# SPREAD SPECTRUM COMMUNICATIONS

O. Julien / C. Macabiau

Ecole Nationale de l'Aviation Civile

Toulouse, France

Email: ojulien@recherche.enac.fr; macabiau@recherche.enac.fr

# Introduction

- Communication using spread spectrum techniques consists in transmitting a useful signal over a bandwidth that is significantly greater than the bandwidth strictly necessary to transmit this information.
- Such techniques are used, for instance, for
  - increasing resistance to interference
  - Secure communication
  - Resistance to fading
  - Adapted to multiple access

# Introduction

- There are 2 main ways to increase the useful signal bandwidth:
  - Direct Sequence (DS), which consists in multiplying the useful signal with a high frequency binary pseudo-random signal based on a pseudo random binary sequence (PRBS). This results in an overall signal with a wider spectrum than the useful signal.
  - Frequency Hopping (FH), which consists in changing the carrier frequency of the useful signal over many frequency channels (thus widening the spectral occupation of the signal) according to a pseudo-random sequence
- In both cases, the pseuso-random sequence is key and has to be known by the emitter and the receiver.

# Introduction

- The present course will focus on Direct Sequence Spread Spectrum (DS-SS) as it is the most widely used in telecommunication (WiFi, 3G, military com.) and navigation (GNSS).

- The objective of this course is to introduce
  - the temporal and spectral structure of the signals using spread spectrum techniques
  - the way to process these signals (acquisition, synchronization and data demodulation)
  - the performance of these processing blocks in presence of thermal noise.

# Outline

1. Structure and properties of pseudo random sequences
2. Temporal and spectral structure of a transmitted signal using DS-SS
3. Received DS-SS signal model
4. Correlation operation in a DS-SS receiver
5. Acquisition of a DS-SS signal
6. Carrier phase tracking in a DS-SS signal
7. Code delay tracking in a DS-SS

# 1. Structure and Properties of Pseudo-Random Codes

1. Definition of Pseudo-Random Binary Sequence
2. LFSR Definition
3. M-Sequence from an LFSR
4. Digital Signal Corresponding to an M-Sequence and Properties

# 1.1 Definition of Pseudo-Random Binary Sequence

- A PRBS is a known (deterministic) binary sequence that possesses statistical properties similar to those of a random White sequence in terms of autocorrelation (a Dirac function) and Power Spectral Density (constant over all frequencies).
- In the case of a DS-SS, these sequences are used to spread the useful signal spectral occupation.

# 1.1 Definition of Pseudo-Random Binary Sequence

- In DS-SS, they can also be used to distinguish the useful signal of several emitters of a same system sharing the same frequency band
  - This type of multiple access is referred to Code Division Multiple Access (CDMA), and is based on the assignment to each emitter of a specific PRBS that is as uncorrelated as possible with the other PRBS used by the other emitters.
  - In this case, a PRBS can then be seen as an identifier modulating the useful signal.

# 1.1 Definition of Pseudo-Random Binary Sequence

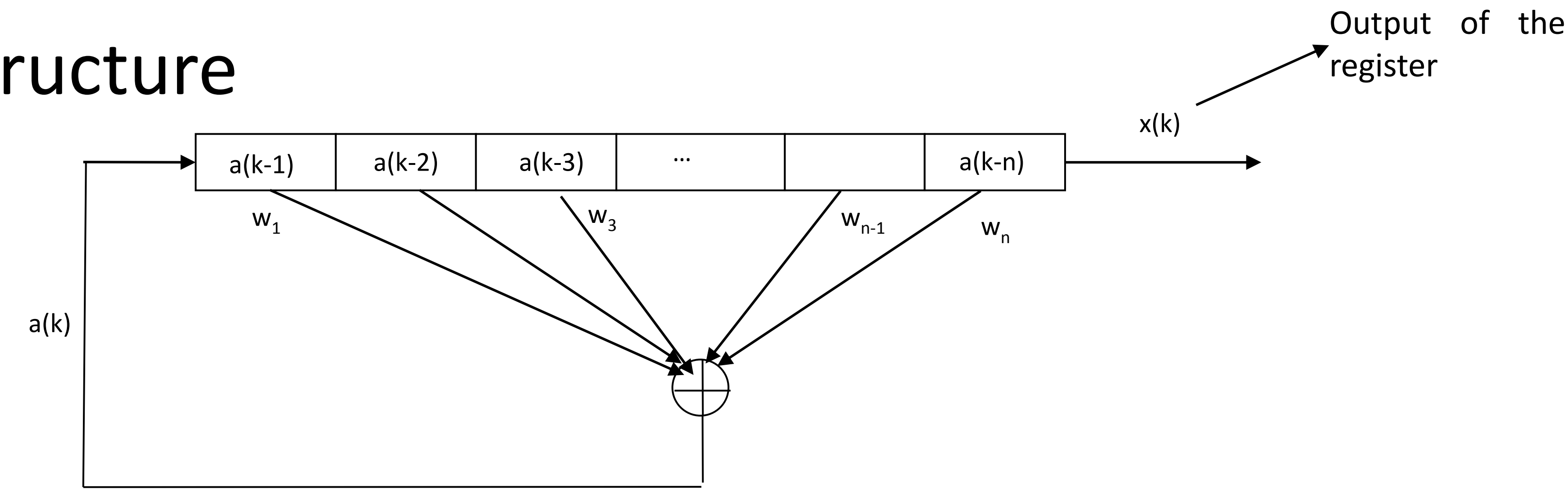
- DS-SS systems use PRBS that are finite and periodic.
- These sequences are also referred to as Pseudo Noise (PN) sequences or **Pseudo-Random Noise (PRN)** sequences.
  - In the following, we will use the term PRN sequences
- The statistical properties of the PRN sequences are key to the functioning of a DS-SS system
- The most used pseudo-random sequences are
  - m-sequences,
  - Gold codes,
  - Kasami codes
  - Barker codes.

# 1.1 Definition of Pseudo-Random Binary Sequence

- There are different ways to generate PRN sequences:
  - Memory codes (codes are stored in emitter/receiver memory)
    - Could be optimally chosen,
    - But require memory at the transmitter and receiver side
  - Codes generated by a Linear Feedback Shift Register (LFSR)
    - Restrict the choice, but can still provide high performance
    - Very efficient to generate the codes « on the fly »
- The next section will focus on codes generated by LFSR

# 1.2 LFSR Definition

- Structure




where :

- $k$ is the current time index $(k \geq 0)$
- $n$ is the number of register bins (register size)
- $w_i$ is a binary weight associated to the i$^{th}$ bin of the register ($w_i = 1$ ou $w_i = 0$)
- $a(k-i)$ is the value stored in the i$^{th}$ bin at time index k
- $a(k) = \oplus_{i=1}^{n} w_i a(k-i), \forall k > 0$
- $x(k)$=$a(k-n)$ is the register output at time index k

# 1.2 LFSR Definition

- The bins of the LFSR are updated at instant (k+1) by shifting to the right the values of all the register bins.
  - The first register bin takes the value of $a(k)$, which is a linear combination of all the register bins.
  - $a(k)$ will be at the register output after n iterations:

$$x(k+n) = a(k)$$

- We can write:

$$x(k) = a(\mathrm{k}-n) = \oplus_{i=1}^{n} w_i a(k-n-i), \forall k > 0$$

- Then:

$$x(k) = \oplus_{i=1}^{n} w_i x(k-i), \forall k > 0$$

- This kind of register is said to be linear as <u>its output is a linear combination of its n previous outputs</u>.

# 1.2 LFSR Definition

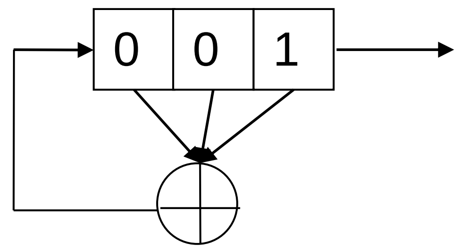
- The binary word representing the binary values in the register bins at time instant k is referred to as <u>register state</u> and can be noted $X(k)$:

$$X(k) = [a(\mathrm{k}-1) \quad a(\mathrm{k}-2) \quad \ldots \quad a(\mathrm{k}-\mathrm{n}+1) \quad a(\mathrm{k}-\mathrm{n})]$$

- Since there is finite number of possible register states ($2^n$), it is easy to understand that the succession of values of $X(k)$ will always be periodic.
  - <u>The generated sequence</u>, constituted of the last bit of the register state, is thus also <u>periodic</u>

# 1.2 LFSR Definition

- **Example 1:**




- Starting sequence: 0 0 1; all weights are set to 1.
- Successive values of register bins: 0 0 1

  1 0 0

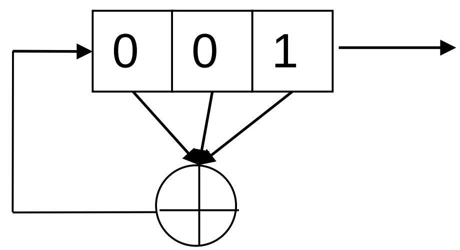
  1 1 0

  0 1 1

  0 0 1 --> same as state 1
- The sequence generated by this LFSR is [1 0 0 1] and then repeats.
  - It has a length of 4.

# 1.2 LFSR Definition

- **Example 2: same LFSR, Different starting sequence**



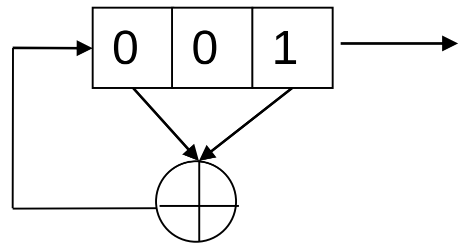

- Starting sequence: 1 0 1
- Successive values of register bins: 1 0 $\boxed{1}$

  0 1 $\boxed{0}$

  1 0 1 --> same as state 1
- The sequence generated by this LFSR is [1 0] and then repeats.
  - It has a length of 2

# 1.2 LFSR Definition

- **Example 3: LFSR with different weights**




- Starting sequence: 0 0 1; central weight set to 0.
- Successive values of register bins:

0 0 1
1 0 0
1 1 0
1 1 1
0 1 1
1 0 1
0 1 0
0 0 1 --> same as state 1

- The sequence generated by this LFSR is [1 0 0 1 1 1 0 1]. It has a length 7

# 1.2 LFSR Definition

- A LFSR can thus generate a periodic sequence which characteristics are defined by:
  - the LFSR size
  - the weights associated to each register bin
  - the initial state of the register

# 1.3 M-Sequence from an LFSR

- The maximum number of different values that $X(k)$ can take is the maximum number of words that can be coded by the register bins.
  - However, we do not consider the state $X(k) = [00 \dots 0]$, as this state can only lead to the generation of itself at the next register epoch → It is referred to as an <u>absorbing state</u>.
  - The maximum number of acceptable register states is thus $2^n$-1.

  → This means that the <u>maximum length of the generated sequence is L=$2^n$-1</u>.
- If the generated sequence length is L=$2^n$-1, then the sequence is said to be of <u>maximum length</u>, also referred to as <u>m-sequence</u>.
  - Note that an m-sequence always has a <u>length that is an odd number </u>($2^n$-1)
  - Since the state [000...0] is the only one excluded, this means that there is <u>always one more '1' than '0' in an m-sequence</u>.

