Carrier and Timing Synchronization of BPSK via LDPC Code Feedback

Esteban Vallés, Richard Wesel
and John Villasenor

Electrical Engineering Department
Univ. California, Los Angeles

Christopher Jones
Marvin Simon

Jet Propulsion Laboratory
Pasadena, CA

Introduction

- This work addresses the problem of joint carrier-phase estimation and symbol timing recovery under low SNR conditions in a ***soft decision-directed*** LDPC-coded system.

- Timing Recovery:
  - Previous treatments in the literature focus on the use of output codewords
  - We exploit information available from constraint node metrics of an LDPC code

- Carrier-Phase Synchronization: Two basic trends:
  1. Classis Synchronization techniques: Costas Loop, PLL, squaring loops.
  2. Iterative Synchronization techniques
     - Modifying iterative detection/decoding algorithms and/or graph structure to include parameter estimation
     - **Turbo Synchronization**: Pass messages between an independent phase estimation block and an essentially unmodified iterative decoder [Noels05]

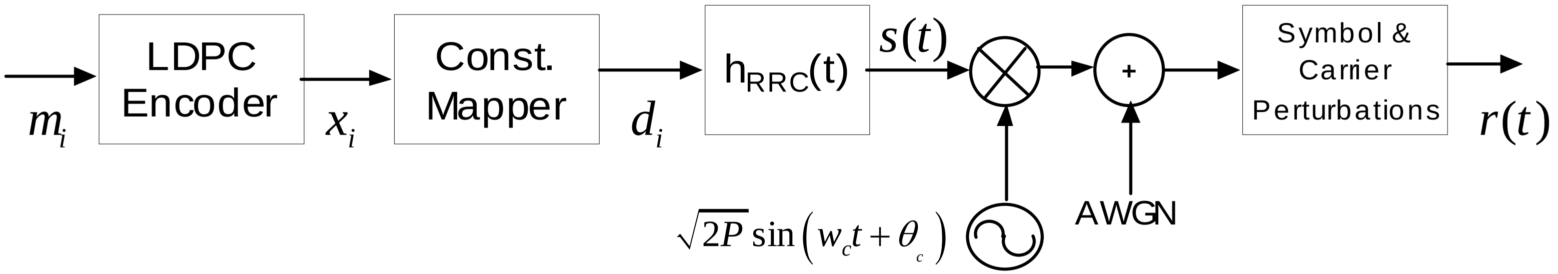
- We propose a ***soft decision-directed pilotless*** joint carrier-phase and symbol-timing recovery circuit with little modification to either the iterative decoder or the timing recovery block.

Joint Estimation Algorithm

- Decoder has a carrier phase-synchronization block and a timing-recovery block that interact as follows:

1. Estimate carrier phase
   - This can be done using a Costas Loop (CL) or an iterative Decision Directed Carrier Synchronization (DDCS) algorithm
   - For BPSK and QPSK constellation, BER/FER performance is similar using any of CL or DDCS since these constellations are not severely affected by carrier phase jitter.
2. Estimate Symbol Timing parameters
   - For symbol frequency and symbol time-delays use iterative timing recovery circuit that uses information from the constraint side of an LDPC graph to aid recovery process
   - For random walk effects, use a phase-lock loop (PLL)
3. Restart the iterative estimation process

Transmission Model


m_i
LDPC Encoder
x_i
Const. Mapper
d_i
$h_{RRC}(t)$
$s(t)$
+
Symbol & Carrier Perturbations
$r(t)$
$\sqrt{2P}\sin\left(w_c t+\theta_c\right)$
AWGN

- Parameters:
  - d_i = BPSK symbols (Code length = *n*)
  - $h_{RRC}(t)$ = Root-raised cosine (RRC) pulse
  - T = Symbol period
  - $\tau(t)$ = Timing perturbation
  - θ_c = Carrier Phase

$$s(t)=\sum_{i=0}^{n-1} d_i h_{RRC}\left(t-iT\right)$$

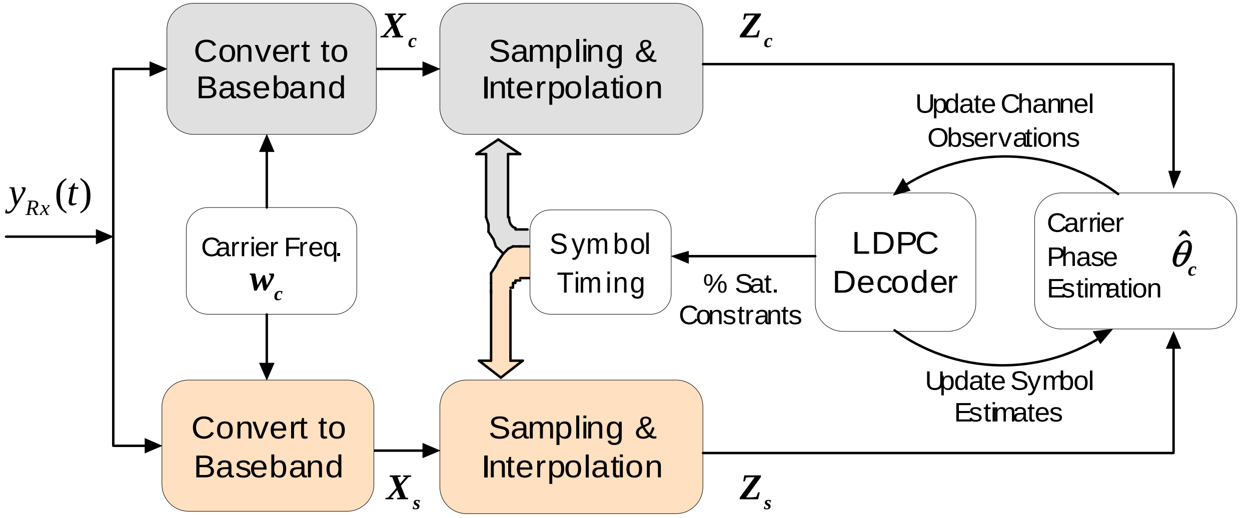
⇨ Analog transmitted waveform *s(t)* comprised of a superposition of RRC pulses

$$r(t)=\left(\sum_{i=0}^{n-1} d_i h_{RRC}\left(t-iT+\tau(t)\right)\right)\sqrt{2P}\sin\left(w_c t+\theta_c\right)+N\left(t\right)$$

⇨ Received waveform

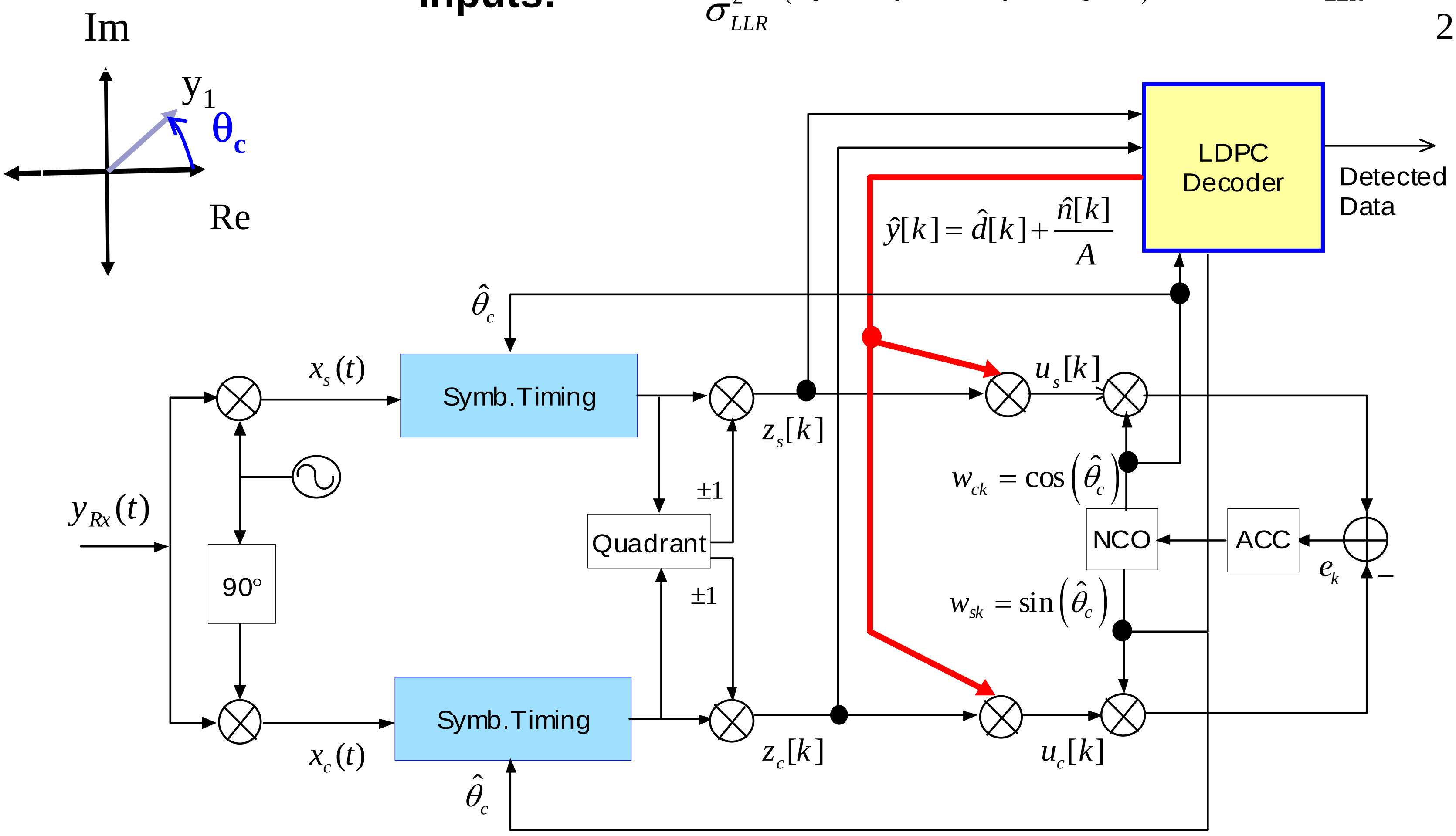
Joint Symbol Timing and Carrier Phase Recovery

