integrate-and-dump filter is 2 dB worse than the MLSD bound (373) and 0.8 dB worse than the MLSD bound (373) in AWGN.

A.3 The 4-State Trellis Detector

As mentioned above, the ML detector is a maximum likelihood sequence detector (MLSD) with 512 states. To reduce the computational complexity of the detector, a trellis derived from an approximation of SOQPSK-TG is used. The approximations are discussed in [21]. The only approximation highlighted here is the one used in formulating the sequence detectors used in our simulations.

The pulse amplitude modulation (PAM) representation of SOQPSK-TG was derived by Perrins and Rice [52] and is

$$s(t) = \sum_{k=0}^{R-1} \sum_{i} b_{k,i} c_k(t - iT_b) \tag{377}$$

where $R = 2 \times 3^{L-1} = 4374$. Equation (377) represents SOQPSK-TG as a linear combination of 4374 pulses $c_k(t)$ whose amplitudes are modulated by *pseudosymbols* $b_{k,i}$. Formulae for the pulses and pseudosymbols are given in [52]. The R pseudosymbols $b_{k,i}$ are derived from the ternary SOQPSK-TG symbols α_i [see (365) and (367)] by a nonlinear mapping. Hence, the nonlinear

[10] Interoperability in this context means that FPQPSK-JR and SOQPSK-TG occupy essentially the same bandwidth and have nearly equivalent bit error rate performance when using an ordinary OQPSK detector with an integrate-and-dump detection filter. Furthermore, the same bit sequences are produced by this detector with both modulations. This means that the OQPSK-based SxS detection can be modulator agnostic — it does not know, nor does it care — which of the interoperable modulations were used.

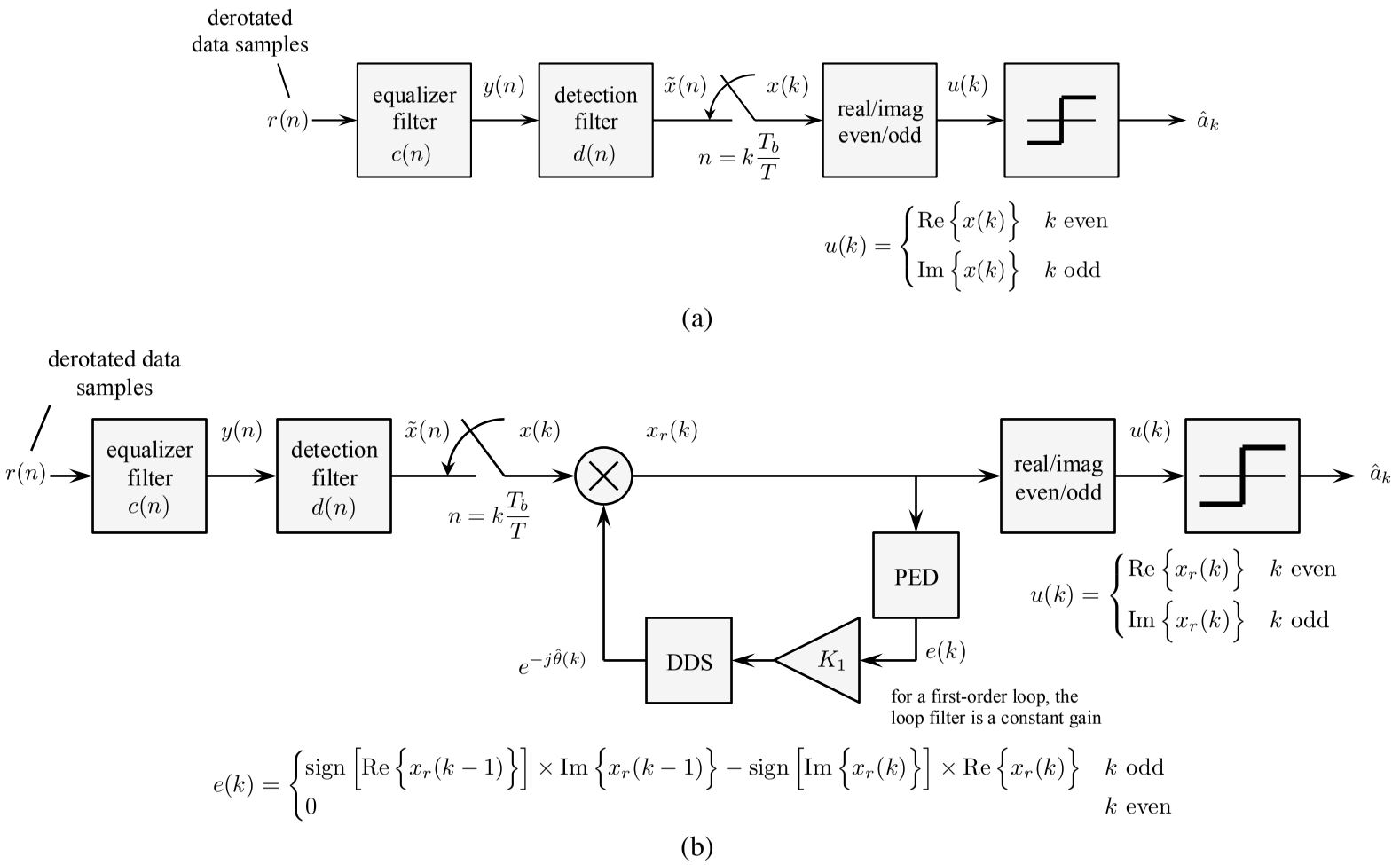

Figure 157: Symbol-by-symbol (SxS) detectors for SOQPSK-TG: (a) the ideal detector assuming perfect timing and phase synchronization; (b) the detector that assumes perfect timing with a small carrier phase frequency offset.