# 1.3 M-Sequence from an LFSR

- It is possible to show that the number of different m-sequences that can be generated by a LFSR of size n is:

$$N_m = \frac{\Phi(2^n - 1)}{n}$$

where

- $\Phi(\mathrm{n}) = \begin{cases} 1 & \text{if } \mathrm{n} = 1 \\ \prod_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{p}_\mathrm{i}^{\alpha_\mathrm{i}-1}(\mathrm{p}_\mathrm{i} - 1) & \text{if } \mathrm{n} > 1 \text{ and not a prime number} \\ \mathrm{p} - 1 & \text{if } \mathrm{n} > 1 \text{ and is a prime number} \end{cases}$
- $\mathrm{p}_\mathrm{i}$ and $\alpha_\mathrm{i}$ are the prime numbers and the associated powers that compose n: $\mathrm{n} = \prod_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{p}_\mathrm{i}^{\alpha_\mathrm{i}}$

- Note that $N_m$ is always an even number

# 1.3 M-Sequence from an LFSR

- Examples:
  - For a register of size n=3:

$$N_m =$$

  - For a register of size n=4:

$$N_m =$$

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

- We can associate any discrete binary sequence $x$ composed of the elements {1,0} and governed by the modulo-2 addition operator to an equivalent discrete digital signal $c$ composed of the elements {-1,1} and governed by the traditional multiplication
  - This is done by replacing every '1' of $x$ by a '-1' and every '0' by a '1'.
  - This is possible since there is an isomorphism between the fields ({1,0},⊕) and ({-1,1},x).

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

- Although it is important to differentiate between the sequence (composed of elements {0,1}) and its associated binary antipodal digital signal (composed of elements {-1,1}), the name 'sequence' will also be used to refer to the binary digital signal, for simplicity.

- In order to verify the 'random' properties of an m-sequence (to be considered as a PRN sequence), its autocorrelation and Power Spectral Density (PSD) must now be calculated.

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

## Autocorrelation of an m-sequence (1/3)

- The autocorrelation of the digital signal corresponding to the periodic binary sequence is defined as:

$$K(m) = \frac{1}{N} \sum_{i=0}^{N-1} c_i c_{i+m}$$

where $N$ is the period of the digital m-sequence.

- Note: an autocorrelation function can be interpreted as a measure of the resemblance of a function with a shifted version of itself

- Since the digital signal is periodic, then the autocorrelation function is also periodic with the same period as the digital signal.

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

## Autocorrelation of an m-sequence (2/3)

- In the case of an m-sequence, L= $2^n - 1$ :
- For $m$ =0: Since the digital signal takes only the values 1 or -1:

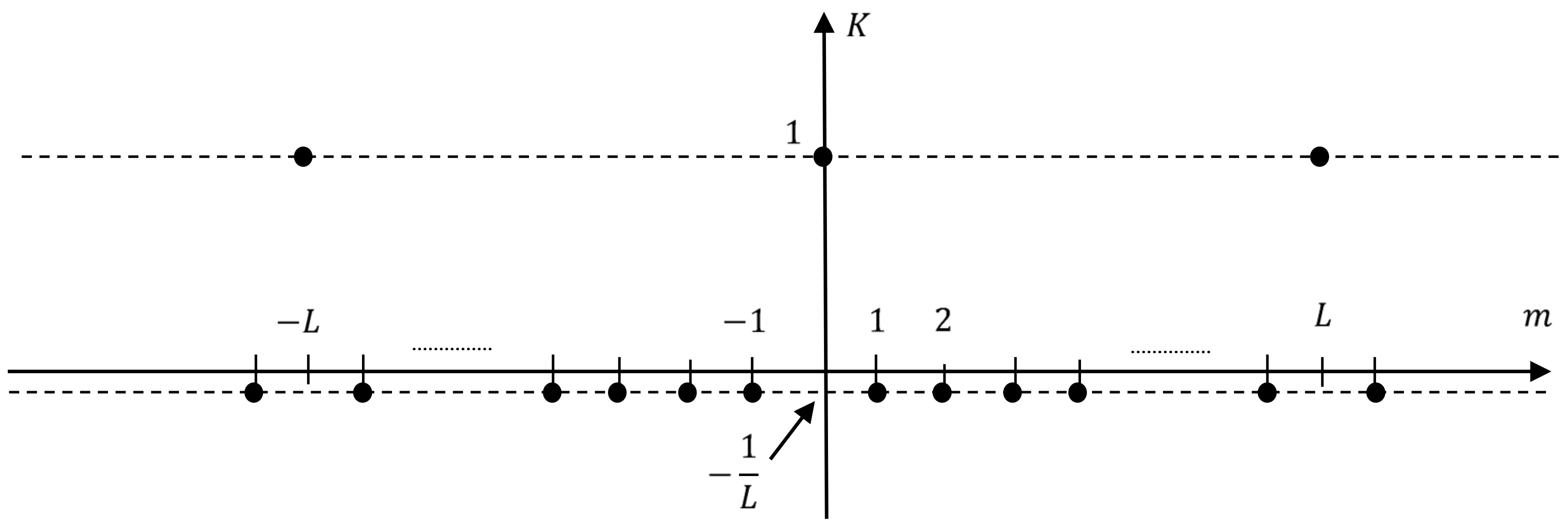
$$K(0) = \frac{1}{L}\sum_{i=0}^{L-1} c_i^2 = 1$$

- For $1 \leq m \leq L-1$: It can be shown that:

$$K_c(m) = \frac{1}{L}\sum_{i=0}^{L-1} c_i c_{i+m} = \frac{1}{L}\sum_{i=0}^{L-1} c_{i+m+p} = -\frac{1}{L}$$

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

## Autocorrelation of an m-sequence (3/3)




- If the length of the m-sequence increases, <u>the autocorrelation of the m-sequence will look like a Dirac</u> (impulse) function
  - Such type of autocorrelation function is characteristic of white noise.
  - <u>Interpretation: A long m-sequence does not look like any shifted version of itself</u> (except for shifts equal to a multiple of the period).

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

## PSD of an m-sequence (1/3)

- The PSD of a digital signal can be obtained by applying a Discrete Fourier Transform (DFT) to its autocorrelation function.
- In the case of a periodic signal based on an m-sequence of length $L$, this means:

$$S(k) = DFT(K)(k) = \sum_{i=0}^{L-1} K(i) e^{-2j\pi\frac{ki}{L}}$$

- Due to the inherent behavior of the DFT, <u>the PSD of a periodic signal of period $L$ is periodic with a period $L$</u>.

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

## PSD of an m-sequence (2/3)

- As seen before, the autocorrelation function of the m-sequence only takes 2 values. Then:
  - for $0 \leq k \leq N-1$:

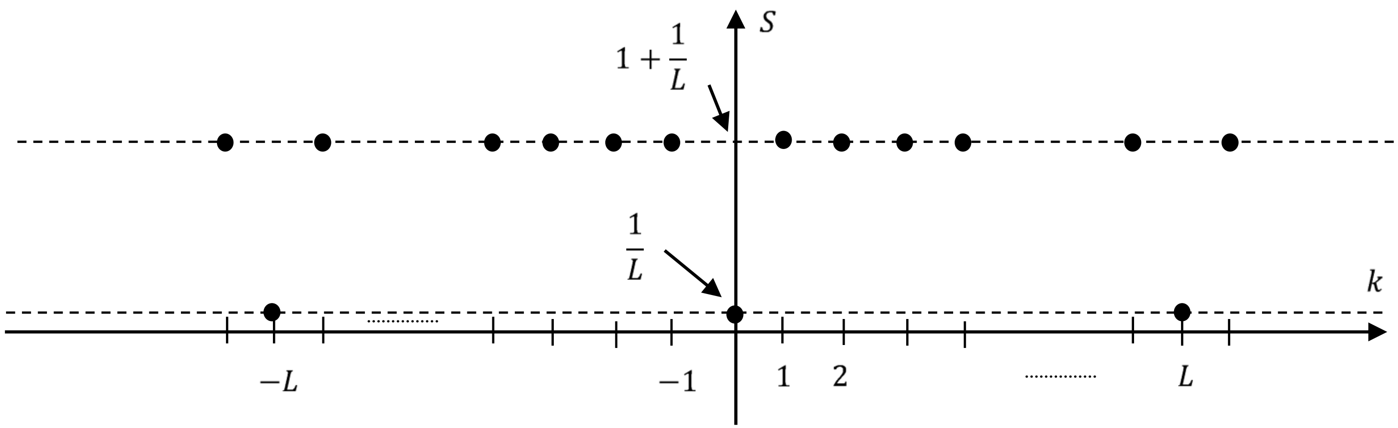
$$S(k) = 1 - \frac{1}{N}\sum_{i=1}^{N-1} e^{-2j\pi\frac{ki}{N}} = 1 + \frac{1}{N} - \frac{1}{N}\sum_{i=0}^{N-1} e^{-2j\pi\frac{ki}{N}} = 1 + \frac{1}{N}\big(1 - N\delta(k)\big)$$

  - Thus:

$$S(k) = \begin{cases} \dfrac{1}{N} & if\ k = 0 \\ 1 + \dfrac{1}{N} & 0 \leq k \leq N-1 \end{cases}$$

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

## PSD of an m-sequence (3/3)




The PSD of the digital signal associated with an m-sequence is, as expected, very similar to the PSD of a white noise (constant over the frequencies).

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

- An m-sequence can thus be considered as a PRN sequence since the digital signal associated to this m-sequence has properties similar to those of White noise
  - This is especially true when the m-sequence used is long.

- Notation: In the continuation, the discrete periodic signal associated to a PRN sequence will be referred to as discrete PRN signal.

# 1.4 Digital Signal Corresponding to an M-Sequence and Properties

- When DS-SS is used in CDMA, the PRN sequences used by each emitter also have to have excellent cross-correlation properties
  - A family of PRN sequences then has to be selected
  - M-sequences are then less adapted to find such PRN sequences
  - Gold codes (used by GPS) are, for instance, a type of code families generated based on the summation of ‘preferred’ m-sequences.

# 2. Temporal and Spectral Structure of a Transmitted Signal using DS-SS

# 2.1 Introduction

- Now that the discrete PRN signal has been presented, it is necessary to understand the process to build the continuous DS-SS signal that will be transmitted. This is based on 2 main operations:
  - The physical shaping of the PRN sequence
  - The modulation of the useful information by the PRN signal
- The next sections provide
  - the method used to obtain the physical shaping of the discrete PRN signal and the new properties of the resulting continuous signal, which will be referred to as the PRN signal.
  - The way the data is modulated by the PRN signal and transmitted to the user

# 2.1 Introduction

- Notations:
  - The bits of a PRN sequence will be referred to as chips (a piece of the PRN sequence)
    - This allows to distinguish PRN chips from useful data bits.
  - When generating the PRN sequence, the duration between two PRN bits will be referred to as PRN bit duration or chip duration
    - It will be noted as $T_c$
    - The inverse of $T_c$ is referred to as chipping rate and is noted $f_c$

# 2.2 Pulse Shaping of a PRN Sequence

- To use this signal in a communication system, it is important to transmit the PRN sequence via an analogue signal, or in other words, to represent these discrete binary values using a continuous shape, also referred to as a <u>waveform</u>.
- The action of providing a physical support or a waveform to the PRN chips is referred to as <u>pulse shaping</u>.

