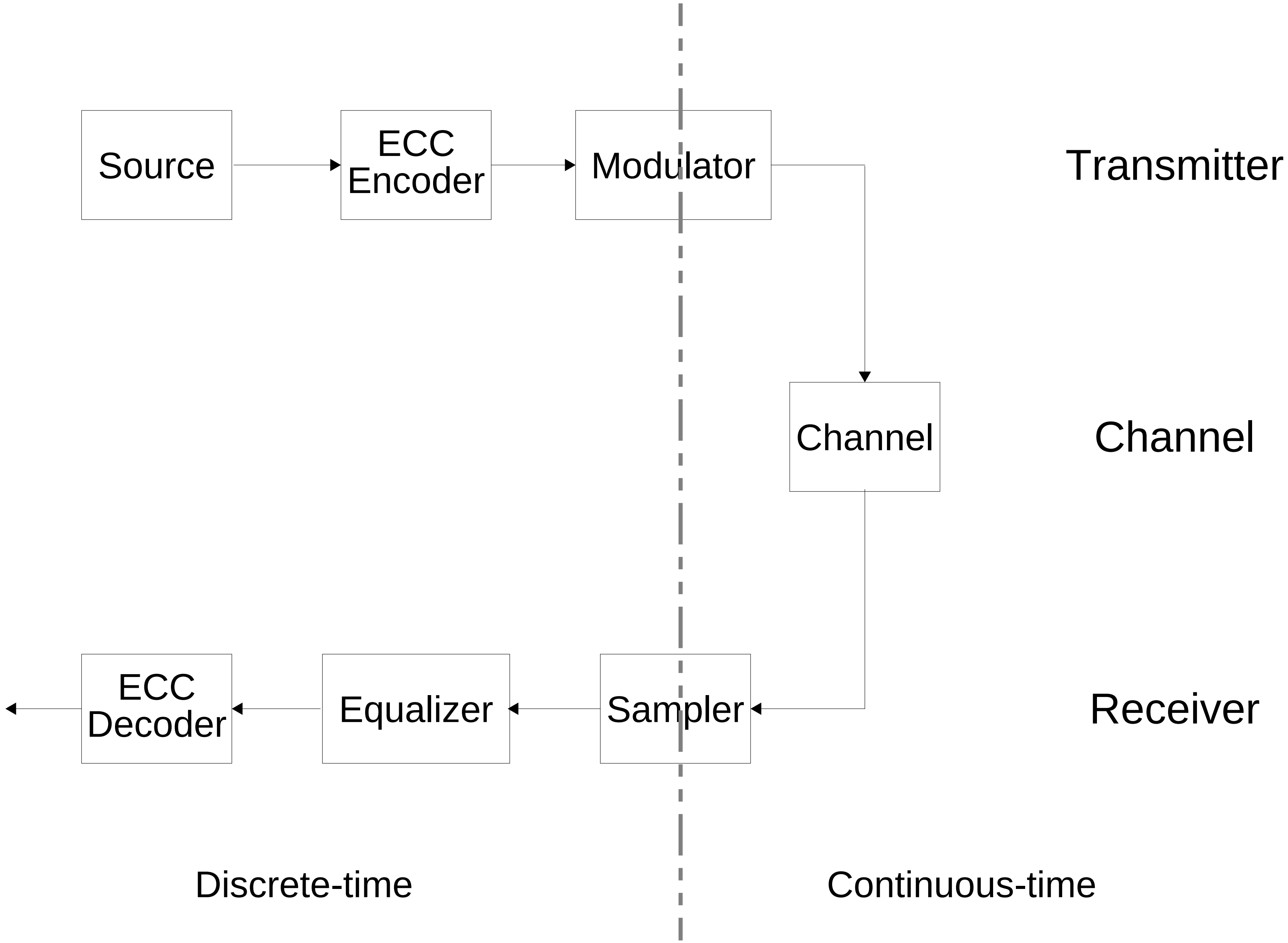
Timing Recovery at Low SNR

Cramer-Rao bound, and outperforming the PLL

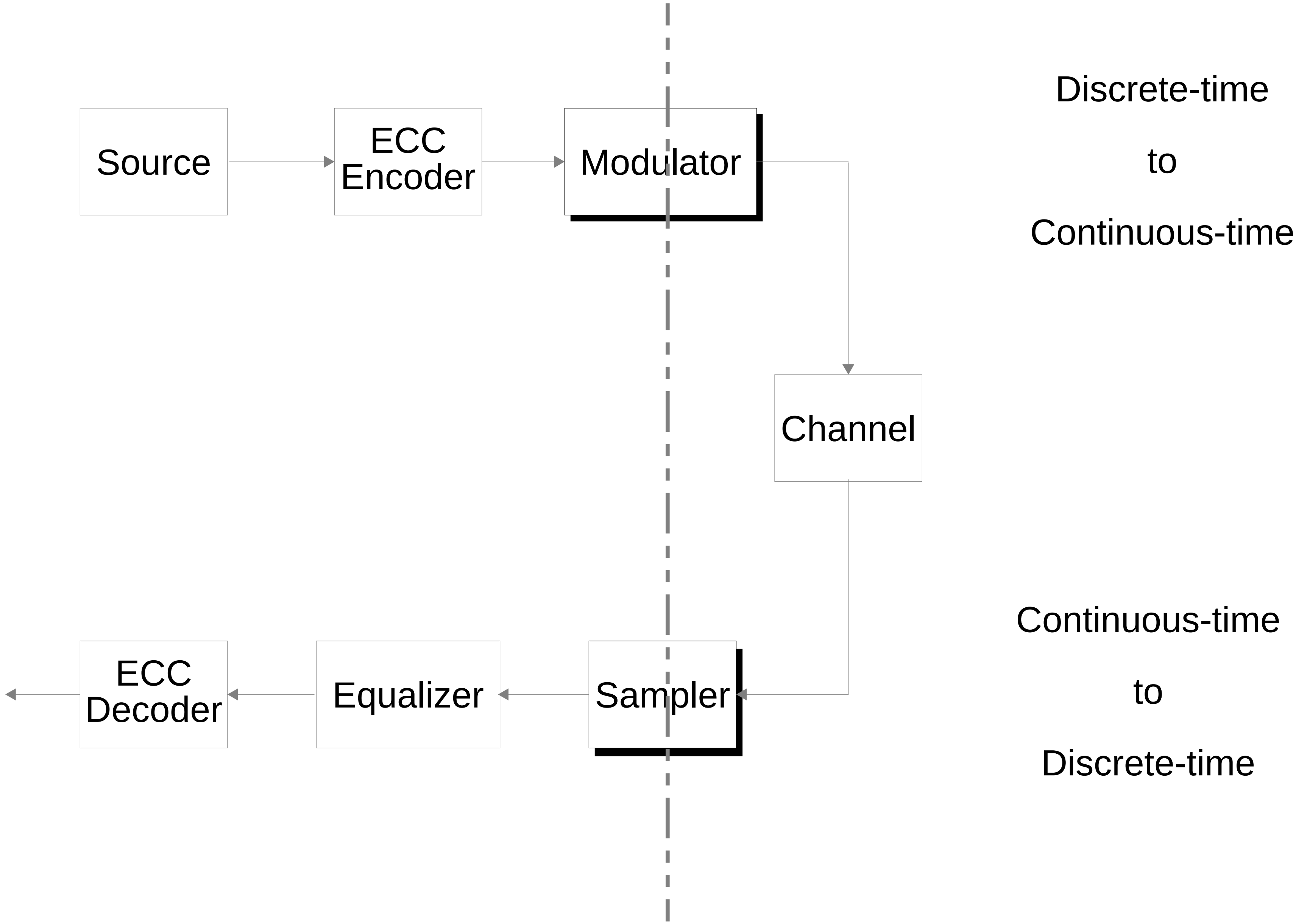
Aravind R. Nayak
John R. Barry
Steven W. McLaughlin