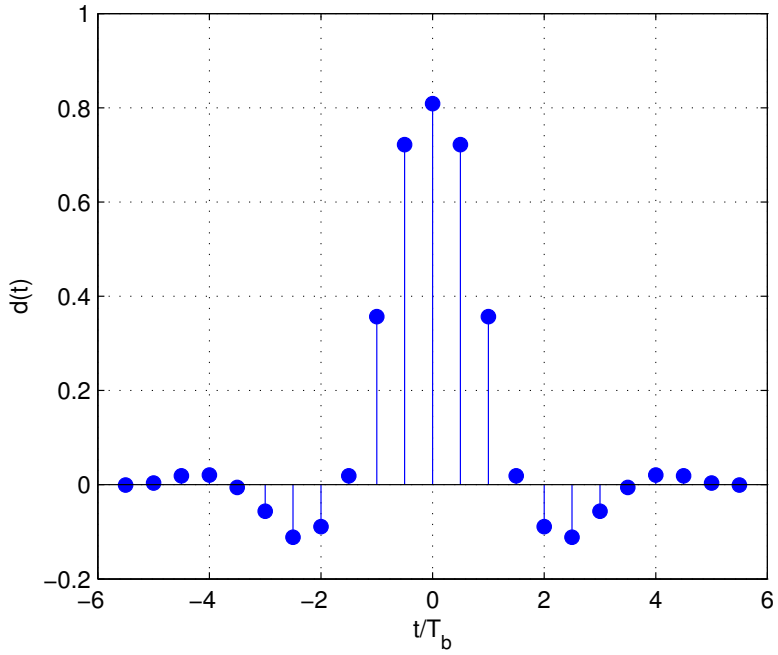

Figure 158: The impulse response of the numerically optimized detection filter $d(n)$ (cf., Figure 157) at an equivalent sample rate of 2 samples/bit.

nature of CPM is isolated in the pseudosymbols.

For the purposes of constructing a reduced-complexity detector for SOQPSK-TG, the approximation resulting from retaining the first two terms of the outer summation of (377) is used:

$$s(t) \approx \sum_i b_{0,i} c_0(t - iT_b) + \sum_i b_{1,i} c_1(t - iT_b). \tag{378}$$

The two pulses of interest, $c_0(t)$ and $c_1(t)$, are

$$c_0(t) = \left(\prod_{\ell=0}^{L-1} \psi(t + \ell T_b) \right)^2 \tag{379}$$

$$c_1(t) = 2 \left(\prod_{\ell=0}^{L-1} \psi(t + \ell T_b) \right) \left(\prod_{\ell=0}^{L-1} \psi(t + \ell T_b + T_b) \right) \tag{380}$$

where

$$\psi(t) = \begin{cases} \dfrac{\sin\left(\dfrac{\pi}{2}q(t)\right)}{\sin\left(\dfrac{\pi}{4}\right)} & 0 \le t < LT_b \\ \dfrac{\sin\left(\dfrac{\pi}{4} - \dfrac{\pi}{2}q(t-LT_b)\right)}{\sin\left(\dfrac{\pi}{4}\right)} & LT_b \le t < 2LT_b \\ 0 & \text{otherwise.} \end{cases} \tag{381}$$

These pulses, known as *principal pulses* in [52], are plotted in Figure 159. Principal pulse $c_0(t)$ has support on $0 \le t \le 9T_b$ and principal pulse $c_1(t)$ has support on $0 \le t \le 8T_b$. The pseudosymbols of (378) are

$$b_{0,i} = \begin{cases} -je^{j\theta_{i-1}} & \alpha_i = -1 \\ e^{j\theta_{i-1}} & \alpha_i = 0 \\ je^{j\theta_{i-1}} & \alpha_i = +1 \end{cases} \qquad b_{1,i} = \begin{cases} \dfrac{1}{\sqrt{2}}(1-j)e^{j\theta_{i-1}} & \alpha_i = -1 \\ \dfrac{1}{\sqrt{2}}e^{j\theta_{i-1}} & \alpha_i = 0 \\ \dfrac{1}{\sqrt{2}}(1+j)e^{j\theta_{i-1}} & \alpha_i = +1. \end{cases} \tag{382}$$

where $\theta_i \in \{3\pi/2, \pi, 0, \pi/2\}$ is the phase state of SOQPSK-TG at the end of bit time i. When working with the Viterbi algorithm, it is most convenient to express the pseudosymbols in the form

$$b_{0,i} = \beta_0(\alpha_i)e^{j\theta_{i-1}} \qquad b_{1,i} = \beta_1(\alpha_i)e^{j\theta_{i-1}} \tag{383}$$

where, after quick inspection of (382), we have

$$\beta_0(\alpha_i) = \begin{cases} -j & \alpha_i = -1 \\ 1 & \alpha_i = 0 \\ j & \alpha_i = +1 \end{cases} \qquad \beta_1(\alpha_i) = \begin{cases} \dfrac{1}{\sqrt{2}}(1-j) & \alpha_i = -1 \\ \dfrac{1}{\sqrt{2}} & \alpha_i = 0 \\ \dfrac{1}{\sqrt{2}}(1+j) & \alpha_i = +1. \end{cases} \tag{384}$$

The reduced-state trellis detector follows from the maximum likelihood principle. Let $\mathbf{b}$ be a sequence of N_b bits transmitted during the interval $0 \le t \le N_bT_b$. Using continuous-time notation for the moment, let $y(t)$ be the equalizer output in a continuous-time version of the system in