- To facilitate the upcoming mathematical developments, the discrete PRN signal can be expressed in the time-continuous domain by :

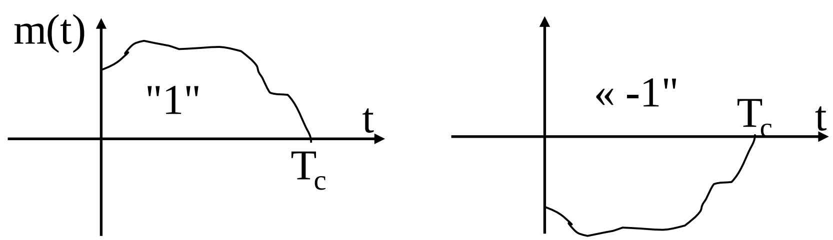
$$c(t) = \sum_{k=-\infty}^{+\infty} c(k)\delta(t - kT_c)$$

where

- $\delta$ is the Dirac function
- $c(k) = c_k$

# 2.2 Pulse Shaping of a PRN Sequence

- Let us denote $m_{-1}$ and $m_1$ the continuous functions (waveforms) representing a PRN bit with value "-1" and "+1", respectively.
  - There are many different options to select $m_{-1}$ and $m_1$.
  - Only <u>antipodal functions</u> ($m_1 = -m_{-1} = m$) are investigated.

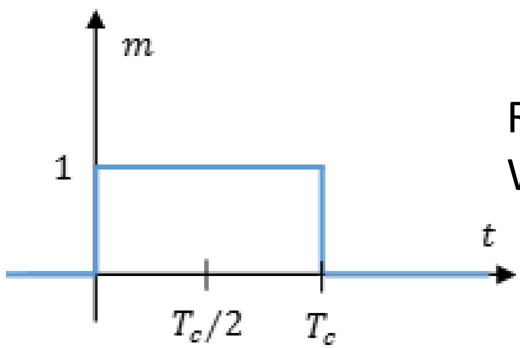



$m_1(t) = m(t)$,
$m_{-1}(t) = -m(t)$

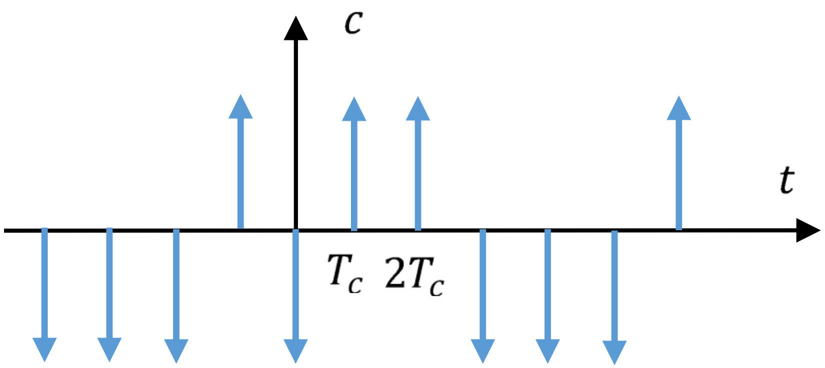
  - Examples of well-known waveforms are:
    - the rectangular function (time-limited)
    - the raised cosine shape (not time-limited)
    - the square root raised cosine (not time-limited)

# 2.2 Pulse Shaping of a PRN Sequence

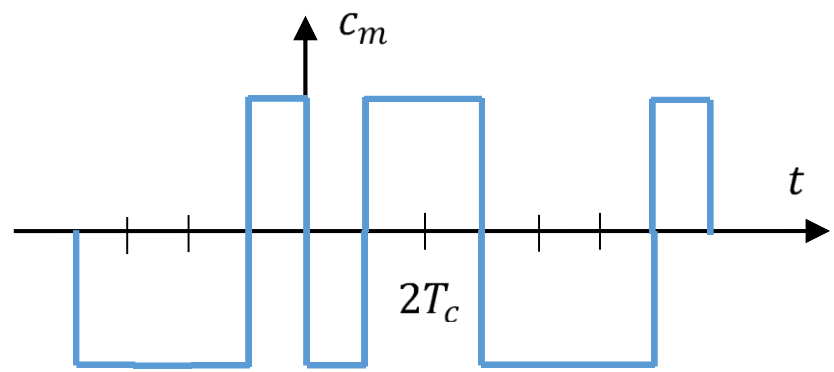
- Example: Case of a discrete PRN signal based on an m-sequence and shaped by a rectangular waveform




Rectangular Shaping Waveform




Discrete PRN Signal

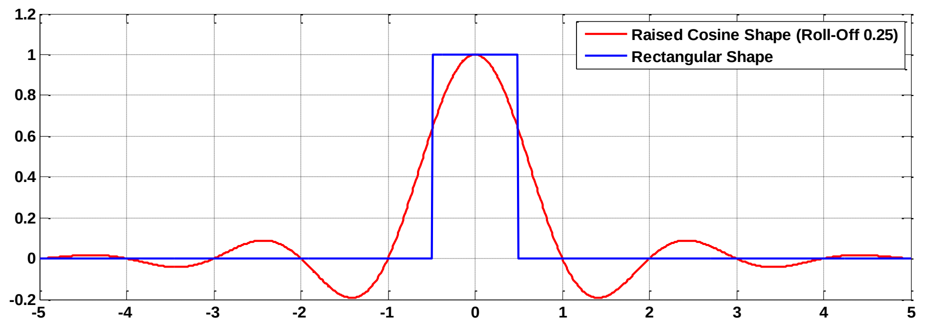



Continuous PRN Signal After Shaping

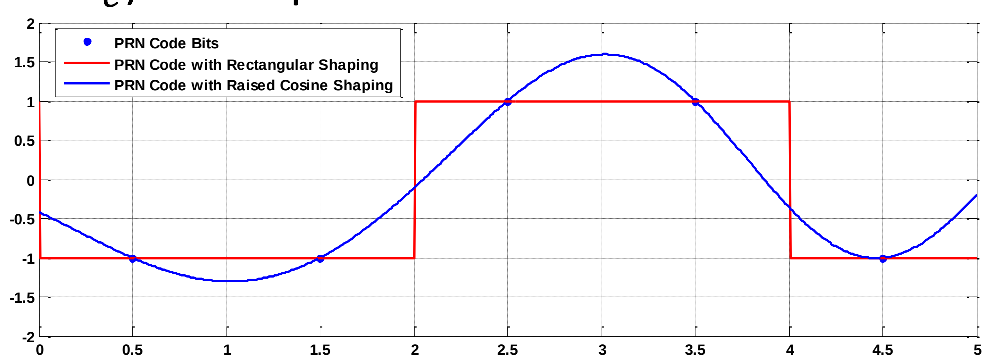
# 2.2 Pulse Shaping of a PRN Sequence

- Example: Case of a discrete PRN signal based on an m-sequence and shaped by a raised cosine
- The shape of a Raised Cosine with a roll-off $\beta$ is:

$$m(t) = \operatorname{sinc}\left(\pi \frac{t}{T_c}\right) \frac{\cos\left(\pi \frac{\beta t}{T_c}\right)}{1 - 4\left(\frac{\beta t}{T_c}\right)^2}$$




  - The more $\beta$ tends towards 0, the more the amplitude of the secondary lobes (after $T_c$) are important.




Example of a PRN signal after shaping (five first PRN chips)

# 2.2 Pulse Shaping of a PRN Sequence

- The pulse shaping process of the discrete PRN signal $c$ results in an continuous signal $c_m$, referred to as the <u>PRN signal</u>, that can be modeled as:

$$c_m(t) = \sum_{k=-\infty}^{+\infty} c(k)m(t - kT_c)$$

where

- $T_c$ is PRN bit duration
- $c$ represents the sequence of the discrete PRN signal bits

- This can be written as:

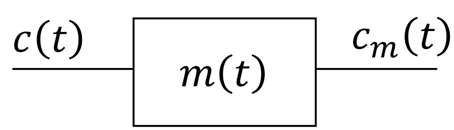
$$c_m(t) = \left(\sum_{k=-\infty}^{+\infty} c(k)\delta(t - kT_c)\right) * m(t - kT_c)$$

# 2.2 Pulse Shaping of a PRN Sequence

- The shaping process thus consists in the convolution between the waveform $m$ and the discrete PRN signal:

$$c_m(t) = (c * m)(t)$$

  - The process of shaping the PRN sequence bits is thus equivalent to filtering the discrete PRN
  - The filter impulse response is the waveform $m$.




- <u>The statistical properties of the PRN signal will thus be influenced by the choice of the shaping waveform</u>

# 2.2 Pulse Shaping of a PRN Sequence

- Due to the “random-like” nature of the PRN sequence, the literature usually models the PRN signal in two different ways:
  - as a periodic and deterministic signal, <u>which is the truth</u>
  - as a random binary signal (the PRN sequence is assumed infinite and random)
    - This represents a common simplification generally acceptable as long as the PRN sequence is long

- The properties of the PRN signal will be investigated considering these 2 models

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (1/14)

- In this case, the PRN sequence is deterministic, finite (length $L$) and periodic.
- It can be modeled as:

$$c_m(t) = \left[\left[\left(\sum_{k=0}^{L-1} c(k)\delta(t - kT_c)\right) * \left(\sum_{j=-\infty}^{+\infty} \delta(t - jT_R)\right)\right] * m(t)\right]$$

where

- $T_R = LT_c$ is the <u>PRN code repetition period</u>
- $F_R = \frac{1}{T_R}$ is the <u>PRN code repetition rate</u>

Periodic discrete PRN sequence

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (2/14)

### PRN Signal Autocorrelation (1/7)

- Reminder on autocorrelations of deterministic signals:
  - The autocorrelation function of a deterministic real signal $m$ is:

$$K_m(\tau) = \int_{-\infty}^{+\infty} m(t)m(t-\tau)dt$$

  - The autocorrelation function of a periodic and deterministic real signal $c$ that has a period $T_R$ can be written as:

$$K_c(\tau) = \frac{1}{T_R}\int_{0}^{T_R} c(t)c(t-\tau)dt$$

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (3/14)

### PRN Signal Autocorrelation (2/7)

- It can be shown that the autocorrelation of the PRN signal is equal to:

$$K_{c_m}(\tau) = (K_c * K_m)(\tau)$$

where

- $K_m$ is the autocorrelation function of the shaping waveform $m$
- $K_c$ is the autocorrelation function of the discrete PRN signal expressed in the continuous domain

→ As expected, the autocorrelation of the PRN signal depends upon

- the autocorrelation of the “continuous time discrete PRN signal” and
- the autocorrelation of the selected shaping waveform

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (4/14)

### PRN Signal Autocorrelation (3/7)

- Because $c$ represents the discrete PRN signal in the continuous domain, its autocorrelation function $K_{\mathrm{c}}$ is thus related to that of the discrete PRN sequence $(K)$ expressed in section 1.4 (for an m-sequence)

- Taking into account the differences between the definitions of the autocorrelation function for a discrete and continuous function, the relation between $K_{\mathrm{c}}$ and $K$ is:

$$K_c(t) = \frac{1}{T_c}\left(\sum_{i=0}^{L-1} K(i)\delta(\tau - iT_c)\right) * \left(\sum_{j=-\infty}^{+\infty} \delta(\tau - jT_R)\right)$$