- Block Diagram for DDCS



BPSK Digital Circuit Implementation

LDPC LLR Inputs: $Q[k] = \frac{2}{\sigma^2_{LLR}}\left(z_s[k]w_c[k] + z_c[k]w_s[k]\right)$ where $\sigma^2_{LLR} = \frac{\left(\sqrt{PT_s}\right)^2}{2\frac{E_s}{N_o}}$



$x_c(t;\theta_c) = \sqrt{PT_s}m(t)\sin(\theta_c) + Noise(t;\theta_c)$

$x_s(t;\theta_c) = \sqrt{PT_s}m(t)\cos(\theta_c) + Noise(t;\theta_c)$

$z_c[k] = d[k]\sin(\theta c) + Noise$

$z_s[k] = d[k]\cos(\theta c) + Noise$

$u_c[k] = A\sin(\theta_c) + Noise$

$u_s[k] = A\cos(\theta_c) + Noise$

Carrier Phase Recovery: Motivation

- When designing communication systems engineers need to decide whether or not to suppress the transmitted carrier power

- Total power = Carrier Power + Data Power → $P_t = P_c + P_d$
  - Carrier Power: related to the accuracy of the *carrier synch* process
  - Data Power: related to the accuracy of the *data detection* process (in the presence of perfect carrier synchronization).

- System design requires a proper trade off between these power requirements to minimize the *average error probability* of the system.

- **Suppressed-carrier circuits:**
  - Costas loop
    - Require a larger SNR to track the carrier with a given accuracy.
    - Phase estimation has jitter
  - Decision Directed Carrier Synchronization:
    - Parameter estimation improves with the iteration process
    - Squaring loss tends to unity as iterations increase

Decision Directed Carrier-Synchronization (DDCS)

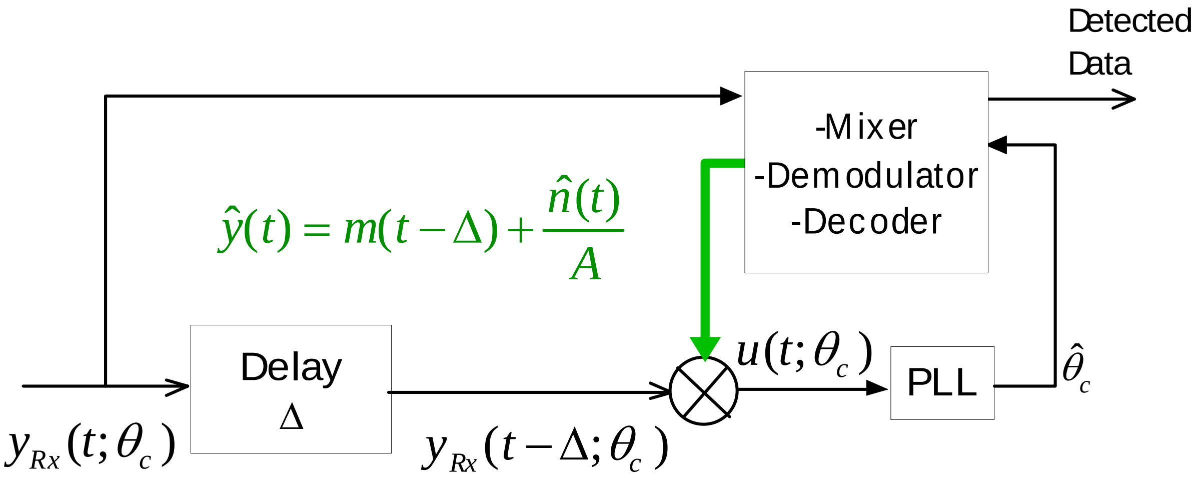
- Use a ***soft estimate*** of the instantaneous data symbol (and thus of the instantaneous phase modulation) to reduce the amount of randomness (information) in the signal being processed in the carrier loop.

- LDPC symbol estimates “wipe-off ” modulated symbols in a ***decision directed*** loop to ***enhance the carrier information*** such that a classic PLL can provide increasingly accurate (very low jitter) phase estimates over LDPC iterations.

- Latency penalty: tracking improves with increased iterations

- System complexity: No significant modifications to the current residual carrier recovery techniques used for BPSK/ QPSK modulation in NASA's deep-space network.

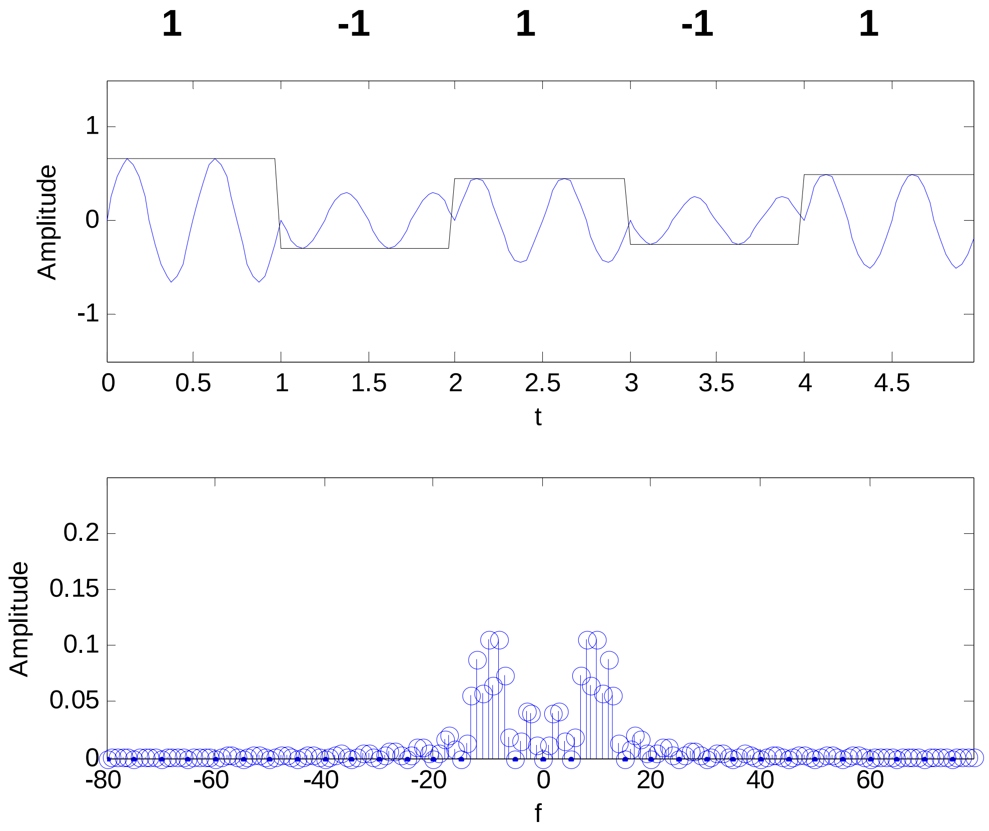
DDCS BPSK System Description



- BPSK Modulation: $m(t) = \sum_{k=-\infty}^{\infty} d_k p\left(t - kT_s\right)$
- Signal Input: $y_{Rx}\left(t;\theta_c\right) = \sqrt{2P}\,m\left(t\right)\sin\left(\omega_c t + \theta_c\right) + n_1\left(t\right)$
- AWGN Noise: $n_1\left(t\right) = \sqrt{2}\left[N_{c1}(t)\cos\left(w_c t + \theta_c\right) - N_{s1}(t)\sin\left(w_c t + \theta_c\right)\right]$
- IR Signal: $u\left(t;\theta_c\right) = \sqrt{2P}\sin\left(w_c t + \theta_c\right) + \text{ [Noise terms]}$

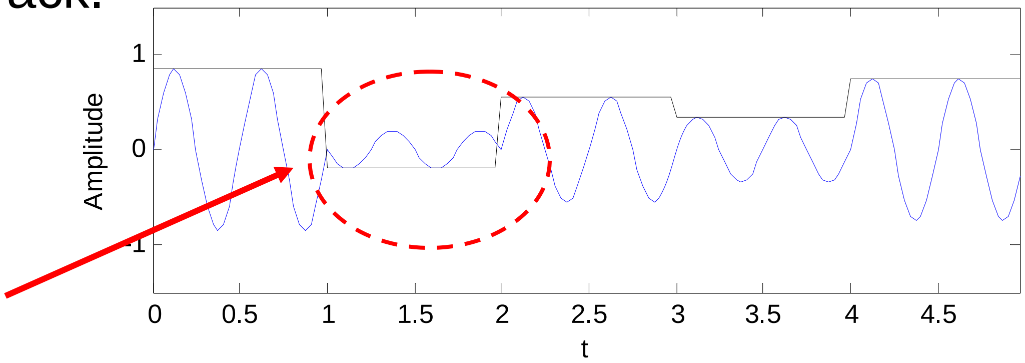
Carrier Detection Process

- Assume N=5 transmitted BPSK symbols
- Sample transmitted waveform = [1, -1, 1, -1 ,1]
- Plot shows received waveform *affected by symbol-wise noise*

