Synchronization Techniques for Burst-Mode Continuous Phase Modulation

Ph.D. Dissertation Defense
Ehsan Hosseini

Department of Electrical Engineering & Computer Science
University of Kansas
Outline

• Chapter 1: *Introduction*

• Chapter 2: *The Cramér-Rao Bound for Training Sequence Design for Burst-Mode CPM*

• Chapter 3: *Timing, Carrier and Frame Synchronization for Burst-Mode CPM*

• Chapter 4: *Applications to SOQPSK*

• Chapter 5: *Conclusions*
External Contributions

• Chapter 1: Introduction

• Chapter 2: Training Sequence Design for Burst-Mode CPM

• Chapter 3: Timing, Carrier and Frame Synchronization for Burst-Mode CPM

• Chapter 4: Applications to SOQPSK

• Chapter 5: Conclusions
Motivation: Aeronautical Telemetry

- integrated Network Enhanced Telemetry (iNET) project
  - Packet-based telemetry in contrast to streaming telemetry.
  - Benefits: spectrum conservation, multi-hop, enhanced mobility,
  - A major challenge in the RF link is detection and synchronization of packets, i.e. bursts
  - Physical layer waveform: SOQPSK, a subset of CPM

- In this work:
  1. Synchronization for general CPM in burst-mode TX
  2. Apply the results to SOQPSK, i.e. iNET standard
What is CPM?

- Continuous phase modulation:
  \[ s(t; \alpha) = \sqrt{\frac{E_s}{T_s}} \exp \left\{ 2\pi h \sum_i \alpha_i q(t - iT_s) \right\} \]

  - \( \alpha_i \): M-ary data symbols
  - \( h \): modulation index
  - \( q(t) \): phase response

- Frequency pulse:
  - \( g(t) = \frac{dq(t)}{dt} \)
  - Duration of \( LT_s \)

- Properties:
  - Constant envelope
  - Bandwidth efficient
  - **Drawback**: complex at Rx

\[ L=1: \text{full-response} \]
\[ L>1: \text{partial-response} \]
Signal Transmission Model

- **Burst-Mode Transmission:**
  - **Burst Structure:**
    - Preamble (training sequence): sequence of $L_0$ known symbols
    - Unique Word (UW): burst identification
    - Payload: information bits
  - Received baseband signal:
    \[ r(t) = e^{j(2\pi f_d t + \theta)} s(t - \tau) + w(t) \]
    - $f_d$: Frequency offset
    - $\theta$: Carrier phase
    - $\tau$: time delay
  - Noise model: AWGN

*Synchonization Parameters (Unknown)*
Synchronization Problem

- **Challenge:** Acquisition time must be kept minimum in burst-mode communications to save bandwidth, thus:
  1. Feedforward approach for shorter acquisition time
  2. Data-aided (DA) algorithms to assist synchronization

- **Problems:**
  1. Preamble design: What is the best preamble?
  2. Timing & carrier recovery: Estimation of \([f_d, \theta, \tau]\)
  3. Frame synchronization: Detection of preamble boundaries
Comparison with Related Works

This Work

- CPM
  - Frequency
  - Phase
  - Symbol Timing

Related Work

- PSK
- Debora (2002)
- MSK
- Only Symbol Timing

- Zhao (2006)
- CPM
- Phase & Timing (No Freq.)
- Morelli (2007)
- PSK

Best Preamble

Timing & Carrier Recovery

Frame Synchronization

- Choi (2002)
- PSK
- Continuous Transmission

E. Hosseini Ph.D. Defense: Synchronization Techniques for Burst-Mode CPM
The Cramér-Rao Bound for Training Sequence Design for Burst-Mode CPM
Optimum Training Sequence Design: Cramér-Rao Bound Method

- Two approaches can be taken in order to derive the optimum (best) training sequence for estimation of $\mathbf{u} = \left[ f_d, \theta, \tau \right]^T$
  
  1. Minimize the estimation error variance:

  $$\mathbf{a}_{opt} = \arg\min_{\alpha} E \left[ (\hat{u}_i - u_i)^2 \mid \alpha \right]$$

  2. Minimize the CRB, i.e. a theoretical lower bound on estimation error variance of any unbiased estimator:

  $$\text{MSE} \rightarrow E \left[ (\hat{u}_i - u_i)^2 \mid \alpha \right] \geq \text{CRB} \left( u_i \mid \alpha \right) \Rightarrow \mathbf{a}_{opt} = \arg\min_{\alpha} \{ \text{CRB} \left( u_i \mid \alpha \right) \}$$

- CRB is computed via

  $$\text{CRB} \left( u_i \mid \alpha \right) = \mathbf{I}(\mathbf{u})^{-1}_{i,i} \quad \text{where} \quad \mathbf{I}(\mathbf{u})_{i,j} = -\mathbf{E} \left[ \frac{\partial^2}{\partial u_i \partial u_j} \ln(p(r; \mathbf{u}, \alpha)) \right]$$

  \[\text{Conditional CRB} \quad \text{Fisher Info. Mat. (FIM)} \quad \text{Likelihood Function}\]
Overview of CRB Calculations

- Closed-form FIM for CPM and joint estimation

\[
I(u) = \frac{1}{T_s} \left( \frac{E_s}{N_0} \right) \begin{bmatrix}
8\pi^2 T_0^3 / 3 & 2\pi T_0^2 & -8\pi^2 hA \\
2\pi T_0^2 & 2T_0 & -2\pi hB \\
-8\pi^2 hA & -2\pi hB & 8\pi^2 h^2 C
\end{bmatrix}
\]

- It is shown the optimum training sequence must satisfy these two conditions at the same time
  1. \( A' = B' = 0 \)
      \[
      A' = \sum_{i=1}^{L_0} \alpha_{i-1} = 0
      \]
      \[
      B' = \sum_{i=1}^{L_0} \alpha_{i-1} = 0
      \]
  2. Maximize \( C \)
      \[
      C \approx R_g(0) \sum_{i=0}^{L_0-1} \alpha_i^2 + 2R_g(T_s) \sum_{i=0}^{L_0-2} \alpha_i a_{i+1} \quad \text{where} \quad R_g(t) = \int_{-\infty}^{+\infty} g(u)g(u+t)du
      \]
Proposed Optimum Training Sequence