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (5/14)

### PRN Signal Autocorrelation (4/7)

- Finally, the closed-form expression of the PRN signal autocorrelation is:

$$K_{c_m}(\tau) = \frac{1}{T_c}\left(\left[\sum_{i=0}^{L-1} K(i)\delta(\tau - iT_c)\right] * \sum_{j=-\infty}^{+\infty} \delta(\tau - jT_R)\right) * K_m(\tau)$$

One period of the PRN sequence's autocorrelation function (discrete)

Periodicity of the whole PRN sequence

Shaping waveform autocorrelation

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (6/14)

### PRN Signal Autocorrelation (5/7)

Example (1/3): m-sequence with rectangular shaping

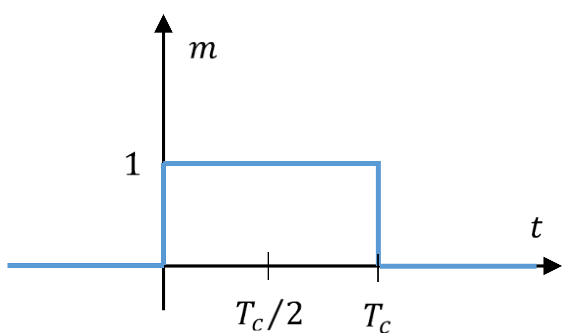
- Rectangular waveform: $m(t) = rect_{T_c}\left(t - \frac{T_c}{2}\right)$

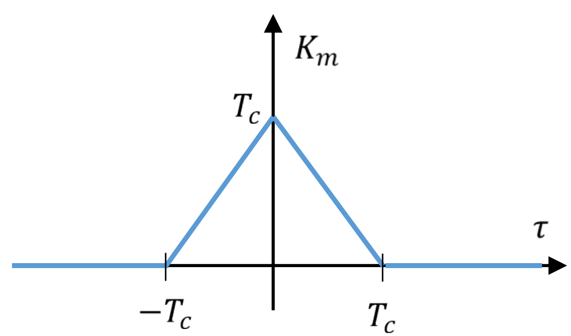
  where $rect_y(t-x)$ is defined as a unit rectangular function centered in $x$ and of duration $y$

- The autocorrelation of this rectangular waveform is:

$$K_m(x) = T_c \times tri_{T_c}(x)$$

where $tri_y(t-x)$ is defined as a triangular function of unit amplitude centered in $x$ and of base $2y$



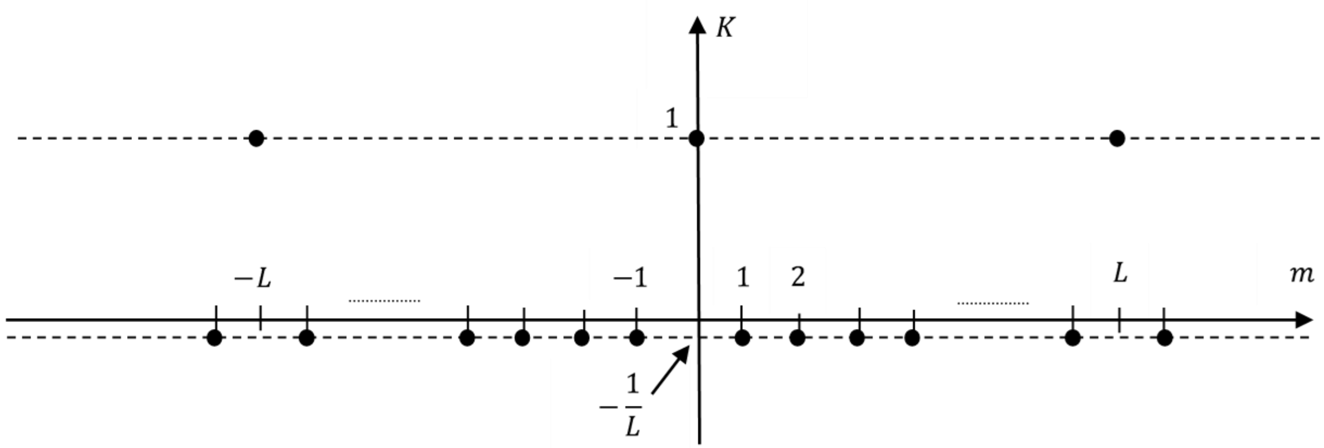
# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (7/14)

### PRN Signal Autocorrelation (6/7)

Example (2/3): m-sequence with rectangular shaping

- Reminder of an m-sequence autocorrelation function $K$ (see in section 1.4):



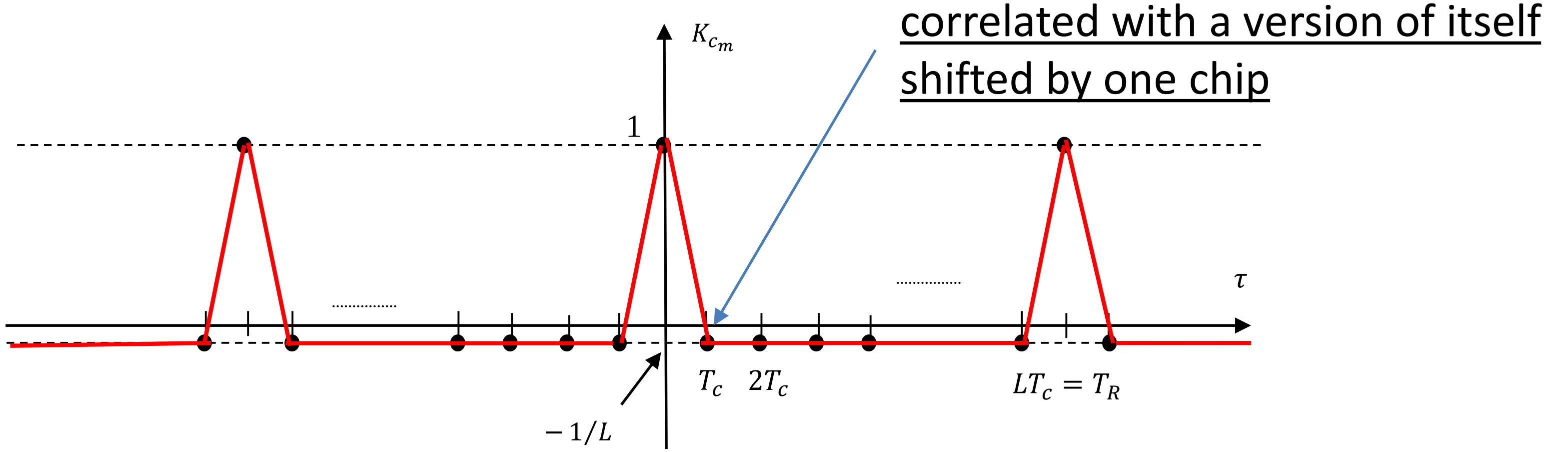
# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (8/14)

### PRN Signal Autocorrelation (7/7)

Example (3/3): m-sequence with rectangular shaping

- PRN signal autocorrelation function:




- The PRN signal autocorrelation function is driven by
  - the discrete PRN sequence autocorrelation for the global shape
  - the shaping waveform autocorrelation function drives for the local shape

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic and Deterministic (9/14)

### PRN Signal PSD (1/6)

- The PSD of the PRN signal (after shaping) can now be computed using the well-known relationship:

$$S_{c_m}(f) = \mathrm{FT}\left[K_{c_m}\right](f) = S_{\mathrm{c}}(f) * S_m(f)$$

where FT is the Fourier Transform operator

- Then:

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic & Deterministic (10/14)

### PRN Signal PSD (2/6)

- Finally:

$$S_{c_m}(f) = \frac{1}{LT_c^2} \times \left( \sum_{i=0}^{L-1} K(i) e^{-j2\pi f i T_c} \right) \times \left( \sum_{k=-\infty}^{+\infty} \delta(f - kF_R) \right) \times S_m(f)$$

where $S_m(f) = \mathrm{FT}[K_m](f)$ is the PSD of the shaping waveform

- The PSD of the PRN signal is a discrete PSD (non-zero every $F_R$)
  - expected from a periodic signal → The PSD is a peak or line spectrum

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic & Deterministic (11/14)

### PRN Signal PSD (3/6)

- The PSD of the PRN signal can be simplified into:

$$S_{c_m}(f) = \frac{1}{LT_c^2} \times \left( \left( \sum_{\mathrm{i}=0}^{\mathrm{L}-1} S(i)\delta(f - iF_R) \right) * \left( \sum_{k=-\infty}^{+\infty} \delta(f - kf_c) \right) \right) \times S_m(f)$$

PSD of the discrete PRN sequence (one period) that takes value only at frequencies that are multiple of $F_R = \frac{1}{T_R}$

Dirac comb that duplicates the left term every <u>chipping rate</u> $f_c = \frac{1}{T_c}$

PSD of the shaping waveform

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic & Deterministic (12/14)

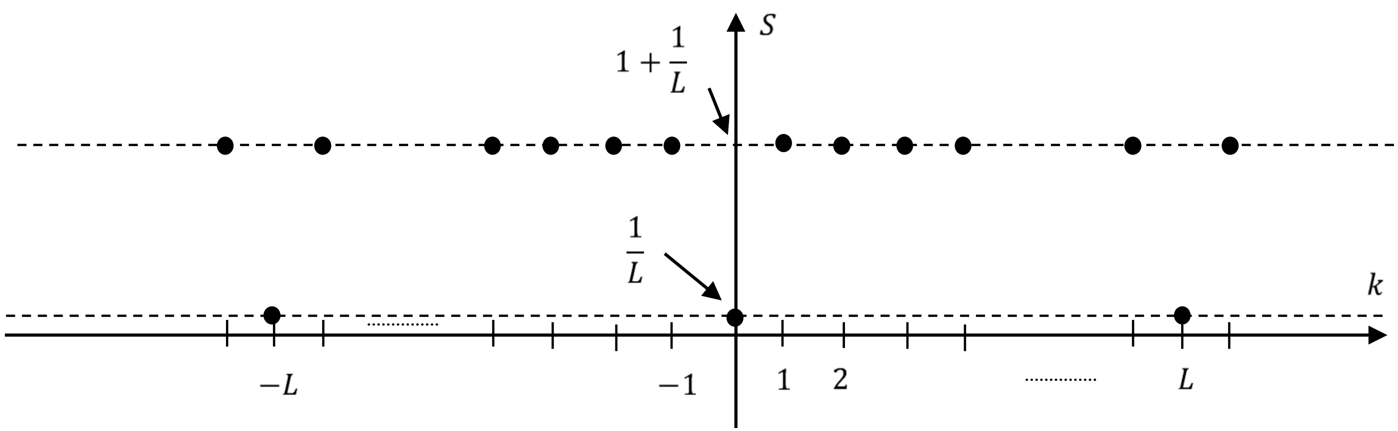
### PRN Signal PSD (4/6)

Example (1/3): m-sequence with rectangular shaping

- The PSD of the rectangular shaping waveform of time support $T_c$ is:

$$S_m(f) = \left(T_c \frac{\sin(\pi f T_c)}{\pi f T_c}\right)^2$$

- The PSD of the discrete PRN sequence $S$ (see section 1.4) is



# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic & Deterministic (13/14)

