USC Viterbi School of Engineering
Center for Cyber-Physical Systems and the Internet of Things

SOFTWARE DEFINED RADIO

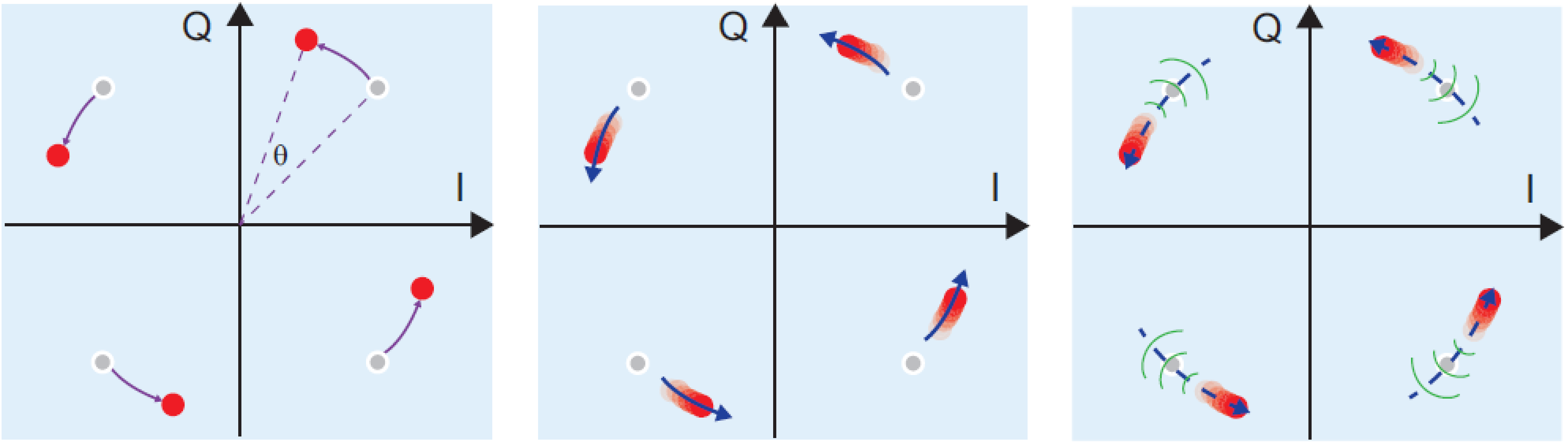
USR SDR WORKSHOP, SEPTEMBER 2017

PROF. MARCELO SEGURA

SESSION 4: TIME AND FREQUENCY SYNCHRONIZATION

PHASE SYNC

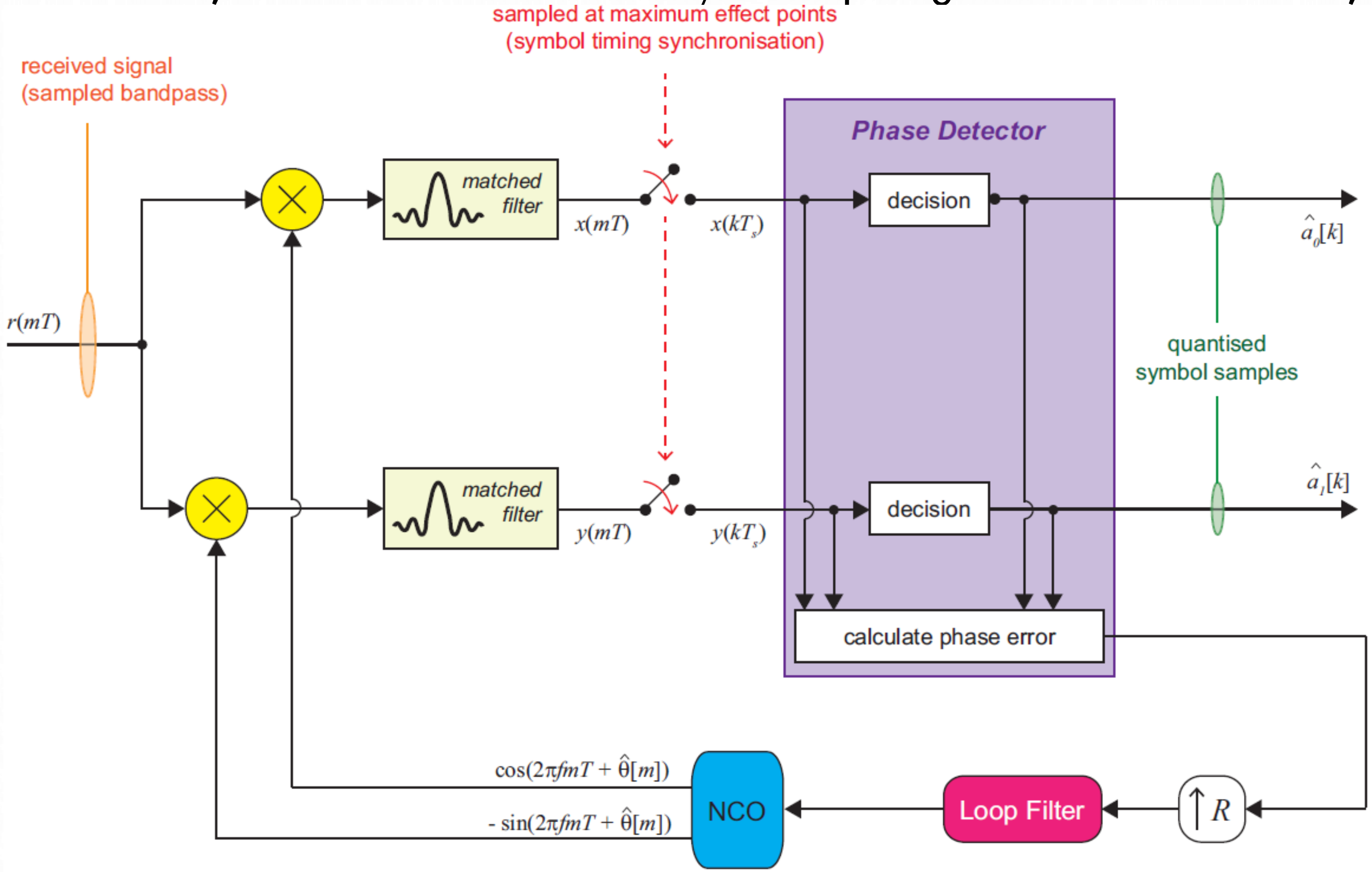
- Problems:
  - LO same frequency but out of phase.
  - LO slightly different frequency.
  - LO phase and frequency changing relative each other.



(a) Received constellation (***fixed phase offset***) (b) Received constellation (***fixed frequency offset***) (c) Received constellation (***changing frequency offset***)

AT DEMO

- We will consider at this stage that there is perfect timing sync.
- The input comes from ADC, and after demodulation I/Q samples goes to Phase detector, one sample per symbol.

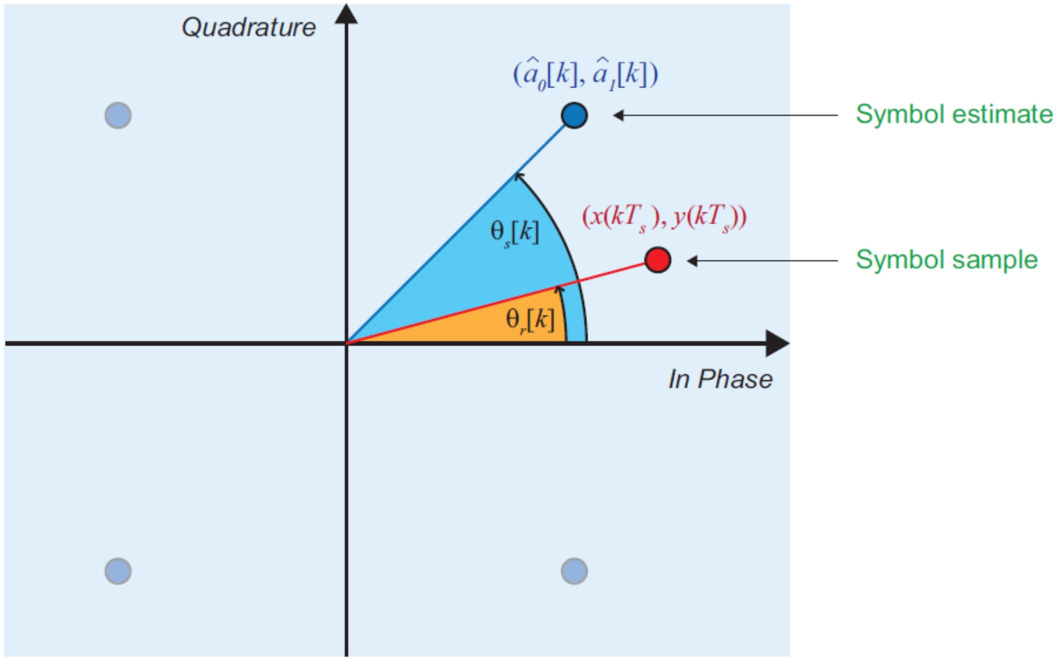


AT DEMO

- The phase of the samples are compared with a reference constellation to detect the error.
- There are two possible algorithm to do it:
  - **Decision Direct:** the transmitted symbols are unknown and error is generated on symbol decision.
  - **Data Aided:** the RX know in advance the symbols and correct the phase accordingly.
- In QPSK Decision Direct, samples are mapped to the nearest symbol. Then the phase after and before quantization is calculated and finally error could be estimated.

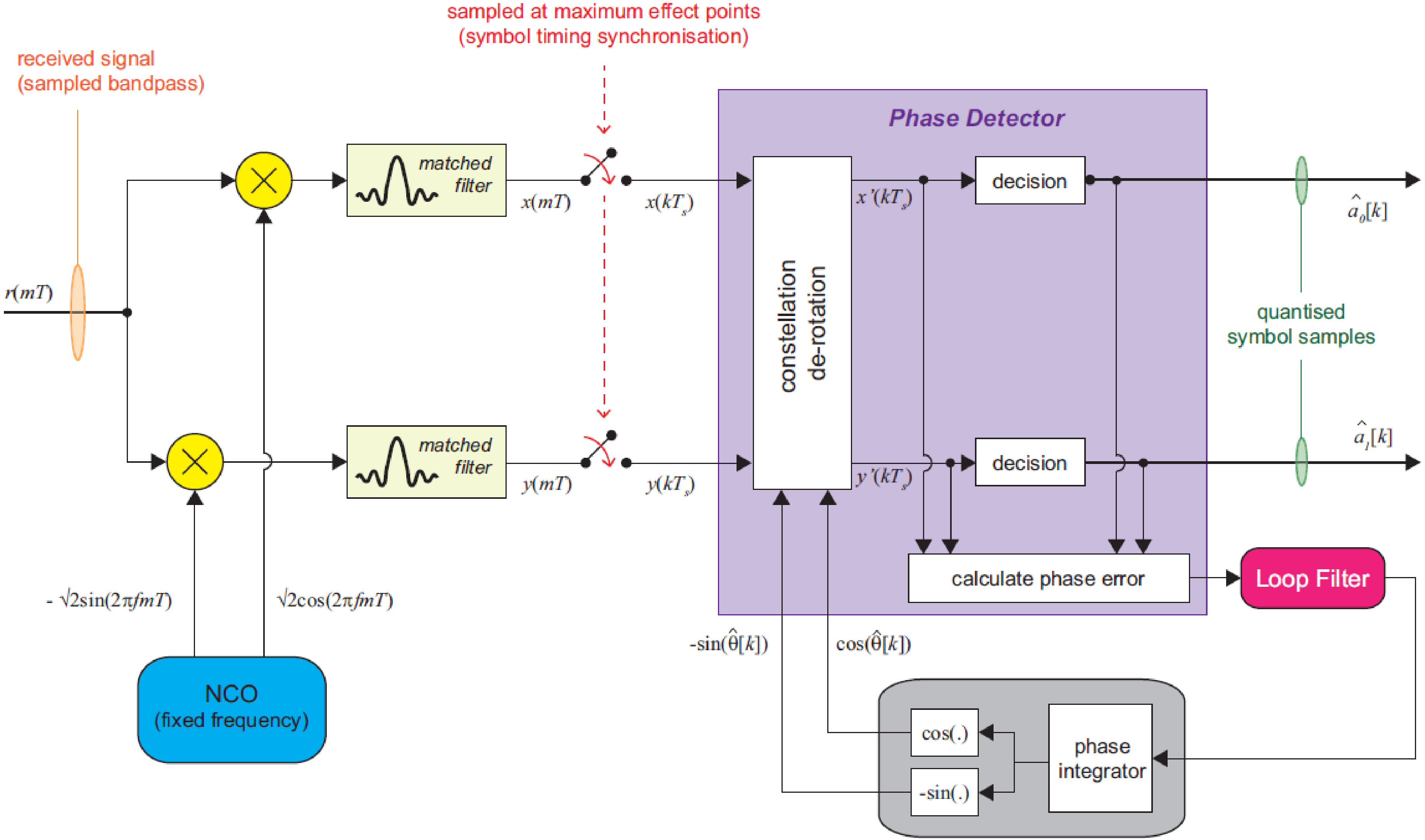
$$\theta_r = \tan^{-1}\left(\frac{y(kT_s)}{x(kT_s)}\right) \qquad \theta_r = \tan^{-1}\left(\frac{y(kT_s)}{x(kT_s)}\right)$$

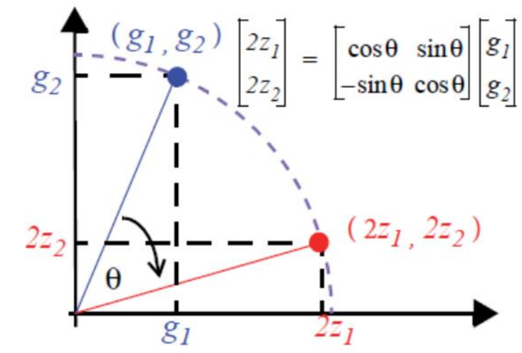
$$\theta_e[k] = \theta_r[k] - \theta_s[k]$$



AT BASEBAND

- Advantages: this implementation require less computational power since operate and symbol rate.