- Proposed sequence based on our analysis

- It satisfies $A' = B' = 0$,
  1. Optimum sequence for full-response CPM.
  2. Only 2 transitions => Near-optimum sequence for partial-response CPM

\[
(1) A' = \sum_{i=1}^{L_0} i \alpha_{i-1} = 0 \quad (2) B' = \sum_{i=1}^{L_0} \alpha_{i-1} = 0
\]

(3) Maximize $C \approx R_g(0) \sum_{i=0}^{L_0-1} \alpha_i^2 + 2R_g(T_s) \sum_{i=0}^{L_0-1} \alpha_i a_{i+1}$ = 0 for full-response CPM

- We can reduce the approximations for partial-response by using $L_1 = L_0 - \lfloor L/2 \rfloor$ rather than $L_0$ to reduce the edge-effect (EF)
Findings and Properties of the Optimum Sequence

1. The same sequence is optimum for frequency, phase, and timing, which is opposed to linear modulations.
2. It follows the same pattern for all CPMs.
3. The minimized frequency and phase CRBs are independent of underlying CPM parameters.
4. The minimized symbol timing CRB do depend on CPM parameters such as frequency pulse.
5. It decouples symbol timing estimation from phase and frequency offset estimates.

\[ I^*(\mathbf{u}) = \frac{1}{T_s} \left( \frac{E_s}{N_0} \right) \begin{bmatrix} 8\pi^2T_0^3 / 3 & 2\pi T_0^2 & 0 \\ 2\pi T_0^2 & 2T_0 & 0 \\ 0 & 0 & 8\pi^2h^2C^* \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \]

- \( u_1 \) : frequency
- \( u_2 \) : phase
- \( u_3 \) : timing
More Findings

6. The proposed sequence is \textit{asymptotically} optimum for partial-response CPMs

\[
\frac{\text{CRB}(\tilde{\alpha} \mid \tau)}{\text{CRB}(\alpha^* \mid \tau)} = \frac{\tilde{C} - 8R_g(T_s)}{\tilde{C}} = 1 - \frac{8R_g(T_s)}{L_0 R_g(0) + 2(L_0 - 1)R_g(T_s)} \quad L_0 \to \infty
\]

\(\tilde{\alpha}: \) Hypothetical sequence

\(\alpha^*: \) Proposed sequence

7. Computer search confirms the proposed sequence at small \(L_0\)

8. We also computed CRB for random sequences and compared with the optimum one:

- The optimum sequence delivers significant gain for symbol timing estimation of partial-response and/or M-ary CPMs.
- The frequency and phase estimation are less sensitive to the selection of training sequence especially for long sequences.
Timing, Carrier, and Frame Synchronization of Burst-Mode CPM
Timing and Carrier Recovery Framework

- Received Signal: \( r(t) = e^{j(2\pi f_d t + \theta)} s(t - \tau) + w(t) \)
- Decompose timing delay:

\[
\tau = \mu T_s + \varepsilon T_s \quad \text{where} \quad \begin{cases} 
\mu \geq 0 : \text{integer delay} \\
-0.5 < \varepsilon < 0.5 : \text{fractional delay}
\end{cases}
\]

- For timing and carrier recovery, we assume integer delay is known
- Joint maximum likelihood (ML) estimator:

\[
\left[ \hat{f}_d, \hat{\theta}, \hat{\varepsilon} \right] = \arg \max_{[f_d, \theta, \varepsilon]} \left\{ \Lambda[r(t); f_d, \theta, \varepsilon] = \text{Re} \left[ \int_{T_0 + \varepsilon T_s}^{T_0 + (\varepsilon + 1) T_s} e^{-j(2\pi f_d t + \theta)} r(t) s^*(t - \varepsilon T_s) dt \right] \right\}
\]

- \( f_d \) and \( \varepsilon \) are embedded inside the above integral \( \rightarrow \) It requires a 2-dimensional grid search
- **Objective**: decouple timing and frequency in the LLF
The Trick: Piecewise Linear Approximation

- Phase response of the optimum sequence

\[ \text{GMSK (BT_=0.3, M=2, h=1/2)} \]

- Similar and consequent symbols cause the phase to increase/decrease uniformly

- We use linear approximation of the phase in each segment

- There is a lag for partial-response CPMs → \( T_l \)
Approximated Log-Likelihood Function (LLF)

- **Discrete-time LLF:**
  \[
  \Lambda(r; \nu, \theta, \varepsilon) \approx \Re \left[ e^{j(2\pi\nu + \theta)} \sum_{n=0}^{NL_0 - 1} e^{-j(2\pi\nu + \theta)} r[n] s^*_\varepsilon[n] \right]
  \]
  \(\nu\): normalized freq. offset
  \(\varepsilon\): fractional delay
  \(s^*_\varepsilon[n]\): baseband signal shifted by \(\varepsilon\)

- **Approximated baseband signal:**
  \[
  \phi(t, \alpha) = 2\pi h \sum_{i=0}^{L_0 - 1} \alpha^*_i q(t - i T_s)
  \]
  **Linear phase approximation**
  **Original phase**
  \[
  \phi(t, \alpha) \approx \begin{cases} 
  -(M - 1)\pi h(t - T_i) / T_s & T_i \leq t < T_0 / 4 + T_i \\
  (M - 1)\pi h(t - T_0 / 2 - T_i) / T_s & T_0 / 4 + T_i \leq t < 3T_0 / 4 + T_i \\
  -(M - 1)\pi h(t - T_0 - T_i) / T_s & 3T_0 / 4 + T_i \leq t < T_0 + T_i 
  \end{cases}
  \]
  
  \[
  s^*_\varepsilon[n] \approx \begin{cases} 
  \exp[-j(M - 1)\pi h(n / N - \varepsilon)] & n \leq n < NL_0 / 4 \\
  \exp[+j(M - 1)\pi h(n / N - L_0 / 2 - \varepsilon)] & NL_0 / 4 \leq n < 3NL_0 / 4 \\
  \exp[-j(M - 1)\pi h(n / N - L_0 - \varepsilon)] & 3NL_0 / 4 \leq n < NL_0
  \end{cases}
  \]
  will be used in the above LLF
Maximization of the LLF

- Our approximation decouples timing and frequency offsets:
  \[
  \Lambda^* (r; \nu, \theta, \varepsilon) \approx \text{Re} \left\{ e^{-j\theta} \left[ e^{-j(M-1)\pi \varepsilon} \lambda_1 (\nu) + e^{j(M-1)\pi \varepsilon} \lambda_2 (\nu) \right] \right\}
  \]