### PRN Signal PSD (5/6)

Example (2/3): m-sequence with rectangular shaping

- The theoretical PRN signal PSD is:

$$S_{c_m}(f) = \begin{cases} \left(\frac{L+1}{L^2}\right)\text{sinc}^2\left(\frac{\pi\text{k}}{L}\right) & for\ f = kF_R\ with\ k \in \mathbb{Z}\ and\ k\ not\ multiple\ of\ L \\ \frac{1}{L^2} & for\ f = 0 \\ 0 & \text{elsewhere} \end{cases}$$

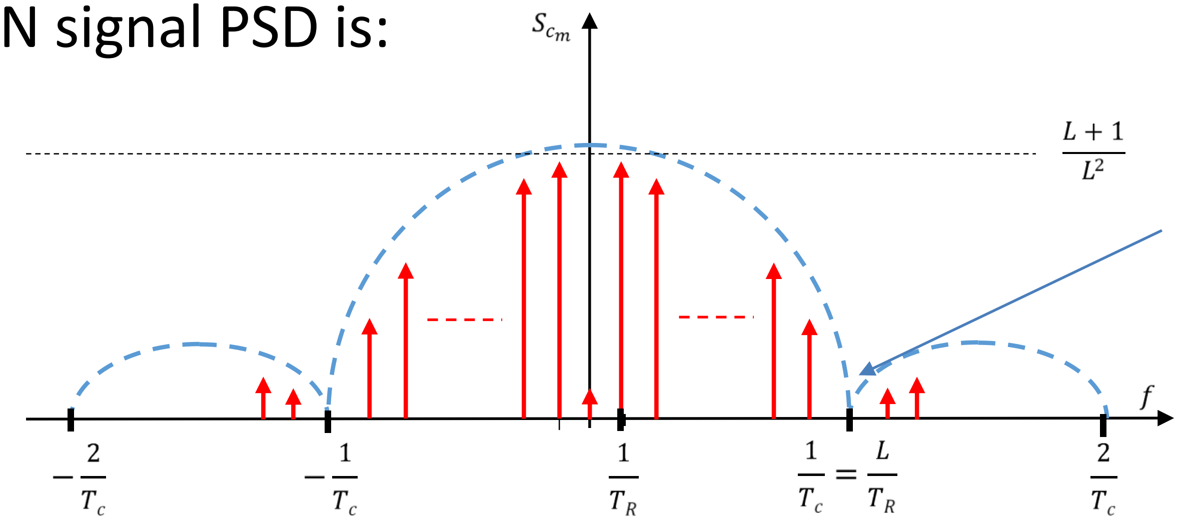
# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Periodic & Deterministic (14/14)

### PRN Signal PSD (6/6)

<u>Example (3/3):</u> m-sequence with rectangular shaping

- The PRN signal PSD is:




The width of the PRN signal PSD is driven by the choice of the PRN chip duration:
→ <u>the shorter $T_c$ (the higher the chipping rate $f_c$), the wider the PRN signal PSD</u>

- The general shape of the PRN signal PSD follows
  - the PSD of the shaping waveform,
  - its local shape is driven by the PSD of the PRN sequence.

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Random (1/6)

- The PRN sequence is now assumed random, uncorrelated, binary and without periodicity.
  - This assumption is taken when the PRN sequence is long, typically several hundred bits

- The model for the PRN signal after shaping is then:

$$c_m(t) = \sum_{k=-\infty}^{+\infty} c(k)m(t - kT_c)$$

where c is the infinite PRN sequence and $\mathrm{E}[c(i)c(j)] = \delta(i - j)$

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Random (2/6)

### PRN Signal Autocorrelation (1/3)

- The autocorrelation function of a random signal is given by:

$$K_{c_m}(\tau) = E[c_m(t)c_m{}^*(t-\tau)]$$

where $E$ represents the expectation operator

- It is possible to make the PRN signal stationary by inserting a random phase $\varphi$ that is uniformly distributed over $[0; T_c]$ in the term $m(t-kT_c)$.
  - $\varphi$ is assumed independent from $c$

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Random (3/6)

### PRN Signal Autocorrelation (2/3)

- Finally:

$$K_{c_m}(\tau) = \frac{1}{T_c} K_m(\tau)$$

→ The resulting PRN signal autocorrelation function can be approximated by the autocorrelation of the shaping waveform (with a coefficient)

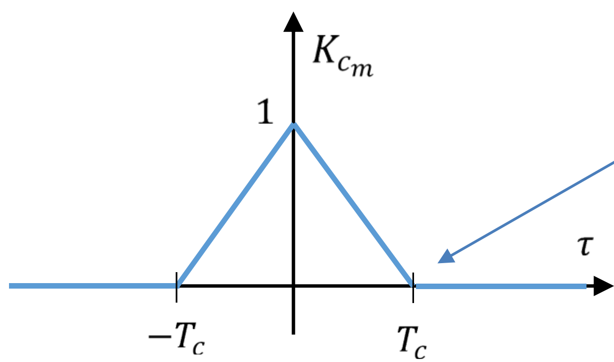
# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Random (4/6)

### PRN Signal Autocorrelation (3/3)

Example: Discrete PRN signal shaped by a rectangular waveform

- The rectangular waveform expression and its autocorrelation function $K_m$ were already given in section 2.3




The PRN signal is assumed not correlated with a version of itself shifted by more than one chip

- The autocorrelation function of the PRN signal is thus a triangular function with a support equal to $[-T_c;\ +T_c]$.
  - Corresponds to the “determinist case” with $L \rightarrow \infty$

# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Random (5/6)

### PRN Signal PSD (1/2)

- The PSD of the PRN signal is the Fourier Transform of its autocorrelation function
- Thus:

$$S_{c_m}(f) = \mathrm{FT}\left[K_{c_m}\right] = \frac{S_m(f)}{T_c}$$

- The resulting PRN signal PSD can be approximated by the PSD of the shaping waveform (with a coefficient).
  - Note that this PSD function is a continuous function (not discrete).

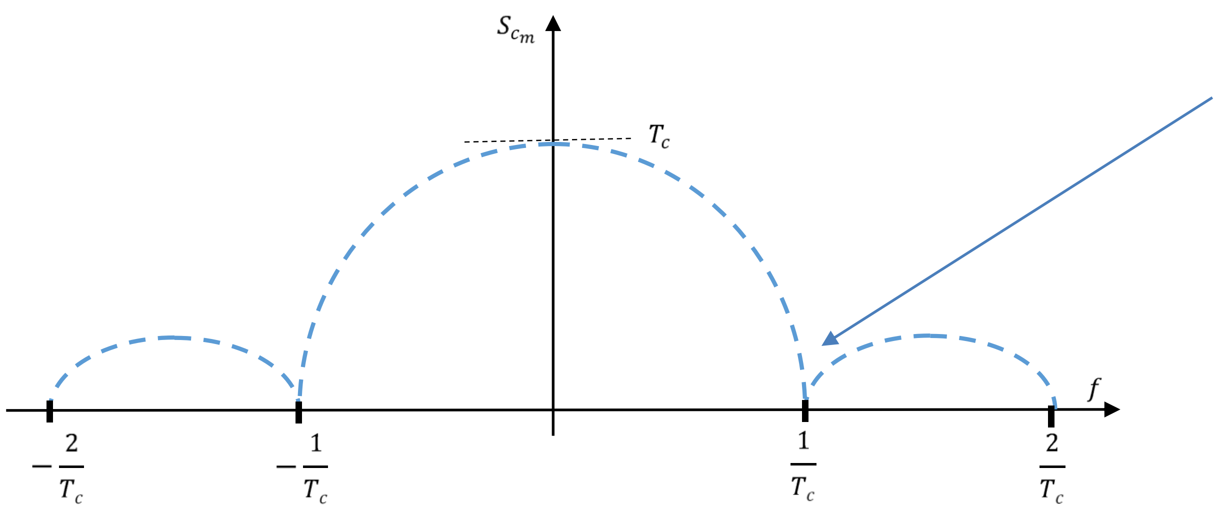
# 2.3 PRN Signal Properties

## PRN Sequence is Assumed Random (6/6)

### PRN Signal PSD (2/2)

Example: Discrete PRN signal shaped by a rectangular waveform

- The PSD of a rectangular waveform was already given in section 2.3
- the PSD of the PRN signal is: $S_{c_m}(f) = T_c \operatorname{sinc}^2(\pi f T_c)$




PRN chipping rate controls the PRN signal PSD width

The PSD of the PRN signal is thus the enveloppe of the "deterministic case"

# 2.4 Transporting Useful Information using DS-SS

- Now that the properties of the PRN sequence and its associated PRN signal have been presented, let us look at how it is used to transmit the useful information.

- Notations:
  - $d$ is the data signal carrying the binary information data to be transmitted to the user via the propagation channel.
  - $T_d$ the duration of a data symbol
  - $f_d = \frac{1}{T_d}$ is the data symbol rate.

# 2.4 Transporting Useful Information using DS-SS

- The binary data transmission using DS-SS is defined as the transmission of a signal that is the product of
  - the data signal
  - a PRN signal that has a chip duration $T_c$ much smaller than the data symbol duration $T_d$:

$$\underline{T_d \ll T_c \text{ or equivalently } f_c \gg F_d}$$

- The general time-domain structure of the transmitted DS-SS signal is:

$$s_T(t) = \mathrm{Re}\left[Ad(t)c_m(t)e^{-j2\pi f_L}\right]$$

where

- $A$ is the amplitude of the transmitted signal
- $f_L$ is the carrier frequency

# 2.4 Transporting Useful Information using DS-SS

- The specific time-domain structure of a transmitted DS-SS signal is thus found by customizing the previous equation

