Timing Recovery at Low SNR
Cramer-Rao bound, and outperforming the PLL

Aravind R. Nayak
John R. Barry
Steven W. McLaughlin

{nayak, barry, swm}@ece.gatech.edu
Georgia Institute of Technology
Continuous-time discrete-time interface

Source → ECC Encoder → Modulator

Channel

ECC Decoder ← Equalizer ← Sampler

Discrete-time to Continuous-time
Continuous-time to Discrete-time
Sampling: Timing recovery

\[ \begin{align*}
0 & \quad a_0 \\
T & \quad a_1 \\
2T & \quad a_2 \\
3T & \quad a_3
\end{align*} \]

\( \tau_0, \tau_1, \tau_2, \ldots \) \quad \text{Timing offsets}

\( T \) \quad \text{Symbol duration}

\( a_0, a_1, a_2, \ldots \) \quad \text{Data symbols}

**Timing Recovery Problem:** Estimate \( \tau_0, \tau_1, \tau_2, \ldots \)
Timing offset models

Constant offset:
\[ \tau_k = \tau \]

Frequency offset:
\[ \tau_k = \tau_0 + k\Delta T = \tau_{k-1} + \Delta T \]

Random walk:
\[ \tau_{k+1} = \tau_k + w_k = \tau_0 + \sum_{i=0}^{k} w_i \]

where \( w_i \) are i.i.d. zero-mean Gaussian random variables of variance \( \sigma_w^2 \). \( \sigma_w^2 \) determines the severity of the random walk.
Timing recovery in two stages

**Acquisition:**
- Estimate $\tau_0$
- Correlation techniques
- Known preamble sequence at start of packet (Trained mode)
- Parameter $\tau_0$ spans a large range

**Tracking:**
- Keep track of $\tau_1$, $\tau_2$, $\tau_3$, ...
- Based on the phase-locked loop (PLL)
- Data symbols unknown (Decision-directed mode)
- Sufficient to track small signals $\tau_1 - \tau_0$, $\tau_2 - \tau_1$, $\tau_3 - \tau_2$, ...
PLL: Motivation

Consider the simple case of a time-invariant offset:

\[ \tau_k = \tau \]

Let \( \hat{\tau}_i \) be the current timing estimate.

Timing error:

\[ \varepsilon_i = \tau_i - \hat{\tau}_i = \tau - \hat{\tau}_i . \]

With a perfect timing error detector (TED), we get \( \hat{\varepsilon}_i = \varepsilon_i \).

Update:

\[ \hat{\tau}_{i+1} = \hat{\tau}_i + \hat{\varepsilon}_i = \tau \]

With imperfect TED:

\[ \hat{\tau}_{i+1} = \hat{\tau}_i + \alpha \hat{\varepsilon}_i \]
PLL-based timing recovery

\[ f(t) \rightarrow r(t) \rightarrow y(t) \]

\[ T.E.D. \]

\[ r_k \]

for further processing

First-order PLL

Second-order PLL

\[ \hat{t}_{k+1} = \hat{t}_k + \alpha \hat{\epsilon}_k \]

\[ \hat{t}_{k+1} = \hat{t}_k + \alpha \hat{\epsilon}_k + \beta \sum_{i=0}^{k-1} \hat{\epsilon}_i \]
Timing Error Detector (TED)

- TED is a decision-directed device
- Usually, instantaneous hard quantization
- Better decisions entail delay that destabilizes the loop
Improving timing recovery

- **Improve the quality of decisions** (Approach I)
  - Need to get around the delay induced by better decisions.
  - Feedback from the ECC decoder and equalizer to timing recovery.

  Dr. Barry’s presentation!

- **Improve the timing recovery architecture** (Approach II)
  - Need to assume perfect decisions for tractability.
  - Methods based on gradient search and projection operation.
  - Use Cramer-Rao bound to evaluate competing methods.

  This presentation!
Overview: Approach II

Questions:
• How good is the PLL-based system?
• Can it be improved upon?