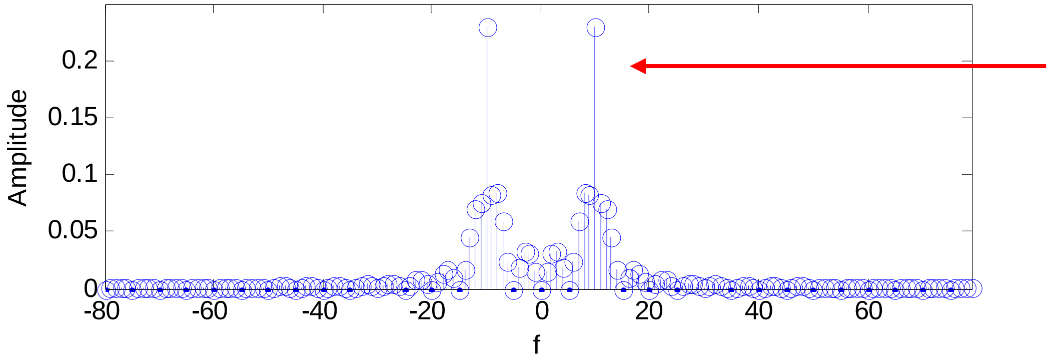


Carrier Detection Process

- Modulation is removed by multiplying the received waveform by soft-estimated symbols from the decoder
- Symbol Information randomness is reduced
- Plot shows “IR” waveform after the first iterations
- As iterations increase, soft-symbol estimation becomes more accurate
- Frequency spectrum has a distinctive tone that a PLL based circuit can now track.



- **Soft Symbol information may still be incorrect**
- **Incorrect information has low reliability**



Unmodulated frequency spectrum begins to show the presence of a carrier tone.

Loop SNR

- PLL:

$$\frac{1}{\sigma_{\phi_c}^{\ 2}} = \rho_{PLL} = \frac{P_c}{N_o B_L}$$

B$_L$: Loop Bandwidth
Pc: Carrier Power,
No: Noise PSD

  - □ Loop SNR exhibits No squaring loss
  - □ Utilizes carrier power for carrier estimation

- Costas Loop:

$$\frac{1}{\sigma_{\phi_c}^{\ 2}} = \rho_C \ .S_{L_C} = \frac{P_t}{N_o B_L}\left(1+\frac{1}{2R_d}\right)^{-1}$$

Pt: Total Power,
SL $_{IRCS}$: Squaring Loss,
Rd: Input Data SNR

  - □ SL is independent of the iteration process, for a given SNR.
  - □ Suppressed carrier circuit

- DDCS:

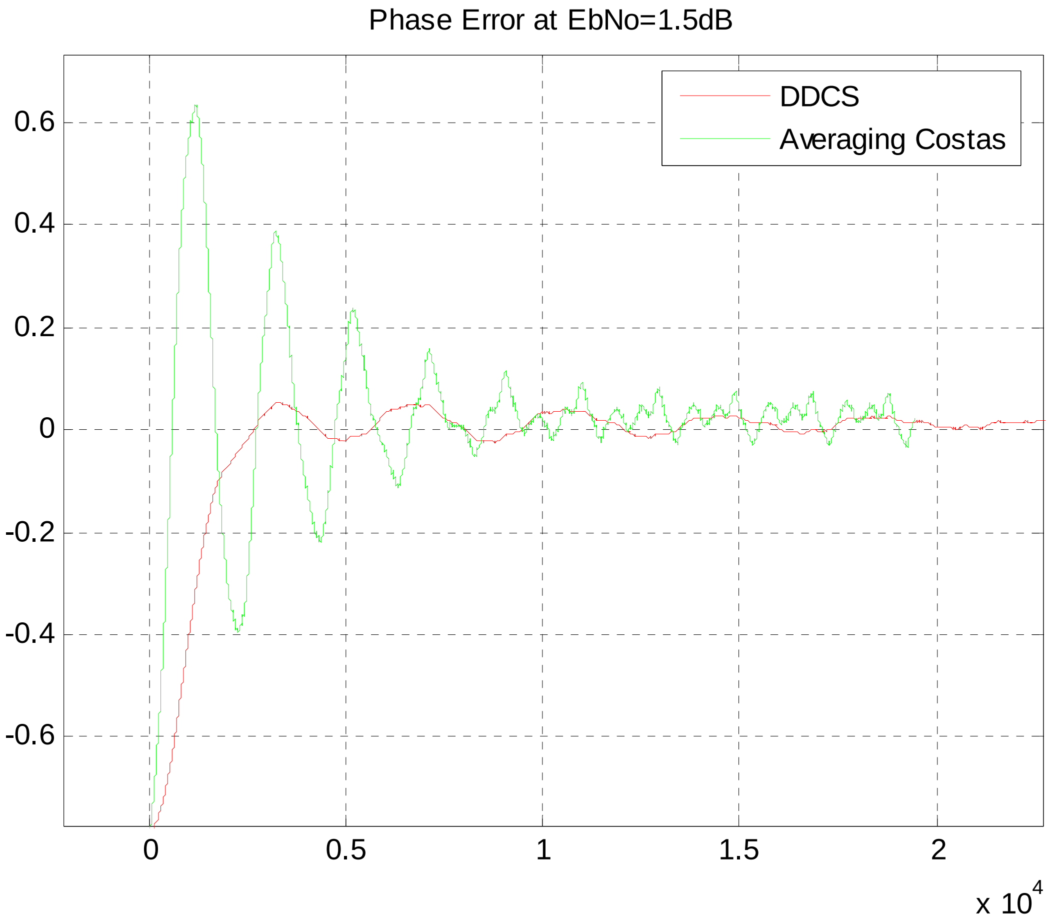
$$\frac{1}{\sigma_{\phi_c}^{\ 2}} = \rho_{DDCS} \ .S_{L_{DDCS}} = \frac{P_T}{N_o B_L}\left(1+\frac{\sigma^2}{A^2}\right)^{-1}$$

B$_L$: Loop Bandwidth
Pt: Total Power,
SL $_{IRCS}$: Squaring Loss,
A: Estimated Decoder Amplitude

  - □ SL approaches unity as the number of LDPC iterations increase
  - □ Suppressed carrier circuit

Averaging Costas Loop

- Instead of rotating the symbols by the “instantaneous” phase estimation, an average over a large number of symbols is used



- The idea is to eliminate the phase jitter present in the estimation
- Observe that the mean of the error is in both cases very close to zero.

Loop SNR: Larger Gains

- Larger gains
- Phase estimation process converges faster

- DDCS Gains:

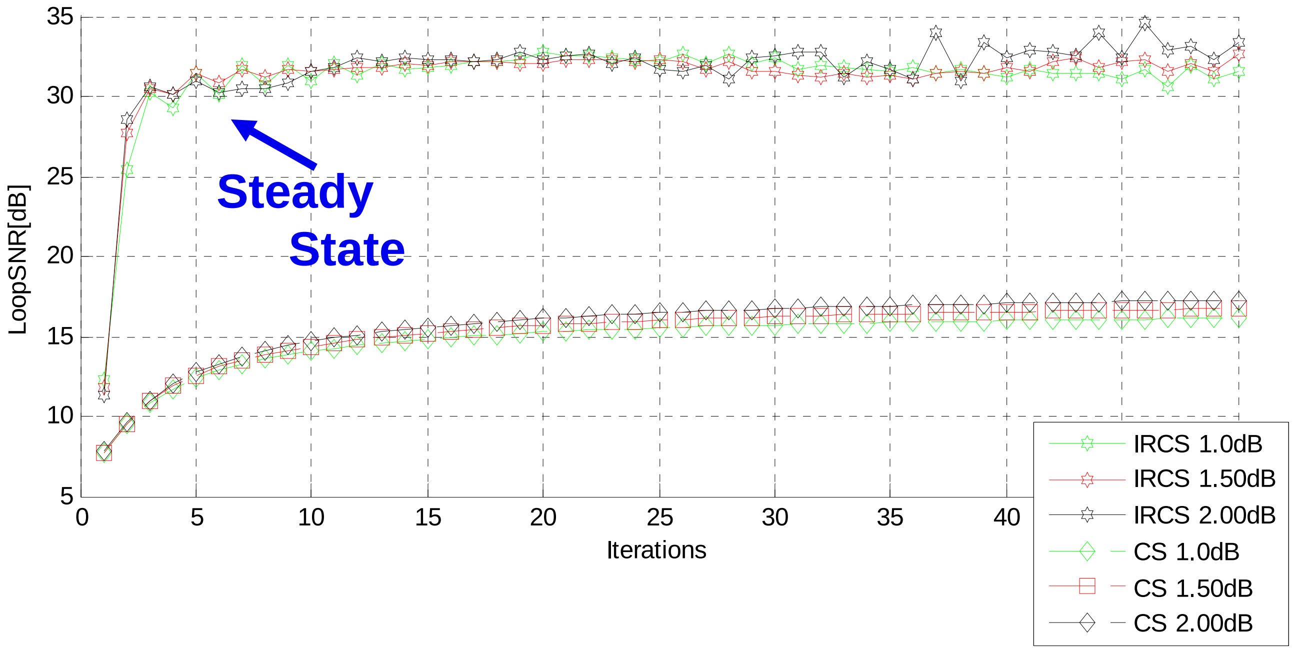
  Kp=8.92E-5

  KI=-8.75E-5

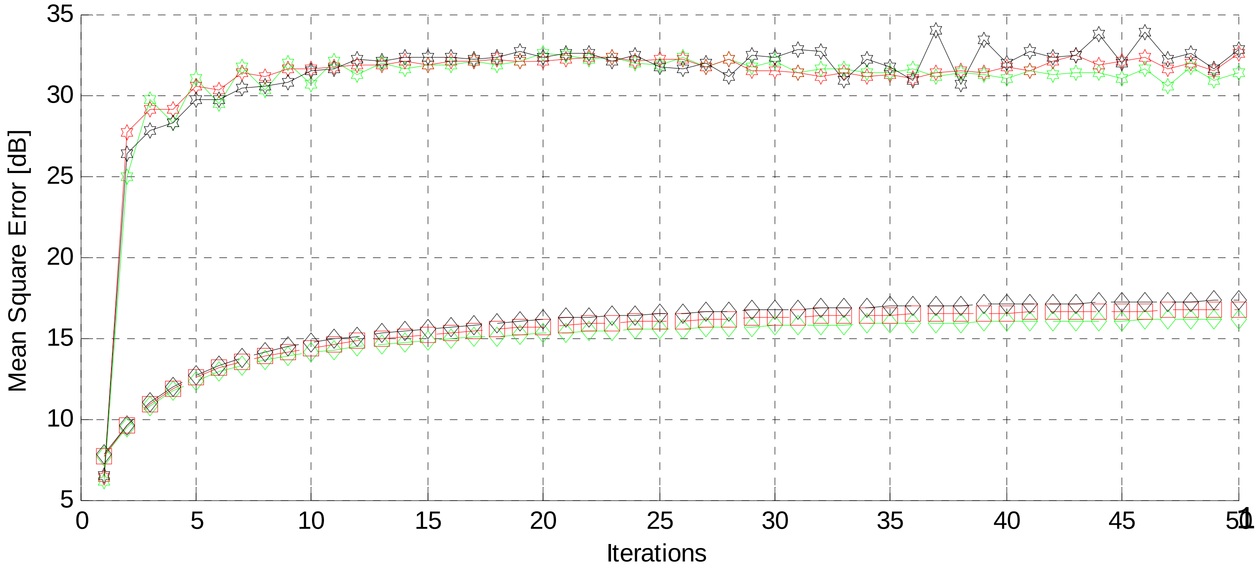
- Costas Gains:

  Kp=8.85E-4

  KI=-8.75E-4



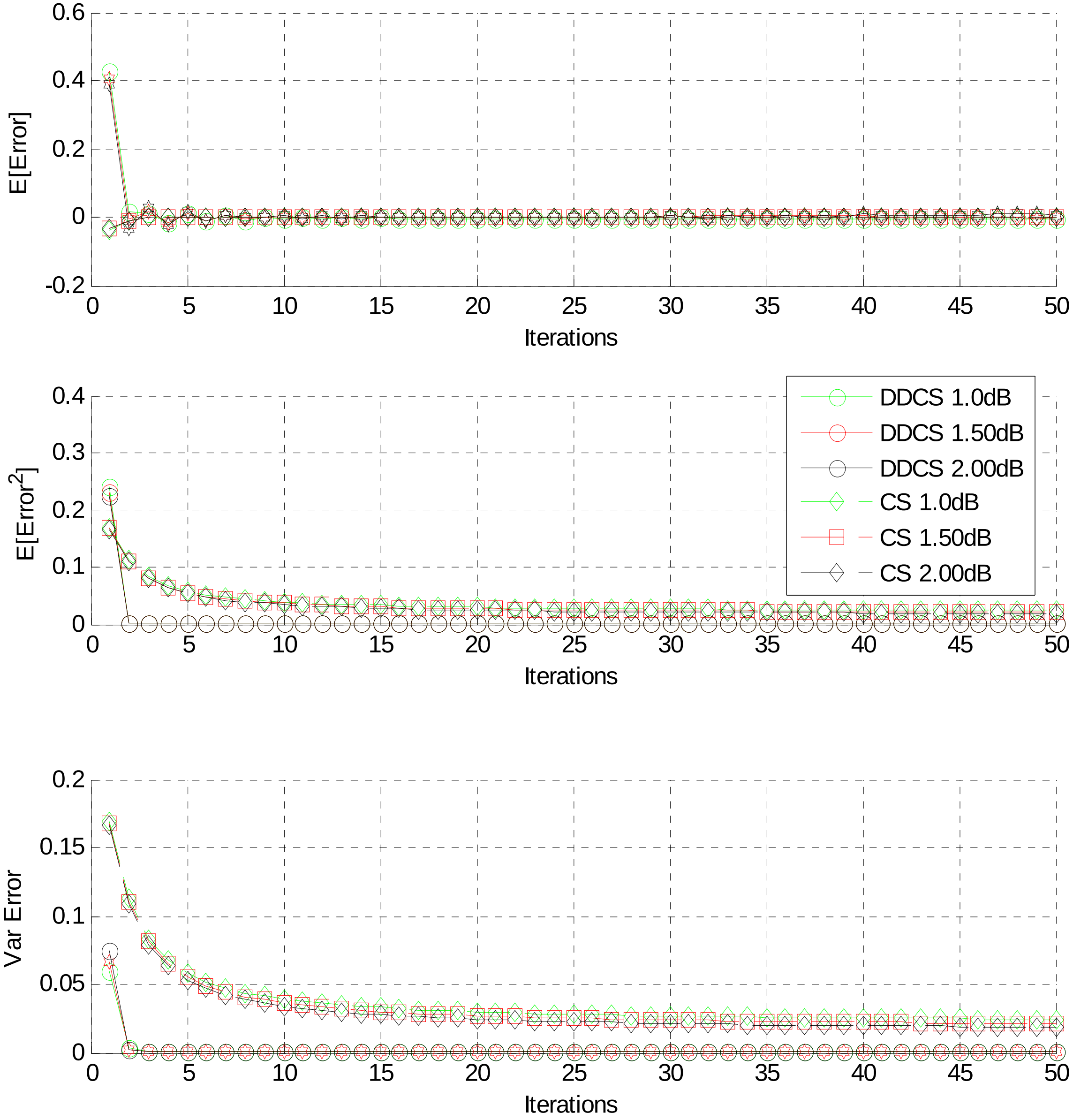
Steady State



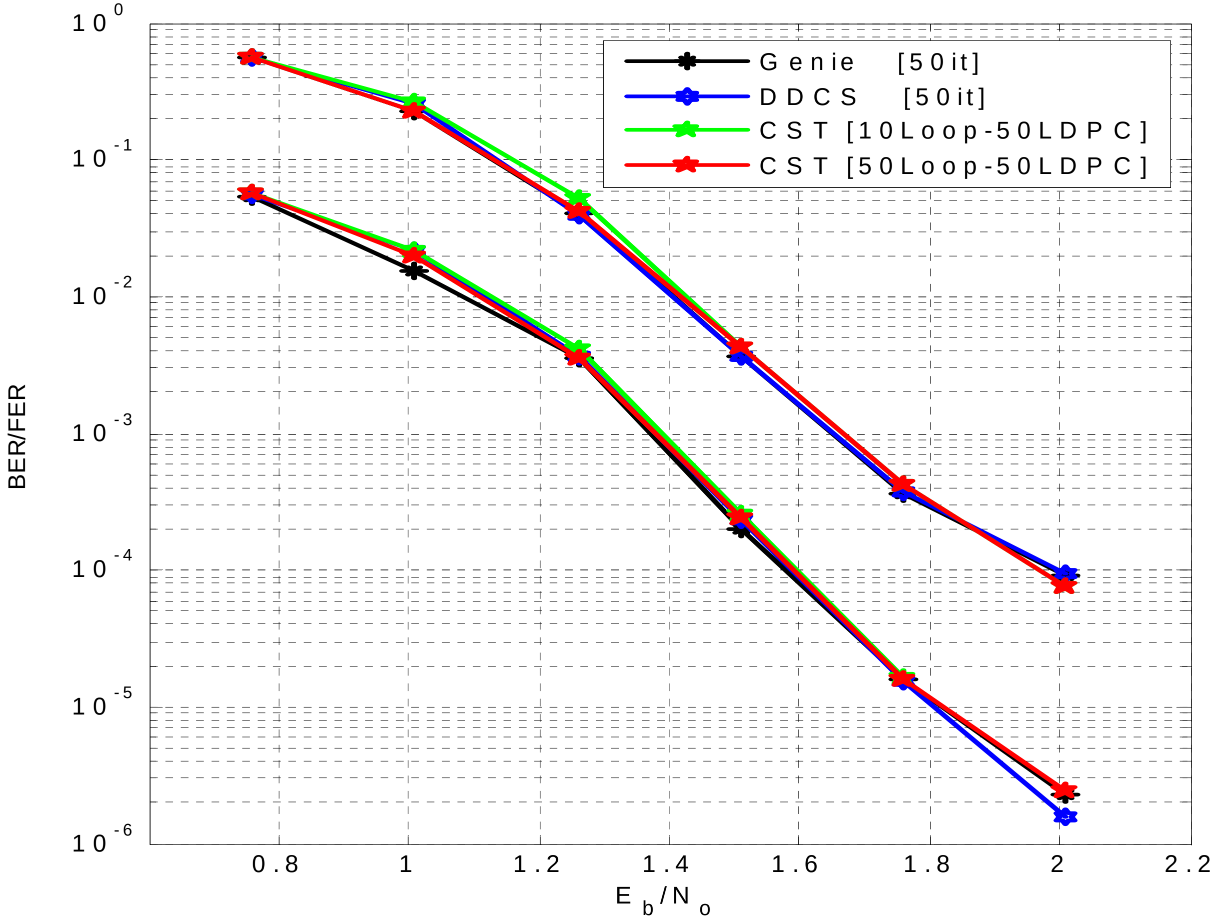
1st , 2nd Moments & Variance

- Costas Variance =
  2.8E-2 →
  LS ≈15 dB
- DDCS Variance =
  6E-4 →
  LS ≈32 dB