{nayak, barry, swm}@ece.gatech.edu
Georgia Institute of Technology

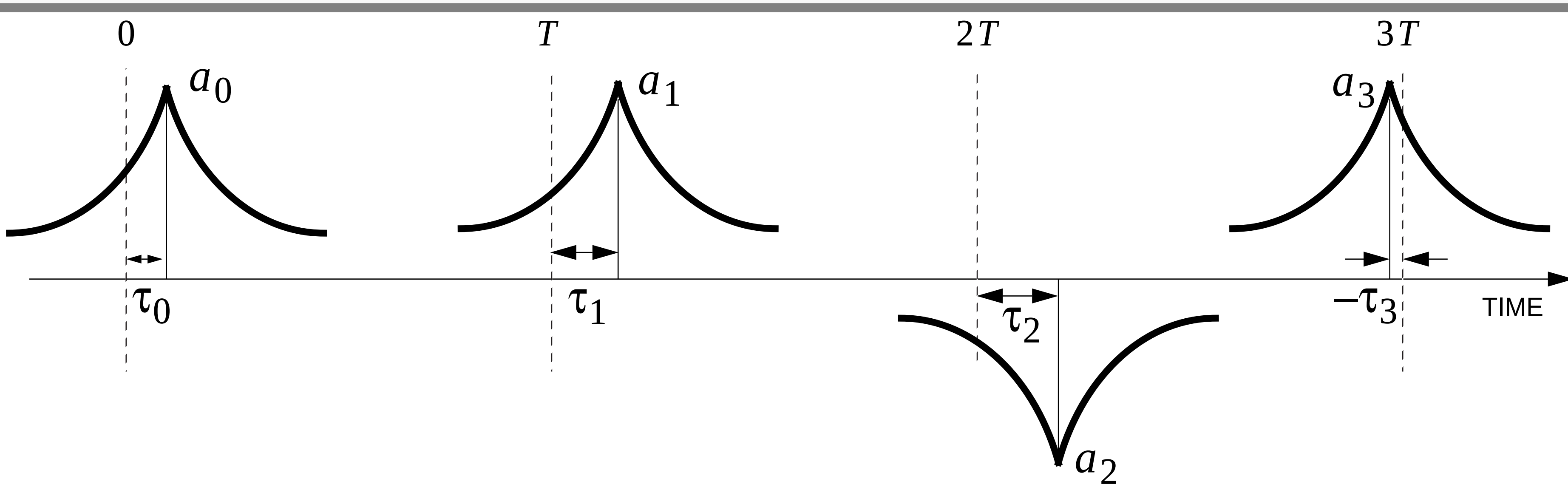
Communication system model



Continuous-time discrete-time interface



Sampling: Timing recovery



T	–	Symbol duration
$a_0, a_1, a_2, \ldots$	–	Data symbols
$\tau_0, \tau_1, \tau_2, \ldots$	–	Timing offsets

Timing Recovery Problem: Estimate $\tau_0, \tau_1, \tau_2, \ldots$

Timing offset models

Constant offset:

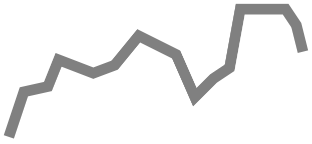
$$\tau_k = \tau$$

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 σ_w^2. σ_w^2 determines the severity of the random walk.

Timing recovery in two stages

Acquisition:

- Estimate τ_0
- Correlation techniques
- Known preamble sequence at start of packet (Trained mode)
- Parameter τ_0 spans a large range

Tracking:

- Keep track of τ_1, τ_2, τ_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.

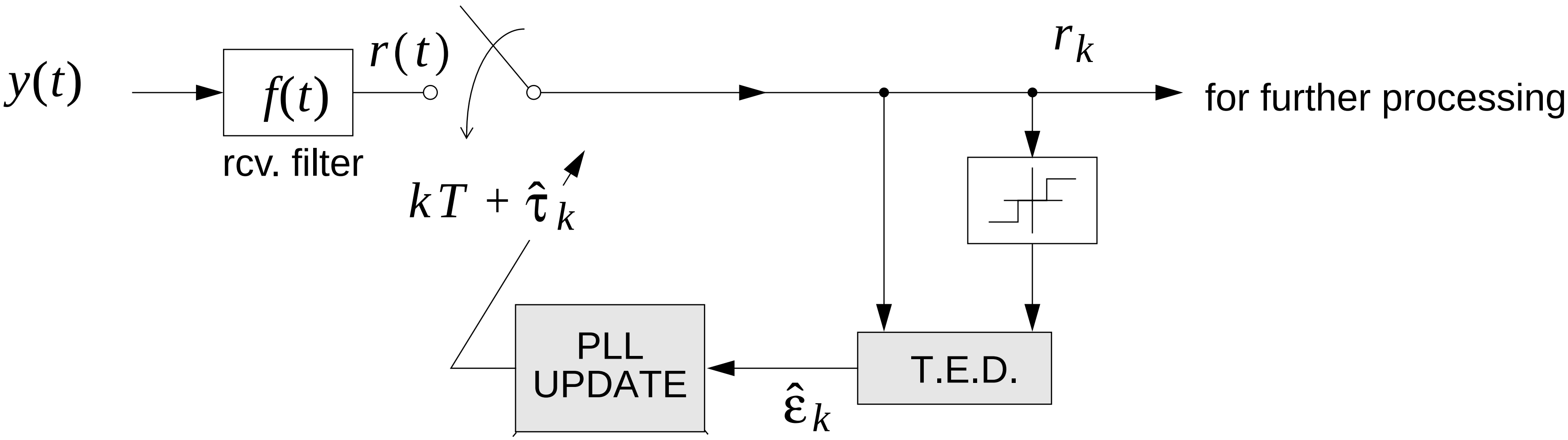
Timing error: $\varepsilon_i = \tau_i - \hat{\tau}_i = \tau - \hat{\tau}_i$.