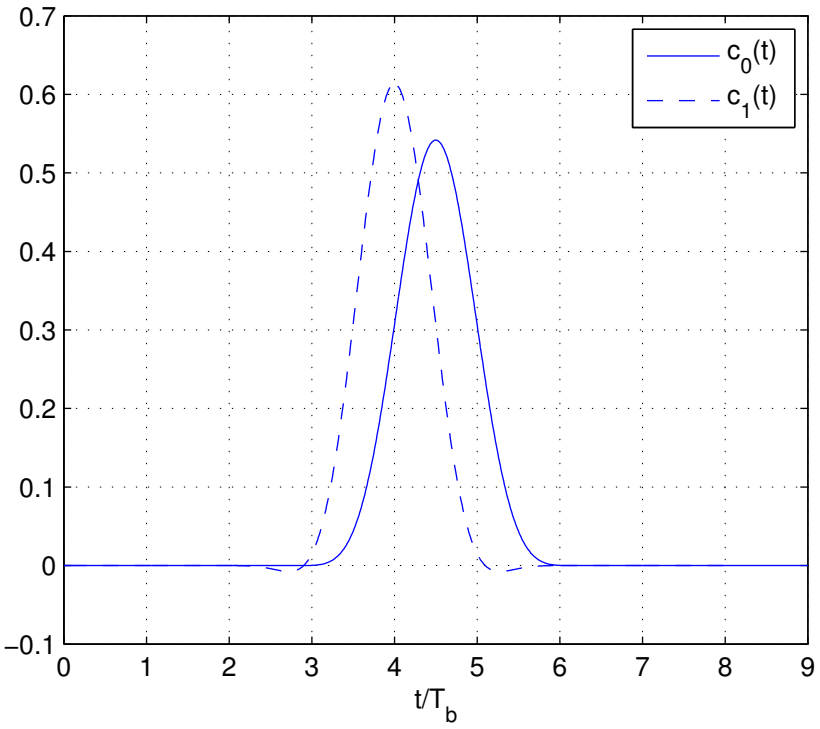

Figure 159: The principle pulses associated with the PAM representation for SOQPSK-TG.

Figure 126. The log-likelihood function is

$$\Lambda(\mathbf{b}) = -\int_0^{N_bT_b} \left|y(t) - s(t)\right|^2 dt \tag{385}$$

$$= -\int_0^{N_bT_b} \left[|y(t)|^2 - 2\text{Re}\left\{y(t)s^*(t)\right\} + |s(t)|^2\right] dt \tag{386}$$

$$= \text{Re}\left\{\int_0^{N_bT_b} y(t)s^*(t)dt\right\} \tag{387}$$

where the last step follows from the fact that $|y(t)|^2$ does not have a functional dependence on the data and $|s(t)|^2$ is a constant. Using the PAM approximation (378), we have for the integral

$$\int_0^{N_bT_b} y(t)s^*(t)dt \approx y(t)\int_0^{N_bT_b}\left[\sum_{i=0}^{N_b-1} b_{0,i}^* c_0(t-iT_b) + y(t)\sum_{i=0}^{N_b-1} b_{1,i}^* c_1(t-iT_b)\right]dt$$

$$= \sum_{i=0}^{N_b-1} b_{0,i}^* \int_0^{N_bT_b} y(t)c_0(t-iT_b)dt + \sum_{i=0}^{N_b-1} b_{1,i}^* \int_0^{N_bT_b} y(t)c_1(t-iT_b)dt$$

$$= \sum_{i=0}^{N_b-1} b_{0,i}^* \underbrace{\int_{iT_b}^{(i+9)T_b} y(t)c_0(t-iT_b)dt}_{x_0(i)} + \sum_{i=0}^{N_b-1} b_{1,i}^* \underbrace{\int_{iT_b}^{(i+8)T_b} y(t)c_1(t-iT_b)dt}_{x_1(i)} \tag{388}$$

$$= \sum_{i=0}^{N_b-1} b_{0,i}^* x_0(i) + \sum_{i=0}^{N_b-1} b_{1,i}^* x_1(i). \tag{389}$$

Using the definitions (384), we have

$$\int_0^{N_bT_b} y(t)s^*(t)dt = \sum_{i=0}^{N_b-1} e^{-j\theta_{i-1}} \left[\beta_0^*(\alpha_i)x_0(i) + \beta_1^*(\alpha_i)x_1(i)\right]. \tag{390}$$

Putting this all together, the log-likelihood function for the sequence of bits $\mathbf{b}$ is

$$\Lambda(\mathbf{b}) = \text{Re}\left\{\sum_{i=0}^{N_b-1} e^{-j\theta_{i-1}} \left[\beta_0^*(\alpha_i)x_0(i) + \beta_1^*(\alpha_i)x_1(i)\right]\right\}. \tag{391}$$

The sequence that maximizes $\Lambda(\mathbf{b})$ is the ML bit sequence estimate. Because there is no closed form expression for the ML bit sequence, as search must be conducted. The most efficient search algorithm for this type of structured problem is the Viterbi Algorithm.

To understand the notational conventions of the Viterbi algorithm, let

$$\tilde{\mathbf{b}}_n = \tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_n \tag{392}$$

be a possible (hypothesized) bit sequence up to bit time n and let

$$\tilde{\boldsymbol{\alpha}}_n = \tilde{\alpha}_0, \tilde{\alpha}_1, \ldots, \tilde{\alpha}_n \tag{393}$$

be the corresponding sequence of ternary symbols. The log-likelihood function is

$$\Lambda(\tilde{\mathbf{b}}_n) = \text{Re}\left\{\sum_{i=0}^{n} e^{-j\theta_{i-1}} \left[\beta_0^*(\tilde{\alpha}_i)x_0(i) + \beta_1^*(\tilde{\alpha}_i)x_1(i)\right]\right\}$$

$$= \underbrace{\text{Re}\left\{\sum_{i=0}^{n-1} e^{-j\theta_{i-1}}\left[\beta_0^*(\tilde{\alpha}_i)x_0(i) + \beta_1^*(\tilde{\alpha}_i)x_1(i)\right]\right\}}_{\Lambda(\tilde{\mathbf{b}}_{n-1})}$$
$$+ \text{Re}\left\{e^{-j\theta_n}\left[\beta_0^*(\tilde{\alpha}_n)x_0(n) + \beta_1^*(\tilde{\alpha}_n)x_1(n)\right]\right\}$$
$$= \Lambda(\tilde{\mathbf{b}}_{n-1}) + \text{Re}\left\{e^{-j\theta_n}\left[\beta_0^*(\tilde{\alpha}_n)x_0(n) + \beta_1^*(\tilde{\alpha}_n)x_1(n)\right]\right\}. \quad (394)$$

This equation shows that the log-likelihood function of the bit sequence $\tilde{\mathbf{b}}_n$ may be computed recursively. This recursion lies at the heart of the Viterbi algorithm.