- Loop SNR performance shows a difference of around **15dB** for DDCS and Costas Loop



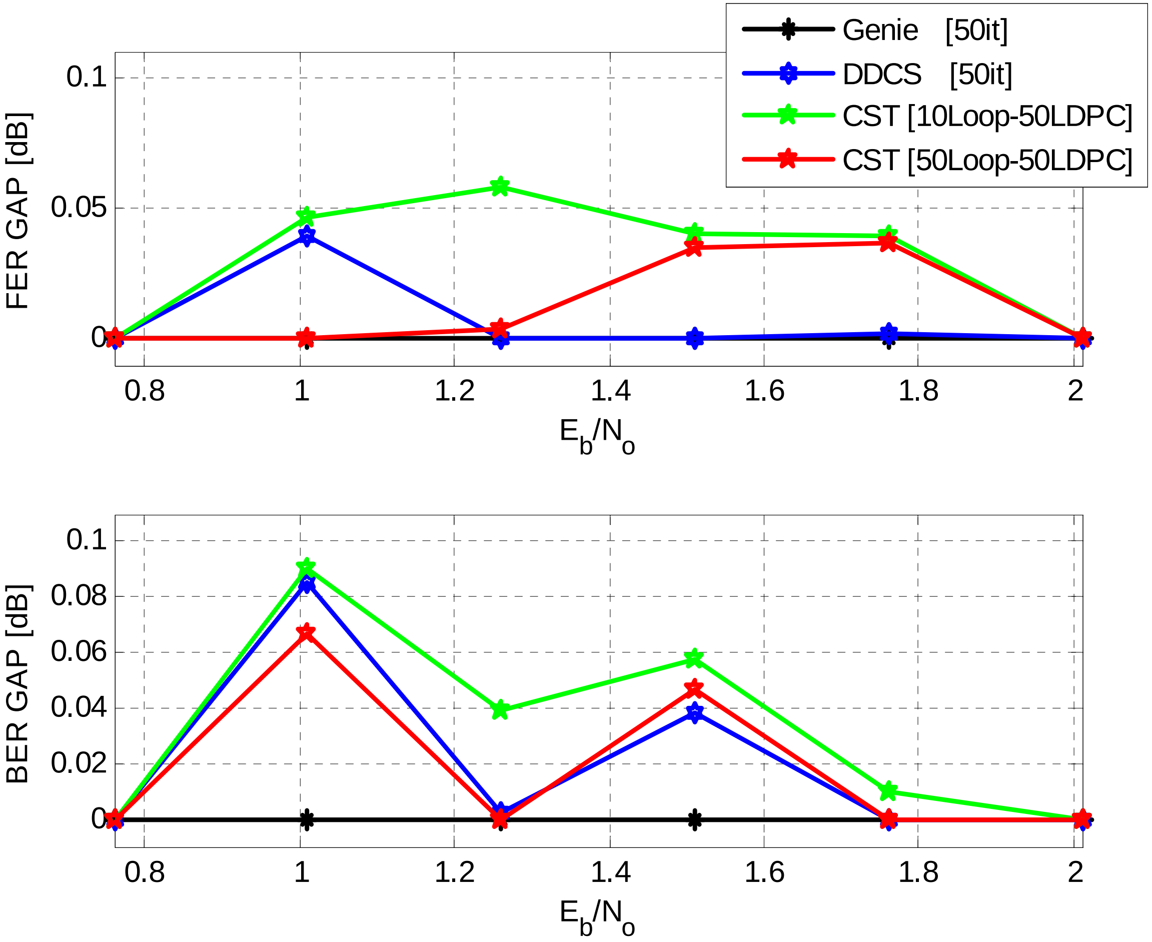
FER/BER



- BER / FER results for 50 iterations ***assuming no symbol timing offsets***

- Results show that for small M-PSK constellations jitter effects do not affect average error symbol performance

BER /FER GAP from Genie Performance



- BER / FER results for 50 iterations ***assuming no symbol timing offsets***

Timing Offset

- **Symbol-wise Offsets**
  - The traditional method used in the literature introduces an offset to each of the conforming pulses of *s(t)* before any superposition is done
  - The major drawback of this method is that it introduces unnecessary ISI

$$r(t) = \sum_{i=0}^{n-1} a_i \cdot h_{RRC}\left(t + \tau_i - iT\right) + N(t)$$

- **Sample-wise Timing Offsets:**
  - Proposed models applies timing shifts to the superimposed signal ***after*** superposition is done. This is done by having a time-varying timing perturbation function t(t) independent of the sample number.

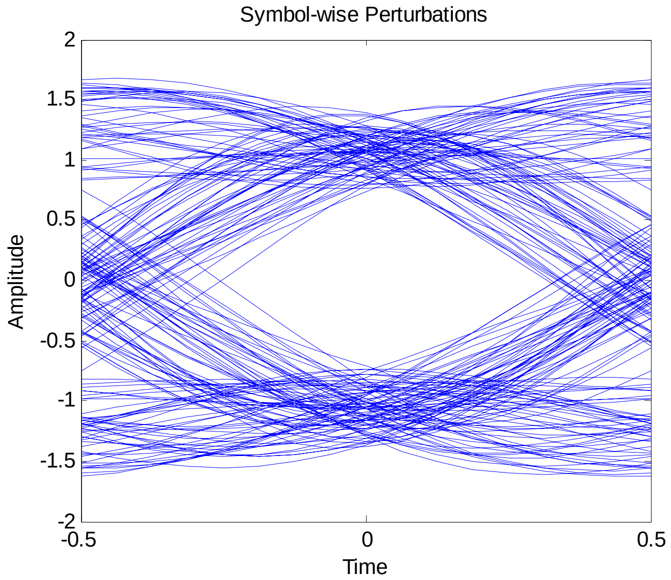
$$r(t) = \sum_{i=0}^{n-1} a_i \cdot h_{RRC}\left(t + \tau(t) - iT\right) + N(t)$$

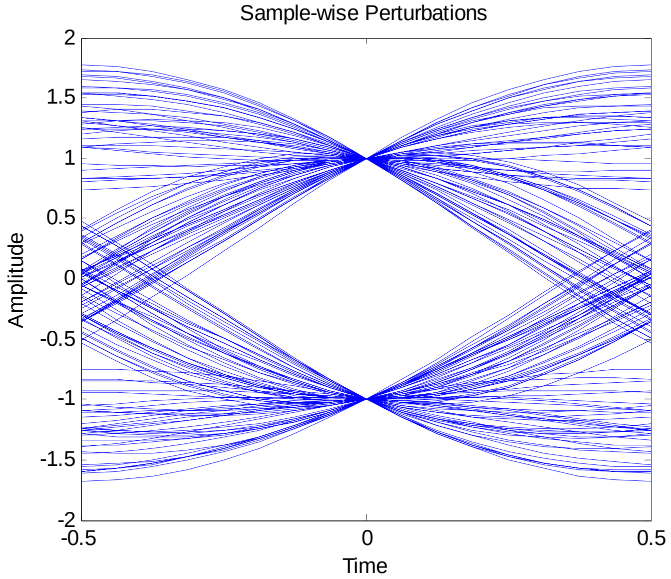
Timing Offset

- Timing perturbations occur in the channel due to:
  - Relative motion between transmitter and receiver
  - Mismatch between clock crystal frequencies
  - Receiver-side timing estimation errors

- Symbol-wise modeling introduces inter-symbol interference (ISI) in the transmitted waveform *(even though an ISI free pulse shape is modulated e.g. RRC)* as shown in the eye diagram

- Symbol-wise perturbation model is not well suited for our channel model assumptions





Timing Error Modalities

- **Constant Time Offset**: All pulses affected by the same constant delay with respect to their ideal sampling time

$$\tau[k] = D$$

where *k* indexes the *k*th received sample.