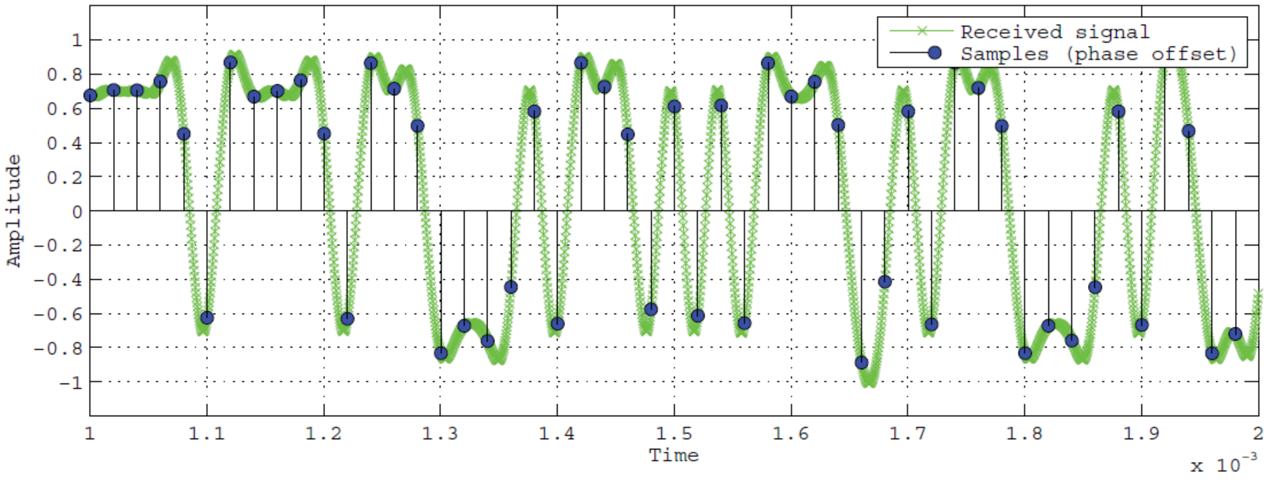


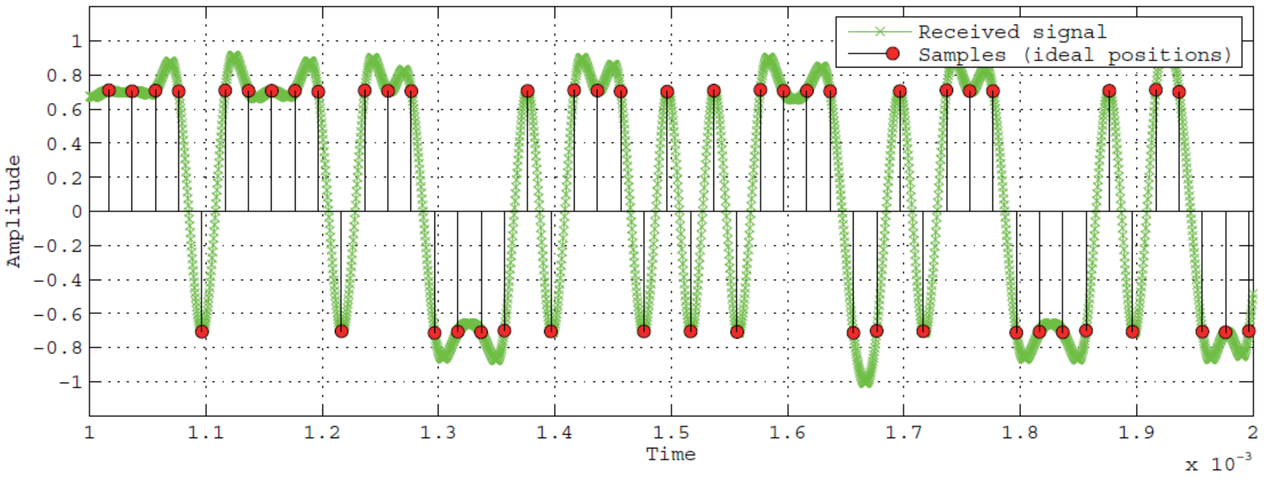


$$\begin{bmatrix} 2z_1 \\ 2z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

TIMING RECOVERY

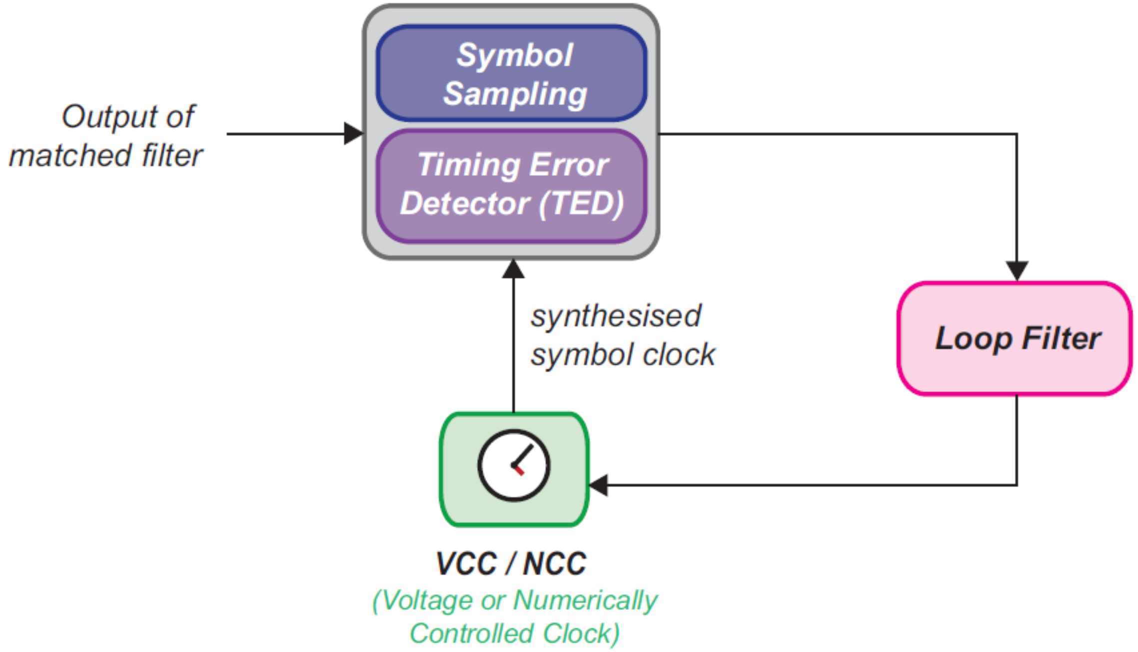
- The accuracy on the de-mapping process, it is direct related to the maximum effective points.
- If RX take the samples at this points, SNR will be maximum and ISI minimized.
- Timing error sources:
  - Timing phase error: taken at correct time but offset phase.
  - Timing frequency error: **most probably**, frequency will be different on TX/RX, so the sampling phase change over time.
  - Timing jitter: from the clock source that affect ADC/DAC.





TIMING SYMBOL RECOVERY

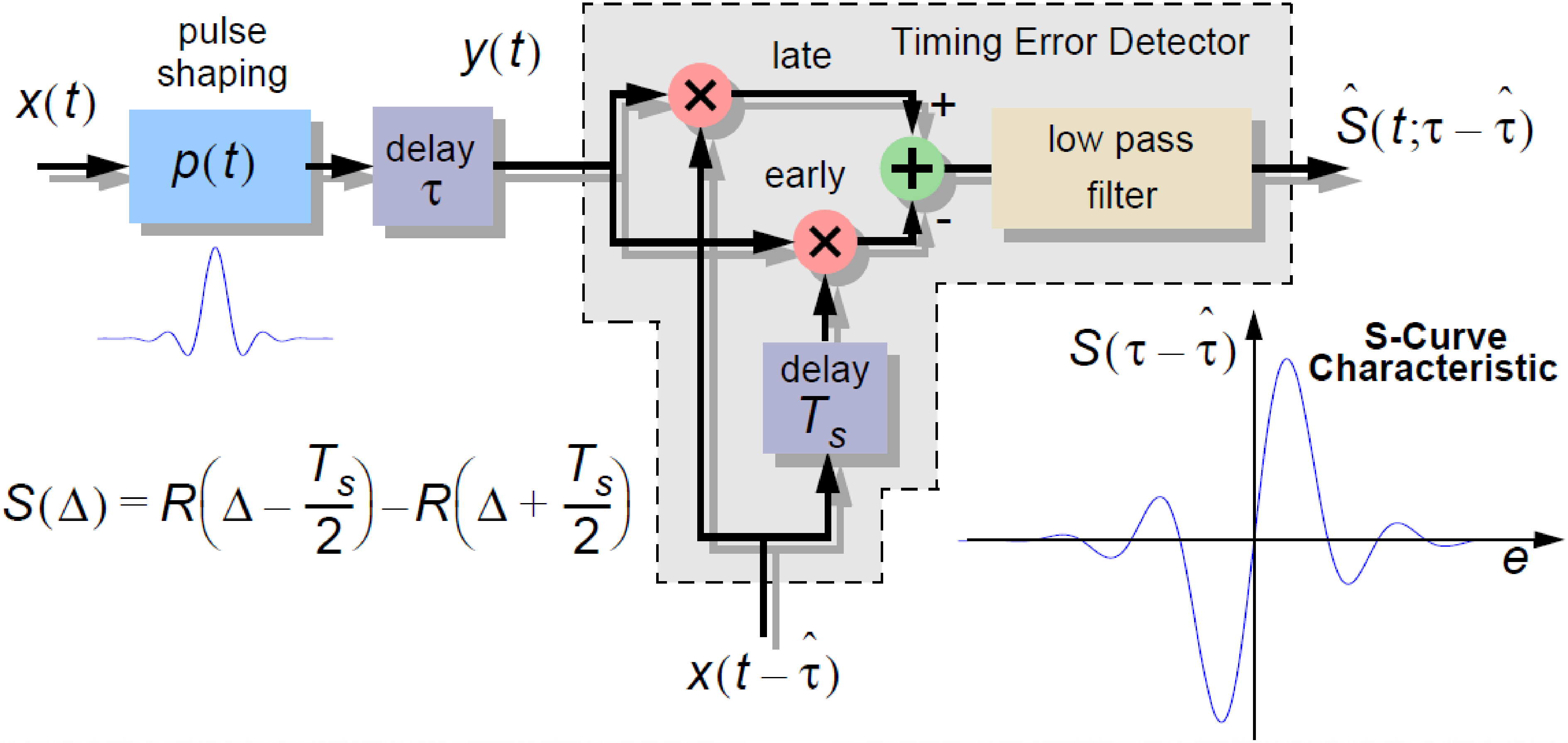
- With time frequency error, not all the samples are taken at the same rate, so the frequency change and need to be followed.
- The timing synchronization has similar blocks than carrier sync:
  - Instead of phase detector, there is a Timing Error detector (TED).
  - There is also a loop filter.
  - Instead of VCO/NCO, there is a voltage or numerically controlled clock. That create the reference signal to sample at right time/phase.