Method:
• Derive fundamental performance limits.
• Compare the PLL performance with these limits.
• Develop methods that outperform the PLL.
We consider the following *uncoded* system:

\[
\{a_k\}_{0}^{N-1} \xrightarrow{\text{uncoded i.i.d.}} \tau \xrightarrow{\text{}} h(t) \xrightarrow{\text{}} \text{LPF} \xrightarrow{\text{kT (uniform)}} r_k
\]

The uniform samples are:

\[
r_k = \sum_{l=0}^{N-1} a_l h(kT - lT - \tau_l) + n_k,
\]

where \(\sigma^2\) is the noise variance, and \(h(t)\) is the impulse response.

**Problem:** Given samples \(\{r_k\}\) and knowledge of channel model, estimate

- the \(N\) uncoded i.i.d. data symbols \(\{a_k\}\)
- the \(N\) timing offsets \(\{\tau_k\}\).
Cramer-Rao bound (CRB)

Cramer-Rao bound

- answers the following question:
  
  “What is the best any estimator can do?”

- is independent of the estimator itself.

- is a lower bound on the error variance of any estimator.
CRB, intuitively

\[ f(r|\theta_1) \quad f(r|\theta_2) \]

\[ f(r|\theta_1) \quad f(r|\theta_2) \]

\[ \theta \rightarrow \text{fixed, unknown parameter to be estimated} \]

\[ r \rightarrow \text{observations} \]

- Sensitivity of \( f(r|\theta) \) to changes in \( \theta \) determines quality of estimation.
- If \( f(r|\theta) \) is narrow, for a given \( r \), probable \( \theta \)s lie in a narrow range.
  \[ \Rightarrow \theta \text{ can be estimated better, } i.e., \text{ with lesser error variance.} \]
- CRB uses \( \frac{\partial}{\partial \theta} \log f(r|\theta) \) as a measure of narrowsness.
CRB for a random parameter

If $\theta$ is random as opposed to being fixed and unknown,

- $\theta$ is characterized by a \textit{p.d.f.} $f(\theta)$ and
- $r, \theta$ are characterized by the joint \textit{p.d.f.} $f(r, \theta)$.

The measure for narrowness in this case is

$$\frac{\partial}{\partial \theta} \log f(r, \theta)$$
For any unbiased estimator $\hat{\theta}(r)$, the estimation error covariance matrix is lower bounded by

$$E[(\hat{\theta}(r) - \theta)(\hat{\theta}(r) - \theta)^T] \geq J^{-1}$$

where $J$ is the information matrix given by

$$J = E\left\{ \left[ \frac{\partial}{\partial \theta} \log f(r, \theta) \right] \left[ \frac{\partial}{\partial \theta} \log f(r, \theta) \right]^T \right\}$$

In particular,

$$E[(\hat{\theta}_i(r) - \theta_i)^2] \geq J^{-1}(i, i)$$
Efficient estimators

- An estimator that achieves the CRB is called efficient.
- Efficient estimators do not always exist.

Fixed, unknown $\theta$: ML is efficient

$$\frac{\partial}{\partial \theta} \log f (r | \theta) = 0$$

Random $\theta$: MAP is efficient

$$\frac{\partial}{\partial \theta} \log f (r, \theta) = 0$$
CRB: lower bound on timing error variance

Constant offset:

\[ \sigma^2_\epsilon \geq \frac{\sigma^2}{N E_{h'}} \]

Frequency offset:

\[ \sigma^2_\epsilon \geq \frac{6\sigma^2}{(N - 1)N(2N - 1)E_{h'}} \]
The Cramer-Rao bound on the error variance of any unbiased timing estimator:

\[ E[(\hat{\tau}_k - \tau_k)^2] \geq h \cdot f(k) \]

where

\[ h = \sigma_w^2 \frac{\eta}{\eta^2 - 1} \]

is the steady state value,

\[ f(k) = \tanh((N + 0.5)\log \eta) \left[ 1 - \frac{\sinh((N + 0.5 - 2(k + 1))\log \eta)}{\sinh((N + 0.5)\log \eta)} \right], \]

\[ \eta = \frac{\lambda + \sqrt{\lambda^2 - 4}}{2} \quad \text{and} \quad \lambda = 2 + \left( \frac{2\pi^2}{3} - 1 \right) \frac{\sigma_w^2}{\sigma^2 T^2}. \]
Steady-state value becomes more representative as SNR and N increase.