- **Random Walk**: Models an accumulation process. Timing error is equal to previous error plus a perturbation:

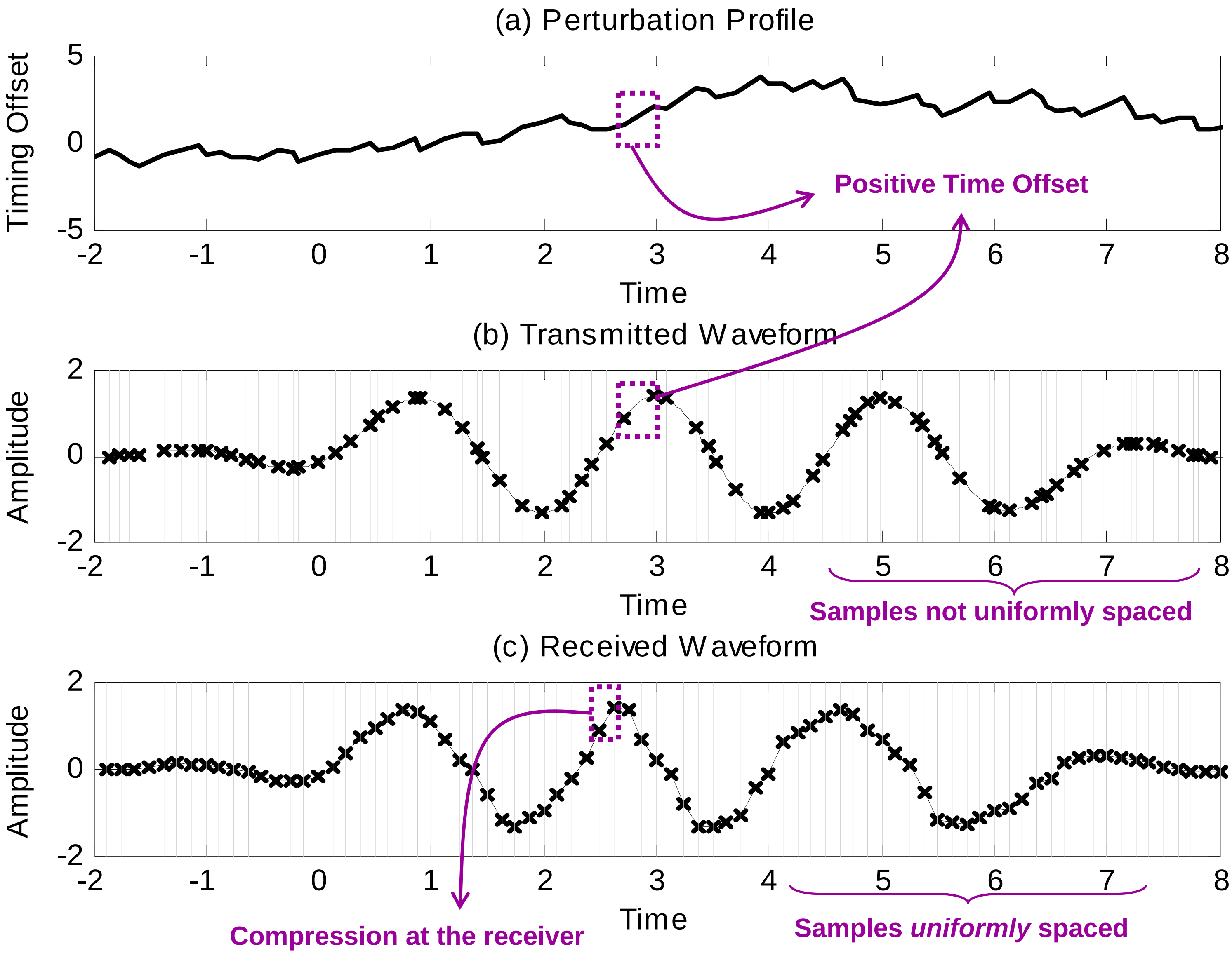
$$\tau[k] = \tau[k-1] + \mathcal{N}\left(0, \sigma_d^2\right)$$

- **Frequency Offset:** Produced by
  - Mismatch between transmitter and receiver clock crystals
  - Doppler shift due to relative motion between transmitter and receiver