The Viterbi algorithm is usually expressed in terms of starting states and ending states. The pseudosymbols (382) include a phase state θ_i. Because there are 4 phase states, the sequence of bits (or ternary symbols) define transitions of a finite state machine with 4 states. The finite state machine may be represented by a state transition diagram with 4 states. "Unrolling" the state transition diagram to make time explicit produces a 4-state trellis. The 4-state trellis corresponding to the reduced complexity SOQPSK-TG waveform is illustrated in Figure 160. Some important observations include the following:

1. The trellis is time-varying: the state structure alternates between even-indexed bits and odd-indexed bits. This has strong corollaries to the real/imaginary, even/odd operation in the symbol-by-symbol detectors of Figure 157.

2. The state labels are the previous two bits. For even-indexed bit intervals ($i = 2k$) the bit labels are (b_{2k-2}, b_{2k-1}) whereas for odd-indexed bit intervals ($i = 2k+1$) the bit labels are (b_{2k}, b_{2k-1}).

3. The phases associated with each state are also shown. Note that the phase assignments go with the state labels, not with the bits, per se.

4. Each transition through the state machine is represented by a *branch* in the trellis. Associated with each branch is an "input/output" pair denoted b_i/α_i. The "input" is the i-th bit $b_i \in \{0, 1\}$ and the "output" is the ternary symbol $\alpha_i \in \{-1, 0, +1\}$.

The hypothesized bit sequence $\tilde{\mathbf{b}}_n$ corresponds to a path through the trellis. The trellis path may be defined in terms of its state sequence. That is, there is a one-to-one mapping between the bit sequence $\tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_n$ and a state sequence $\tilde{S}_0, \tilde{S}_1, \ldots, \tilde{S}_n$ where $S_i \in \{00, 01, 10, 11\}$. The

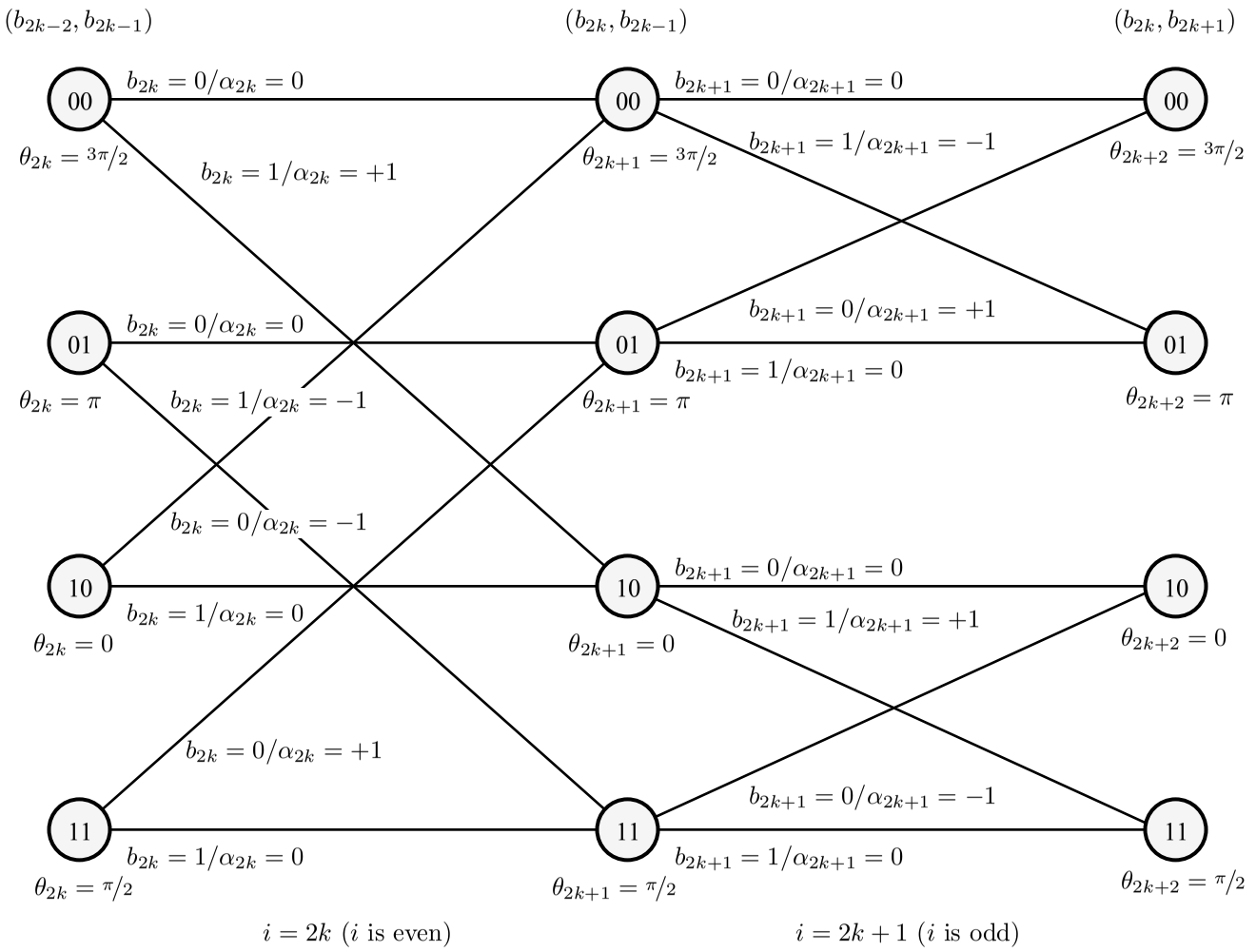

Figure 160: The 4-state trellis associated with the approximate PAM representation for SOQPSK-TG given by (378).