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



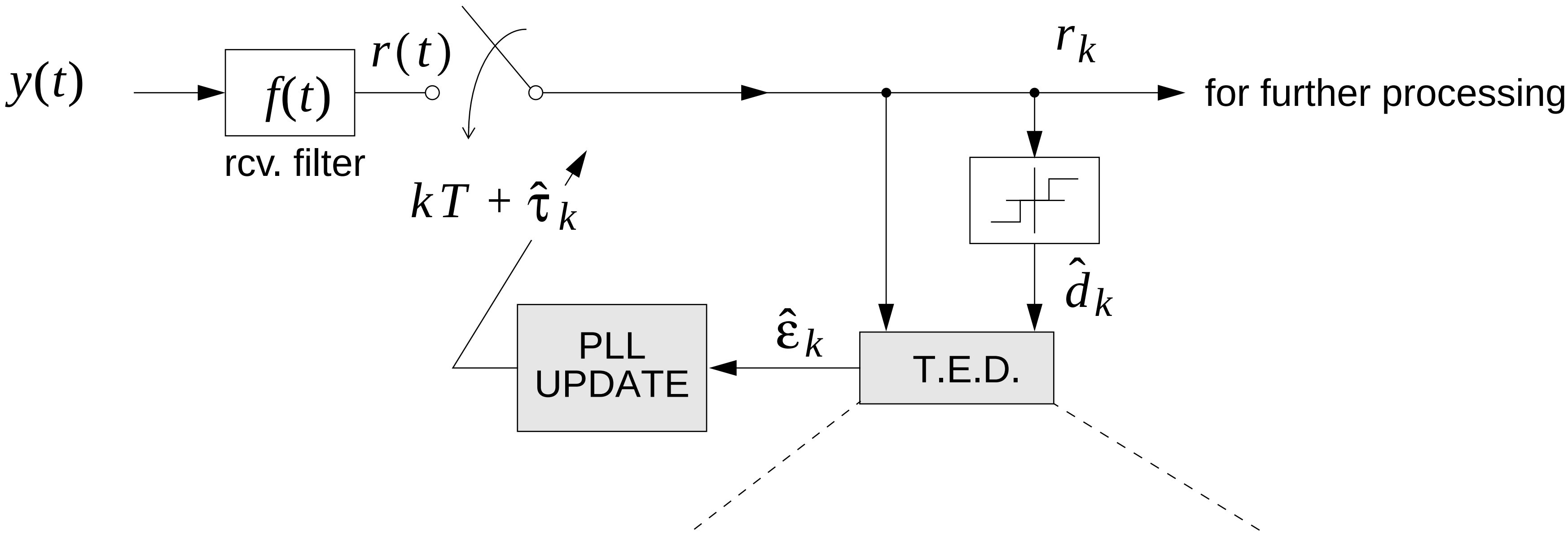
$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \hat{\varepsilon}_k$$

First-order PLL

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \hat{\varepsilon}_k + \beta \sum_{i=0}^{k-1} \hat{\varepsilon}_i$$

Second-order PLL

Timing Error Detector (TED)



$$\hat{\varepsilon}_k = \frac{3T}{16}(r_k \hat{d}_{k-1} - r_{k-1}\hat{d}_k)$$

Mueller & Müller Timing Error Detector

- 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)

  $\Rightarrow$ Need to get around the delay induced by better decisions.

  $\Rightarrow$ Feedback from the ECC decoder and equalizer to timing recovery.

Dr. Barry's presentation!

- **Improve the timing recovery architecture** (Approach II)

  $\Rightarrow$ Need to assume perfect decisions for tractability.

  $\Rightarrow$ Methods based on gradient search and projection operation.

  $\Rightarrow$ Use Cramer-Rao bound to evaluate competing methods.

This presentation!

Overview: Approach II

Questions:

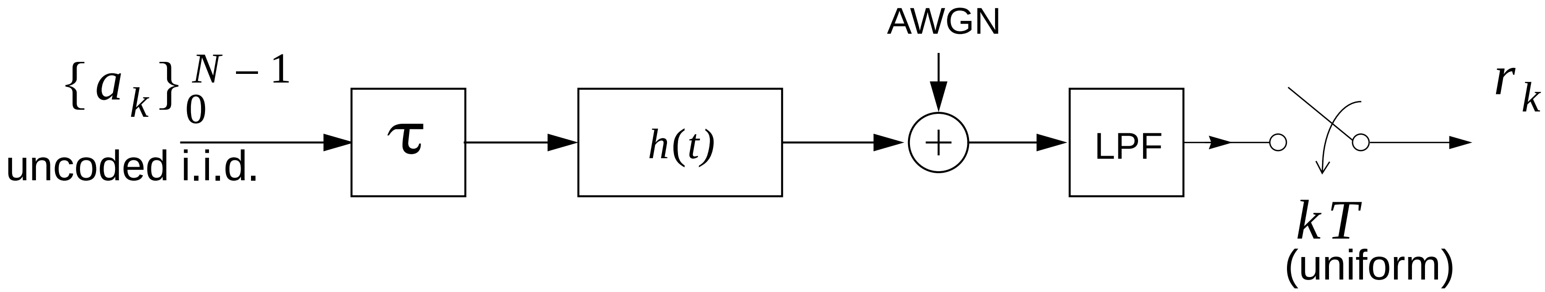
- How good is the PLL-based system?
- Can it be improved upon?

Method:

- Derive fundamental performance limits.
- Compare the PLL performance with these limits.
- Develop methods that outperform the PLL.

Problem statement

We consider the following *uncoded* system:



The uniform samples are:

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

where σ^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

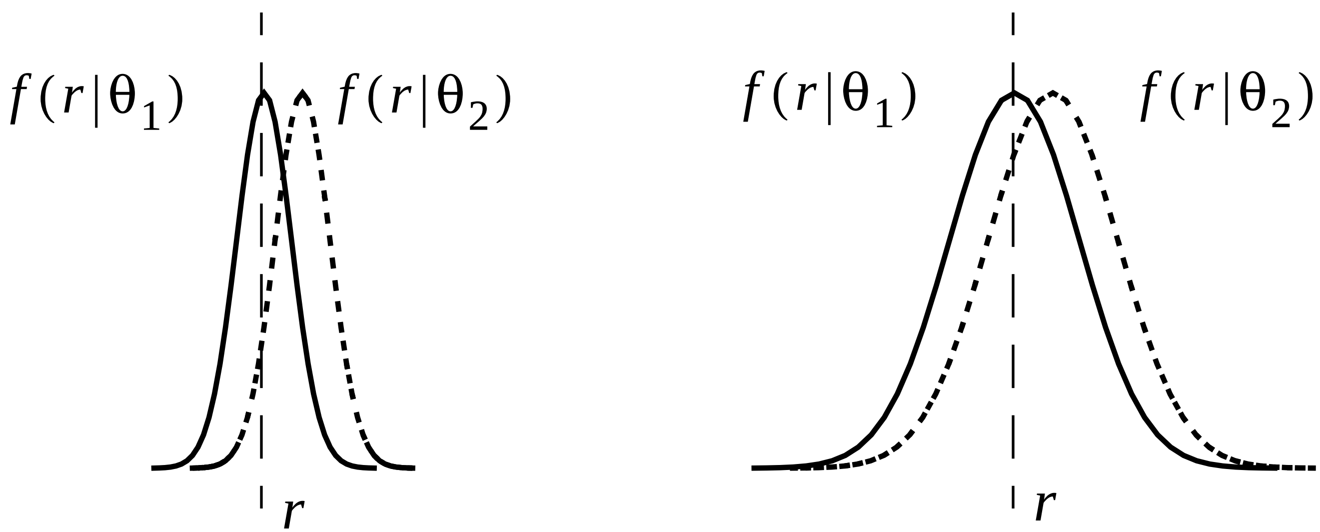
- answers the following question:

  “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



$\theta \rightarrow$ fixed, unknown parameter to be estimated

$r \rightarrow$ observations

- Sensitivity of $f(r|\theta)$ to changes in θ determines quality of estimation.
- If $f(r|\theta)$ is narrow, for a given r, probable θs lie in a narrow range.