TIMING ERROR DETECTOR

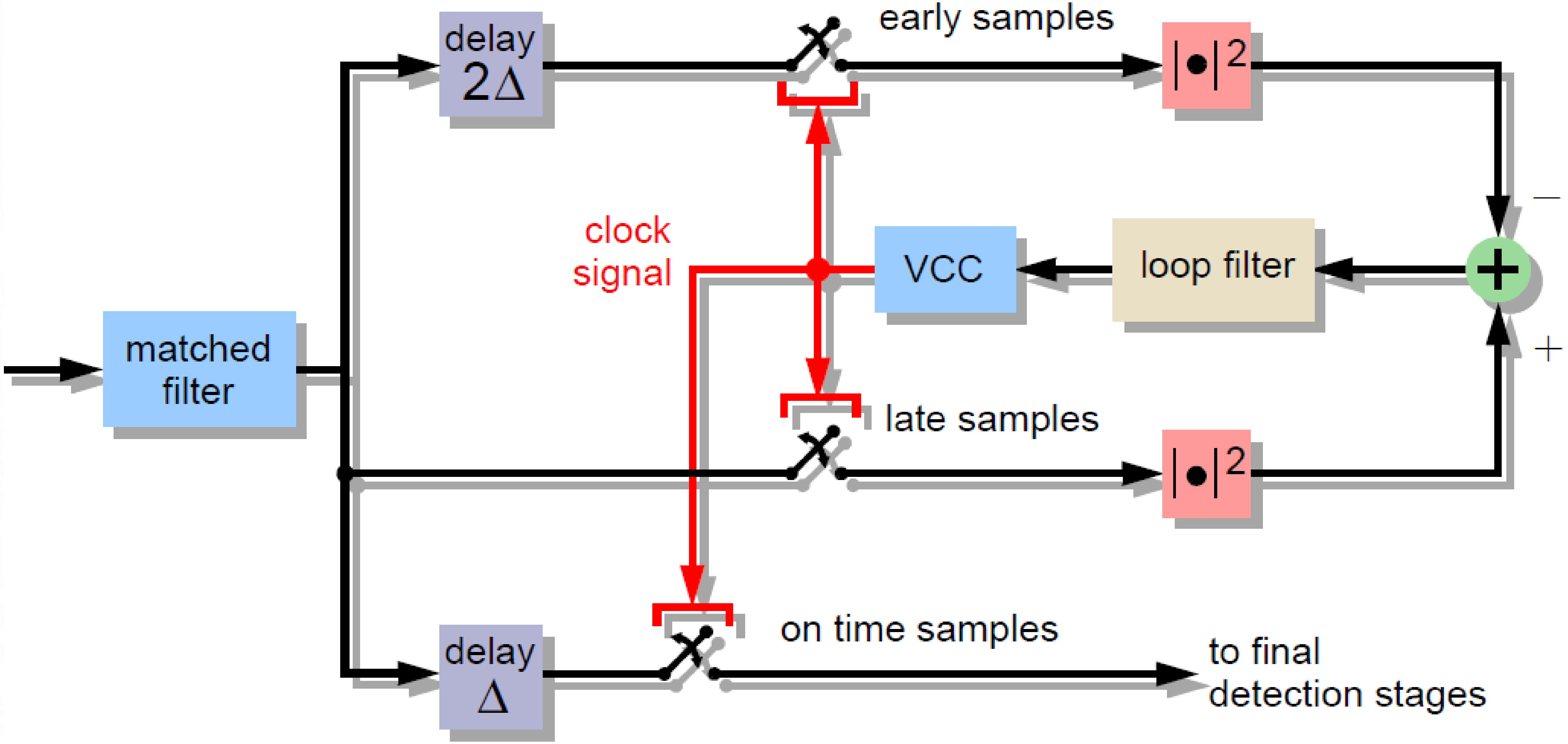
- TED algorithms:
  - **Early-Late Gate**
  - **Zero Crossing (Gardner Loop)**
  - **Maximum likelihood**
  - **Mueller and Meuller**
- Depending on the previous signal knowledge the algorithm can be classified as:
  - **Data Aided**
  - **Decision Direct**
- The timing adjustment can be do using:
  - **Oversampling:** the sampling rate is several times higher than the symbol rate. It chose the closest sample to the optimal.
  - **Interpolation:** the sample rate is two times the symbol rate and after interpolation, the sampling point is obtained.

TED: EARLY LATE ANALOG



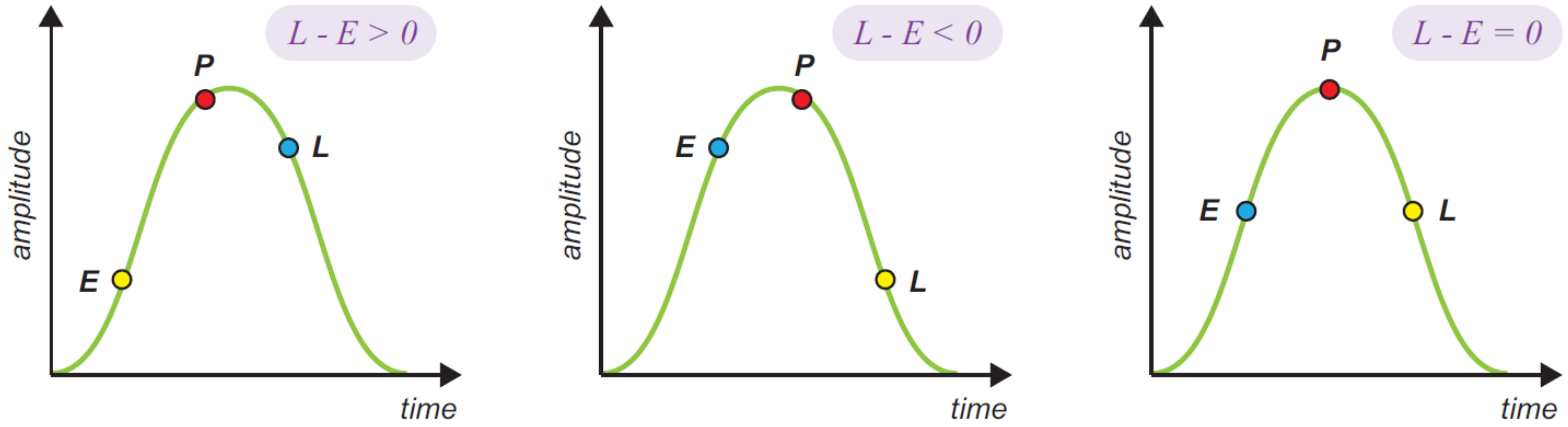
$$S(\Delta) = R\left(\Delta - \frac{T_s}{2}\right) - R\left(\Delta + \frac{T_s}{2}\right)$$

TED: EARLY LATE



TED: EARLY LATE

- The Early-Late TED, take 3 simple per symbol: Late, Early, Punctual.



TED: EARLY LATE

- The S-curve define the relation between the error and the TED output. **It's depend** on the pulse shape, therefore on the **signal amplitude**.
- In order to avoid timing **dependent on signal power**, usually a AGC is used before TED.

$$e[k] = \hat{a}[k]\left[x(kT_s + \Delta T_s + \hat{\tau}) - x(kT_s - \Delta T_s + \hat{\tau})\right]$$

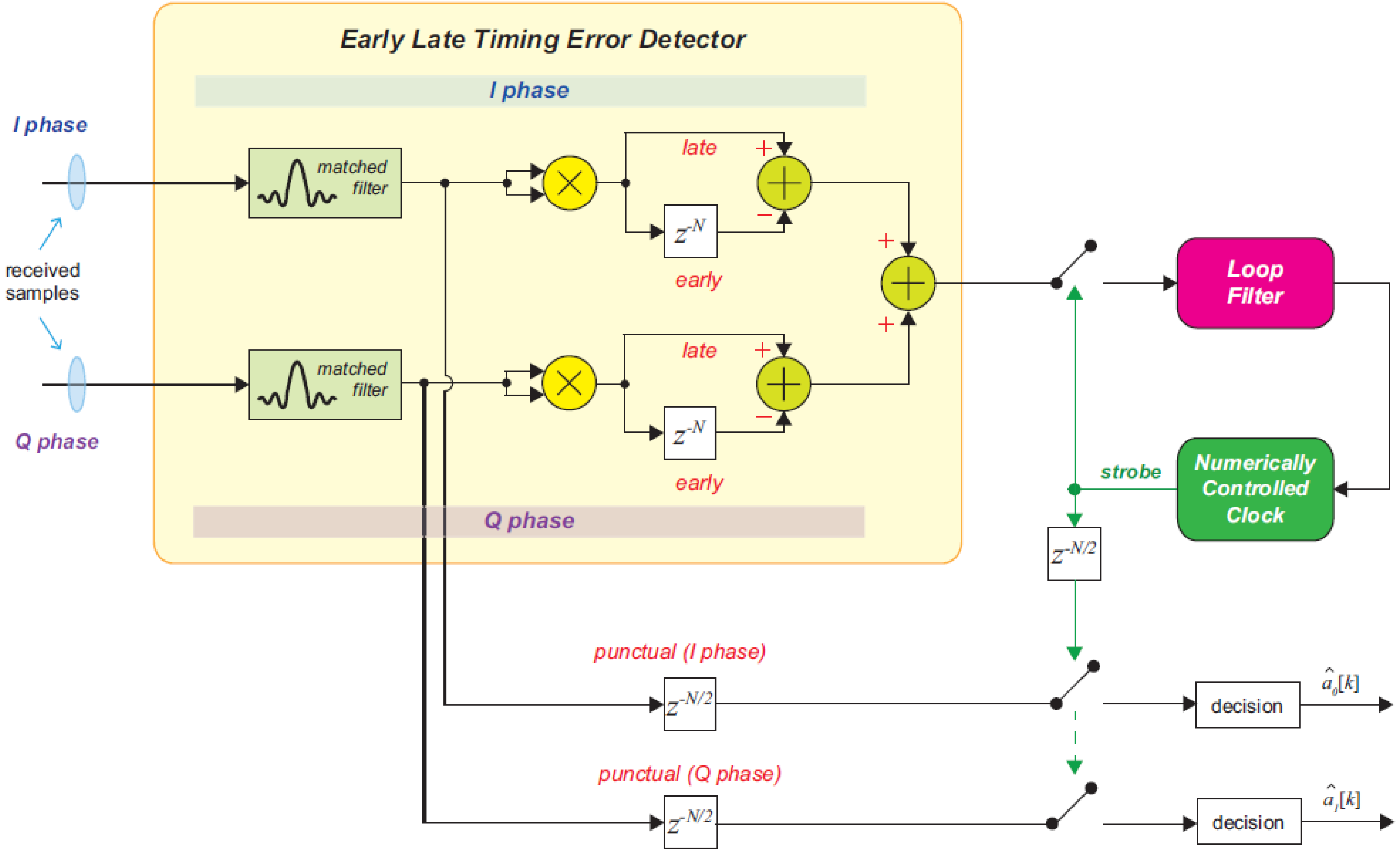
$$e[k] = \hat{a}[k]\left[x((k+0.5)T_s + \hat{\tau}) - x((k-0.5)T_s + \hat{\tau})\right]$$

- Because Early-Late works with positive pulses, we have to square the signal.

$$e[k] = \left[x^2(kT_s + \Delta T_s + \hat{\tau}) - x^2(kT_s - \Delta T_s + \hat{\tau})\right]$$

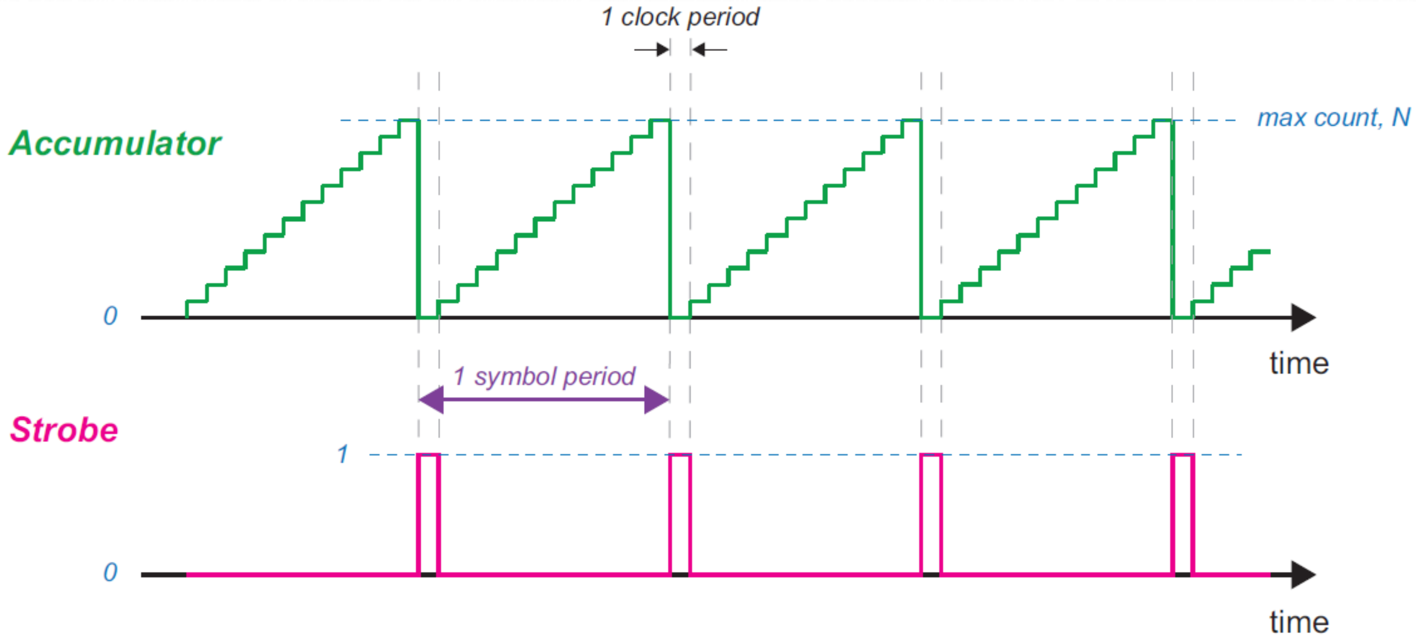
TED: EARLY LATE

- After converge the NCC will create the strobe signal Ts/2 later, respect to the Punctual. Therefore, an additional delay is added to sample the symbol at the maximum effective point.



NCC

- NCC instead of generate a sine signal, it give strobe signal to capture the samples. The accumulator control is done changing the step size. The rate of the NCC is adjusted to be the symbol rate.