recursion (394) may be restated in terms of the state sequence as

$$\lambda_{n+1}(\tilde{S}_{n+1}=q)=\lambda_n(\tilde{S}_n=p)+M\left(n,\tilde{S}_n=p,\tilde{S}_{n+1}=q\right) \tag{395}$$

where $M\left(n,\tilde{S}_n=p,\tilde{S}_{n+1}=q\right)$ is a restatement of the second term of (394):

$$M\left(n,\tilde{S}_n=p,\tilde{S}_{n+1}=q\right)=\text{Re}\left\{e^{-j\theta_p}\left[\beta_0^*\left(\alpha^{(p,q)}\right)x_0(n)+\beta_1^*\left(\alpha^{(p,q)}\right)x_1(n)\right]\right\} \tag{396}$$

where $\alpha^{(p,q)}$ is the output label on the branch connecting state p to state q. This term is called a *branch metric* in the Viterbi algorithm because its value depends only the results of the integrators (during the time interval corresponding to the branch) and the branch labels. The branch metrics are[11]

$$\begin{aligned}
M(2k,\tilde{S}_{2k}=00,\tilde{S}_{2k}=00)&=\text{Re}\left\{e^{-j\theta_{00}}\left[\beta_0^*(0)x_0(2k)+\beta_1^*(0)x_1(2k)\right]\right\}\\
&=\text{Re}\left\{jx_0(2k)+j\frac{1}{\sqrt{2}}x_1(2k)\right\} &(397)\\
M(2k,\tilde{S}_{2k}=00,\tilde{S}_{2k}=10)&=\text{Re}\left\{e^{-j\theta_{00}}\left[\beta_0^*(+1)x_0(2k)+\beta_1^*(+1)x_1(2k)\right]\right\}\\
&=\text{Re}\left\{x_0(2k)+\frac{1}{\sqrt{2}}(1+j)x_1(2k)\right\} &(398)\\
M(2k,\tilde{S}_{2k}=01,\tilde{S}_{2k}=01)&=\text{Re}\left\{e^{-j\theta_{01}}\left[\beta_0^*(0)x_0(2k)+\beta_1^*(0)x_1(2k)\right]\right\}\\
&=-\text{Re}\left\{x_0(2k)+\frac{1}{\sqrt{2}}x_1(2k)\right\} &(399)\\
M(2k,\tilde{S}_{2k}=01,\tilde{S}_{2k}=11)&=\text{Re}\left\{e^{-j\theta_{01}}\left[\beta_0^*(-1)x_0(2k)+\beta_1^*(-1)x_1(2k)\right]\right\}\\
&=-\text{Re}\left\{jx_0(2k)+\frac{1}{\sqrt{2}}(1+j)x_1(2k)\right\} &(400)\\
M(2k,\tilde{S}_{2k}=10,\tilde{S}_{2k}=00)&=\text{Re}\left\{e^{-j\theta_{10}}\left[\beta_0^*(-1)x_0(2k)+\beta_1^*(-1)x_1(2k)\right]\right\}\\
&=\text{Re}\left\{jx_0(2k)+\frac{1}{\sqrt{2}}(1+j)x_1(2k)\right\} &(401)\\
M(2k,\tilde{S}_{2k}=10,\tilde{S}_{2k}=10)&=\text{Re}\left\{e^{-j\theta_{10}}\left[\beta_0^*(0)x_0(2k)+\beta_1^*(0)x_1(2k)\right]\right\}\\
&=\text{Re}\left\{x_0(2k)+\frac{1}{\sqrt{2}}x_1(2k)\right\} &(402)
\end{aligned}$$

[11]The branch metrics not listed, e.g., $M(2k,\tilde{S}_{2k}=00,\tilde{S}_{2k}=11)$, are those for which there is no connecting branch.

$$M(2k, \tilde{S}_{2k} = 11, \tilde{S}_{2k} = 01) = \text{Re}\left\{e^{-j\theta_{10}}\left[\beta_0^*(+1)x_0(2k) + \beta_1^*(+1)x_1(2k)\right]\right\}$$
$$= -\text{Re}\left\{x_0(2k) + \frac{1}{\sqrt{2}}(1+j)x_1(2k)\right\} \quad (403)$$

$$M(2k, \tilde{S}_{2k} = 11, \tilde{S}_{2k} = 11) = \text{Re}\left\{e^{-j\theta_{10}}\left[\beta_0^*(0)x_0(2k) + \beta_1^*(0)x_1(2k)\right]\right\}$$
$$= -\text{Re}\left\{jx_0(2k) + j\frac{1}{\sqrt{2}}x_1(2k)\right\} \quad (404)$$

$$M(2k+1, \tilde{S}_{2k+1} = 00, \tilde{S}_{2k+2} = 00) = \text{Re}\left\{e^{-j\theta_{00}}\left[\beta_0^*(0)x_0(2k+1) + \beta_1^*(0)x_1(2k+1)\right]\right\}$$
$$= \text{Re}\left\{jx_0(2k+1) + j\frac{1}{\sqrt{2}}x_1(2k+1)\right\} \quad (405)$$

$$M(2k+1, \tilde{S}_{2k+1} = 00, \tilde{S}_{2k+2} = 01) = \text{Re}\left\{e^{-j\theta_{00}}\left[\beta_0^*(-1)x_0(2k+1) + \beta_1^*(-1)x_1(2k+1)\right]\right\}$$
$$= -\text{Re}\left\{x_0(2k+1) + \frac{1}{\sqrt{2}}(1-j)x_1(2k+1)\right\} \quad (406)$$

$$M(2k+1, \tilde{S}_{2k+1} = 01, \tilde{S}_{2k+2} = 00) = \text{Re}\left\{e^{-j\theta_{01}}\left[\beta_0^*(+1)x_0(2k+1) + \beta_1^*(+1)x_1(2k+1)\right]\right\}$$
$$= \text{Re}\left\{jx_0(2k+1) - \frac{1}{\sqrt{2}}(1-j)x_1(2k+1)\right\} \quad (407)$$