- The maximization of the LLF becomes straightforward:
  - Step 1: Frequency offset estimation:
    \[
    \hat{\nu} = \arg \max_{\nu} \left\{ |\lambda_1 (\nu)| + |\lambda_2 (\nu)| \right\}
    \]
    Grid search
  - Step 2: Symbol timing estimation:
    \[
    \hat{\varepsilon} = \arg \left\{ \frac{\lambda_1 (\hat{\nu}) \lambda_2^* (\hat{\nu})}{2(M - 1)\pi h} \right\}
    \]
  - Step 3: Carrier phase estimation:
    \[
    \hat{\theta} = \arg \left\{ e^{-j(M-1)\pi \hat{\varepsilon}} \lambda_1 (\hat{\nu}) + e^{j(M-1)\pi \hat{\varepsilon}} \lambda_2 (\hat{\nu}) \right\}
    \]
Implementation of Frequency Estimator

- The received $r[n]$ is pre-processed

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

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

- $\lambda_1(\nu)$ and $\lambda_2(\nu)$ are computed using FFT operations:

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

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

- The precision can be increased via:
  1. Zero padding prior the FFTs $\rightarrow K_f$
  2. Interpolation of FFT results

![True maximum](image-url)
Joint ML Estimator Block Diagram

\[
\exp(j(M-1)\pi h n/N)
\]

\[
n = \frac{NL_0}{4}
\]

\[
r_1[n] \quad r'_1[n]
\]

\[
r_2[n] \quad r'_2[n]
\]

\[
\exp(-j(M-1)\pi h n/N)
\]

\[
\text{FFT} \quad |\cdot|
\]

\[
\text{MAX} \quad \text{Interpolator}
\]

Frequency Offset (\( \hat{\nu} \))

\[
r'_1[n]
\]

\[
\sum_n (\cdot) \quad \lambda_1
\]

\[
\exp(-j2\pi n(\cdot))
\]

\[
\sum_n (\cdot) \quad \lambda_2
\]

\[
(\cdot)^*
\]

Symbol Timing (\( \hat{\varepsilon} \))

\[
r'_2[n]
\]

\[
\exp\{-j(M-1)\pi h(\cdot)\}
\]

\[
\text{arg}(\cdot)
\]

Carrier Phase (\( \hat{\theta} \))

\[
\lambda_1
\]

\[
\lambda_2
\]
Similar performance for different CPM schemes with $L_0=64$
Within 0.5 dB of CRB at low to medium SNRs
Estimator’s Performance: Timing Offset

- The error variance attains the CRB for various CPMs.
- Exception: 1RC due to large deviations from our template signal.
Frame Synchronization: Overview

- Prior to Timing and Carrier recovery we must determine preamble boundaries, i.e. start-of-signal (SoS), in 2 steps:
  1. **SoS detection algorithm**: Detects the arrival of a new burst and *tentative* location of the SoS
  2. **SoS estimation algorithm**: Estimates the exact location of SoS within an observation window

\[ N_w : \text{Observation window length} \]
\[ N_p : \text{Preamble length} \]
\[ \delta : \text{Estimation parameter} \]
SoS Estimation Algorithm

• Apply ML estimation rules to the observation window

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

• Unwanted parameters, \([\nu, \theta, \alpha_d]\), are averaged out

\[
p(r; \delta) \approx C(\delta) \left( \sum_{n=\delta}^{N_w-1} |r[n]|^2 + 2 \sum_{d=1}^{N_p-\delta-1} \sum_{n=\delta}^{N_p-\delta-d-1} r^*[n]r[n + d]s[n - \delta]s^*[n + d - \delta] + R_{ss}(d) \sum_{n=N_p+\delta}^{N_w-d-1} r^*[n]r[n + d] \right)
\]

• Find \(\delta\) for which the likelihood function is maximized

\[
\hat{\delta} = \arg \max_{\delta} C(\delta) \left( \sum_{n=\delta}^{N_w-1} |r[n]|^2 + 2 \sum_{d=1}^{D} \sum_{n=\delta}^{N_p+\delta-d-1} r^*[n]r[n + d]s[n - \delta]s^*[n + d - \delta] + R_{ss}(d) \sum_{n=N_p+\delta}^{N_w-d-1} r^*[n]r[n + d] \right)
\]

\[
C(\delta) = (N_w - \delta)^q \quad q > 0 : \text{optimized via simulations} \quad 1 \leq D < N_p : \text{Complexity factor}
\]
SoS Detection Algorithm

• We apply a simple likelihood ratio test (LRT)

\[
L(r_p) = \frac{p(r_p; H_1)}{p(r_p; H_0)} \geq \gamma
\]

where:
- \( r_p \): vector of \( N_p \) samples
- \( H_0 \): No preamble
- \( H_1 \): Fully-aligned preamble
- \( \gamma \): Test threshold

• Using the same techniques as SoS estimator, the LRT becomes

\[
L_{D'}(r_p) = \sum_{d=1}^{D'} \left[ \sum_{n=0}^{N_p-d-1} r_s[n] r[n+d] s[n] s^*[n+d] \right] \begin{cases} 1 & \text{if } H_1 \\ 0 & \text{if } H_0 \end{cases} \geq \gamma_{D'}
\]

- \( 1 \leq D' < N_p \): determines complexity
- \( \gamma_{D'} \): test threshold; selected for a given probability of false alarm (Neyman-Pearson criterion).
- ROC: \( P_{FA} = \Pr\{L_{D'}(r_p) > \gamma_{D'} \mid H_0\} \) vs. \( P_D = 1 - \Pr\{L_{D'}(r_p) < \gamma_{D'} \mid H_1\} \)
- The LRT is performed using a sliding window.
Frame Synchronization Performance

- Probabilities are computed via simulations
- Probability of false lock: $P_{FL} = \Pr\{\delta \neq \delta\}$
Putting Everything Together: Simulating a Burst-Mode CPM Receiver

- Transmit bursts of optimum preamble \((L_0)\) + 4096 information bits
- Apply AWGN + random frequency, phase and timing offsets
- Receiver’s structure:

\[
\begin{align*}
r[n] & \rightarrow \text{SoS Detector} \rightarrow \text{SoS Estimator} \rightarrow \text{Joint ML Estimator} \rightarrow \text{MFs} \rightarrow \text{Viterbi Algorithm} \rightarrow \hat{a}_i \\
& \quad \quad \quad \quad \text{Frame Synchronization} \quad \quad \quad \quad \text{Timing & Carrier Recovery} \quad \quad \quad \quad \text{CPM Demodulator+ PLL}
\end{align*}
\]

- Design parameters:
  - \(N=2, K_f=2\), Gaussian interpolator
  - \(D=D'=4, N_w = 2NL_0\)
  - \(\gamma_{D'}\) optimized for each scenario using simulations
Bit Error Rate (BER) Performance

GMSK \((h=1/2, \ M=2)\)

2RC \((h=1/4, \ M=4)\)

- A preamble of 64 bits performs within 0.1 dB of the ideal synchronization
Applications to SOQPSK (iNET)
What is SOQPSK?