  $\Rightarrow$ θ can be estimated better, *i.e.*, with lesser error variance.
- CRB uses $\frac{\partial}{\partial\theta}\log f(r|\theta)$ as a measure of narrowness.

CRB for a random parameter

If θ is random as opposed to being fixed and unknown,

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

The measure for narrowness in this case is

$$\frac{\partial}{\partial\theta}\log f(r, \theta)$$

CRB is the inverse of Fisher information

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 θ: ML is efficient

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

Random θ: MAP is efficient

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

CRB: lower bound on timing error variance

Constant offset:

$$\sigma_\varepsilon^2 \geq \frac{\sigma^2}{N E_{h'}}$$

Frequency offset:

$$\sigma_\varepsilon^2 \geq \frac{6\sigma^2}{(N-1)N(2N-1)E_{h'}}$$

CRB for a random walk

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,

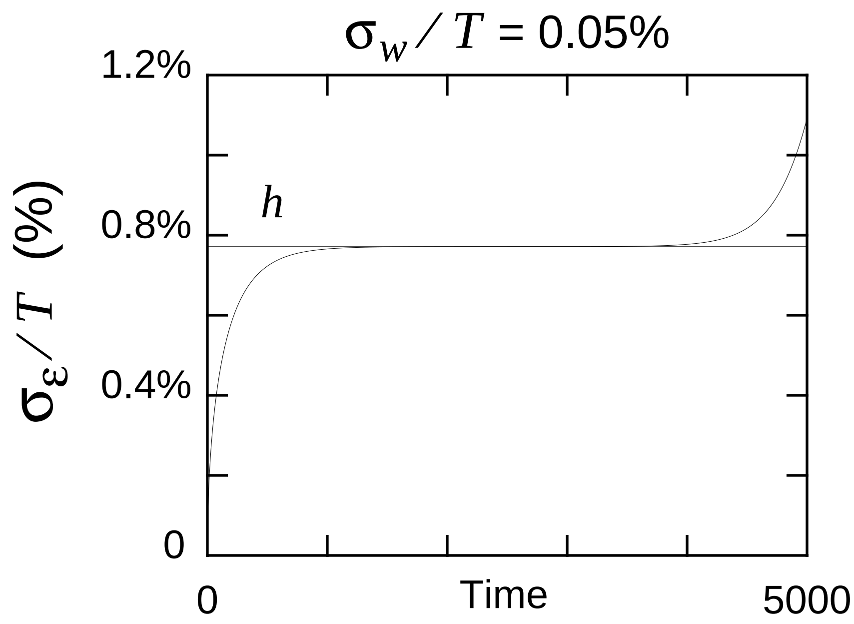
$$f(k) = \tanh((N + 0.5)\log\eta)\left[1 - \frac{\sinh((N + 0.5 - 2(k+1))\log\eta)}{\sinh((N + 0.5)\log\eta)}\right],$$

$$\eta = \frac{\lambda + \sqrt{\lambda^2 - 4}}{2}$$ and $$\lambda = 2 + \left(\frac{2\pi^2}{3} - 1\right)\frac{\sigma_w^2}{\sigma^2 T^2}.$$