- For simplification purpose, we will assume that the <u>data modulation is a BPSK modulation</u> with the data bits represented as rectangles of duration $T_d$.
  - The transmitted DS-SS signal is then

$$s_T(t) = Ad(t)c_m(t)\cos(2\pi f_L t)$$

with

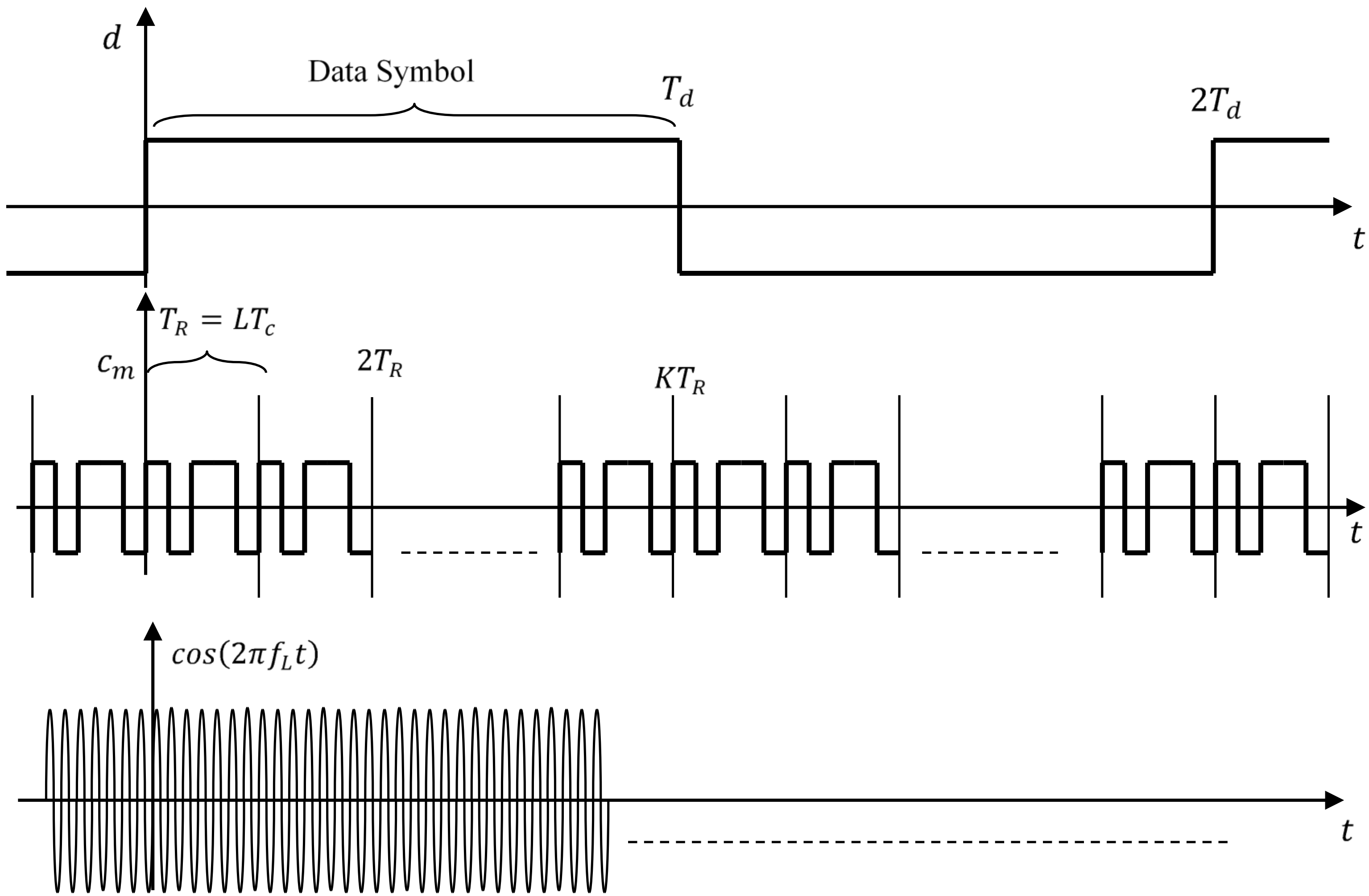
$$d(t) = \left[\sum_{k=-\infty}^{+\infty} d_k \delta(t - kT_d)\right] * rect_{T_d}\left(t - \frac{T_d}{2}\right)$$

where $d_k$ is the binary information, $d_k \in \{-1,1\}$

# 2.4 Transporting Useful Information using DS-SS

- In practice, for a DS-SS signal, the following parameters are chosen:
  - $T_d = KT_R$ with $K$ integer equal or greater than 1.
    - This means that there is an integer number of PRN sequences within one data symbol.
  - $f_L = Mf_c$ with $M$ integer significantly greater than 1.
  - The 3 components (data, PRN sequence and carrier) are synchronous:
    - The data bits start and end systematically at the same instant as a PRN sequence,
    - The PRN sequence starts when the phase of the carrier is equal to 0.

# 2.4 Transporting Useful Information using DS-SS



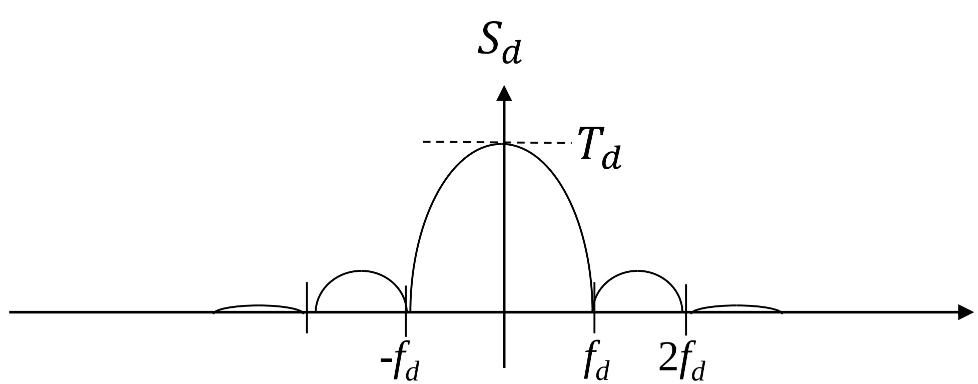
# 2.5 Properties of the Transmitted DS-SS Signal

## PSD of the Data Signal

- The PSD of the data signal $d$ (without DS-SS) is (BPSK modulation):

$$S_d(f) = T_d \,\mathrm{sinc}^2(\pi f T_d)$$

→ the PSD of the data signal is thus mainly contained within the main lobe of the sinc function: $[-f_d; f_d]$ Hz



# 2.5 Properties of the Transmitted DS-SS Signal

## PSD of the Transmitted DS-SS Signal (1/5)

- Assuming that the PRN sequence is fully random and that $d$ and $c$ are independent, it is possible to make the transmitted GNSS signal $s_T$ stationary by adding a random phase $\theta$ with a uniform distribution over $[0; 2\pi]$ in the carrier ($\theta$ being independent from $c$ and $d$).
  - This leads to an autocorrelation function of the transmitted GNSS signal equal to:

$$K_{s_T}(\tau) = \frac{A^2}{2} K_d(\tau) K_{c_m}(\tau) \cos(2\pi f_L \tau)$$

# 2.5 Properties of the Transmitted DS-SS Signal

## PSD of the Transmitted DS-SS Signal (2/5)

- The PSD of the transmitted GNSS signal is then:

$$S_{s_T}(f) = FT\left[K_{s_T}(\tau)\right](f) = \frac{A^2}{4}\left[S_{dc_m}(f - f_L) + S_{dc_m}(-f + f_L)\right]$$

where $S_{dc_m}(f) = S_d(f) * S_{c_m}(f)$

- $S_{dc_m}$ represents the PSD of the data signal after multiplication by the PRN signal:
  - the width of the PSD of the PRN signal is at least twice the PRN chipping rate
  - for a DS-SS signal, $f_c \gg f_d$:

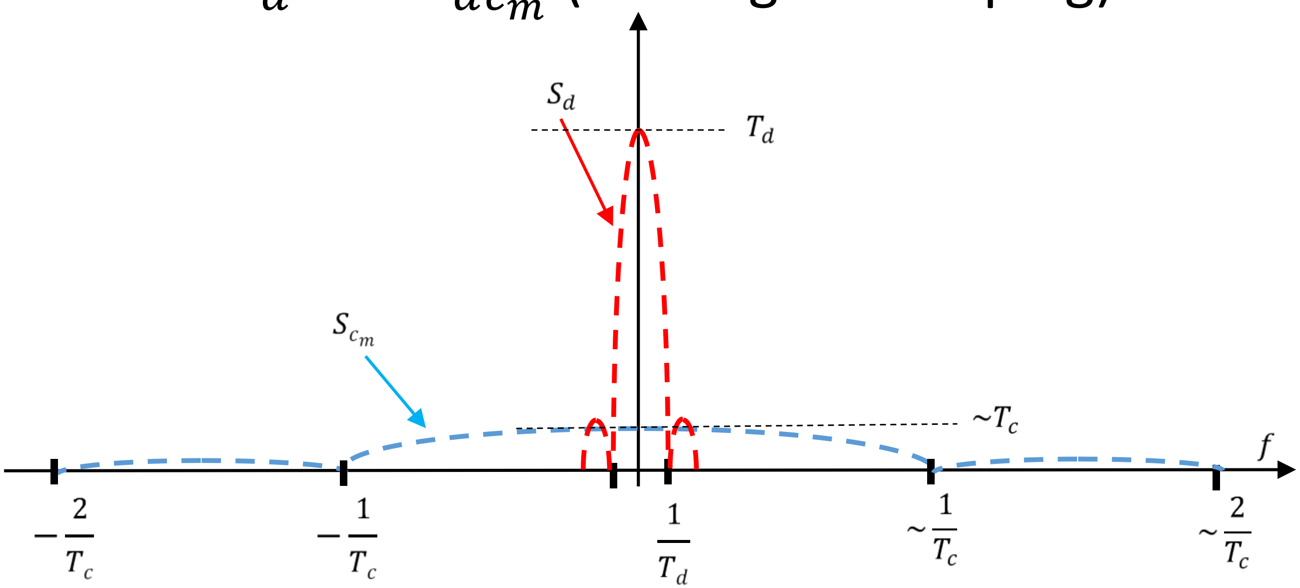
    → <u>$S_{dc_m}$ occupies a much wider frequency range than $S_d$.</u>

    → <u>The transmitted data signal is indeed spread in the frequency domain by the PRN signal</u>

# 2.5 Properties of the Transmitted DS-SS Signal

## PSD of the Transmitted DS-SS Signal (3/5)

- Comparison of $S_d$ and $S_{dc_m}$ (rectangular shaping):

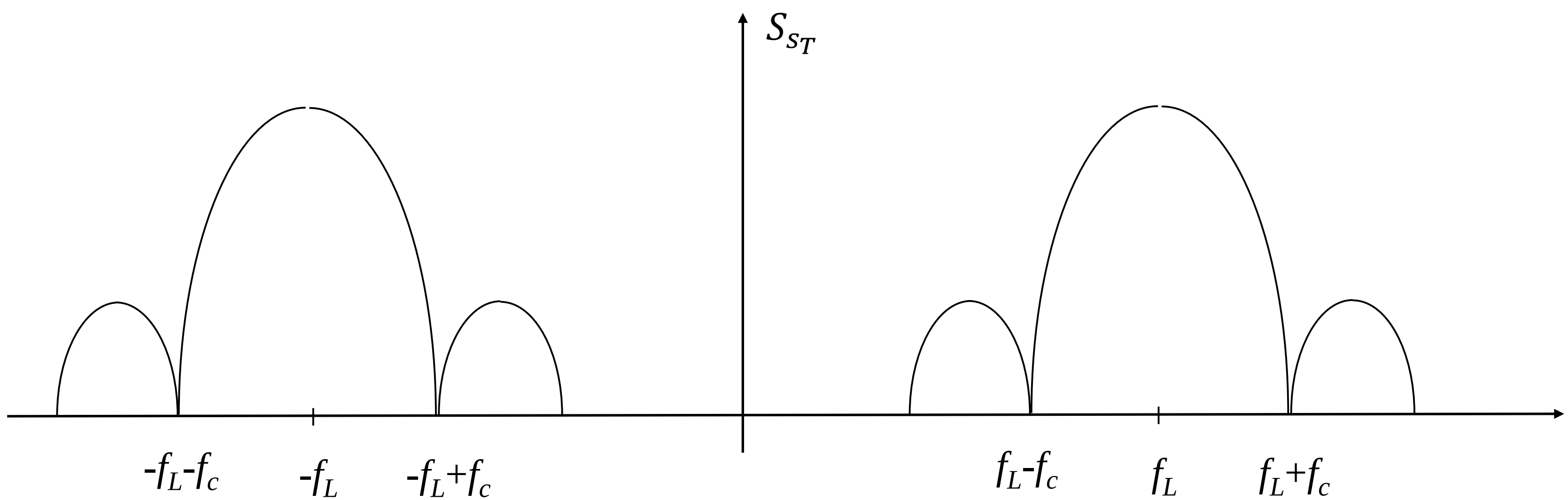



- The bandwidth increase is approximately $KL = T_d/T_c$ → typically a factor of several hundreds or thousands
- The maximum amplitude of $S_{dc_m}$ is significantly lower than that of $S_d$, approximately by the same factor $KL$