- An OQSPK-like modulation, which has a CPM representation

\[ a_n \in \{-1,1\} \] Precoder \[ \alpha_n \in \{-1,0,1\} \] CPM Modulator \[ s(t) \]

- The precoder increases spectral efficiency of SOQPSK compared to a binary CPM by imposing:
  - \( \alpha_{n-1} = 1 \) cannot be followed by \( \alpha_n = -1 \) and vice versa
- Partial-response SOQPSK-TG is adopted by the iNET standard
iNET Synchronization Preamble

- Length of $L_0 = 128$ bits
- Periodic and consists of repeating a sequence of 16 bits 8 times as
  \[
  \begin{align*}
  a_{2k} &= 1,0,1,0,1,0,1,0 \\
  a_{2k+1} &= 1,0,1,1,0,1,0,0
  \end{align*}
  \] for $k = 0, \ldots, 7$

- One period in terms of ternary symbols

![Diagram of ternary symbols over time index]
Burst-Mode Synchronization of SOQSPK-TG: iNET Preamble

- We apply our CPM techniques to SOQPSK by considering ternary symbols
- iNET preamble phase response

- SOQPSK-TG’s response is approximated by SOQPSK-MIL’s
- Derivation of joint ML estimation algorithm is similar to the CPM’s when \( h=1/2 \) and \( M=2 \)
Estimation Error Variances for SOQPSK: Optimum vs. iNET preambles

- Performance degradation due to transitions in the iNET preamble
- Phase error variance has the same behavior as the frequency’s
Overall Performance of SOQPSK: Optimum vs. iNET preambles

- iNET preamble performs very close to the optimum one when $L_0=128$
- The optimum preamble with $L_0=64$ is within 0.2 dB of the iNET’s
- False locks for $L_0=32$ at low SNRs

<table>
<thead>
<tr>
<th>Preamble</th>
<th>dB loss @ BER=10^{-4}</th>
</tr>
</thead>
<tbody>
<tr>
<td>iNET</td>
<td>0.4</td>
</tr>
<tr>
<td>$L_0=128$</td>
<td>0.3</td>
</tr>
<tr>
<td>$L_0=64$</td>
<td>0.6</td>
</tr>
<tr>
<td>$L_0=32$</td>
<td>1</td>
</tr>
</tbody>
</table>
Conclusions

• Gave an overview of feedforward synchronization methods for CPM
• Derived the optimum training sequence for CPM in AWGN
  ▪ Minimizes the CRBs for all three synchronization parameters
  ▪ Applicable to the entire CPM family including M-ary and partial response
• Developed a joint ML DA estimation algorithm for timing and carrier recovery
  ▪ Designed based on the optimized training sequence
  ▪ Performs within 0.5 dB of the CRB at SNRs as low as 0 dB for $L_0 = 64$
• Simulated a burst-mode CPM receiver: only 0.1 dB SNR loss with the optimum preamble of $L_0=64$
• Applied CPM results to SOQSPK for iNET standard
• Final result: developed a complete feedforward DA synchronization scheme for general CPM including: best training sequence, timing and carrier recovery, and frame synchronization
Areas of Future Study

• Application to frequency-selective channels:
  ▪ Training sequence design
  ▪ Timing and carrier recovery
  ▪ Frame synchronization

• Joint synchronization and channel estimation

• Analysis of frame synchronization algorithm as it can be applied to other modulations,
  ▪ Probability of false lock (SoS estimator)
  ▪ Probabilities of false alarm and detection (SoS detector)

• Hardware implementation and trade-offs between complexity and performance
Thank You!

Questions?

I wonder if this technique would work for my problem

Please let it be OVER…

Interesting

Dunno, I’m playing Fruit Ninja

What does that slide say?
Derivation of CRB

- FIM, for CPM and AWGN, is computed via

\[ I_{i,j}(u) = -\frac{2}{N_0} \int_0^{T_0} \text{Re} \left[ s(t, u, \alpha) \frac{\partial^2 s^*(t, u, \alpha)}{\partial u_i \partial u_j} \right] dt \]

- Closed-form FIM for CPM and joint estimation:

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

\[ s(t, u, \alpha) = \sqrt{\frac{E_s}{T_s}} e^{i(2\pi f_d t + \theta)} e^{i\phi(t - \tau, u)} \]

- Received Preamble w/o noise

\[ T_0 = L_0 T_s \]

- Preamble Duration

\[ A = \sum_{i=0}^{L_0-1} \alpha_i \int_0^{T_0} g(t - i T_s - \tau) dt \]

\[ B = \sum_{i=0}^{L_0-1} \int_0^{T_0} g(t - i T_s - \tau) dt \]

\[ C = \sum_{i=0}^{L_0-1} \sum_{j=0}^{L_0-1} \alpha_i \alpha_j \int_0^{T_0} g(t - i T_s - \tau) g(t - j T_s - \tau) dt \]

\[ g(t) : \text{CPM Frequency Pulse} \]
Training Sequence for Symbol Timing

• Symbol Timing CRB should be minimized,

\[
\text{CRB}(\tau \mid \alpha) = \left( 8\pi^2 h^2 C - \left[ -8\pi^2 hA - 2\pi hB \right] \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix}^{-1} \left[ \begin{bmatrix} -8\pi^2 hA \\ -2\pi hB \end{bmatrix} \right] \right)^{-1}
\]

Quadratic form: \( \mathbf{v}^T \mathbf{J} \mathbf{v} \geq 0 \)

• The CRB’s denominator should be maximized, or, simultaneously
  1. Maximize \( C \)
  2. Minimize the quadratic form: \( \mathbf{J} \) is positive-definite \( \Rightarrow \mathbf{v}^T \mathbf{J} \mathbf{v} \geq 0 \) \( \Rightarrow \) It is minimized for \( \mathbf{v} = 0 \) \( \Rightarrow A = B = 0 \)