CRB: Steady-state value

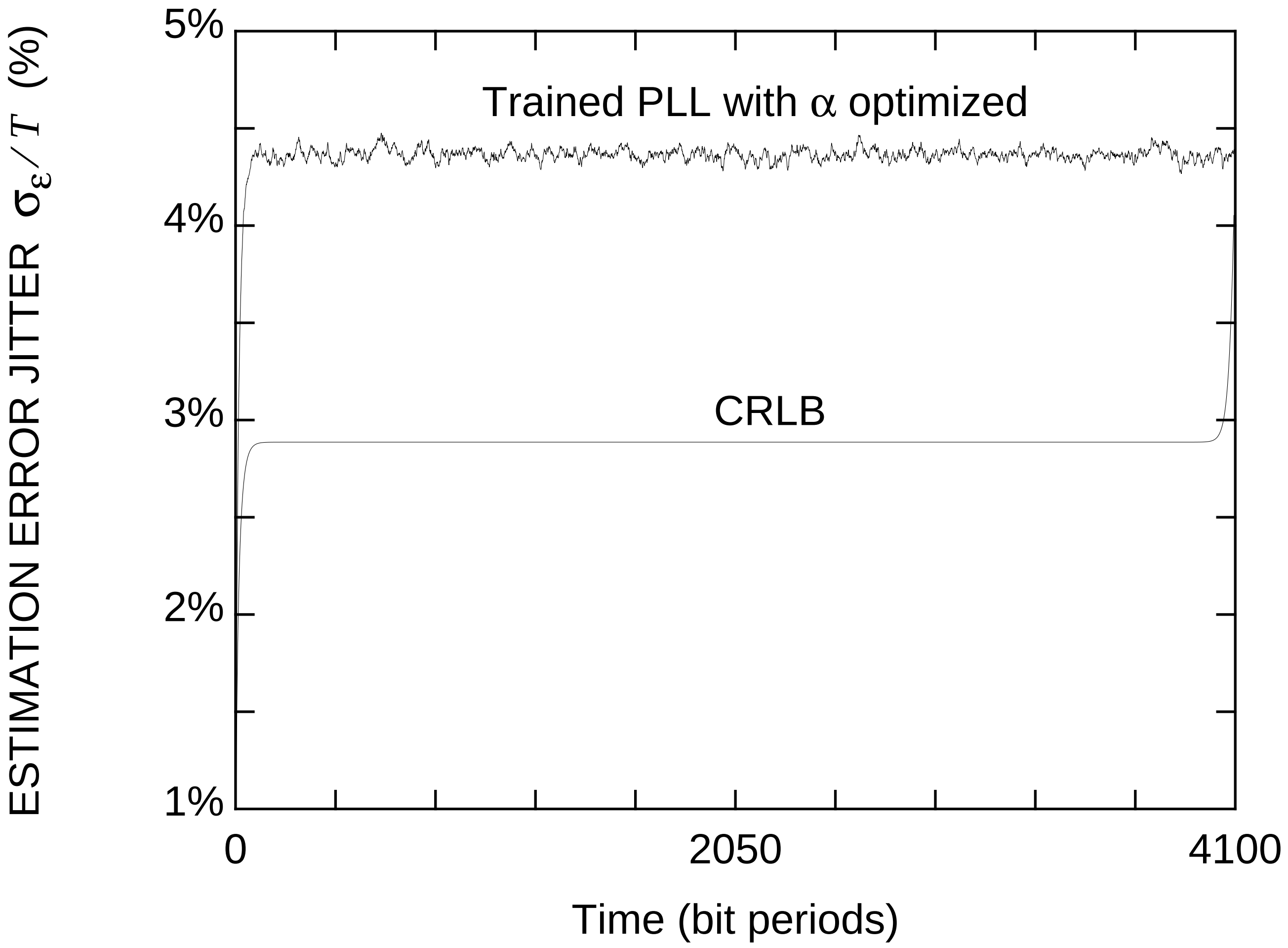
Parameters

SNR_{bit} = 5 dB
N = 5000



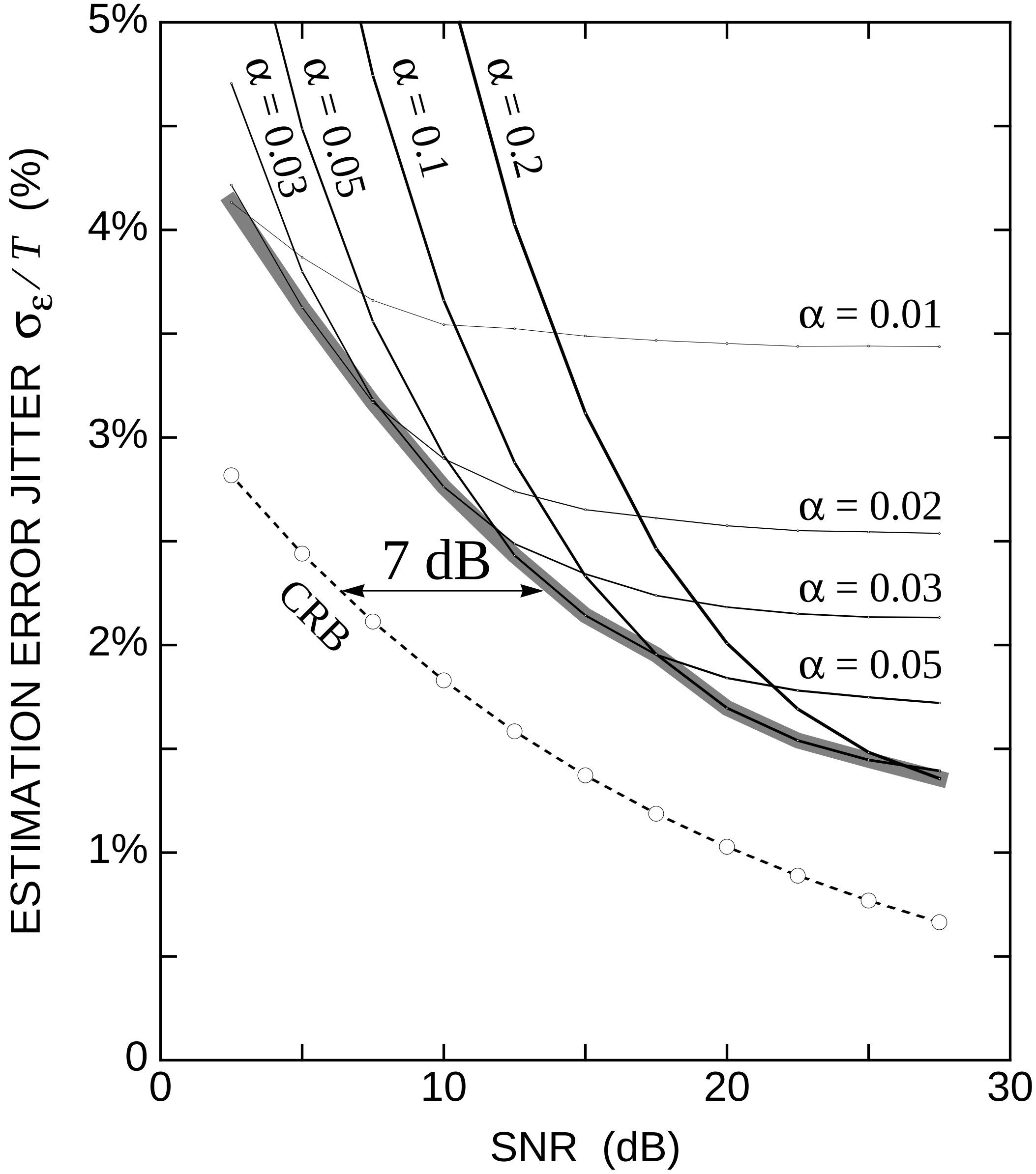
Steady-state value becomes more representative as SNR and N increase.

Trained PLL away from the CRB



Trained PLL does not achieve the steady-state CRB.

Trained PLL vs. Steady-state CRB



Parameters

σ_w / T = 0.5%
1000 trials
N = 500

Outperforming the PLL: Block processing

- 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 τ to minimize

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

This minimization can be implemented using gradient descent:

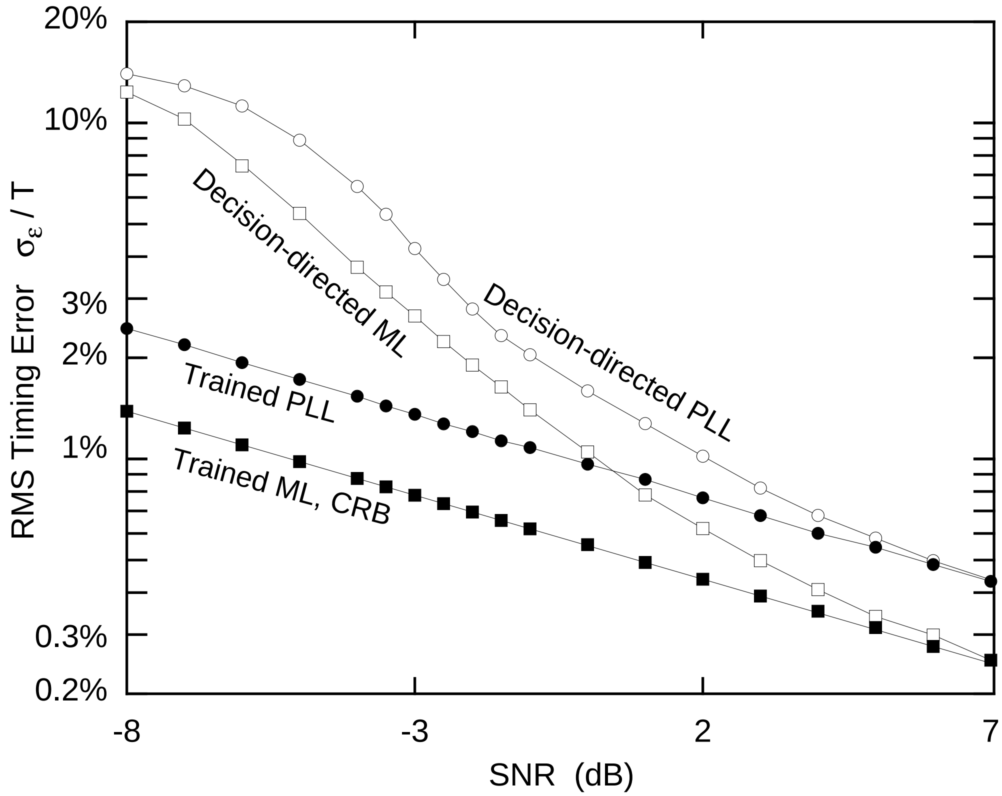
$$\hat{\tau}_{i+1} = \hat{\tau}_i - \mu J'(\hat{\tau}_i;\underline{a})$$

- Initialization using PLL.
- Without training, use $J(\hat{\tau};\hat{\underline{a}})$ instead of $J(\hat{\tau};\underline{a})$.