$$M(2k+1, \tilde{S}_{2k+1} = 01, \tilde{S}_{2k+2} = 01) = \text{Re}\left\{e^{-j\theta_{01}}\left[\beta_0^*(0)x_0(2k+1) + \beta_1^*(0)x_1(2k+1)\right]\right\}$$
$$= -\text{Re}\left\{x_0(2k+1) + \frac{1}{\sqrt{2}}x_1(2k+1)\right\} \quad (408)$$

$$M(2k+1, \tilde{S}_{2k+1} = 10, \tilde{S}_{2k+2} = 10) = \text{Re}\left\{e^{-j\theta_{10}}\left[\beta_0^*(0)x_0(2k+1) + \beta_1^*(0)x_1(2k+1)\right]\right\}$$
$$= \text{Re}\left\{x_0(2k+1) + \frac{1}{\sqrt{2}}x_1(2k+1)\right\} \quad (409)$$

$$M(2k+1, \tilde{S}_{2k+1} = 10, \tilde{S}_{2k+2} = 11) = \text{Re}\left\{e^{-j\theta_{10}}\left[\beta_0^*(+1)x_0(2k+1) + \beta_1^*(+1)x_1(2k+1)\right]\right\}$$
$$= \text{Re}\left\{-jx_0(2k+1) + \frac{1}{\sqrt{2}}(1-j)x_1(2k+1)\right\} \quad (410)$$

$$M(2k+1, \tilde{S}_{2k+1} = 11, \tilde{S}_{2k+2} = 10) = \text{Re}\left\{e^{-j\theta_{11}}\left[\beta_0^*(-1)x_0(2k+1) + \beta_1^*(-1)x_1(2k+1)\right]\right\}$$
$$= \text{Re}\left\{x_0(2k+1) + \frac{1}{\sqrt{2}}(1-j)x_1(2k+1)\right\} \quad (411)$$

$$M(2k+1, \tilde{S}_{2k+1} = 11, \tilde{S}_{2k+2} = 11) = \text{Re}\left\{e^{-j\theta_{11}}\left[\beta_0^*(0)x_0(2k+1) + \beta_1^*(0)x_1(2k+1)\right]\right\}$$
$$= -\text{Re}\left\{jx_0(2k+1) + j\frac{1}{\sqrt{2}}x_1(2k+1)\right\} \quad (412)$$

The hypothesized search sequences are pruned by selecting one of the two paths entering each

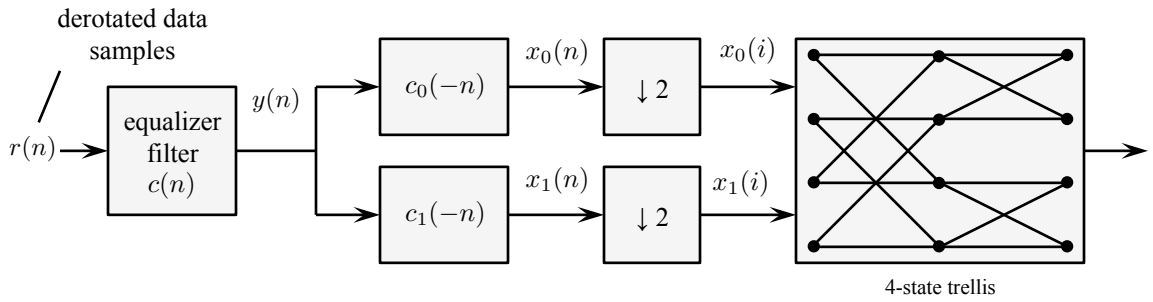

Figure 161: A block diagram of the 4-state trellis detector for SOQPSK-TG. The trellis is described in Figure 160.

state as the survivor. Given the pruning that occurs at each state at the end of each time step, the recursion (395) is interpreted as follows: $\lambda_{n+1}(\tilde{S}_{n+1} = q)$ is the *partial path metric* of the surviving bit sequence terminating in state q. This partial path metric is the sum of the partial path metric of the surviving bit sequence terminating in state p and the branch metric connecting state p to state q. At the end, a backward search through the surviving path identifies the ML bit sequence.

A block diagram of the 4-state trellis detector is illustrated in Figure 161. Here, the integrations defined in (388) are performed by filters matched to the principal pulses $c_0(t)$ and $c_1(t)$. The branch metrics are formed from the matched filter outputs $x_0(i)$ and $x_1(i)$ using (396), [or, in expanded form, (397)–(404) for $i = 2k$ and (405)–(412) for $i = 2k + 1$]. The branch metrics are used by the Viterbi algorithm based on the trellis of Figure 160.

The forgoing analysis assumed perfect timing synchronization and perfect phase synchronization. That is, the detector of Figure 161 based on the trellis shown in Figure 160 and using the branch metrics (396) requires perfect timing and phase synchronization. In the context of data-aided equalization, the equalizer accounts for timing offset in computing the optimum equalizer coefficients (that is, the timing offset is "part of the channel"). Consequently, equalizer output is time synchronized (at least, as best as can be achieved in the given environment), and, when the frequency offset is perfectly known, phase synchronized.

For the case where the frequency offset is not perfectly known (as in the case when a frequency estimator is used), the equalizer output is still time synchronized, but not phase synchronized. This is because any residual frequency offset imparts a linearly increasing (or decreasing) phase shift on the data samples at the equalizer output. This same phase shift applies to the outputs of the filters matched to the principal pulses as well. To compensate for this unknown and linearly increasing (decreasing) phase offset, the branch metric calculations are modified using a technique known as per-survivor processing [53, 54]. The approach taken here is a generalization of the application of

per-survivor processing to CPFSK described by Miller [55]. We begin by introducing an additional variable $\Delta\theta(i)$ for each state at trellis step i. This variable tracks the phase offset at each state for each surviving hypothesis sequence. We use the notation $\Delta\theta_p(i)$ for $p \in \{00, 01, 10, 11\}$.