• Assuming \( C \) is independent of \((A,B)\): \( \alpha_{\text{opt}} = \arg \max_{\alpha} C \) subject to \( A = B = 0 \)

• In practice, \( C \approx R_g(0) \sum_{i=0}^{L_o-1} \alpha_i^2 + 2R_g(T_s) \sum_{i=0}^{L_o-1} \alpha_i \alpha_{i+1} \)
  1. Full-response CPM: \( R_g(T_s) = 0 \) \( \Rightarrow C = \text{const.} \) \( \Rightarrow A = B = 0 \) is sufficient.
  2. Partial-response CPM: \( R_g(T_s) > 0 \) \( \Rightarrow \) maximize \( \sum_{i=0}^{L_o-2} \alpha_i \alpha_{i+1} \) AND \( A = B = 0 \).
Proposing the training sequence

- \( A = B = 0 \) is satisfied when:
  \[ A' = \sum_{i=1}^{L_0} i \alpha_{i-1} = 0 \quad \text{AND} \quad B' = \sum_{i=1}^{L_0} \alpha_{i-1} = 0 \]

- \( C \) is maximized when \( \sum_{i=0}^{L_0-2} \alpha_i \alpha_{i+1} \) is maximized or sign transitions are minimized in the sequence.

- We propose the following binary sequence:

\[ \text{sequence} \]

- It satisfies \( A' = B' = 0 \),
  1. Optimum sequence for full-response CPM.
  2. Only 2 transitions \( \Rightarrow \) Near-optimum sequence for partial-response CPM.
Generalization of the Proposed Sequence

- We can reduce the approximations for partial-response by using $L_1 = L_0 - \lfloor L/2 \rfloor$ rather than $L_0$ to reduce the edge-effect (EF).

- $M$-ary CPM: $C \propto \alpha_i^2 \Rightarrow$ Set $\alpha_i$'s to maximum value in the constellation.

- Frequency and Phase CRBs:

$$\text{CRB}(f_d | \alpha) = \frac{3}{2\pi^2 L_0^2} \times \frac{C L_0 - B^2 / 4}{C L_0 - B^2 / 4 - \frac{3}{4L_0^2} (B L_0 - 4A)^2} \geq \frac{3}{2\pi^2 L_0^2}$$

$$\text{CRB}(\theta | \alpha) = \frac{2}{L_0} \times \frac{C L_0^3 - 3A^2}{C L_0^3 - 3A^2 - \frac{(B L_0 - 3A)^2}{\lambda_i}} \geq \frac{2}{L_0}$$

- Equalities hold, if $\lambda_1 = \lambda_2 = 0$ or $A = B = 0 \rightarrow$ Proposed sequence for symbol timing.

- Approximations:

<table>
<thead>
<tr>
<th></th>
<th>Frequency</th>
<th>Phase</th>
<th>Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-response</td>
<td>Exact</td>
<td>Exact</td>
<td>Exact</td>
</tr>
<tr>
<td>Partial-response</td>
<td>EF</td>
<td>EF</td>
<td>C and EF</td>
</tr>
</tbody>
</table>
Proposed Training Sequence and its Approximations

- We propose the following sequence for *general* CPM and *all* synchronization parameters:

\[
\begin{align*}
(M-1) & \leftarrow \quad L_1/4 \quad \rightarrow \quad L_1/2 \\
(M-1) & \leftarrow \quad \cdots \quad \rightarrow \quad L_0
\end{align*}
\]

- EF is reduced as the sequence length is increased
- Approximations on \( C \) for partial-response CPM and symbol timing:

\[
\text{CRB} \left( \frac{\tilde{\alpha} \mid \tau}{\alpha^* \mid \tau} \right) = \frac{\tilde{C} - 8R_g(T_s)}{\tilde{C}} = 1 - \frac{8R_g(T_s)}{L_0R_g(0) + 2(L_0 - 1)R_g(T_s)} \xrightarrow{L_0 \to \infty} 1
\]

\( \tilde{\alpha} \): Hypothetical sequence

\( \alpha^* \): Proposed sequence

- The proposed sequence is *asymptotically* optimum
PSD of the Optimum Sequence
Computer Search for the Optimum Sequence: Genetic Algorithm (GA)

- The ratio of the minimum CRB found via GA to the proposed sequence CRB:

![Graph showing CRB ratio vs sequence length for different modulation techniques](image)

- The GA is unable to find a better sequence than the proposed one even at short to moderate lengths
Random vs. Optimum Sequences: Symbol Timing Estimation

True CRB vs. Opt. CRB for L=32
Comparison of estimator performance for “unknown and random” with “known and optimum” training sequences.

UCRB vs. Opt. CRB
Comparison of estimator performance for “known but randomly selected” with “known and optimum” training sequences.
Random vs. Optimum Sequences: Frequency and Phase Estimation

- UCRB vs. Optimum CRB

- Frequency and phase estimation are less sensitive to the selection of training sequence
Derivation of CRB

- **Assumption**: Joint estimation of $u = [f_d, \theta, \tau]^T$ in AWGN, 

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

- **The FIM elements**:

$$I(u)_{i,j} = \frac{2}{N_0} \int_{0}^{T_0} \text{Re} \left[ \frac{\partial s(t,u)}{\partial u_i} \frac{\partial s^*(t,u)}{\partial u_j} \right] dt$$

- Observation interval: $T_0 = L_0 T_s$ or $L_0$ symbols
- Computation of three partial derivatives:

$$\frac{\partial s(t,u)}{\partial f_d} = j2\pi s(t,u), \quad \frac{\partial s(t,u)}{\partial \theta} = js(t,u), \quad \frac{\partial s(t,u)}{\partial \tau} = j \frac{\partial \phi(t-\tau; \alpha)}{\partial \tau} s(t,u)$$

- **Note for CPM**: $s(t,u)s^*(t,u) = 1$
- **Frequency pulse**: 

$$\frac{\partial \phi(t-\tau; \alpha)}{\partial \tau} = -\sum_{i=0}^{L-1} \pi \alpha_i g(t-iT_s-\tau) \quad \text{where } g(t) = \frac{\partial q(t)}{\partial t} \rightarrow \text{Frequency Pulse}$$
Derivation of CRB
More Simplifications