$$\tau[k] = \tau[k-1] + T_s\left(Freq\ \%\right)$$

Random Walk Timing Perturbation

- Timing offset which is tracked on a *sample-by-sample* basis



Positive Time Offset

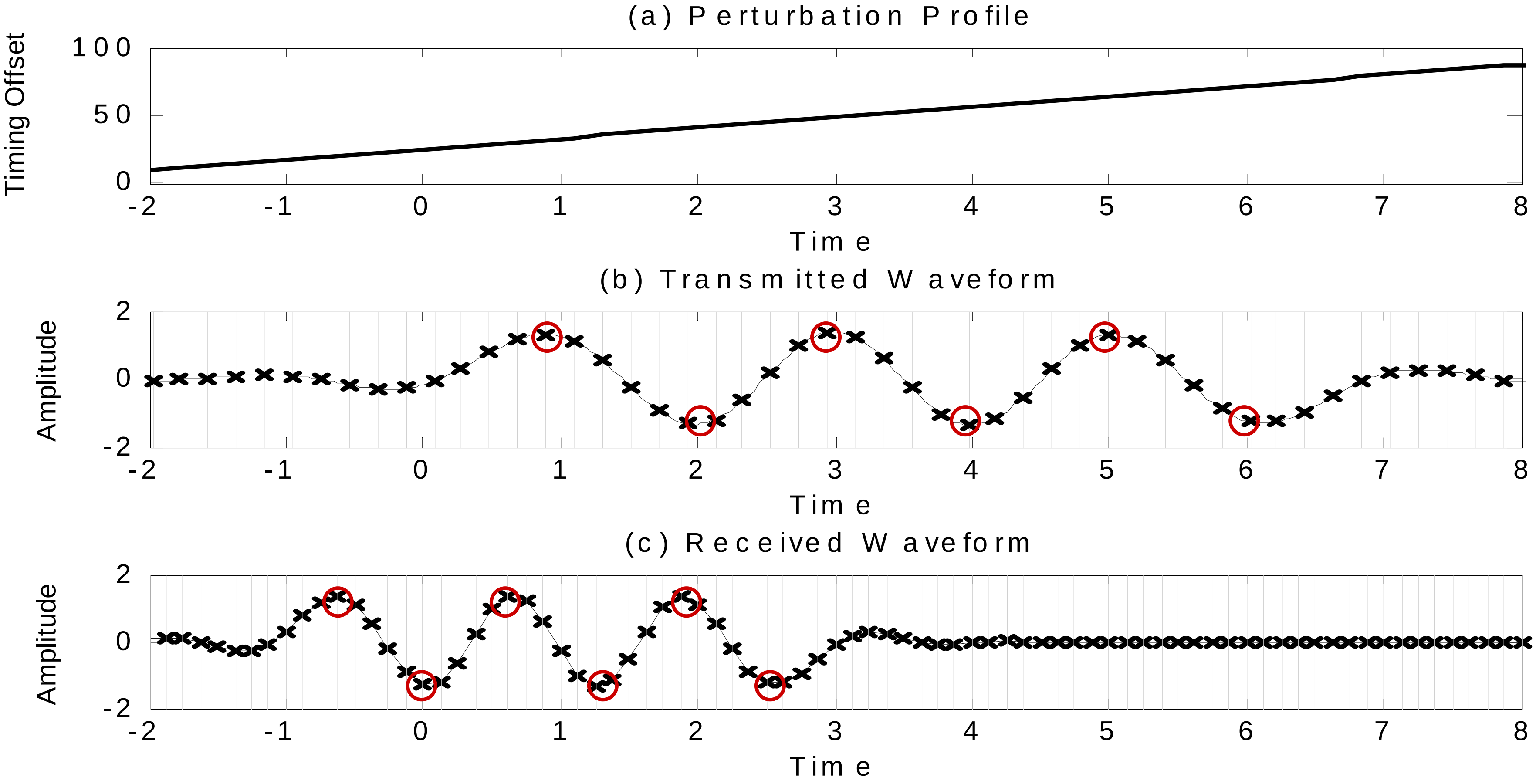
Samples not uniformly spaced

Samples *uniformly* spaced

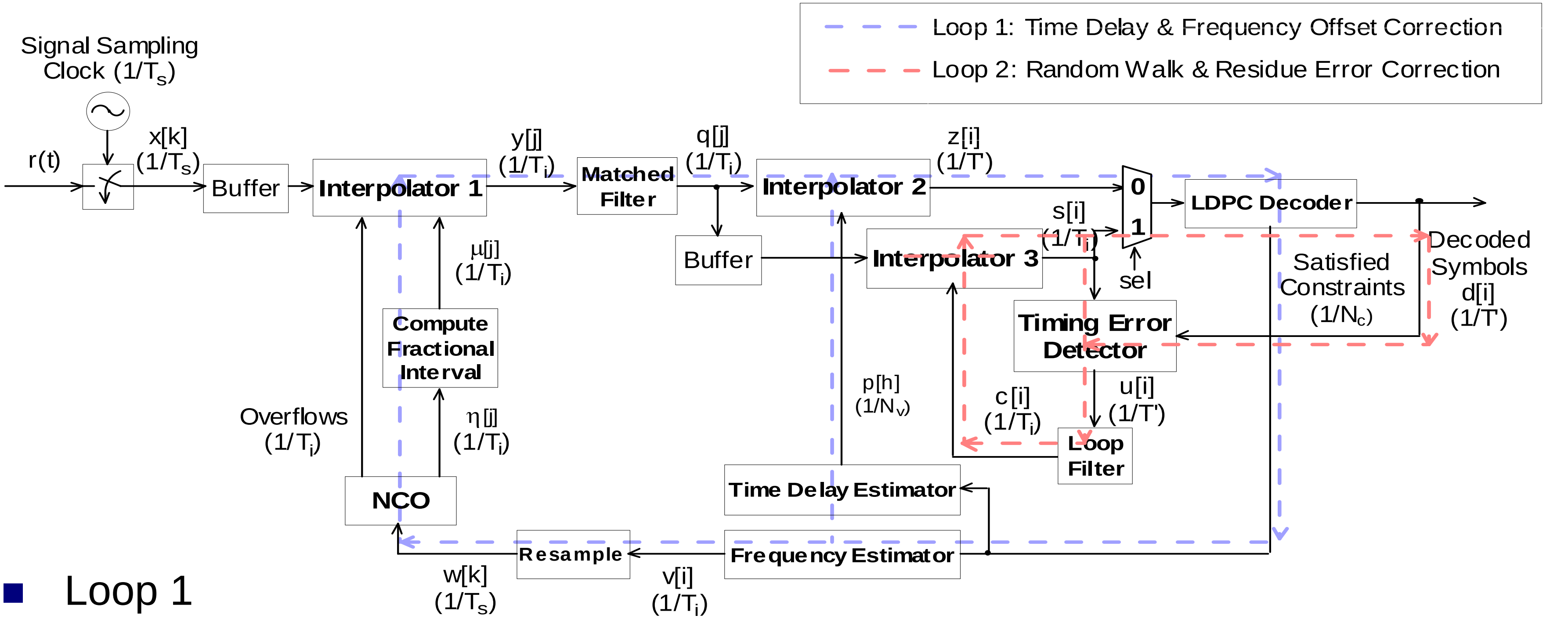
Compression at the receiver

Constant Frequency Offset

- Frequency tracking is done in a Quasi-Static manner (ie. frequency offsets are tracked on a *block-by-block* basis).
- Positive frequency offset causes a time-compression on the received waveform
- Samples are still ISI-free



Symbol Timing Recovery Model



- Loop 1
  - □ Tracks large scale phase and frequency offset
  - □ Interpolator 1 corrects arbitrary frequency mismatch in LDPC block
  - □ Interpolator 2 corrects arbitrary phase offset in LDPC block
- Loop 2
  - □ Fine interpolation of frequency/phase compensated matched filter output
  - □ Dynamic tracking within a block: random walk and residual loop 1 error
  - □ Use Müller-Mueller timing error detector (M&M TED) is used

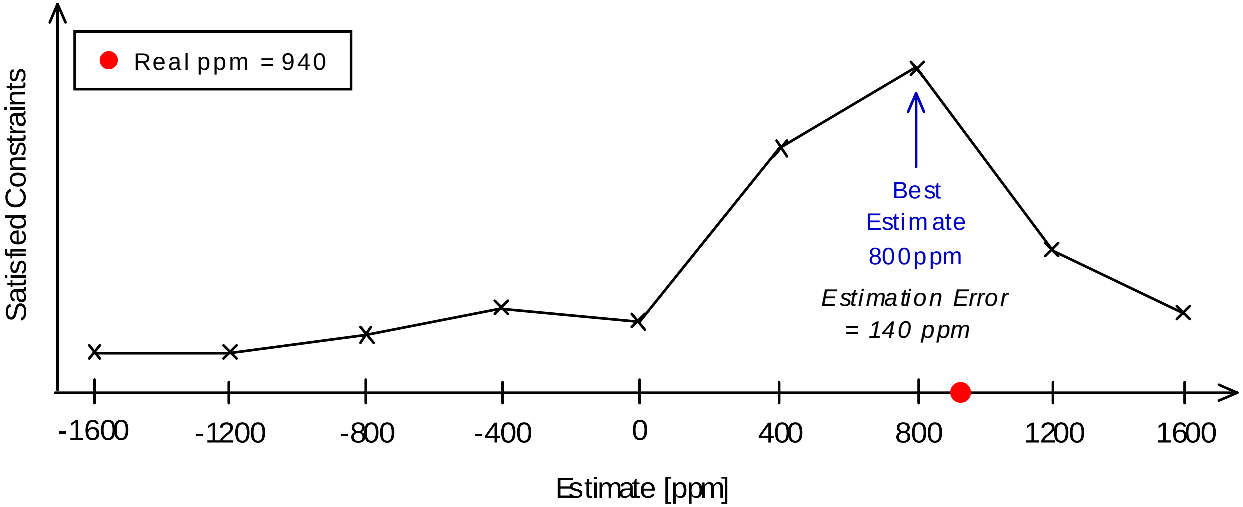
Frequency Offset & Time Delay Search Example: Method A

- Offers very accurate estimations but requires a large number of iterations
- Complexity grows linearly with offset

Search Iteration 1

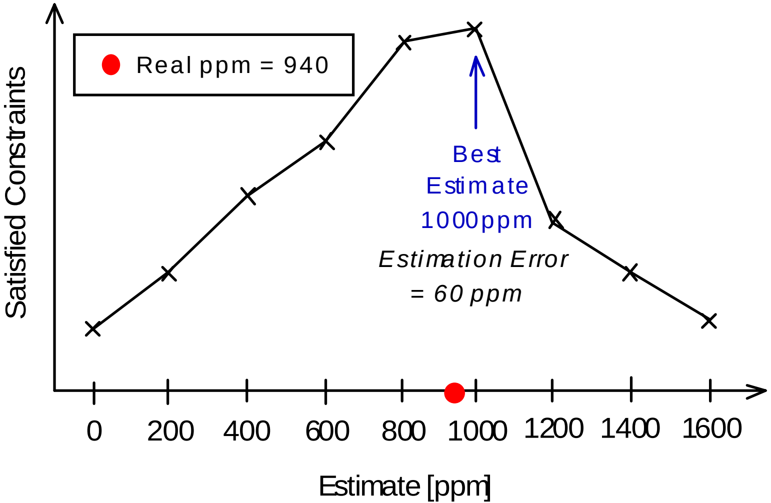
Window	=	3200 ppm
Center	=	0 ppm
Min	=	-1600 ppm
Max	=	1600 ppm
Step	=	400 ppm

- Center next search at 800ppm
- Reduce window and step size e.g. by a factor of 2
- Perform *I* LDPC iterations (*I*<4) for each estimate



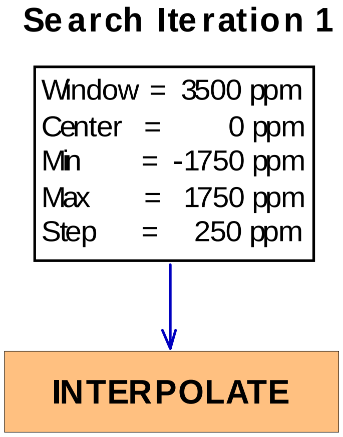
Search Iteration 2

Window	=	1600 ppm
Center	=	800 ppm
Min	=	0 ppm
Max	=	1600 ppm
Step	=	200 ppm



Frequency Offset & Time Delay Search Example: Method B

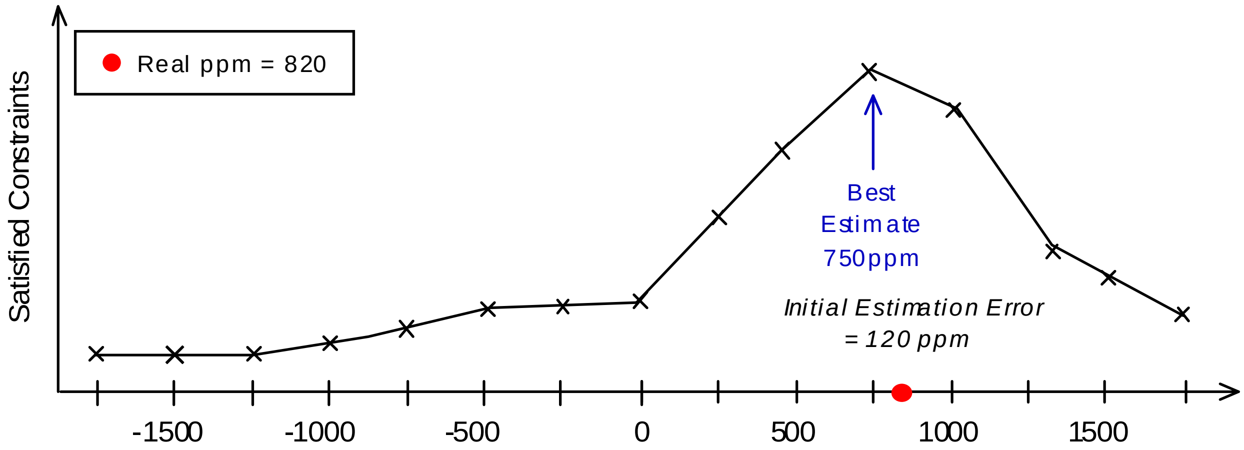
- Use a smaller step but only one frequency search
- Produce closer estimates that allow a curve fitting interpolation to reduce the initial estimation error



Search Iteration 1

Window	=	3500 ppm
Center	=	0 ppm
Min	=	-1750 ppm
Max	=	1750 ppm
Step	=	250 ppm

INTERPOLATE



- Method A
  - □ Diminishing returns after 3 search iterations and 3 LDPC iterations
  - □ Estimation accurate within 40 ppm
  - □ Initial search window of ±1600ppm
    - 3 search iter * 3 LDPC iterations * 9 estimates per search = 81 total LDPC iterations
- Method B
  - □ Estimation accurate within ≈60 ppm
  - □ Initial search window of ±1750ppm
    - 3 search iter * 1 LDPC iterations * 15 estimates per search = 45 total LDPC iterations