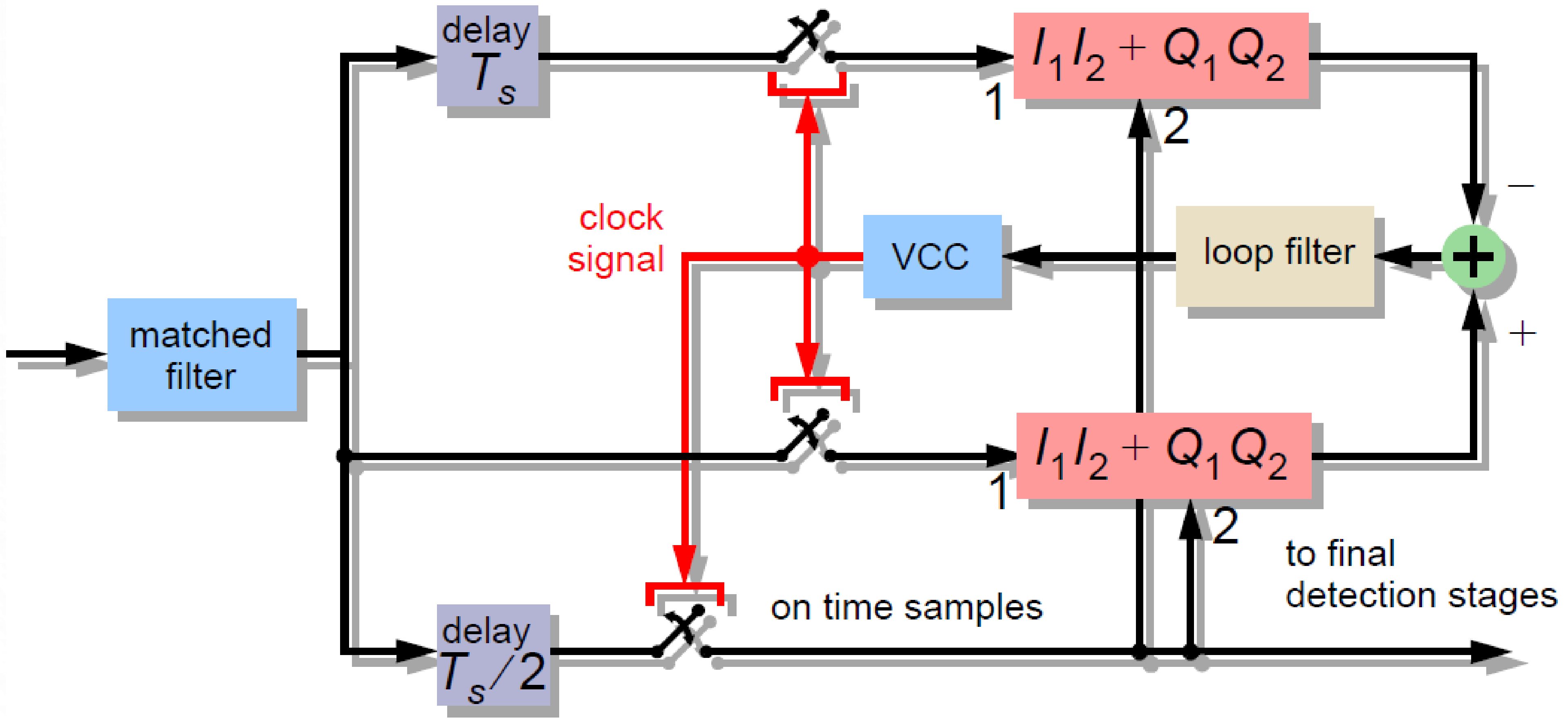
$$r[k] = r[k-1] + 1 + \mu_{fb}[k] \quad mod(N)$$

$$s[k] = \begin{cases} 1 & \text{when} \quad r[k] < r[k-1] \\ 0 & \text{otherwise} \end{cases}$$

- For the Timing adjust **interpolation** case, instead of strobe signal de NCC give a numerical value that select the polyphase filter **branch**.

TED: ZERO CROSSING

- Modified version of Early Late.
- Advantage, no sensitive at carrier offset (no-coherent), so Timing sync is **independent** on frequency sync

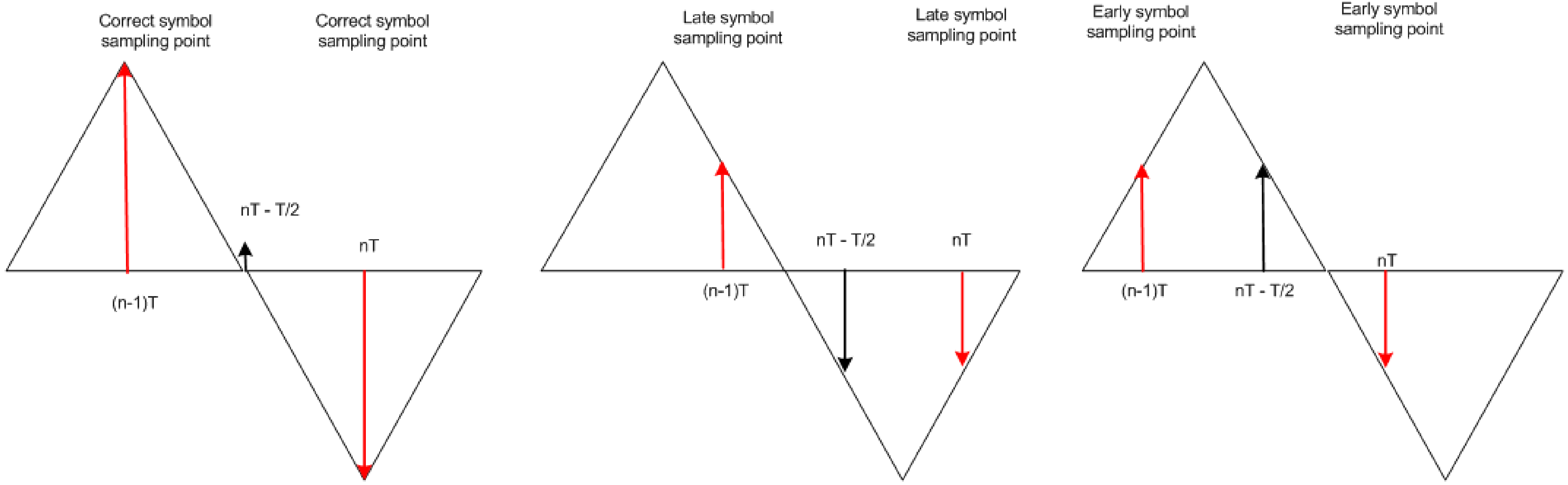


TED: ZERO CROSSING

- Zero Crossing Error:

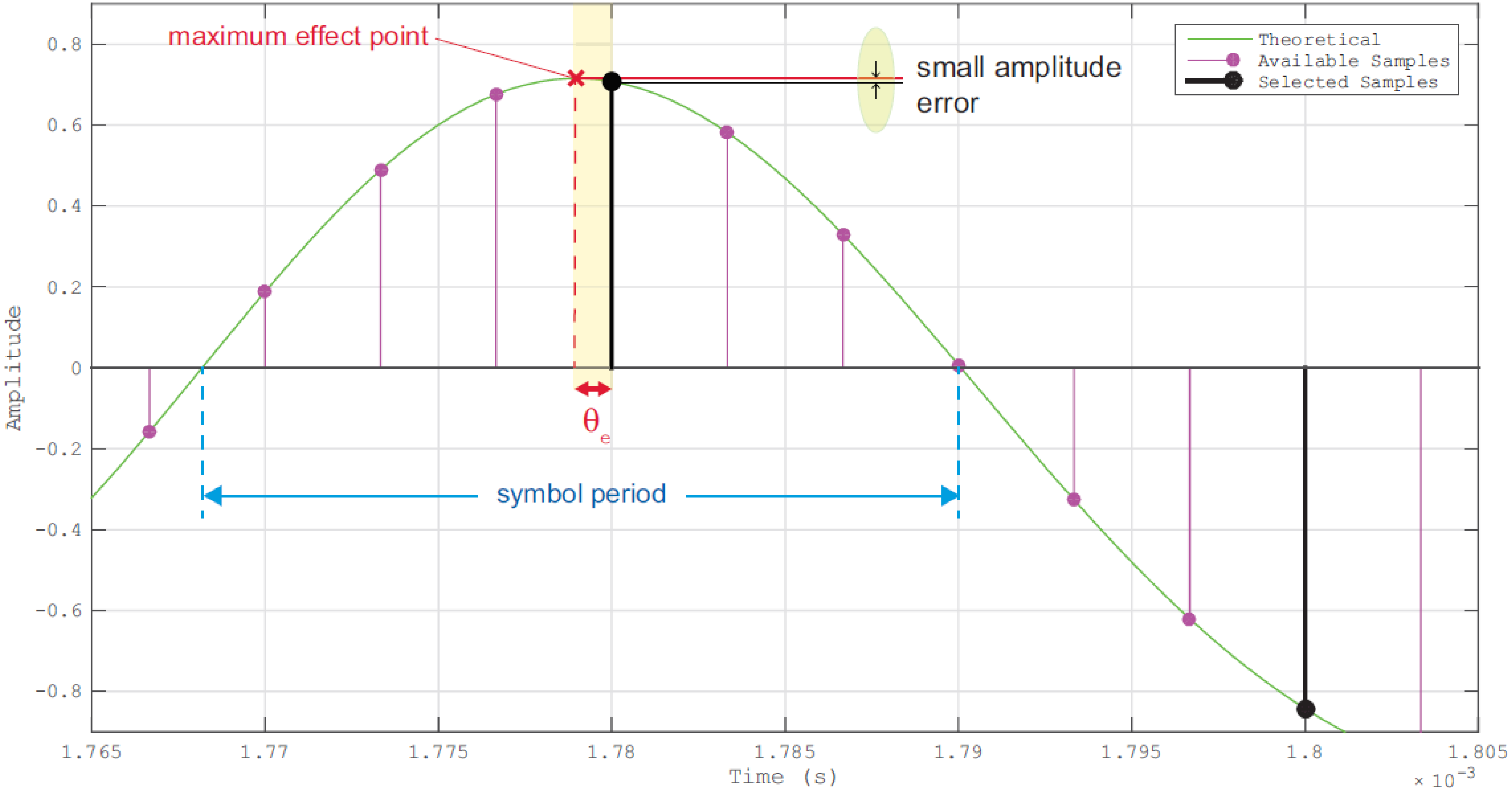
$$e = \{x[nT] - x[(n-1)T]\}x[nT - T/2]$$

- Need only 2 samples per symbol

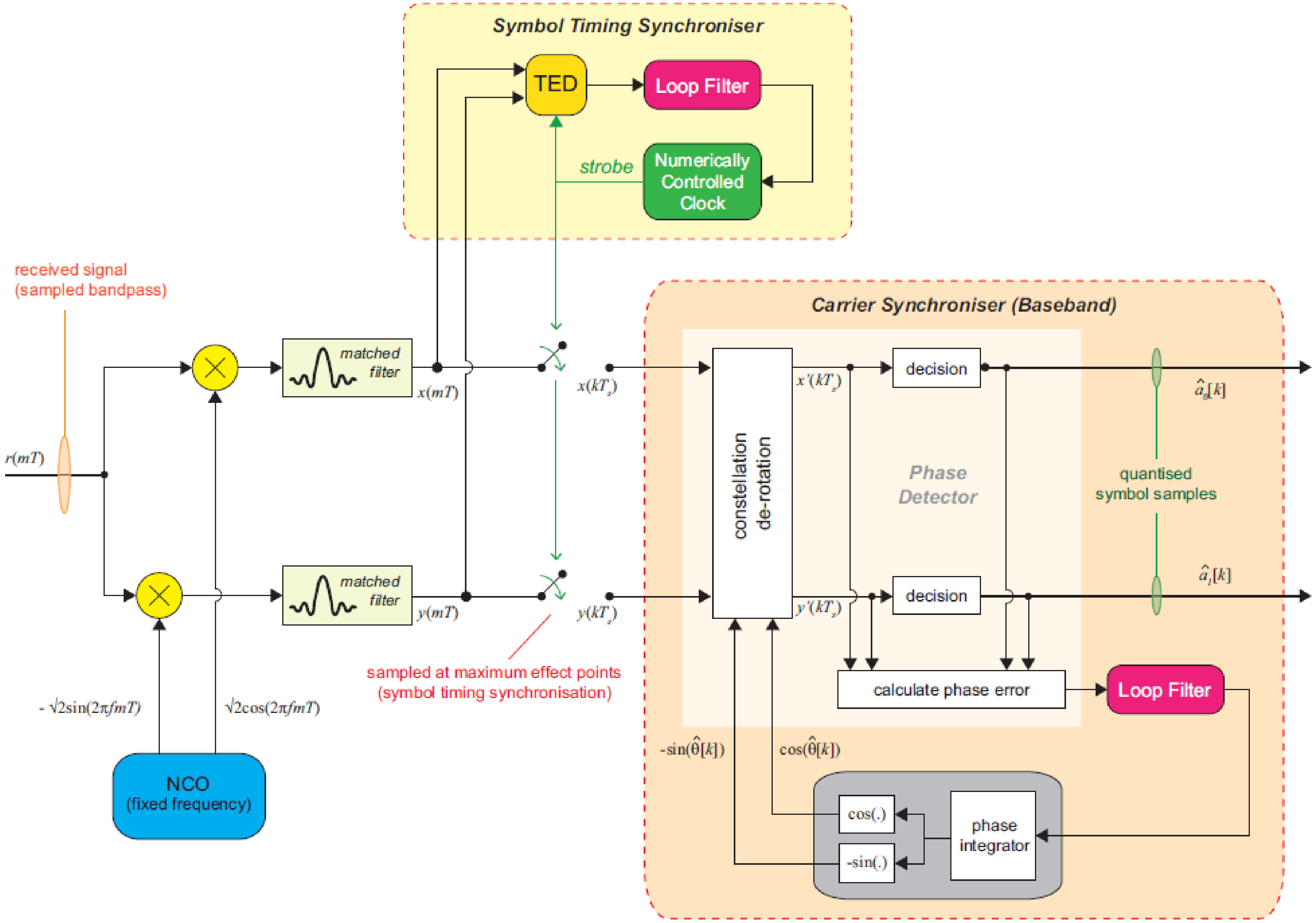


TIMING ADJUSTMENT: OVERSAMPLING

- Oversampling, always small phase error happen and therefore ISI.
- Disadvantage, high rate -> computational loads increase.