• \( A, B, \) and \( C \) relate CRBs to frequency pulse and data symbols. They are approximated as,

\[
A \approx \sum_{i=0}^{L_0-1} \alpha_i \left( \Gamma + i \frac{T_s}{2} \right) \quad \text{where} \quad \Gamma = \int_0^{L_T} u g(u) du \\
B \approx \frac{1}{2} \sum_{i=0}^{L_0-1} \alpha_i \\
C \approx \sum_{i=0}^{L_0-1} \sum_{j=0}^{L_0-1} \alpha_i \alpha_j R_g ((i-j)T_s) \quad \text{where} \quad R_g(t) = \int_{-\infty}^{\infty} g(u) g(u+t) du
\]

• Edge-effect: \( L_1 = L_0 - \left\lfloor \frac{L}{2} \right\rfloor \) for partial-response CPM

• Correlation function of \( g(t) \):

\[
R_g (nT_s) \approx 0 \quad \text{for} \quad n > 1 \quad \Rightarrow \quad C \approx R_g(0) \sum_{i=0}^{L_0-1} \alpha_i^2 + 2 R_g(T_s) \sum_{i=0}^{L_0-1} \alpha_i a_{i+1}
\]

<table>
<thead>
<tr>
<th>Pulse Shape</th>
<th>2REC</th>
<th>3REC</th>
<th>2RC</th>
<th>3RC</th>
<th>GMSK</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_g(0) )</td>
<td>0.125</td>
<td>0.0833</td>
<td>0.1875</td>
<td>0.125</td>
<td>0.1327</td>
</tr>
<tr>
<td>( R_g(\pm T_s) )</td>
<td>0.0625</td>
<td>0.0556</td>
<td>0.0312</td>
<td>0.0589</td>
<td>0.0551</td>
</tr>
<tr>
<td>( R_g(\pm 2T_s) )</td>
<td>0</td>
<td>0.0278</td>
<td>0</td>
<td>0.0036</td>
<td>0.0036</td>
</tr>
</tbody>
</table>

\[ \approx 0 \]
Exhaustive Computer Search for CPM

- *Exact* computer exhaustive search vs. *approximated* proposed sequence:

<table>
<thead>
<tr>
<th>CPM</th>
<th>Exhaustive Search</th>
<th>Proposed Sequence</th>
<th>ΔCRB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1REC</td>
<td><img src="image" alt="Exhaustive Search" /></td>
<td><img src="image" alt="Proposed Sequence" /></td>
<td>0</td>
</tr>
<tr>
<td>2REC</td>
<td><img src="image" alt="Exhaustive Search" /></td>
<td><img src="image" alt="Proposed Sequence" /></td>
<td>0.6%</td>
</tr>
<tr>
<td>3REC</td>
<td><img src="image" alt="Exhaustive Search" /></td>
<td><img src="image" alt="Proposed Sequence" /></td>
<td>0</td>
</tr>
<tr>
<td>1RC</td>
<td><img src="image" alt="Exhaustive Search" /></td>
<td><img src="image" alt="Proposed Sequence" /></td>
<td>0</td>
</tr>
<tr>
<td>2RC</td>
<td><img src="image" alt="Exhaustive Search" /></td>
<td><img src="image" alt="Proposed Sequence" /></td>
<td>0.7%</td>
</tr>
<tr>
<td>3RC</td>
<td><img src="image" alt="Exhaustive Search" /></td>
<td><img src="image" alt="Proposed Sequence" /></td>
<td>0</td>
</tr>
</tbody>
</table>

Length = 16, Binary CPM
CRB for Random Sequences

- Comparison of the optimum sequence and a random sequence
- True CRB:
  1. The likelihood function is averaged over $\alpha$
  2. Only feasible for MSK-type CPMs, i.e. binary, $h=1/2$, and full-response
  
  $$I(u)_{k,l} = \left( \frac{2E_s}{N_0} \right)^2 \sum_{i=0}^{L_a-1} \sum_{j=0}^{L_a-1} E \left\{ \frac{\partial \hat{a}_i}{\partial u_k} \frac{\partial \hat{a}_j}{\partial u_l} \tanh \left( \frac{2E_s}{N_0} \hat{a}_i \right) \tanh \left( \frac{2E_s}{N_0} \hat{a}_j \right) \right\}$$

  3. The above expectation is computed via simulations

- Unconditional CRB (UCRB):
  1. Conditional CRB is averaged over $\alpha$.

  $$E_{r|\alpha} \left[ (\hat{\tau} - \tau)^2 \right] \geq \text{CRB}(\tau \mid \alpha) \quad \text{Averaging over } \alpha \rightarrow E_{r} \left[ (\hat{\tau} - \tau)^2 \right] \geq E_{\alpha} \left[ \text{CRB}(\tau \mid \alpha) \right] = \text{UCRB}(\tau)$$

  2. A closed-form expression was found for symbol timing
CRB for Random Data Sequence: True CRB

- NDA estimators with random (unknown) data sequence:
  - Computation of true CRB, as a nuisance parameter.
  - Quite complex; may not be feasible for general CPM.

- Special case: binary, h=1/2 and full-response:
  1. PAM representation:
     \[ s(t) = \sum_i a_i c_0(t-iT_s) \]
     \[ \text{Receiver} \rightarrow \tilde{a}_n = \begin{cases} 
     \int_{-T_s}^{T_s} \Re \left[ e^{-j(2\pi f t + \theta)} r(t) c_0(t-nT_s-\tau) \right] dt & n - \text{even} \\
     \int_{-T_s}^{T_s} \Im \left[ e^{-j(2\pi f t + \theta)} r(t) c_0(t-nT_s-\tau) \right] dt & n - \text{odd} 
     \end{cases} \]
  2. \( a_i \)'s pseudo symbols: independent Gaussian RVs.
  3. Averaging LF over \( a_i \)'s + partial derivatives \( \rightarrow \) FIM elements:
     \[ I(u)_{k,l} = \left( \frac{2E_s}{N_0} \right)^2 \sum_{i=0}^{L_0-1} \sum_{j=0}^{L_0-1} E \left\{ \frac{\partial \tilde{a}_i}{\partial u_k} \frac{\partial \tilde{a}_j}{\partial u_l} \tanh \left( \frac{2E_s}{N_0} \tilde{a}_i \right) \tanh \left( \frac{2E_s}{N_0} \tilde{a}_j \right) \right\} \]
  4. Monte-Carlo simulations \( \rightarrow \) CRBs.
CRB for Random Data Sequence: Unconditional CRB (UCRB)