# 2.5 Properties of the Transmitted DS-SS Signal

## PSD of the Transmitted DS-SS Signal (4/5)

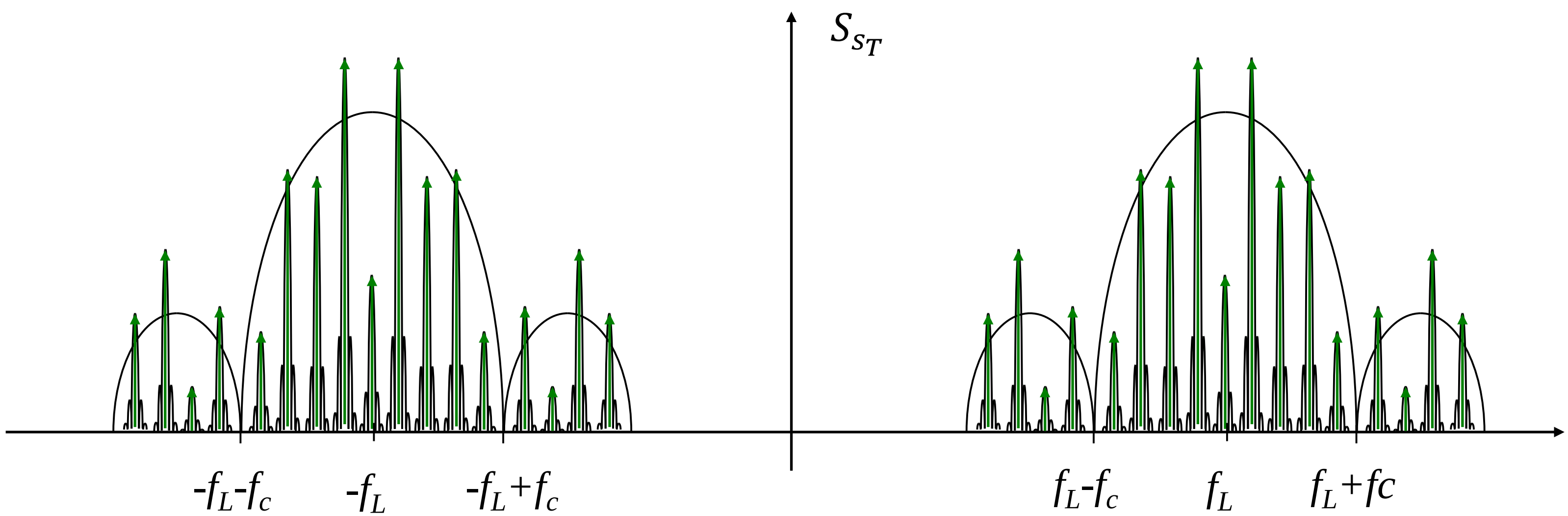
- Representation of the PSD of the transmitted GNSS signal (PRN sequence assumed fully random)



# 2.5 Properties of the Transmitted DS-SS Signal

## PSD of the Transmitted DS-SS Signal (5/5)

- Representation of the PSD of the transmitted GNSS signal (PRN sequence finite and periodic)



# 3. Received DS-SS Signal Model

# 3.1 Introduction

- To process optimally the received signal, it is important to know how the transmitted signal has been distorted until reaching the processing module.
- This includes the distortions brought by
  - the transmitter,
  - the propagation medium,
  - the user antenna, and
  - the receiver RF front end
- It is also important to know what kind of disturbances enter the antenna together with the useful signal (interference).
- These elements are all part of what is considered as the propagation channel. It is thus important to identify and model this propagation channel before deciding upon the processing technique

# 3.2 DS-SS Signal Model at the RF Front-End Output

- The generic received signal can be modeled as:

$$r(t) = h(t) * [g(t) * s_T(t) + b(t)]$$

where

- $g$ is the impulse response of the propagation medium
- $h$ is the impulse response of the equivalent RF front-end
- $b$ is an additive perturbation

# 3.2 DS-SS Signal Model at the RF Front-End Output

## Assumptions (1/2):

- (Reminder) The transmitted GNSS signal of interest is modeled as:
$$s_T(t) = Ad(t)c_m(t)\cos(2\pi f_L t)$$
  - where
    - A is the amplitude of the GNSS signal at the RF front-end input
    - $d$ is the navigation message carried by the GNSS signal
    - $c_m$ is the PRN code signal (see Chapter 7)
    - $f_L$ is the carrier frequency of the signal at the RF front-end input

- The propagation medium behaves only as a pure delay (representative of the propagation time) which can vary in time. It entails that the equivalent propagation filter can be expressed by:
$$g(t) = \delta\big(t - \tau(t)\big)$$

# 3.2 DS-SS Signal Model at the RF Front-End Output

## Assumptions (2/2):

- The RF front-end filter can be modelled as an equivalent filter
  - Its impulse response is $h_{RF}$ and its transfer function is $H_{RF}$.
  - $H_{RF}$ is assumed to be rectangular and to have a two-sided bandwidth $B$ :

$$H_{RF}(f) = rect_B(f) * [\delta(f - f_0) + \delta(f + f_0)]$$

  - The equivalent RF Front-End filter at baseband is thus:

$$H_{RF,BB}(f) = rect_B(f)$$

- The only considered additive perturbation is the noise created by the RF front-end. It creates an additional white Gaussian (thermal) noise $n$ that is modeled as:

$$S_n(f) = \frac{N_0}{2}\ \mathrm{W/Hz}$$

# 3.2 DS-SS Signal Model at the RF Front-End Output

- Based on these assumptions, the signal that will be processed by the receiver signal processing block can be modelled as:

$$r_{RF_{out}}(t) = \Big(\big(s_T\big(u-\tau(u)\big)+n(u)\big) * h_{RF}\Big)(t)$$

Notation:

$$r_{RF_{out}}(t) = \underbrace{(s_r * h_{RF})}_{u_{RF_{out}}}(t) + n_{RF_{out}}(t)$$

with

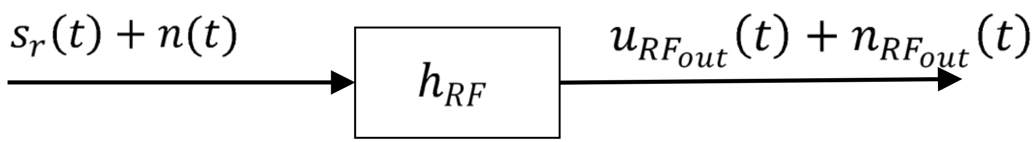
$$s_r(t) = s_T\big(t-\tau(t)\big) = Ad\big(t-\tau(t)\big)c_m\big(t-\tau(t)\big)\cos\big(2\pi f_L\big(t-\tau(t)\big)+\theta_0\big)$$

where

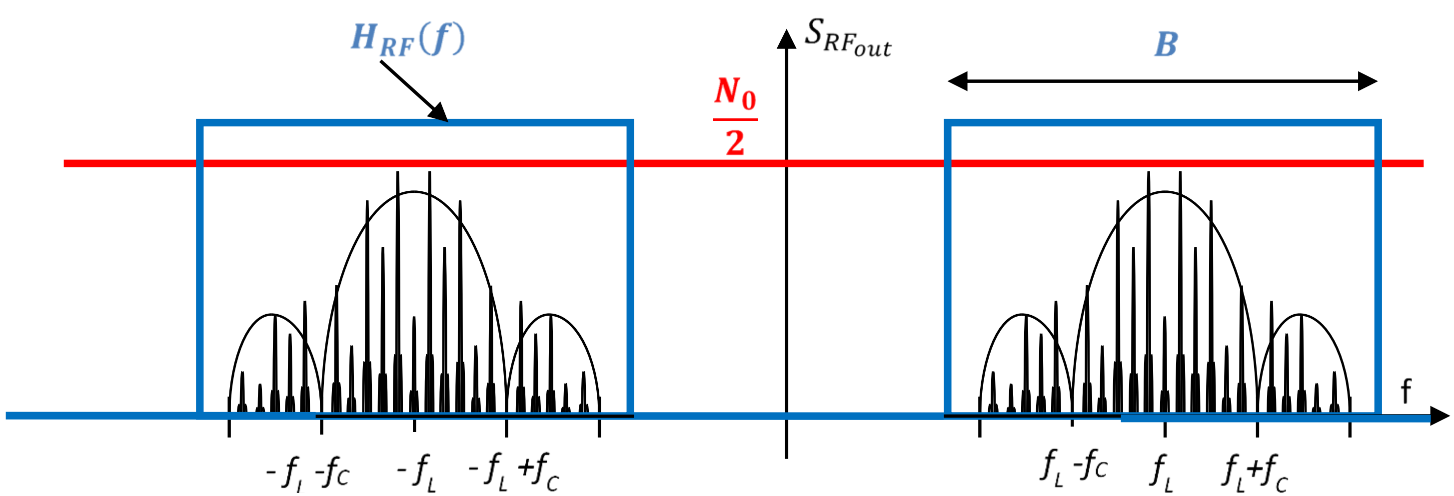
- $u_{RF_{out}}$ is the useful part of the signal coming out of the RF front-end
- $n_{RF_{out}} = (h_{RF} * n)$ is the thermal noise filtered by the RF front-end

# 3.2 DS-SS Signal Model at the RF Front-End Output

- Representation of the effect of the RF front-end




- The components of the Power Spectral Density (PSD) of the received signal, $S_{RF_{out}}$, at the output of the RF front-end are



# 3.3 SNR at the RF Front-End Output

## Power of the Useful Signal at the RF Front-End Input

- Let us denote $C$ the power of the useful signal $s_T$ at the entrance of the RF front-end

- It is easy to show that the power of the useful signal $s_T$ at the entrance of the RF front-end is:

$$C = \frac{A^2}{2}$$

# 3.3 SNR at the RF Front-End Output

## Power of the Useful Signal at the RF Front-End Output (1/6)

- Let us denote $C_{RFout}$ the power of the useful signal $u_{RF_{out}}$ at the RF front-end output, and

- The useful signal only consists in one wide band term: the PRN signal.
  - the RF front-end filter will only affect this component of the useful signal.

$$u_{RF_{out}}(t) = Ad\big(t-\tau(t)\big)\tilde{c}_m\big(t-\tau(t)\big)\cos\big(2\pi f_0\big(t-\tau(t)\big)+\theta_0\big)$$

where $\tilde{c}_m = \left(c_m * h_{RF,BB}\right)$ is the <u>PRN signal filtered by the RF front-end</u>

# 3.3 SNR at the RF Front-End Output

## Power of the Useful Signal at the RF Front-End Output (2/6)

- Assuming that:
  - $d$ is a binary random signal with a bit duration equal to $T_d$. In this case, its autocorrelation function equals

$$K_d(\tau)=\text{tri}\left(\frac{\tau}{T_d}\right)$$

  - $\tilde{c}_m$ is a random signal with an autocorrelation function equal to $K_{\tilde{c}_m}$.
  - $d$, $\tilde{c}_m$ and an additive random phase present in the carrier (uniformly distributed over $[0; 2\pi]$ to make the signal stationary) are independent,