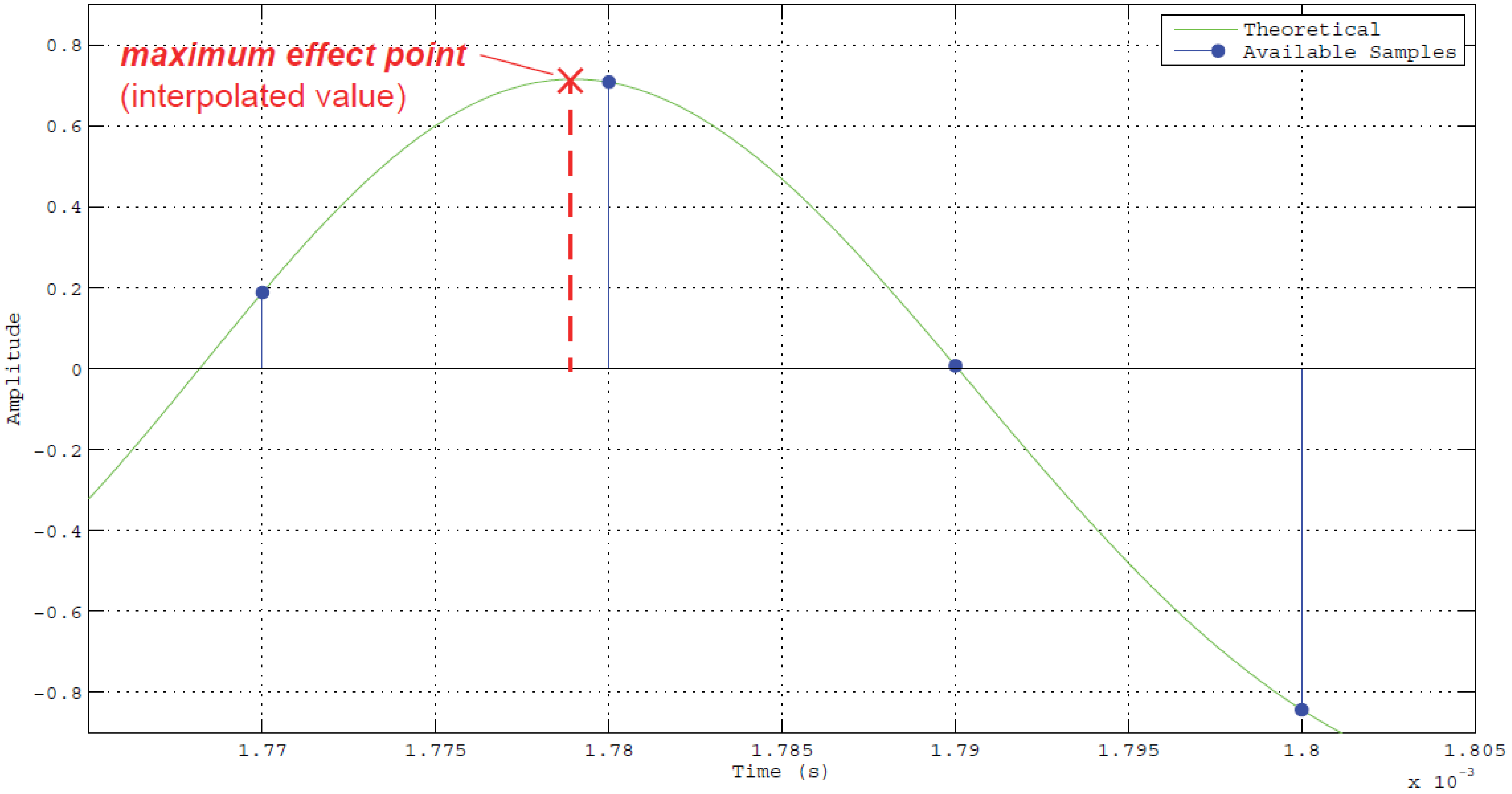


TIMING & FREQUENCY: OVERSAMPLING



TIMING ADJUSTMENT: INTERPOLATION

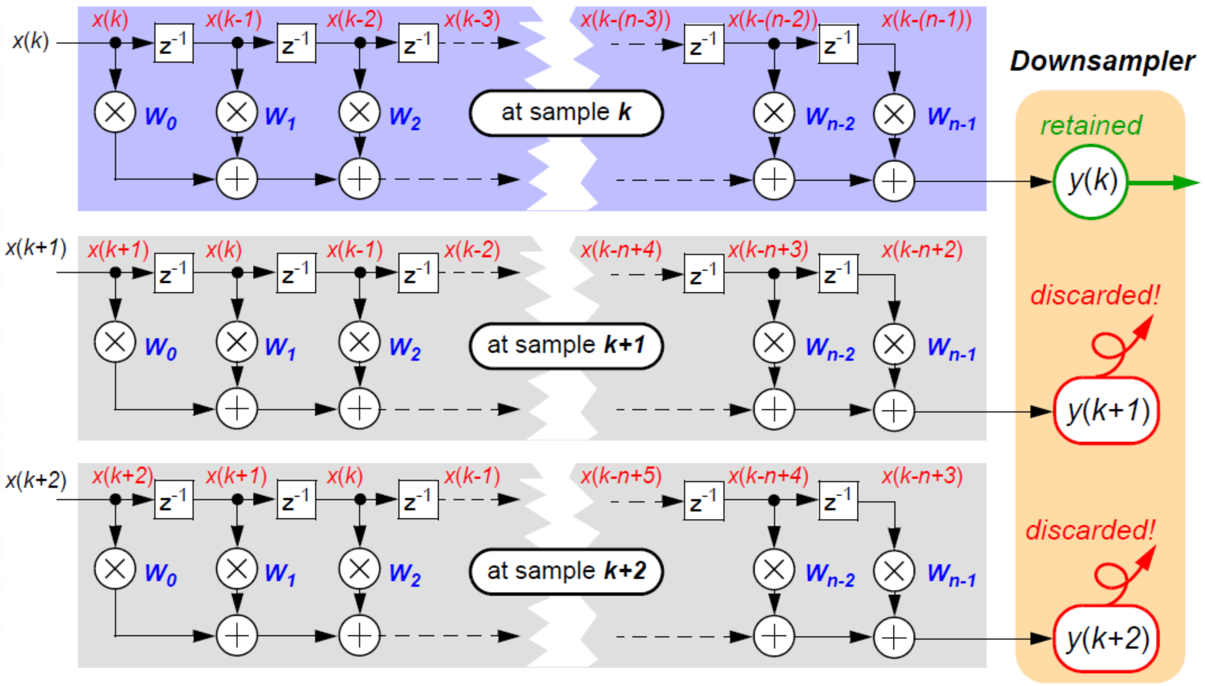
- Interpolation between two samples. Following Nyquist, signal can be recovered since we are sampling twice the max freq. Here interpolated polyphase filters are used.



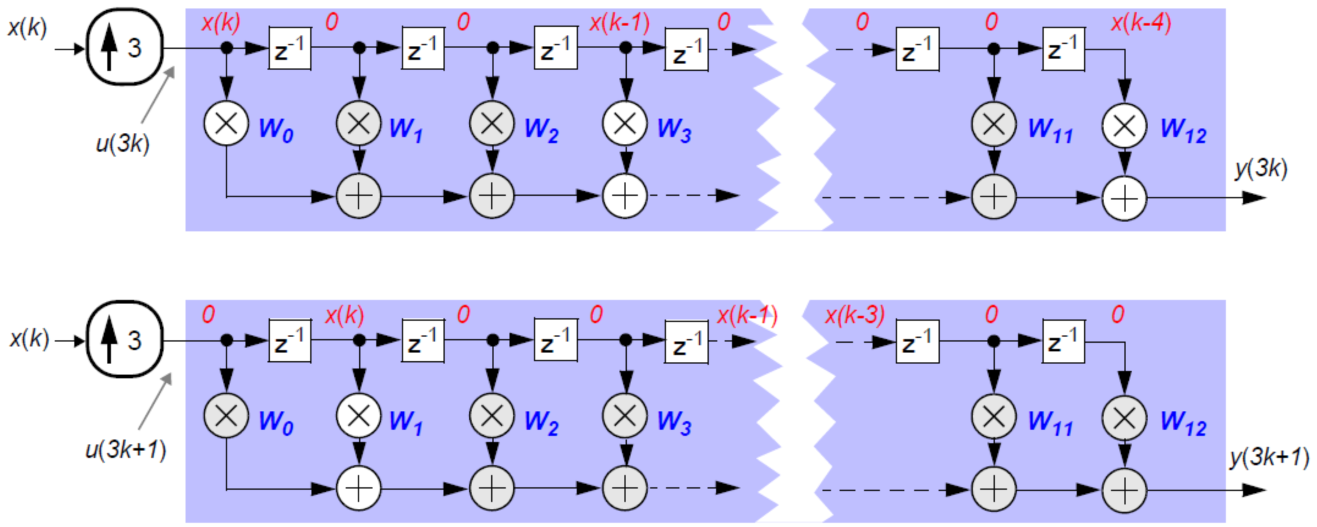
POLYPHASE FILTER

- Efficient implementation of multirate filter. It is used as the interpolation filter on Timing Recovering.