- Averaging the conditional CRB w.r.t the training sequence results in unconditional lower bound on the MSE:

\[ E_{r|\alpha}|(\hat{\tau} - \tau)^2| \geq \text{CRB}(\tau | \alpha) \quad \text{Averaging over } \alpha \rightarrow E_r|(|(\hat{\tau} - \tau)^2| \geq E_a[\text{CRB}(\tau | \alpha)] = \text{UCRB}(\tau) \]

- Symbol timing CRB can be expressed as:

\[
\text{CRB}(\tau | \alpha) = \frac{T_s}{E_s/N_0} \frac{1}{\alpha^T G \alpha - \alpha^T DJD^T \alpha} = \frac{T_s}{E_s/N_0} \frac{1}{\alpha^T Q \alpha}
\]

- Where:

\[
G = 8\pi^2 h^2 \begin{bmatrix}
R_g(0) & R_g(T_s) & \ldots & R_g((L_0 - 1)T_s) \\
R_g(-T_s) & R_g(0) & \ldots & \ldots \\
\vdots & \vdots & \ddots & \vdots \\
R_g((1 - L_0)T_s) & \ldots & \ldots & R_g(0)
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
-8\pi^2 h(\Gamma + \tau/2) & -\pi h \\
-8\pi^2 h(\Gamma + \tau/2 + T_s/2) & -\pi h \\
\vdots & \vdots \\
-8\pi^2 h(\Gamma + \tau/2 + (L_0 - 1)T_s/2) & -\pi h
\end{bmatrix}
\]

\[
J = \begin{bmatrix}
I_{11} & I_{12} \\
I_{21} & I_{22}
\end{bmatrix}^{-1}
\]

- Thus:

\[
\text{UCRB}(\tau | \alpha) = \frac{T_s}{E_s/N_0} E_a \left[ \frac{1}{\alpha^T Q \alpha} \right] \rightarrow \text{Numerical Simplify}
\]
Computing Timing UCRB

- Define: \( Z = \alpha^T Q \alpha \)
- Approximation via Taylor series:
  \[
  \text{UCRB}(\tau) = \frac{T_s}{E_s / N_0} E\left[ \frac{1}{Z} \right] \approx \frac{T_s}{E_s / N_0} E\left[ Z^2 \right]
  \]
- Eigen-decomposition of \( Q \):
  \[
  Z = \alpha^T (V^T \Lambda V) \alpha = x^T \Lambda x = \sum_{i=1}^{L_0} \lambda_i X_i^2
  \quad \text{with} \quad x = V^T \alpha = [X_1, X_2, \cdots, X_{L_0}]^T
  \]
- Expected value of \( Z \)
  \[
  E\{Z\} = \sum_{i=1}^{L_0} \lambda_i E\{X_i^2\} = \sum_{i=1}^{L_0} \lambda_i v_i^T E\{\alpha \alpha^T\} v_i = \frac{M^2 - 1}{3} \sum_{i=1}^{L_0} \lambda_i
  \]
- Using the same approach and employing central limit theorem:
  \[
  E\{Z^2\} \approx \frac{(M^2 - 1)^2}{9} \left[ \sum_i \lambda_i \right]^2 + \frac{2(M^2 - 1)^2}{9} \sum_i \lambda_i^2 = \frac{(M^2 - 1)^2}{9} \text{trace}(Q)^2 + \frac{2(M^2 - 1)^2}{9} \text{trace}(Q^2)
  \]
UCRB:
Computation of $E\{Z^2\}$

- Recall $Z = \sum_{i=1}^{L_0} X_i^2$, the second moment of $Z$ is written as

$$E\{Z^2\} = \lambda^T E\{x_2 x_2^T\} \lambda = \lambda^T C_{x_2} \lambda$$

- Where $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_{L_0}]^T$ and $x_2 = [X_1^2, X_2^2, \ldots, X_{L_0}^2]^T$
- $C_{x_2}$ requires computation of $E\{X_i^4\}$ and $E\{X_i^2 X_j^2\}$.
- Approximation:

$$X_i = v_1^{(i)} \alpha_1 + v_2^{(i)} \alpha_2 + \cdots + v_{L_0}^{(i)} \alpha_{L_0}$$

Central limit theorem $\Rightarrow X_i \sim N\left(0, \frac{M^2 - 1}{3}\right)$

- $X_i$'s are uncorrelated and hence independent.
- Finally

$$[C_{x_2}]_{ij} \approx \begin{cases} 
\frac{(M^2 - 1)^2}{3} & i = j \\
\frac{3}{(M^2 - 1)^2} & i \neq j
\end{cases}$$
Maximization of LLF
(Details 1/2)

1. Discrete-time LLF:

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

2. Approximated CPM signal:

\[ s^*_\varepsilon[n] \approx \begin{cases} \exp[-j(M-1)\pi \vartheta(n/N - \varepsilon)] & n \leq n < NL_0/4 \\ \exp[+j(M-1)\pi \vartheta(n/N - L_0/2 - \varepsilon)] & NL_0/4 \leq n < 3NL_0/4 \\ \exp[-j(M-1)\pi \vartheta(n/N - L_0 - \varepsilon)] & 3NL_0/4 \leq n < NL_0 \end{cases} \]

3. Approximated LLF

\[ \Lambda(r; \nu, \theta, \varepsilon) \approx \text{Re} \left\{ e^{-j\theta} \left[ \sum_{n=0}^{NL_0/4-1} e^{-j2\pi n \nu} r[n] e^{j(M-1)\pi \vartheta(n/N - \varepsilon)} + \sum_{n=NL_0/4}^{3NL_0/4-1} e^{-j2\pi n \nu} r[n] e^{-j(M-1)\pi \vartheta(n/N - L_0/2 - \varepsilon)} + \sum_{n=3NL_0/4}^{NL_0-1} e^{-j2\pi n \nu} r[n] e^{j(M-1)\pi \vartheta(n/N - L_0 - \varepsilon)} \right] \right\} \]
Maximization of LLF
(Details 2/2)

4. Rearrange LLF as

\[ \Lambda(r; \nu, \theta, \varepsilon) \approx \text{Re}\left\{ e^{-j\theta} \left[ e^{-j(M-1)\pi\varepsilon} \lambda_1(\nu) + e^{j(M-1)\pi\varepsilon} \lambda_2(\nu) \right] \right\} \]