The Viterbi algorithm proceeds as usual except with the following changes. At time step n, compute the temporary variable

$$Z\left(n, \tilde{S}_n = p, \tilde{S}_{n+1} = q\right) = e^{-j\theta_p}\left[\beta_0^*\left(\alpha^{(p,q)}\right) x_0(n) + \beta_1^*\left(\alpha^{(p,q)}\right) x_1(n)\right] \tag{413}$$

where, as before, $\alpha^{(p,q)}$ is the output label on the branch connecting state p to state q. The partial path metric for the path terminating in state q at time $n+1$ is

$$\lambda_{n+1}(\tilde{S}_{n+1} = q) = \lambda_n(\tilde{S}_n = p) + M\left(n, \tilde{S}_n = p, \tilde{S}_{n+1} = q\right) \tag{414}$$

where

$$M\left(n, \tilde{S}_n = p, \tilde{S}_{n+1} = q\right) = \text{Re}\left\{e^{-j\Delta\theta_p(n)} Z\left(n, \tilde{S}_n = p, \tilde{S}_{n+1} = q\right)\right\}. \tag{415}$$

In addition to updating the partial path metric, the phase shift is also updated:

$$\Delta\theta_q(n+1) = \Delta\theta_p(n) + K_1 \times \text{Im}\left\{e^{-j\Delta\theta_p(n)} Z\left(n, \tilde{S}_n = p, \tilde{S}_{n+1} = q\right)\right\}. \tag{416}$$

The phase increment $\text{Im}\{\cdots\}$ in (416) is identical to the phase error term in the PLL of the SxS detector shown in Figure 157. Consequently, the constant K_1 in (416) can be the same as the constant used in the PLL of the SxS detector.

In summary, the 4-state trellis detector based on per-survivor processing is given by the block diagram in Figure 161. The corresponding trellis is that of Figure 160. The partial path metrics and branch metrics are given by (414) – (416).

The bit error rate performance using the 4-state trellis defined by the PAM approximation (378) is presented in Figure 10 of [21]. These results show that the detector is 0.1 dB worse than the MLSD bound (373) in AWGN.

A.4 Concluding Remarks

In real operating environments, the SOQPSK-TG detector must deal with frequency offsets and phase noise. The SxS and 4-state trellis detectors each have advantages and disadvantages in such environments. Experience with these detectors suggests the best performance in the presence of a frequency offset and phase noise is obtained using a hybrid approach. The SxS detector with a

second-order PLL is used to remove the frequency offset, but is not used for detection. Detection is performed using the 4-state trellis detector with per-survivor processing based on a first order PLL as described in the previous section.

The SxS detector with a second-order PLL is a generalization of the system in Figure 157 (b) where the gain block — with gain K_1 — is replaced by proportional-plus-integrator loop filter with transfer function

$$F(z) = K_1 + \frac{K_2}{1 - z^{-1}}. \tag{417}$$

An analysis of the second order loop together with the relationship between the filter constants (K_1, K_2) and the loop bandwidth is presented in Appendix C of [25].

References

[1] M. Rice, "Final report: Multipath modeling and mitigation using multiple antennas (M4A)," tech. rep., Brigham Young University, 2013. Submitted to the Spectrum Efficient Technologies (SET) Office of the Science & Technology, Test & Evaluation (S&T/T&E) Program, Test Resource Management Center (TRMC). Also available on-line at http://hdl.lib.byu.edu/1877/3241.

[2] R. Barker, "Group synchronization of binary digital systems," in *Communication Theory* (W. Jackson, ed.), London: Butterworth, 1953.

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

[4] G. Lui and H. Tan, "Frame synchronization for gaussian channels," *IEEE Transactions on Communications*, vol. 35, pp. 818–829, August 1987.

[5] J. Gansman, M. Fitz, and J. Krogmeier, "Optimum and suboptimum frame synchronization for pilot-symbol-assisted modulation," *IEEE Transactions on Communications*, vol. 45, pp. 1327–1337, October 1997.

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

[7] 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, April 2010.

[8] D. Rife and R. Boorstyn, "Single tone parameter estimation from discrete-time observations," *IEEE Transactions on Information Theory*, vol. 20, p. 591, September 1974.

[9] S. Kay, "A fast and accurate single frequency estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1987–1990, December 1989.

[10] M. Fitz, "Further results in the fast estimation of a single frequency," *IEEE Transactions on Communications*, vol. 42, p. 862, February/March/April 1994.

[11] M. Luise and R. Reggiannini, "Carrier frequency recovery in all-digital modems for burst-mode transmissions," *IEEE Transactions on Communications*, vol. 43, p. 1169, February/March/April 1995.

[12] U. Mengali and M. Morelli, "Data-aided frequency estimation for burst digital transmission," *IEEE Transactions on Communications*, vol. 45, p. 23, January 1997.

[13] O. Baksalary, D. Bernstein, and G. Trenkler, "On the equality between rank and trace of an idempotent matrix," *Applied Mathematics and Computation*, vol. 217, pp. 4076–4080, 2010. Also available at `http://www-personal.umich.edu/~dsbaero/library/BaksalaryBernstein.pdf` (accessed 5 April 2014).

[14] J. Proakis and M. Salehi, *Digital Communications*. New York: McGraw-Hill, 5th ed., 2008.

[15] M. Hayes, *Statistical Digital Signal Processing and Modeling*. New York: John Wiley & Sons, 1996.

[16] Z. Ye, E. Satorius, and T. Jedrey, "Enhancement of advanced range telemetry (ARTM) channels via blind equalization," in *Proceedings of the International Telemetering Conference*, (Las Vegas, NV), October 2001.

[17] T. Hill and M. Geoghegan, "A comparison of adaptively equalized PCM/FM, SOQPSK, and multi-h CPM in a multipath channel," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 2002.

[18] M. Geoghegan, "Experimental results for PCM/FM, Tier I SOQPSK, and Tier II Multi-h CPM with CMA equalization," in *Proceedings of the International Telemetering Conference*, (Las Vegas, NV), October 2003.

[19] E. Law, "How well does a blind adaptive CMA equalizer work in a simulated telemetry multipath environment?," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 2004.

[20] 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.