DECIMATION

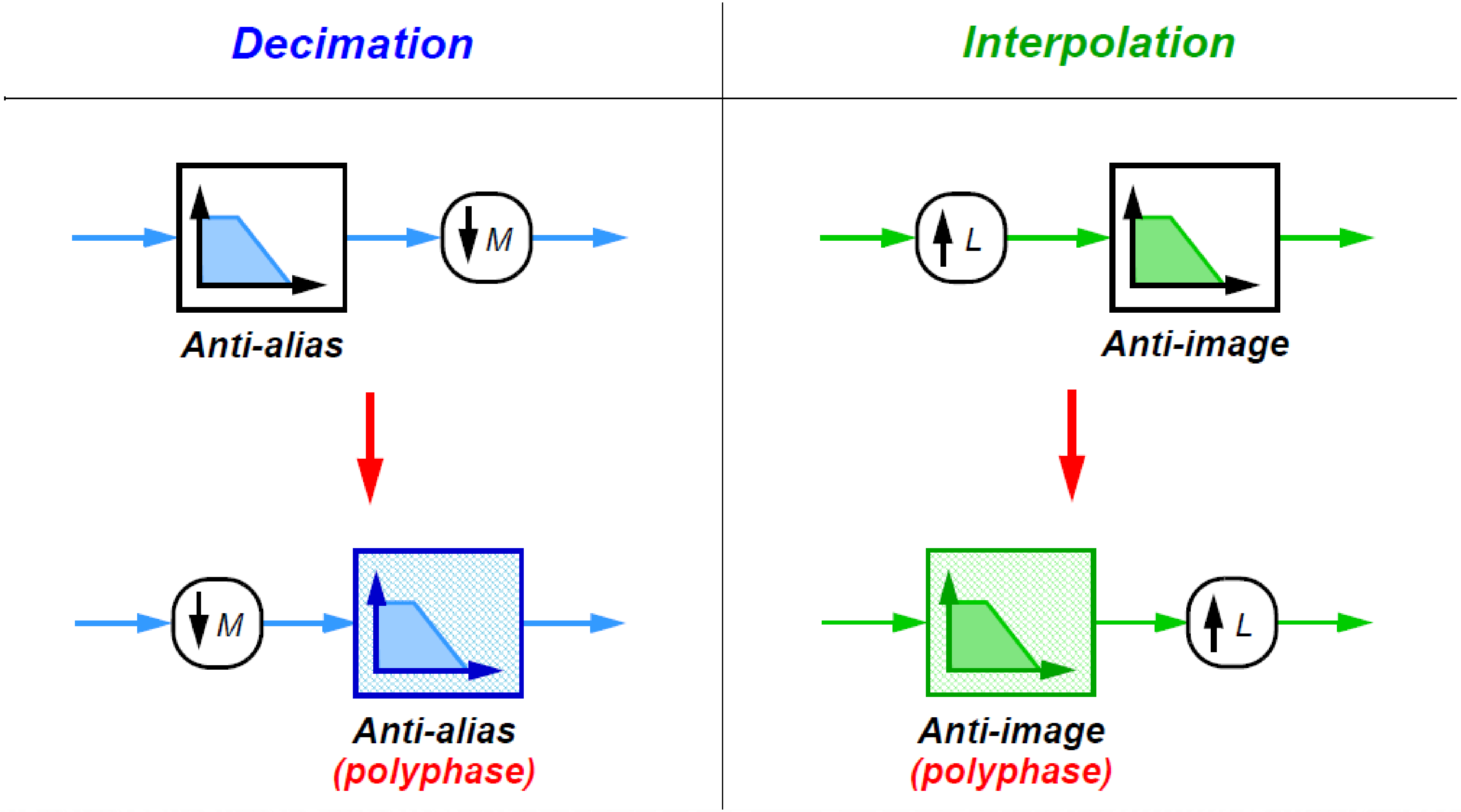


INTERPOLATION



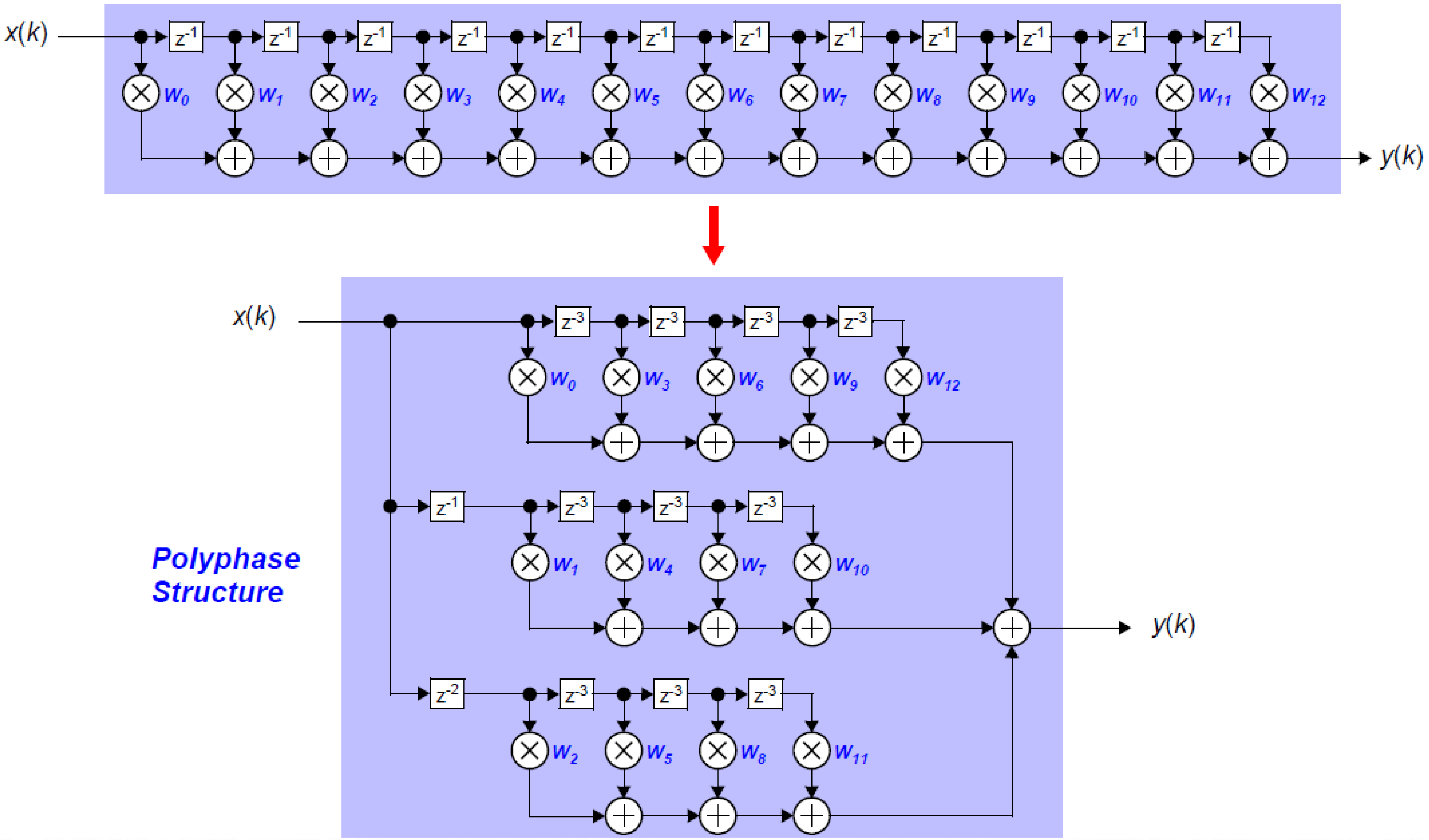
POLYPHASE FILTER

- IMPLEMENTATION OF MULTIRATE FILTERS

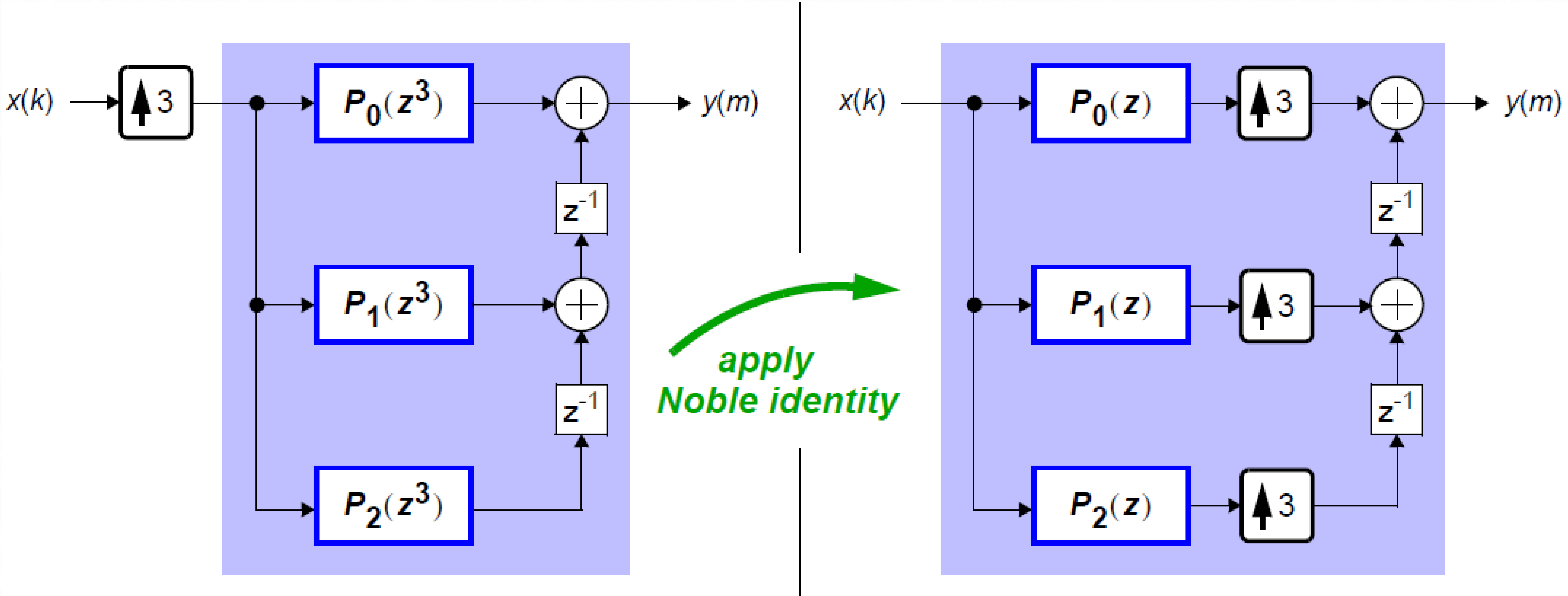


POLYPHASE FILTER

- FIR POLYPHASE DECOMPOSITION



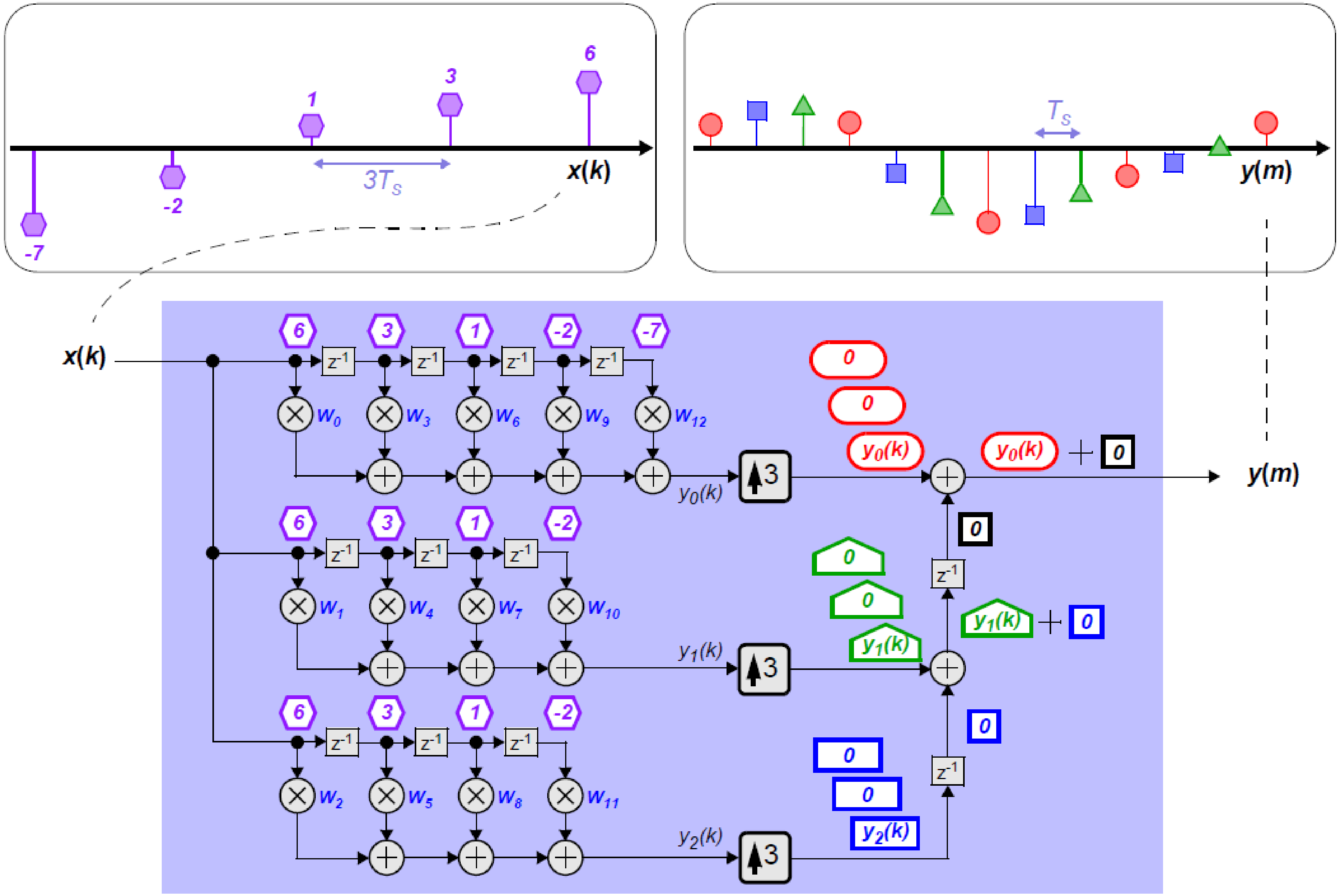
POLYPHASE INTERPOLATOR



***Upsampler** followed by*
Polyphase Filter

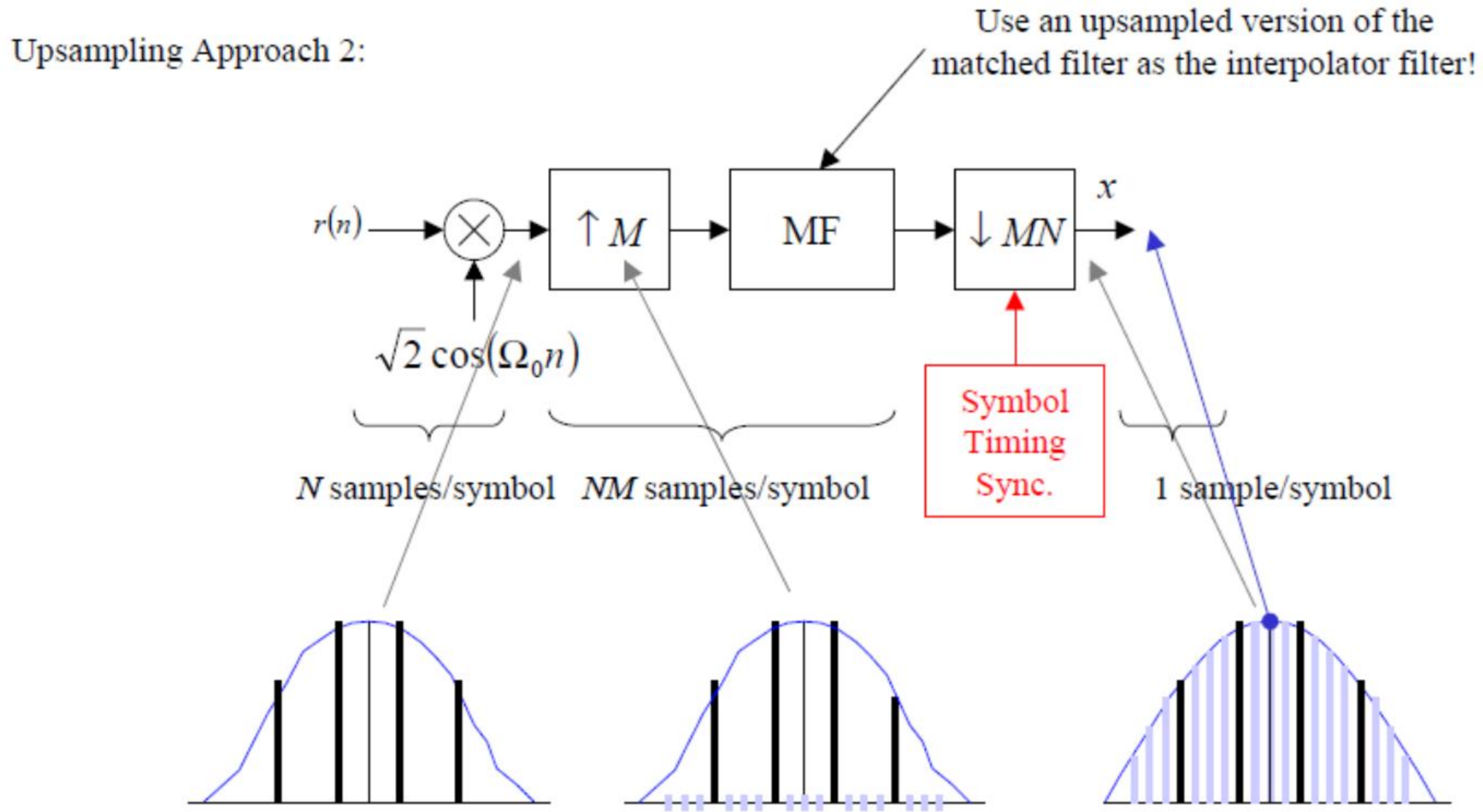
Efficient Polyphase Interpolator

POLYPHASE INTERPOLATOR FOR TIME RECOVERY