5. If \( \Gamma(\nu, \varepsilon) = e^{-j(M-1)\pi\varepsilon} \lambda_1(\nu) + e^{j(M-1)\pi\varepsilon} \lambda_2(\nu) \), the LLF is maximized when phase is

\[ \tilde{\theta} = \text{arg}\{\Gamma(\nu, \varepsilon)\} \]

6. The LLF becomes \( |\Gamma(\tilde{\nu}, \tilde{\varepsilon})|^2 \), which is maximized by maximizing

\[ |\Gamma(\tilde{\nu}, \tilde{\varepsilon})|^2 = |\lambda_1(\tilde{\nu})|^2 + |\lambda_2(\tilde{\nu})|^2 + 2 \text{Re}\left[ e^{-j2(M-1)\pi\varepsilon} \lambda_1(\tilde{\nu}) \lambda_2^*(\tilde{\nu}) \right] \]

7. That is,

\[ \tilde{\varepsilon} = \frac{\text{arg}\{\lambda_1(\tilde{\nu}) \lambda_2^*(\tilde{\nu})\}}{2(M-1)\pi h} \]

\[ \hat{\nu} = \text{arg max}_{\tilde{\nu}} \left\{ |\lambda_1(\tilde{\nu})| + |\lambda_2(\tilde{\nu})| \right\} \]
Linear Phase Approximation: Other CPM Examples

1RC (M=2, h=1/2)

2RC (M=4, h=1/4)
**FFT and Interpolation Precision**

- Frequency MSE for different interpolators and zero padding factors.
- Both interpolation and zero padding must be present based on simulations.

---

- No zero padding
- No interpolation

Our choice: $K_f = 2$, Gaussian
SoS Estimation Algorithm
(Details 1/3)

1. Likelihood function of all unknown parameters

\[ p(r; \delta, \nu, \theta, \alpha_d) = \frac{1}{(\pi \sigma^2)^N} \exp \left( -\frac{1}{\sigma^2} \sum_{n=0}^{\delta-1} |r[n]|^2 \right) \exp \left( -\frac{1}{\sigma^2} \sum_{n=\delta}^{N_{w}-1} |r[n] - s[n - \delta]| e^{j(2\pi n + \theta)} \right) \]

2. Drop constant factors

\[ p(r; \delta, \nu, \theta, \alpha_d) = \exp \left( -\frac{\delta - N_w}{\sigma^2} \right) \exp \left( \frac{2}{\sigma^2} \sum_{n=\delta}^{N_{w}-1} \text{Re} \{r^*[n]s[n - \delta]|e^{j(2\pi n + \theta)}\} \right) \]

3. Approximate the exponential function using 2nd degree Taylor series

\[ p(r; \delta, \nu, \theta, \alpha_d) \approx C(\delta) \left( 1 + \frac{2}{\sigma^2} \sum_{n=\delta}^{N_{w}-1} \text{Re} \{r^*[n]s[n - \delta]|e^{j(2\pi n + \theta)}\} \right) \]

\[ \quad \quad + \frac{1}{\sigma^4} \sum_{n=\delta}^{N_{w}-1} \sum_{m=\delta}^{N_{w}-1} \text{Re} \{r^*[n]r^*[m]s[n - \delta]s[m - \delta]|e^{j(2\pi (m+n) + 2\theta)}\} \]

\[ \quad \quad + \frac{1}{\sigma^4} \sum_{n=\delta}^{N_{w}-1} \sum_{m=\delta}^{N_{w}-1} \text{Re} \{r^*[n]r[m]s[n - \delta]s[m - \delta]|e^{j(2\pi (n-m) + 2\theta)}\} \]
SoS Estimation Algorithm
(Details 2/3)

4. Average the likelihood function w.r.t $\theta$

$$p(r; \delta, \nu, \alpha_d) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Lambda(r; \delta, \nu, \theta, \alpha_d) d\theta$$

$$\approx C(\delta) \frac{1}{\sigma^4} \sum_{n=\delta}^{N_u-1} \sum_{m=\delta}^{N_u-1} \text{Re}\{r^*[n]r[m]s[n-\delta]s^*[m-\delta]e^{j2\pi\nu(n-m)}\}$$

5. Rearrange the above likelihood function

$$p(r; \delta, \nu, \alpha_d) \approx C(\delta) \left( \sum_{n=\delta}^{N_u-1} |r[n]|^2 + 2 \sum_{d=1}^{N_u-\delta-1} \text{Re}\left\{e^{-j2\pi\nu} \sum_{n=\delta}^{N_u-d-1} r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta]\right\} \right)$$

6. Take the expectation w.r.t data symbols

$$p(r; \delta, \nu) \approx C(\delta) \left( \sum_{n=\delta}^{N_v-1} |r[n]|^2 + 2 \sum_{d=1}^{N_v-\delta-1} \text{Re}\left\{e^{-j2\pi\nu} \sum_{n=\delta}^{N_v-d-1} r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta] + R_{ss}(d) \sum_{n=N_v+\delta}^{N_v-d-1} r^*[n]r[n+d]\right\} \right)$$

\text{Preamble} \quad \text{Random Data}
SoS Estimation Algorithm
(Details 3/3)

7. A frequency estimate can be derived from each term in the LLF

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

8. If we use each estimate in its corresponding term, the LLF becomes independent of frequency offset

\[ p(r; \delta) \approx C(\delta) \left( \sum_{n=\delta}^{N_p-1} |r[n]|^2 + 2 \sum_{d=1}^{N_p-1} \sum_{n=\delta}^{N_p+\delta-d-1} r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta] + R_{ss}(d) \sum_{n=N_p+\delta}^{N_w-d-1} r^*[n]r[n+d] \right) \]

9. Since the LLF is now only a function of \( \delta \), it can be used for estimation of SoS

\[ \hat{\delta} = \arg \max_{\delta} \left\{ C(\delta) \left( \sum_{n=\delta}^{N_p-1} |r[n]|^2 + 2 \sum_{d=1}^{N_p-1} \sum_{n=\delta}^{N_p+\delta-d-1} r^*[n]r[n+d]s[n-\delta]s^*[n+d-\delta] + R_{ss}(d) \sum_{n=N_p+\delta}^{N_w-d-1} r^*[n]r[n+d] \right) \right\} \]

\[ C(\delta) = (N_w - \delta)^q \]
Optimization of $C(\delta)$

GMSK
$N_p = 64$
$E_s/N_0 = 1$ dB

$C(\delta) = (N_w - \delta)^q$

- $P_{FL}$ is computed via simulations
- $C(\delta)$ is more important for larger values of $D$