Trained ML achieves CRB

Parameters

$\tau/T = \pi/20$
$\alpha = 0.01$
$N = 5000$



Two ways to improve performance over conventional PLL:

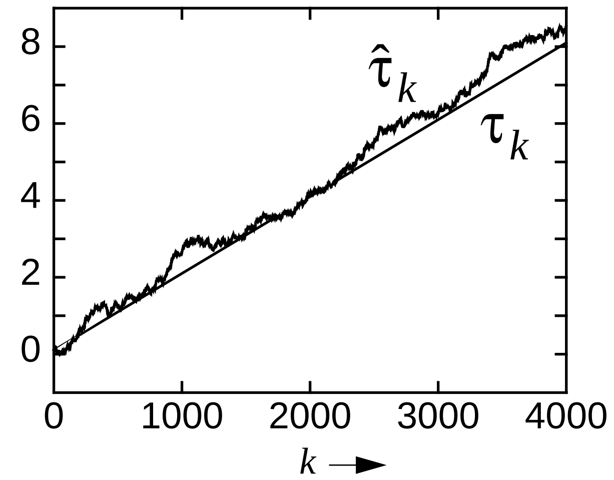
- * Better architecture – ML for example.
- * Better decisions – exploit error correction codes.

Frequency offset: Least squares estimation

Let $\boldsymbol{k} = [0, 1, \ldots, N-1]^T$,

$\tau = [\tau_0, \tau_1, \ldots, \tau_{N-1}]^T$,

$\hat{\tau} = [\hat{\tau}_0, \hat{\tau}_1, \ldots, \hat{\tau}_{N-1}]^T$ from PLL.



Model:

$$\tau = (\Delta T)\boldsymbol{k} + \tau_0$$

Problem:

Find $\hat{\Delta T}$ and $\hat{\tau}_0$ to minimize

$$\left\| \hat{\tau} - ((\hat{\Delta T})\boldsymbol{k} + \hat{\tau}_0) \right\|^2$$

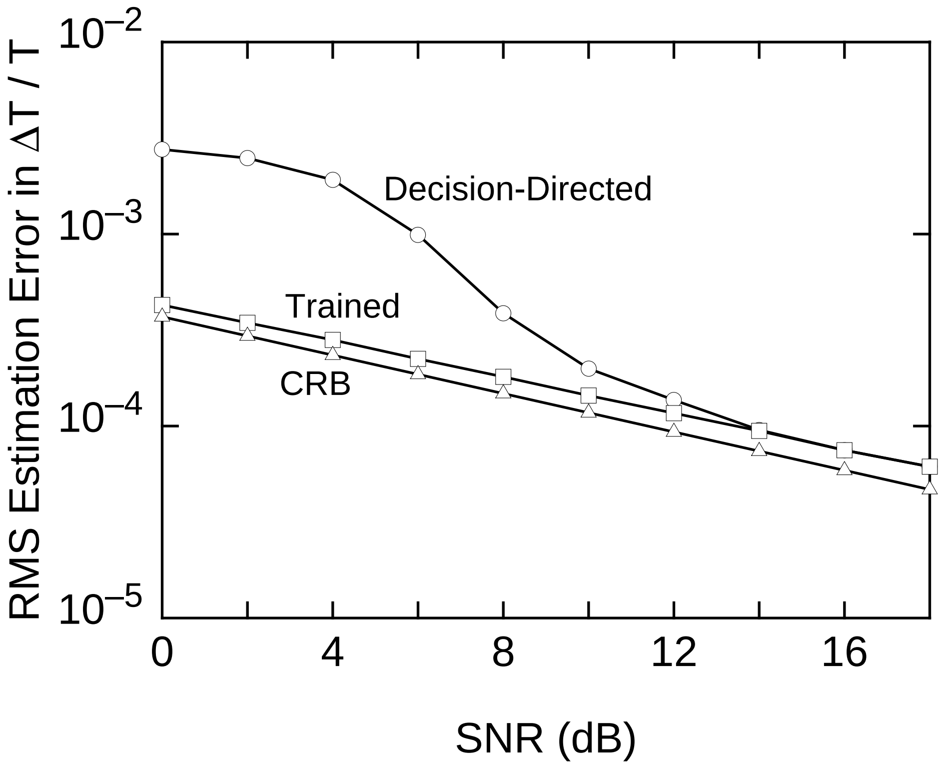
Solution:

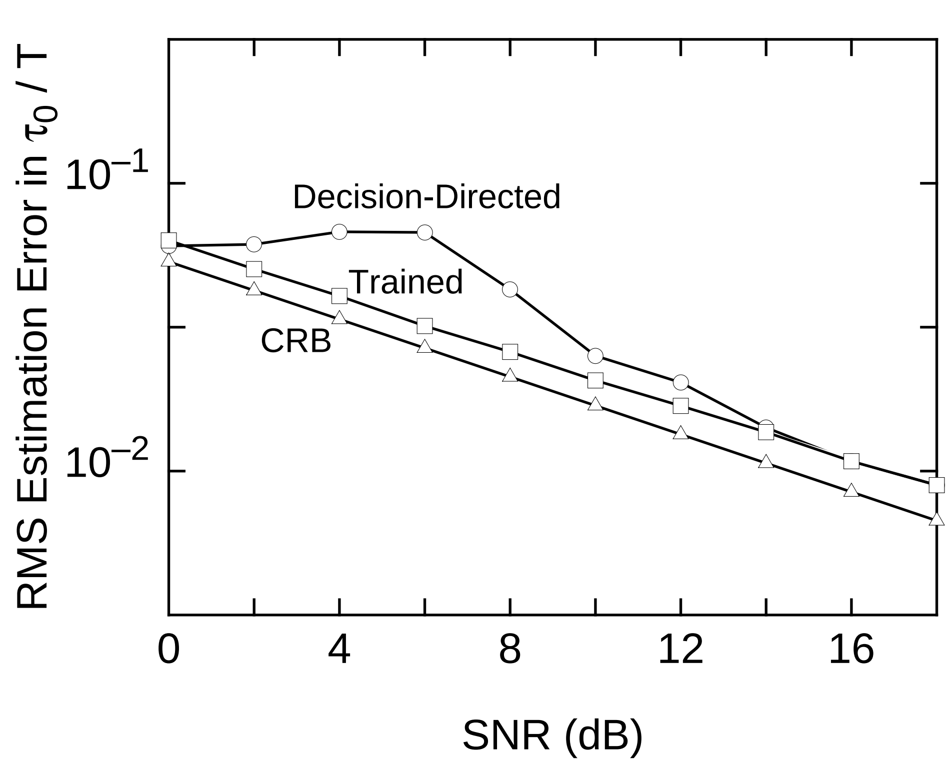
$$\hat{\Delta T} = \frac{N\sum k\hat{\tau}_k - \sum k \sum \hat{\tau}_k}{N\sum k^2 - (\sum k)^2} \text{ and } \hat{\tau}_0 = \frac{1}{N}\sum(\hat{\tau}_k - k\hat{\Delta T})$$