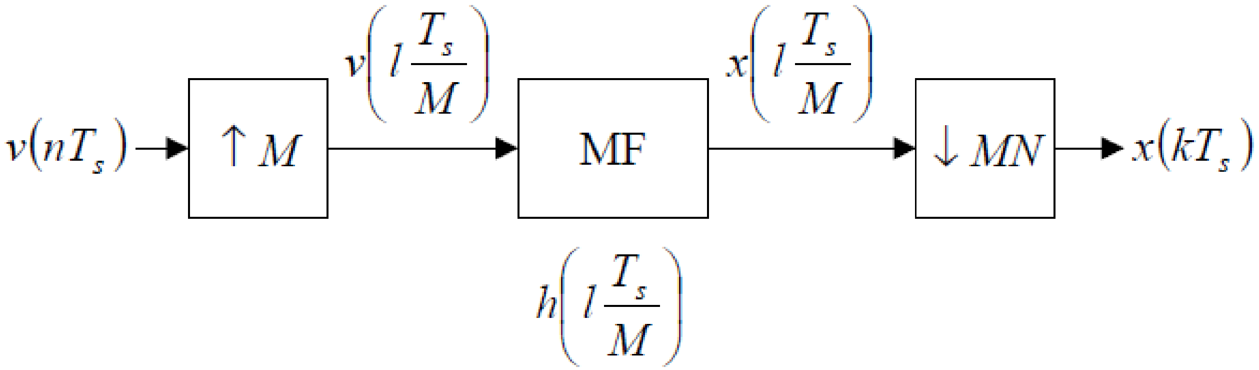


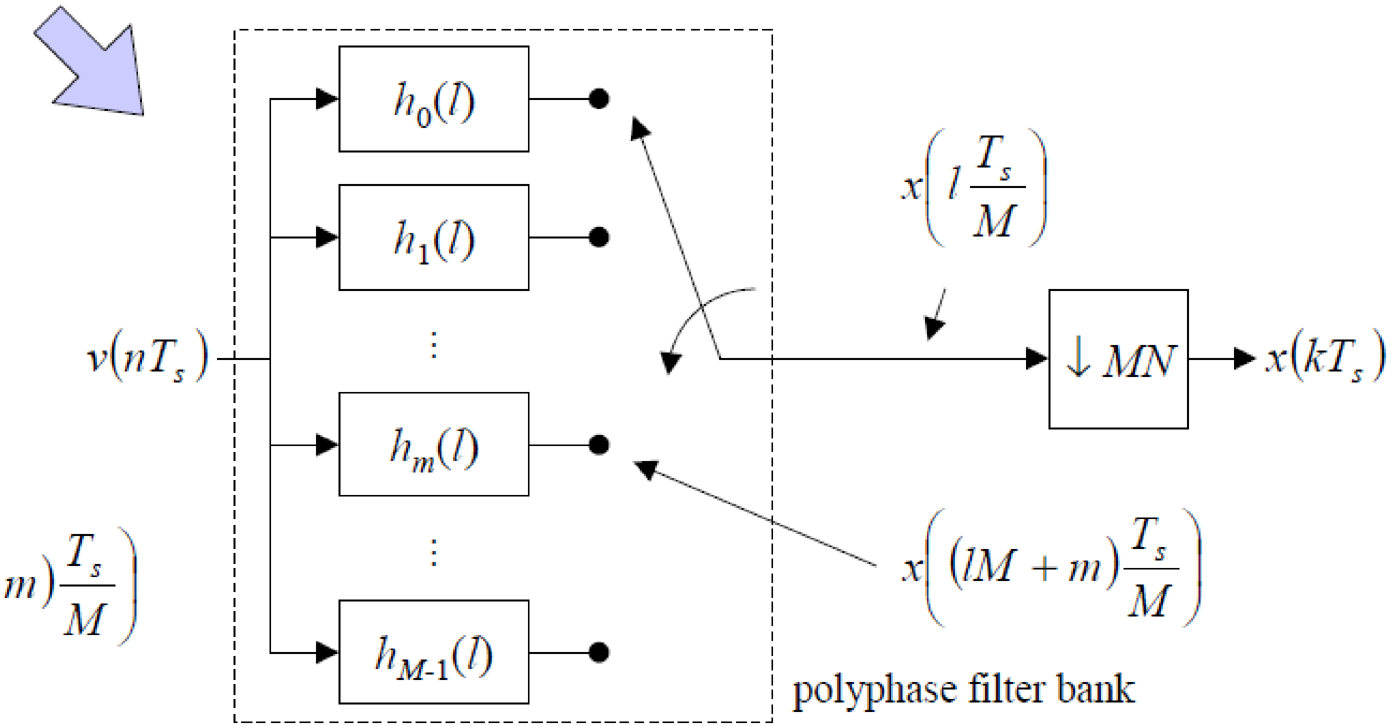
POLYPHASE TIME RECOVERY

- Instead of oversampling, the filter interpolation MF is implemented as a polyphase structure..
- Timing resolution depend on the **number of branches** or phase of Polyphase filter.



POLYPHASE TIME RECOVERY

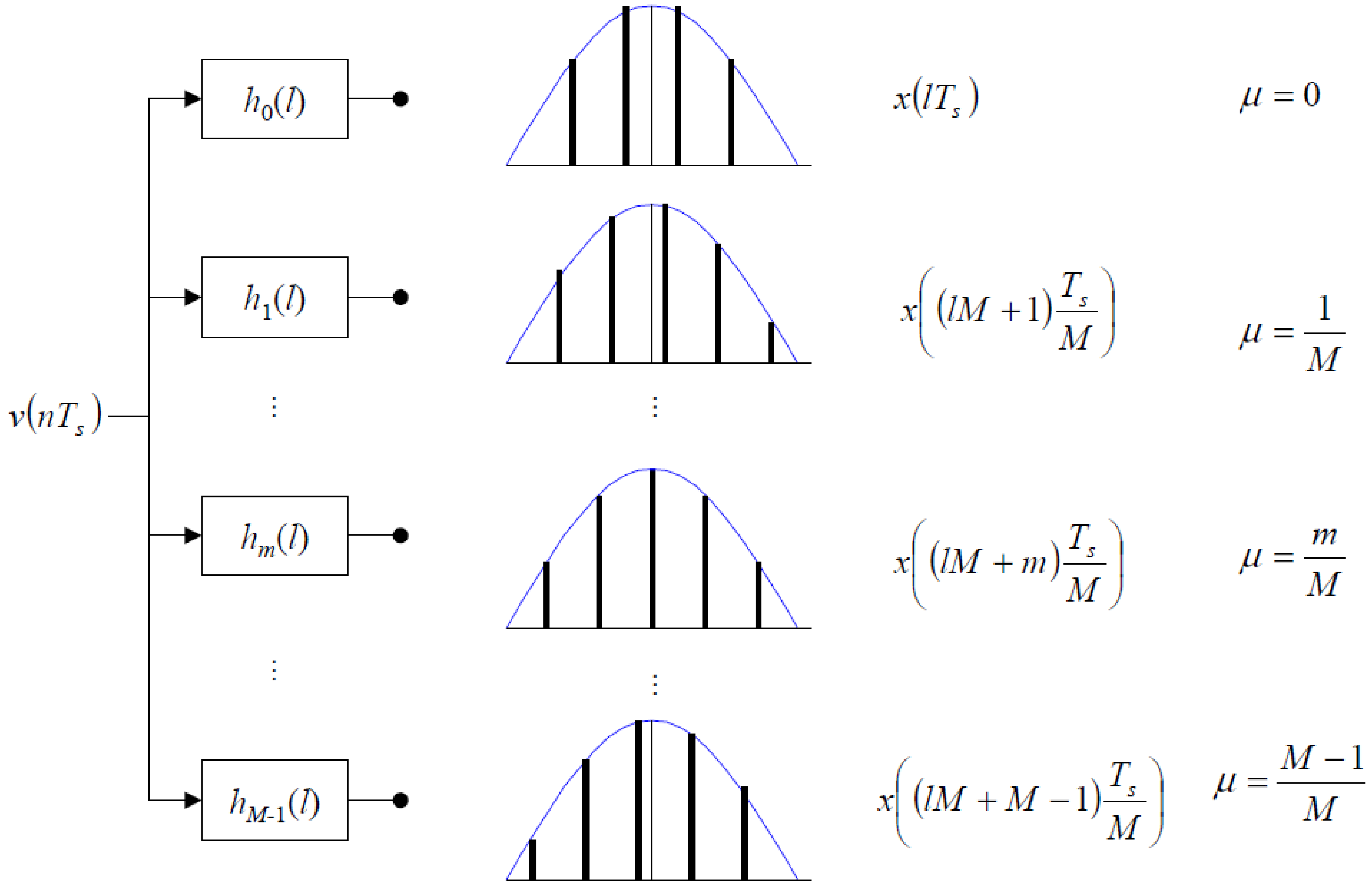




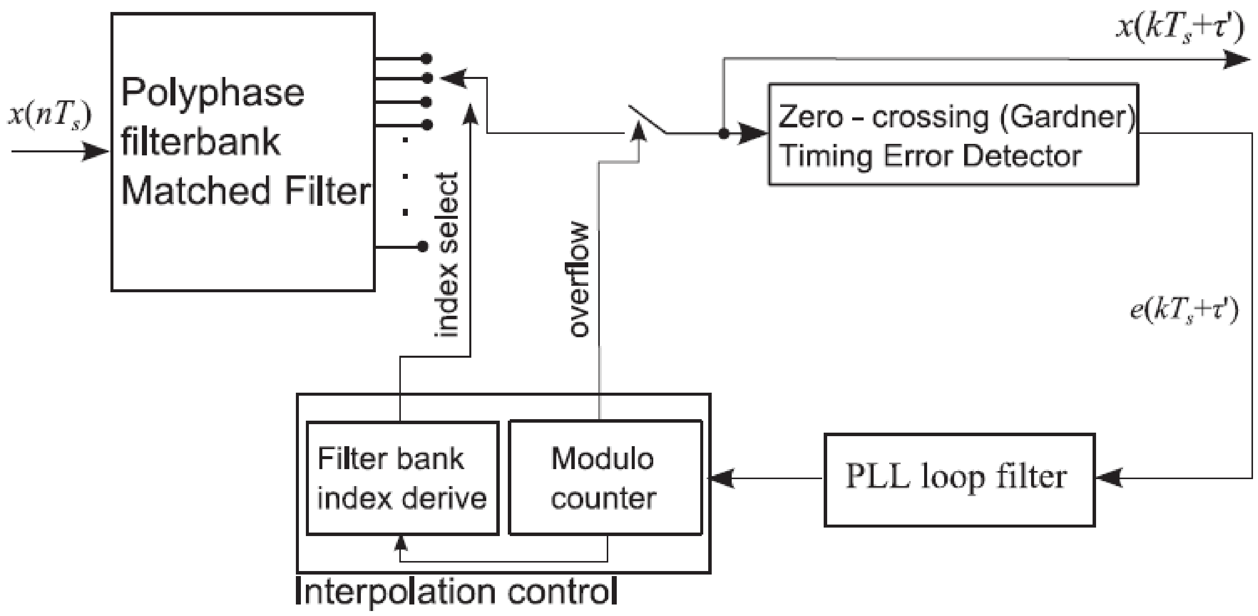
$$h_m(l) = h\left((lM+m)\frac{T_s}{M}\right)$$