Parameters

\[ \text{SNR}_\text{bit} = 5 \text{ dB} \]
\[ N = 5000 \]
Trained PLL away from the CRB

Parameters
- SNR = 5 dB
- $N = 4095$
- $\sigma_w/T = 0.7\%$
- $\alpha = 0.03$
- 10000 sectors

Trained PLL does not achieve the steady-state CRB.
Trained PLL vs. Steady-state CRB

Parameters
\( \sigma_w / T = 0.5\% \)
1000 trials
N = 500

- \( \alpha = 0.01 \)
- \( \alpha = 0.02 \)
- \( \alpha = 0.03 \)
- \( \alpha = 0.05 \)
- \( \alpha = 0.1 \)
- \( \alpha = 0.2 \)
• As in the random walk case, the PLL does not achieve the CRB in the constant offset and the frequency offset cases.

• Using Kalman filtering analysis, we can show that PLL is the optimal causal timing recovery scheme.
  ⇒ Eliminate causality constraint to improve performance.
  ⇒ Block processing.
Constant offset: Gradient search

The trained maximum-likelihood (ML) estimator picks \( \tau \) to minimize

\[
J(\hat{\tau}; \hat{a}) = \sum_{k = -\infty}^{\infty} \left( r_k - \sum_l a_l h(kT - lT - \hat{\tau}) \right)^2
\]

This minimization can be implemented using gradient descent:

\[
\hat{\tau}_{i+1} = \hat{\tau}_i - \mu J'(\hat{\tau}_i; \hat{a})
\]

- Initialization using PLL.
- Without training, use \( J(\hat{\tau}; \hat{a}) \) instead of \( J(\hat{\tau}; a) \).
Trained ML achieves CRB

Parameters
\( \tau/T = \pi/20 \)
\( \alpha = 0.01 \)
\( N = 5000 \)

Two ways to improve performance over conventional PLL:

* Better architecture – ML for example.
* Better decisions – exploit error correction codes.
Frequency offset: Least squares estimation

Let \( \mathbf{k} = [0, 1, \ldots, N - 1]^T \),
\( \tau = [\tau_0, \tau_1, \ldots, \tau_{N - 1}]^T \),
\( \hat{\tau} = [\hat{\tau}_0, \hat{\tau}_1, \ldots, \hat{\tau}_{N - 1}]^T \) from PLL.

Model:  
\( \tau = (\Delta \hat{T}) \mathbf{k} + \tau_0 \)

Problem: 
Find \( \Delta \hat{T} \) and \( \hat{\tau}_0 \) to minimize
\[
\left\| \hat{\tau} - ((\Delta \hat{T}) \mathbf{k} + \hat{\tau}_0) \right\|^2
\]

Solution: 
\[
\Delta \hat{T} = \frac{N \sum k \hat{\tau}_k - \sum k \sum \hat{\tau}_k}{N \sum k^2 - (\sum k)^2} \quad \text{and} \quad \hat{\tau}_0 = \frac{1}{N} \sum (\hat{\tau}_k - k \Delta \hat{T})
\]
Least Squares away from CRB

Parameters

\( N = 250 \)
\( 10000 \) packets
\( \Delta T/T \sim \text{unif}[0, 0.005] \)
\( \tau_0/T \sim \text{unif}[0, 0.1] \)
\( \alpha \) optimized

\[ \text{RMS Estimation Error in } \frac{\Delta T}{T} \]

\[ \text{RMS Estimation Error in } \frac{\tau_0}{T} \]