Least Squares away from CRB

Parameters

N = 250
10000 packets
$\Delta T/T \sim$ unif[0, 0.005]
$\tau_0/T \sim$ unif[0, 0.1]
α optimized





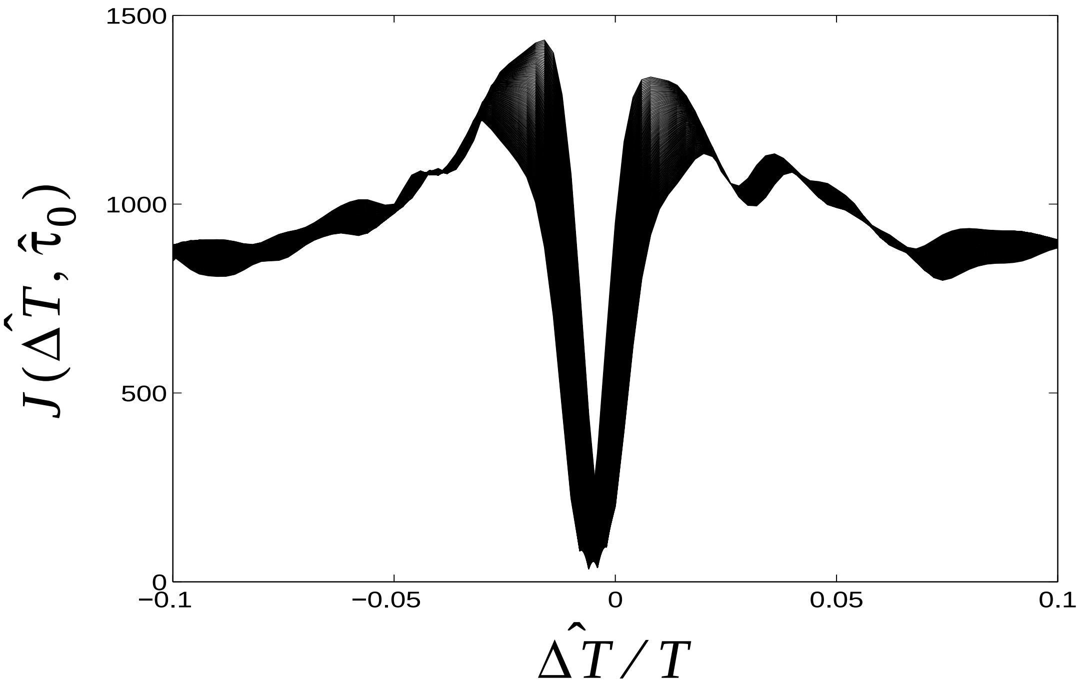
* Trained MM + PLL + LS about 2 dB away from the CRB

$\Rightarrow$ *Gradient descent?*

Gradient descent not suitable

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

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



$J(\hat{\Delta T}, \hat{\tau}_0)$ parabolic in $\hat{\tau}_0$.

Gradient descent $\rightarrow$ sensitive to initialization

$\rightarrow$ proceeds along greatest gradient: rattling in the bowl

Levenberg-Marquardt method

Gradient descent:

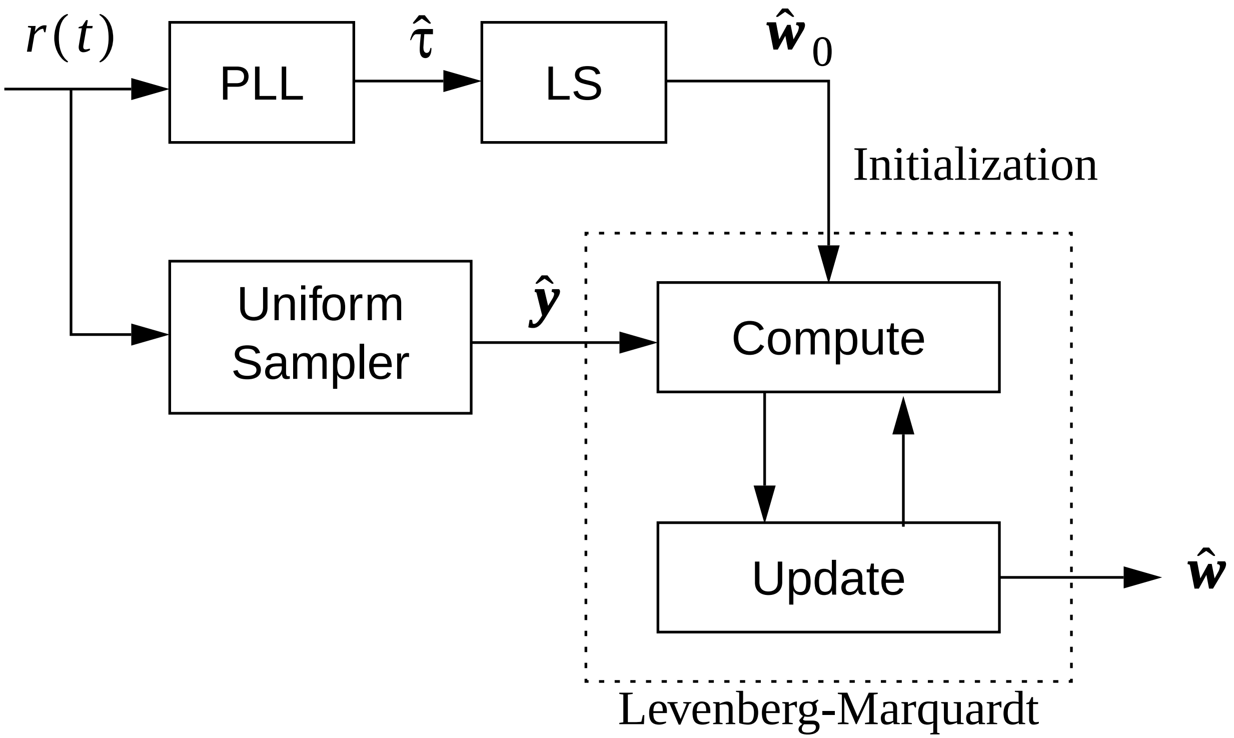
- Moves along the direction of greatest gradient,
- Long, narrow valley $\Rightarrow$ this is not a good idea.