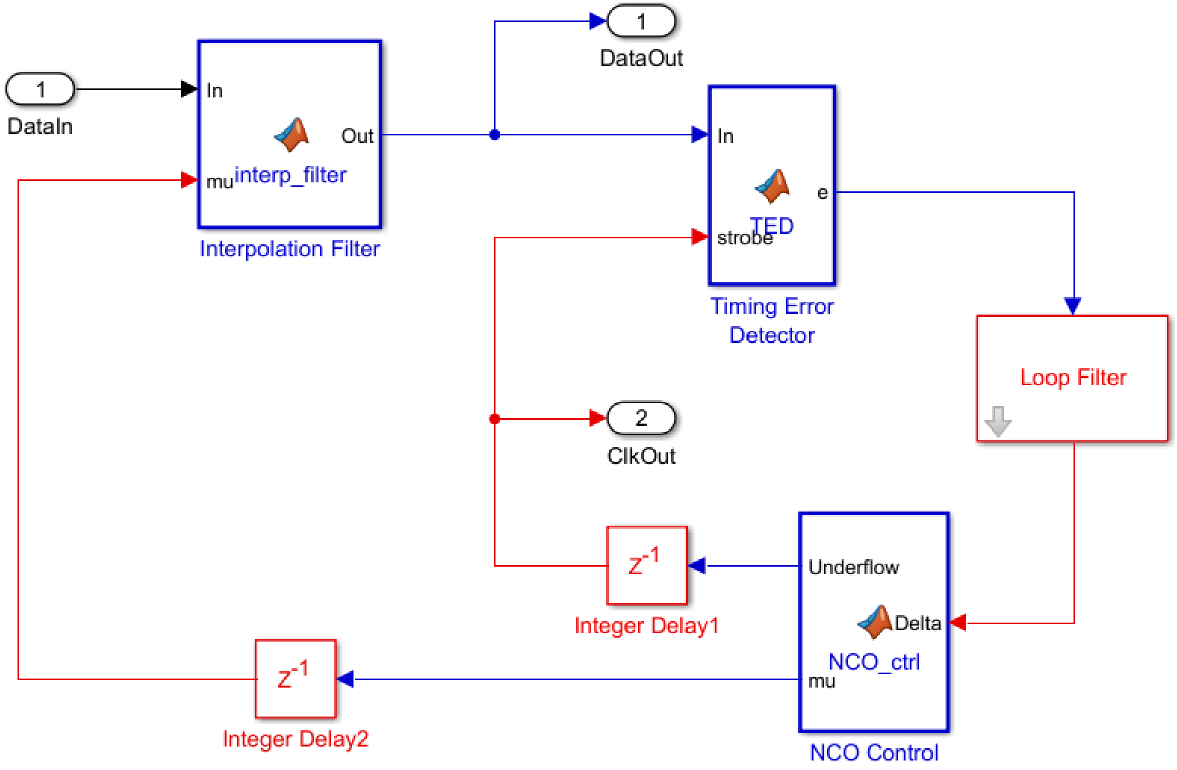
POLYPHASE TIME RECOVERY

- POLIPHASE INTERPOLATORS



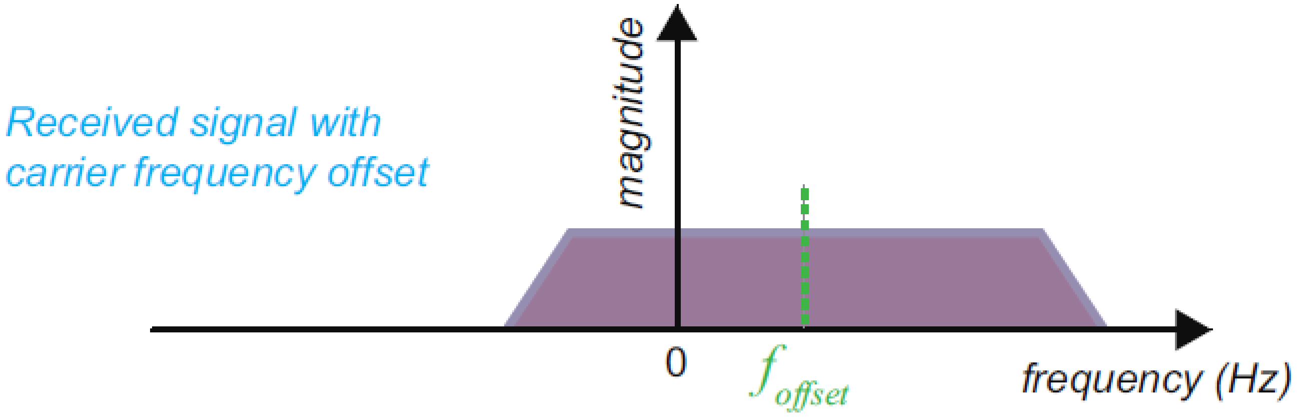
POLYPHASE TIME RECOVERY



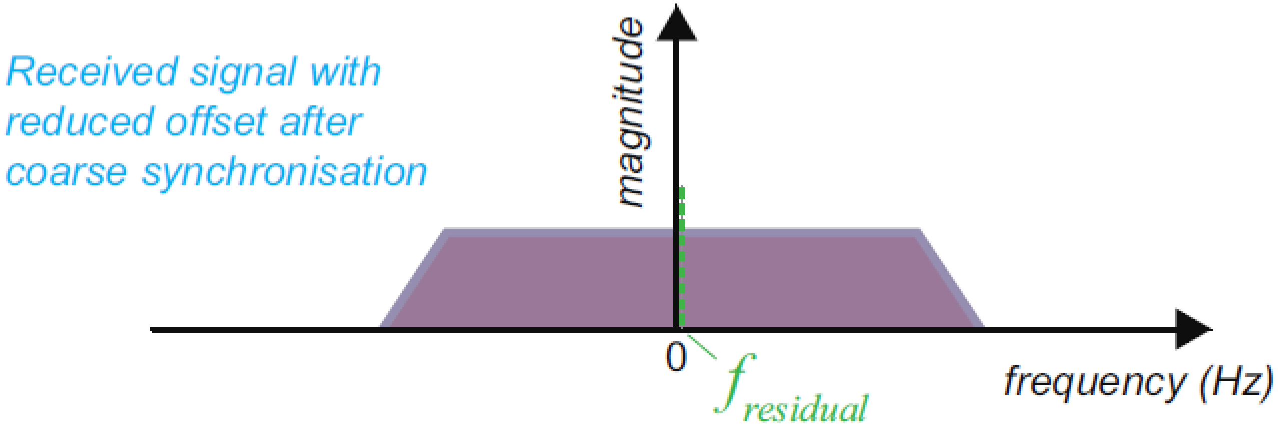


COARSE FREQUENCY: FLL

- If TX/RX LO offset is bigger that the PLL bandwidth capabilities, so carrier sync will fail.
- In that case a coarse frequency estimation has to be implemented.



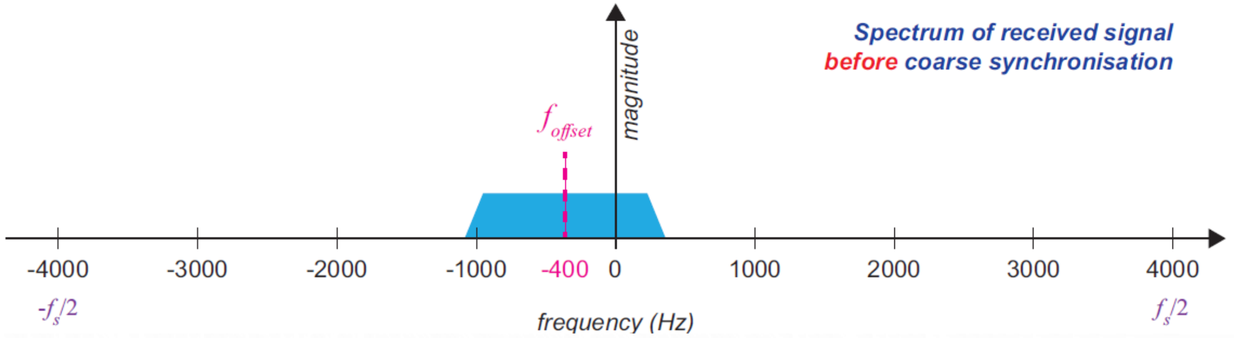
FLL

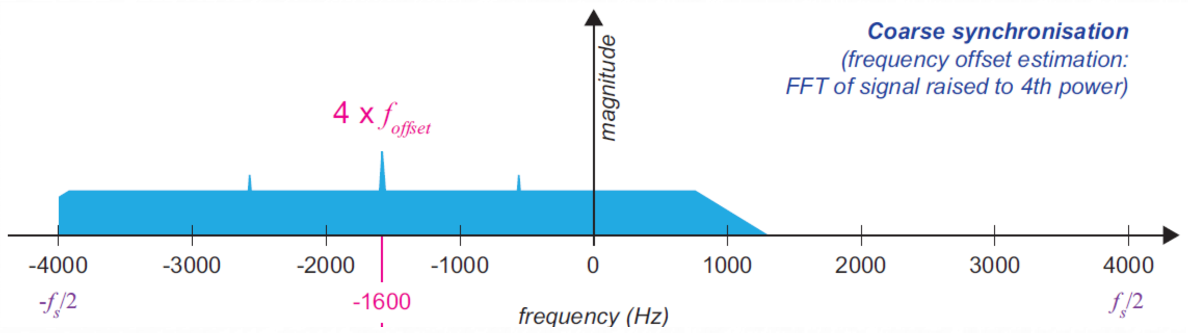


PLL

FLL

- The algorithm use a previous knowledge of the **phase modulation index M.**
- For the case of QPSK, M=4, so the input signal is raised to the power of 4. This produce significant tone power at M time the F_offset.
- The frequency offset estimation algorithm is implemented through **FFT selecting the Bin with the maximum power.**



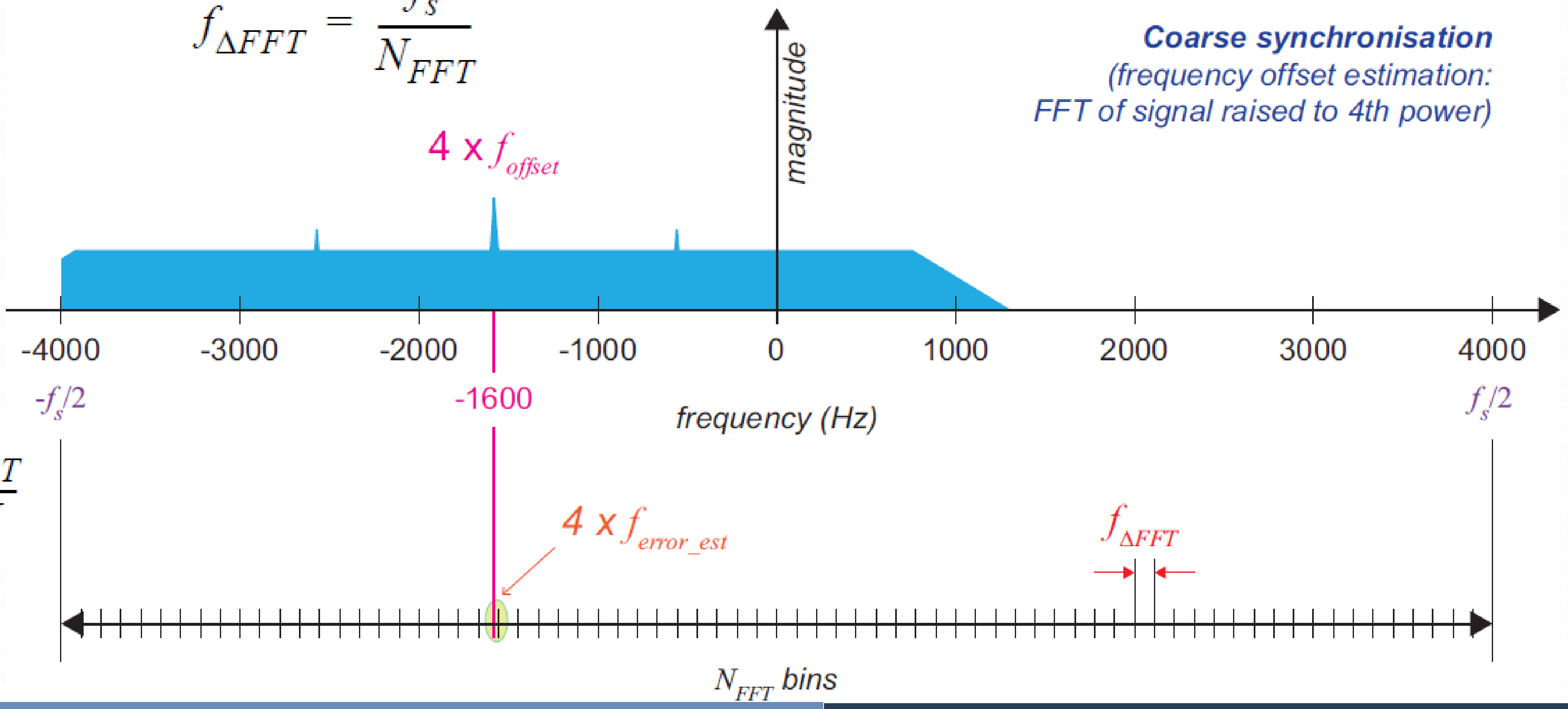


FLL

- The sampling frequency should be selected considering the maximum expected offset. Ex: offset 400hz, M=4, tone 1600Hz, Fs= 3200Hz
- The number of FFT it is also important because the Bin width determine the frequency resolution.

$$f_{\Delta FFT} = \frac{f_s}{N_{FFT}}$$

$$f_{error\_est} = \frac{f_{\Delta FFT}}{2M}$$



FLL IMPLEMENTATION