[21] E. Perrins, "FEC systems for aeronautical telemetry," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, pp. 2340–2352, October 2013.

[22] R. Johnson *et al.*, "Blind equalization using the constant modulus criterion: A review," *Proceedings of the IEEE*, vol. 86, pp. 1927–1950, October 1998.

[23] S. Barbarossa and A. Scaglione, "Blind equalization using cost functions matched to the signal constellation," in *Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers*, vol. 1, (Pacific Grove, CA), pp. 550–554, November 1997.

[24] A. Beasley and A. Cole-Rhodes, "Performance of an adaptive blind equalizer for QAM signals," in *Proceedings of the IEEE Military Communications Conference*, (Atlantic City, NJ), October 2005.

[25] M. Rice, *Digital Communications: A Discrete-Time Approach*. Upper Saddle River, NJ: Pearson Prentice-Hall, 2009.

[26] W. Lee and F. Hill, "A maximum-likelihood sequence estimator with decision feedback equalization," *IEEE Transactions on Communications*, vol. 25, pp. 971–979, September 1977.

[27] J. Bergmans, S. Rajput, and F. V. D. Laar, "On the use of decision feedback for simplifying the Viterbi decoder," *Philips Journal of Research*, vol. 42, no. 4, pp. 399–428, 1987.

[28] K. Wesolowski, "Efficient digital receiver structure for trellis-coded signal s transmitted through channels with intersymbol interference," *Electronics Letters*, vol. 23, pp. 1265–1267, November 1987.

[29] T. Hashimoto, "A list-type reduced-constraint generalization of the Viterbi algorithm," *IEEE Transactions on Information Theory*, vol. 33, pp. 866–876, November 1987.

[30] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Transactions on Communications*, vol. 37, pp. 428–436, May 1989.

[31] P. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Transactions on Communications*, vol. 37, pp. 669–676, July 1989.

[32] M. Eyuboğlu and S. Qureshi, "Reduced-state sequence estimation for coded modulation on intersymbol interference channels," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 989–995, August 1989.

[33] M. Eyuboğlu and S. Qureshi, "Reduced-state sequence estimation with set partition and decision feedback," *IEEE Transactions Communications*, vol. 36, pp. 13–20, January 1988.

[34] R. Martin *et al.*, "Unification and evaluation of equalization structures and design algorithms for discrete multitone modulation systems," *IEEE Transactions on Signal Processing*, vol. 53, pp. 3880–3894, October 2005.

[35] P. Bello and K. Pahlavan, "Adaptive equalization for SQPSK and SQPR over frequency selective microwave LOS channels," *IEEE Transactions on Communications*, vol. 32, pp. 609 – 615, May 1984.

[36] J. Tu, "Optimum MMSE equalization for staggered modulation," in *Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA), pp. 1401 – 1405, November 1993 1993.

[37] M. Rice and E. Satorius, "A comparison of MMSE and CMA equalization techniques for ARTM Tier-1 waveforms," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 2004.

[38] M. Rice and E. Satorius, "Equalization techniques for multipath mitigation in aeronautical telemetry," in *Proceedings of the IEEE Military Communications Conference*, (Monterey, CA), November 2004.

[39] M. Rice, G. Narumanchi, and M. Saquib, "A new approach to multipath mitigation in aeronautical telemetry," in *Proceedings of the International Telemetering Conference*, (Las Vegas, NV), October 2011.

[40] J. Coon, M. Sandell, M. Beach, and J. McGeehan, "Channel and noise variance estimation and tracking algorithms for unique-word based single-carrier systems," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 1488–1496, June 2006.

[41] Secretariat, Range Commanders Council, White Sands Missile Range, New Mexico, *IRIG Standard 106-09: Telemetry Standards*, 2009. (Available on-line at http://www.irig106.org/docs/106-09).

[42] P. S. Leung and K. Feher, "F-QPSK–a superior modulation technique for mobile and personal communications," *IEEE Transactions on Broadcasting*, vol. 39, pp. 288–294, June 1993.

[43] R. P. Jefferis, "Evaluation of constant envelope offset quadrature phase shift keying transmitters with a software based signal analyzer," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 2004.

[44] T. J. Hill, "An enhanced, constant envelope, interoperable shaped offset QPSK (SOQPSK) waveform for improved spectral efficiency," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 2000.

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

[46] L. Li and M. K. Simon, "Performance of coded OQPSK and MIL-STD SOQPSK with iterative decoding," *IEEE Transactions on Communications*, vol. 52, pp. 1890–1900, November 2004.

[47] M. Geoghegan, "Optimal linear detection of SOQPSK," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 2002.

[48] M. Geoghegan, "Implementation and performance results for trellis detection of SOQPSK," in *Proceedings of the International Telemetering Conference*, (Las Vegas, NV), October 2001.

[49] M. Geoghegan, "Bandwidth and power efficiency trade-offs of SOQPSK," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 2002.

[50] T. Nelson, E. Perrins, and M. Rice, "Near optimal common detection techniques for shaped offset QPSK and Feher's QPSK," *IEEE Transactions on Communications*, vol. 56, pp. 724–735, May 2008.

[51] B. Widrow and J. McCool, "A comparison of adaptive algorithms based on the methods of steepest descent and random search," *IEEE Transactions on Antennas and Propagation*, vol. 24, pp. 615–637, September 1976.

[52] E. Perrins and M. Rice, "PAM representation of ternary CPM," *IEEE Transactions on Communications*, vol. 56, pp. 2020–2024, December 2008.

[53] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Transactions on Communications*, vol. 43, pp. 354–364, February/March/April 1995.

[54] E. Esteves and R. Sampaio-Neto, "A per-survivor phase acquisition and tracking algorithm for detection of TCM signals with phase jitter and frequency error," *IEEE Transactions on Communications*, vol. 45, pp. 1381–1384, November 1997.

[55] M. Miller, "Detection of CPFSK signals using per survivor processing," in *Proceedings of the IEEE Military Communications Conference*, (Boston, MA), pp. 524–528, October 1998.