* Trained MM + PLL + LS about 2 dB away from the CRB

\( \Rightarrow \) Gradient descent?
Gradient descent not suitable

Given uniform samples \( \{r_k\} \), pick \( \Delta \hat{T} \) and \( \hat{\tau}_0 \) to minimize

\[
J(\Delta \hat{T}, \hat{\tau}_0; a) = \sum_k \left( r_k - \sum_l a_l h(kT - lT - l\Delta \hat{T} - \hat{\tau}_0) \right)^2
\]

Gradient descent → sensitive to initialization
→ proceeds along greatest gradient: rattling in the bowl

\( J(\Delta \hat{T}, \hat{\tau}_0) \) parabolic in \( \hat{\tau}_0 \).
Levenberg-Marquardt method

Gradient descent:
- Moves along the direction of greatest gradient,
- Long, narrow valley $\Rightarrow$ this is not a good idea.

Newton’s method:
- Makes parabolic approximation,
- Directly computes the location of the minimum,
- Efficacy depends on how good the parabolic approximation is.

LM combines these two estimates using a weight factor $\lambda$. 
The update box

* updates the estimate $\mathbf{w}_i \rightarrow \mathbf{w}_{i+1}$,
* increases $\lambda$ if error increased; decreases $\lambda$ if error decreased.
Trained MM + PLL + LS + LM method achieves the CRB.

* Trained LM achieves CRB

Parameters
- \( N = 250 \)
- 10000 packets
- \( \Delta T / T \sim \text{unif}[0, 0.005] \)
- \( \tau_0 / T \sim \text{unif}[0, 0.1] \)
- \( \alpha \) optimized
Random walk: Linearization and Projection

- N-dimensional estimation problem,
- ML estimation prohibitively complex.

Instead:
- Linearize the PLL-based system,
- Apply projection operator.
TED equation:

\[ \hat{\varepsilon}_k = \varepsilon_k + n_k = \tau_k - \hat{\tau}_k + n_k \]

Define:

\[ y_k = \hat{\tau}_k + \hat{\varepsilon}_k \]

Therefore, we get the following linear Gaussian model:

\[ y_k = \tau_k + n_k \]

- Output \( y_k \) is the sum of the PLL and the TED outputs.
- Validity of model depends on linearity of TED characteristics.
- \( \hat{\tau}_k \) is an estimate based on previous observations (\textit{a priori}).
- \( y_k \) is based on previous and present observations (\textit{a posteriori}).
For the linear Gaussian model, the MAP estimator is

\[ \hat{\tau}_{\text{map}}(y) = (K_\tau + \sigma_n^2 I)^{-1} K_\tau y \]

where

- \( y \) is the vector of a posteriori observations
- \( K_\tau \) is the covariance matrix of the timing offset vector \( \tau \)
- \( \sigma_n^2 \) is the variance of the noise \( n_k \)
• 5.5 dB gain over PLL.
• 1.5 dB away from CRB.
• CRB not attainable with the timing model chosen. (The \textit{a posteriori} density $f(\theta|\mathbf{r})$ needs to be Gaussian, which is not the case here.)
• Gap partly due to loss due to linearization of the TED characteristics.
MAP estimator takes the form of a matrix operation:

\[
\hat{t}_{\text{map}}(y) = (K_\tau + \sigma_n^2 I)^{-1} K_\tau y
\]

Using the structure of the matrices involved, we can rewrite this as

\[
\hat{t}_{\text{map}}(y) \approx A_1 A_2 y
\]

where

- \( A_2 \) is a convolution matrix,
  \( \Rightarrow \) implemented as a \textit{time-invariant filter},
- \( A_1 \) is diagonal matrix with different diagonal entries,
  \( \Rightarrow \) implemented as \textit{time-varying scaling} of the filter output.
Summary

- Conventional timing recovery based on the PLL.
- Cramer-Rao bound gives a bound on performance of any timing estimator.
- Derived the CRB for different timing offset models.
- PLL does not achieve the CRB.

- With constant offset, gradient descent achieves the CRB.
- With frequency offset, the Levenberg-Marquardt method achieves the CRB.
- With a random walk, the MAP estimator significantly outperforms the CRB.
  (Caveat: With a random walk, the CRB is not achievable.)
Questions?

Thank you!