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 λ.

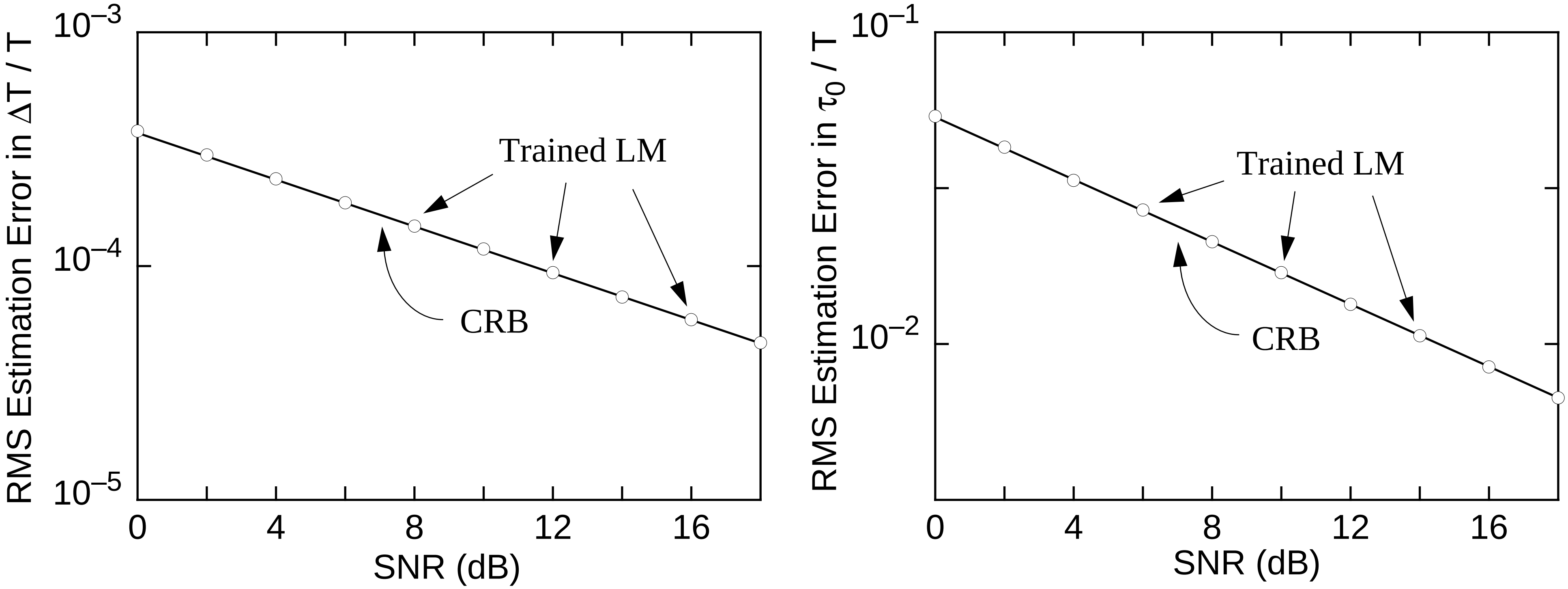
Levenberg-Marquardt (LM) method



The update box

- updates the estimate $\boldsymbol{w}_i \rightarrow \boldsymbol{w}_{i+1}$,
- increases λ if error increased; decreases λ if error decreased.

Trained LM achieves CRB



* Trained MM + PLL + LS + LM method achieves the CRB.

Random walk: Linearization and Projection

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

Instead:

- Linearize the PLL-based system,
- Apply projection operator.

Linear Gaussian model from PLL

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 (*a priori*).
- y_k is based on previous and present observsations (*a posteriori*).

MAP estimator

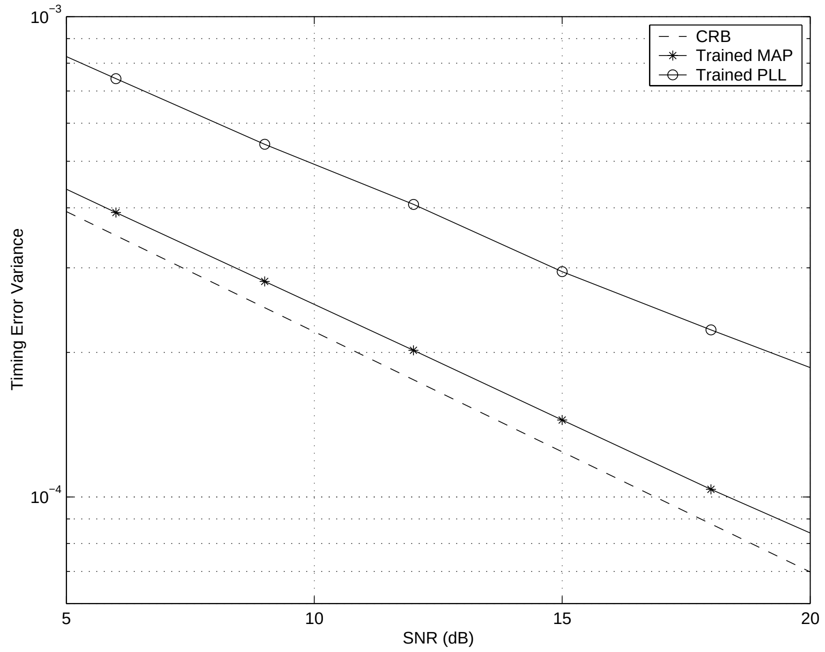
For the linear Gaussian model, the MAP estimator is

$$\hat{\tau}_{\text{map}}(\boldsymbol{y}) = (\boldsymbol{K}_\tau + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{K}_\tau \boldsymbol{y}$$

where

- y is the vector of a posteriori observations
- $\boldsymbol{K}_\tau$ is the covariance matrix of the timing offset vector τ
- σ_n^2 is the variance of the noise n_k

MAP estimator: Performance



Parameters

N = 500
1000 packets
random walk model
$\sigma_w/T = 0.33\%$
first order PLL
MM TED
α optimized

- 5.5 dB gain over PLL.
- 1.5 dB away from CRB.
- CRB not attainable with the timing model chosen. (The *a posteriori* density $f(\theta|r)$ needs to be Gaussian, which is not the case here.)
- Gap partly due to loss due to linearization of the TED characteristics.

MAP estimator: Reduced-complexity Implementation

MAP estimator takes the form of a matrix operation:

$$\hat{\tau}_{\text{map}}(\boldsymbol{y}) = (\boldsymbol{K}_\tau + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{K}_\tau \boldsymbol{y}$$

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

$$\hat{\tau}_{\text{map}}(\boldsymbol{y}) \approx \boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{y}$$

where

- $\boldsymbol{A}_2$ is a convolution matrix,

  $\Rightarrow$ implemented as a **time-invariant filter**,
- $\boldsymbol{A}_1$ is diagonal matrix with different diagonal entries,

  $\Rightarrow$ implemented as **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!