Tracking Random Walks

- Tracked with a 1st order PLL-based circuit (loop 2)
  - □ Müller-Mueller timing error detector (M&M TED) is used
    - M&M TED updates its timing error estimate according to:

$$u[i] = s[i]d[i-1] - s[i-1]d[i]$$

where s[i] is the symbol from the interpolator and d[i] is the decoded symbol from LDPC decoder

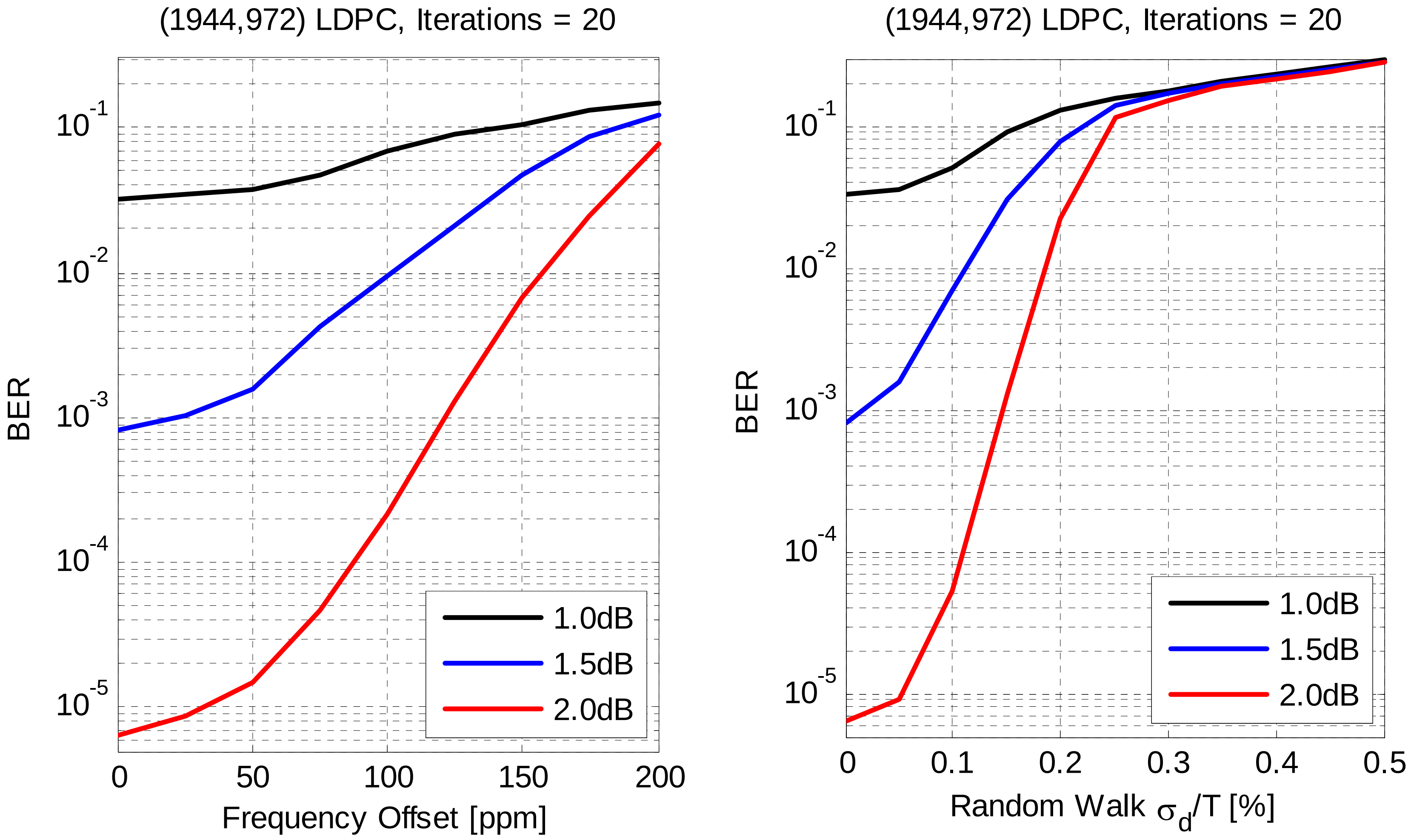
  - □ u[i] is noisy, hence attenuated with a 1st order loop filter:

$$c[i] = c[i-1] + K_p \times u[i]$$

where K_p is the proportional gain and u[i] is the output of the M&M TED

- This circuit also corrects residual time delay and frequency offset

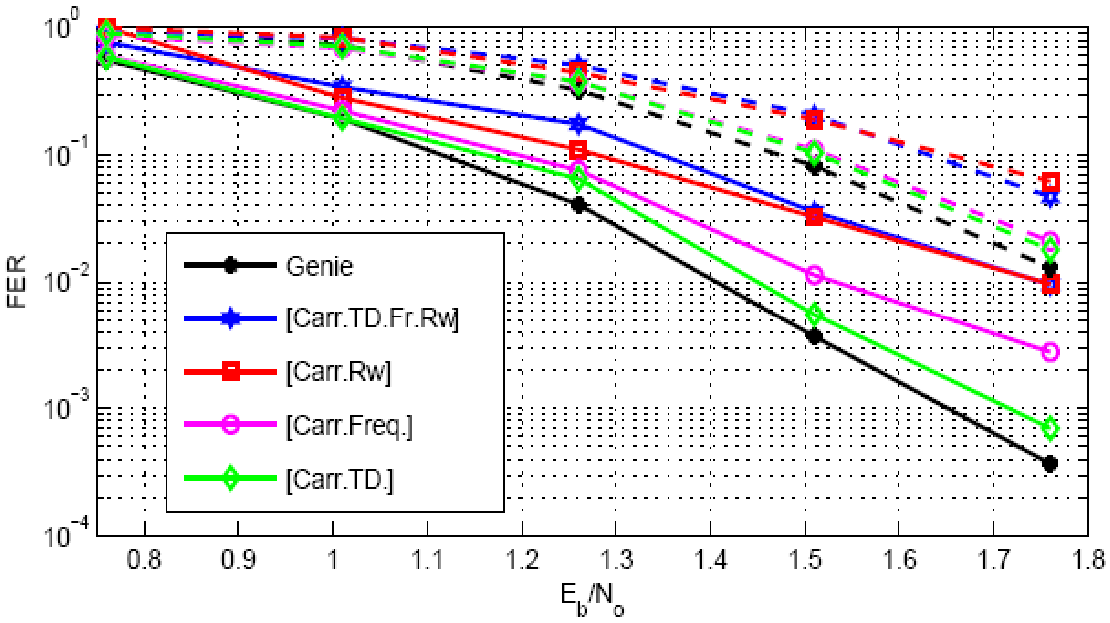
Effects of Timing Errors (No Correction)



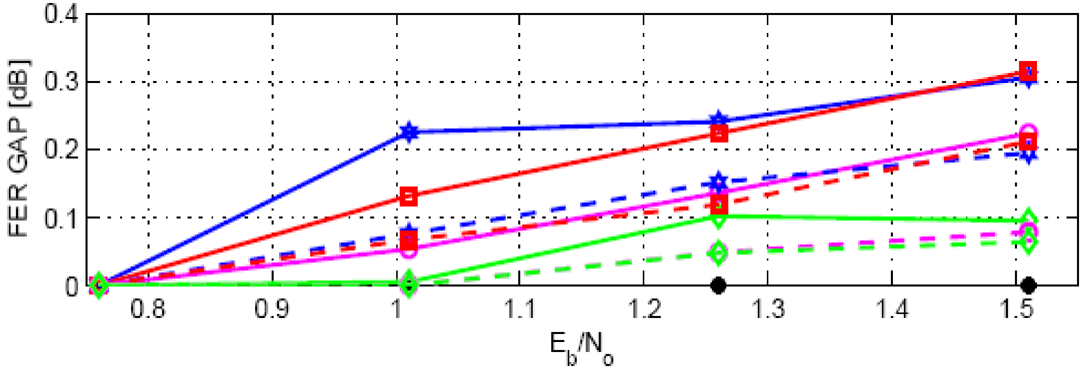
SNR given in E_b/N_0

Symbol Timing & Carrier Recovery

- Fig.(a) FER 50 Iterations(solid) / FER 20 Iterations(dashed) performance.
- Legend format
    - □ [Carr.] indicates the presence of a =π/4 carrier phase offset,
    - □ (Freq.) Symbol-frequency offset
    - □ (TD) Symbol Time delay
    - □ (Rw.) Random walk
- Fig.(b) Shows the SNR gap with respect to the “genie-aided” performance for the same set of curves



(a)



(b)

Summary

- Carrier synchronization circuit overcomes the performance loss due to a noisy signal reference at low SNRs, characteristic of suppressed carrier loops such as the Costas loop.
- Steady state operation is reached after around 15 joint DDCS/LDPC iterations.
- Two-stage pilotless symbol *timing recovery* model for tracking time delay, frequency offsets and random walks
- 1st stage: LDPC constraint feedback to track large-scale time delays and symbol frequency offsets
- Windowed search method
  - Complexity grows linearly with offset
  - Can track any time delay frequency offset as long as it is within the initial search window
- 2nd stage: decoded LDPC symbols to track random walks and residual errors from the 1st stage
- Simulations results show within 0.4 dB of the ideal code performance for time delays and frequency offsets