- The autocorrelation function of $u_{RF_{out}}$ can be calculated as:

$$K_{u_{RF_{out}}}(x) = \mathrm{E}\left[u_{RF_{out}}(t)u_{RF_{out}}(t-x)\right]$$

# 3.3 SNR at the RF Front-End Output

## Power of the Useful Signal at the RF Front-End Output (3/6)

- Finally, the autocorrelation function of $u_{RF_{out}}$ equals:

$$K_{u_{RF_{out}}}(x) = C \times K_d(x) \times K_{\tilde{c}_m}(x) \times \cos(2\pi f_L x)$$

# 3.3 SNR at the RF Front-End Output

## Power of the Useful Signal at the RF Front-End Output (4/6)

- The power of a signal is given by the value of its autocorrelation function in 0:

$$C_{RF_{out}} = K_{u_{RF_{out}}}(0) = C \times \underbrace{K_d(0)}_{=1} \times K_{\tilde{c}_m}(0) = C \times K_{\tilde{c}_m}(0)$$

- In general, the RF front-end filter can be considered wide enough to let the PRN signal through without distortions:

$$K_{\tilde{c}_m}(0) = K_{c_m}(0) = 1$$

- Then:

$$C_{RF_{out}} = C$$

# 3.3 SNR at the RF Front-End Output

## Power of the Useful Signal at the RF Front-End Output (5/6)

- If the RF front-end filter distorts the PRN signal, then $K_{\tilde{c}_m}(0)$ can be computed as:

$$K_{\tilde{c}_m}(0) = IFFT\left(S_{\tilde{c}_m}\right)(0) = \int_{-\infty}^{+\infty} S_{\tilde{c}_m}(f)df$$

and using Wiener-Lee

$$S_{\tilde{c}_m}(f) = \left|H_{RF,BB}(f)\right|^2 S_{c_m}(f) = |rect_B(f)|^2 S_{c_m}(f)$$

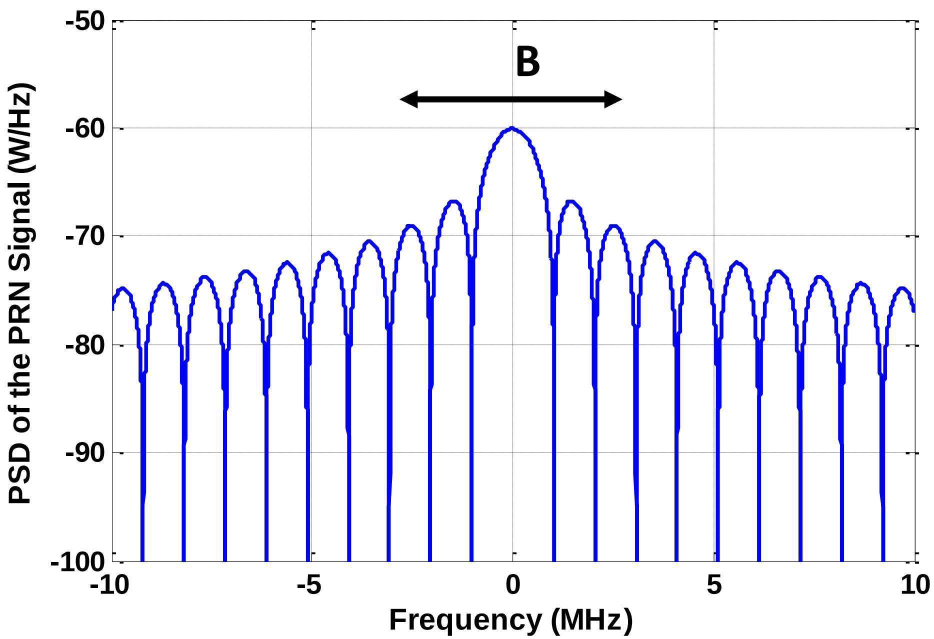
- Then the closed form expression of $C_{RF_{out}}$ is:

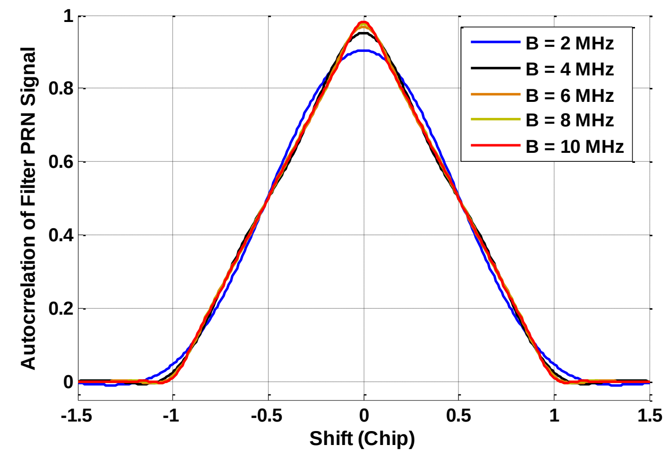
$$C_{RF_{out}} = C\left(\int_{-B/2}^{B/2} S_{c_m}(f)df\right)$$

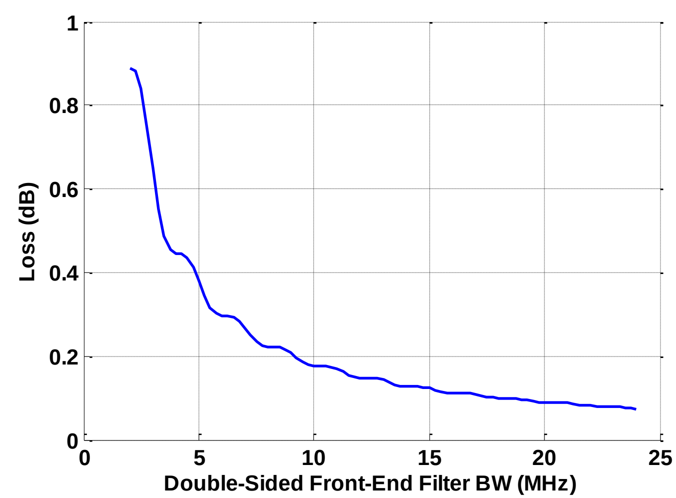
# 3.3 SNR at the RF Front-End Output

## Power of the Useful Signal at the RF Front-End Output (6/6)

- Representation of $K_{\tilde{c}_m}$ assuming
  - an infinite PRN sequence
  - a chipping rate of 1,023 Mcps
  - rectangular shaping waveform



# 3.3 SNR at the RF Front-End Output

## Power of the Noise at the RF Front-End Output

- Let us denote $P_{n_{RF_{out}}}$ the power of the noise $n_{RF_{out}}$ at the RF front-end output
- Using Wiener-Lee and the expression of $H_{RF}$:

$$S_{n_{RF_{out}}}(f) = |H_{RF}(f)|^2 S_n(f) = \frac{N_0}{2}|rect_B(f)|^2 * [\delta(f-f_0) + \delta(f+f_0)]$$

- As a consequence:

$$P_{n_{RF_{out}}} = IFFT\left(S_{n_{RF_{out}}}\right)(0) = \int_{-\infty}^{+\infty} S_{n_{RF_{out}}}(f)df$$

$$P_{n_{RF_{out}}} = N_0 B$$

# 3.3 SNR at the RF Front-End Output

- From what has been done before, the generalized SNR at the RF front-end output is:

$$SNR_{RF_{out}} = \frac{C_{RF_{out}}}{P_{n_{RF_{out}}}} = \frac{C}{N_0 B} \times \left( \int_{-B/2}^{B/2} S_{c_m}(f) df \right)$$

- Assuming that the RF front-end filter is wide with respect to the PRN signal (it does not distort the PRN signal), this can approximated as:

$$SNR_{RF_{out}} = \frac{C}{N_0 B}$$

# 4. Correlation Operation

# 4.1 Presentation of a Correlator

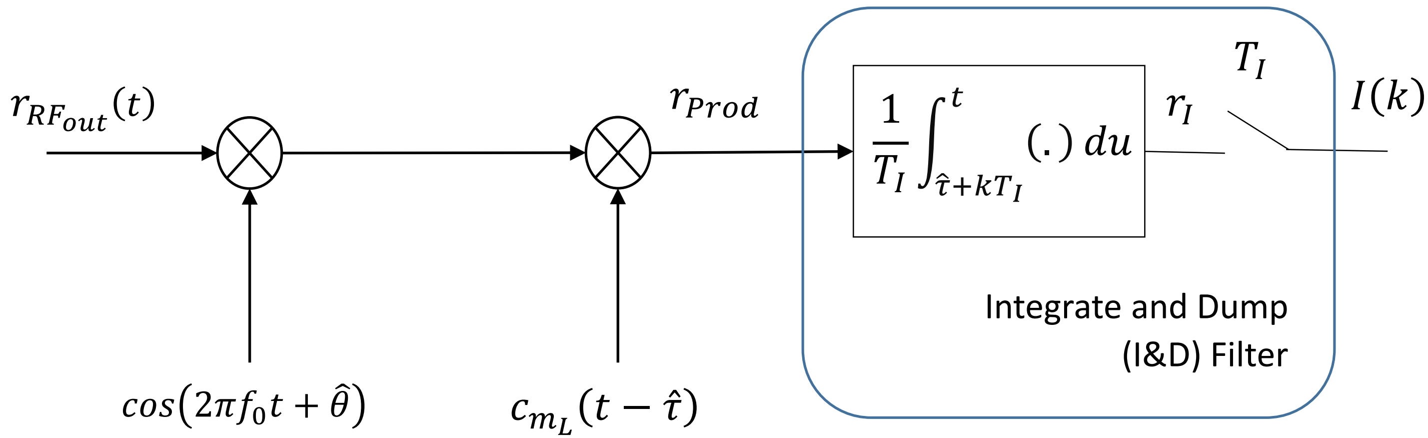
- In order to recover the useful information $d$ present in the transmitted signal, the receiver must perform several steps such as:
  - synchronization with the incoming carrier phase
  - synchronization with the incoming PRN signal
  - symbol demodulation
- These steps are all based on a specific operation referred to as the correlation operation.

# 4.1 Presentation of a Correlator

- This operation consists in computing the correlation between the incoming signal and a local replica of the useful signal.
- The local replica is composed exclusively of a
  - local carrier, and
  - a local PRN signal $c_{m_L}$. Note that the local PRN signal is defined as the PRN code after shaping.

# 4.1 Presentation of a Correlator

- Representation of the correlation operation




where

- $\hat{\tau}$ represents the estimated delay used to generate the local PRN signal
- $\hat{\theta}$ represents the estimated phase used to generate the local carrier replica
- $T_I$ is the <u>integration (or correlation) duration</u>
- $r_{Prod}$, $r_I$ and $I$ represent the signal at different levels of the correlation process.

<u>The correlator output rate</u> is the inverse of the correlation duration $T_I$ which <u>will be</u> much lower than the sampling frequency at the correlator input

# 4.2 Modeling the Integrate and Dump Filter

- From the previous figure, the correlator output is:

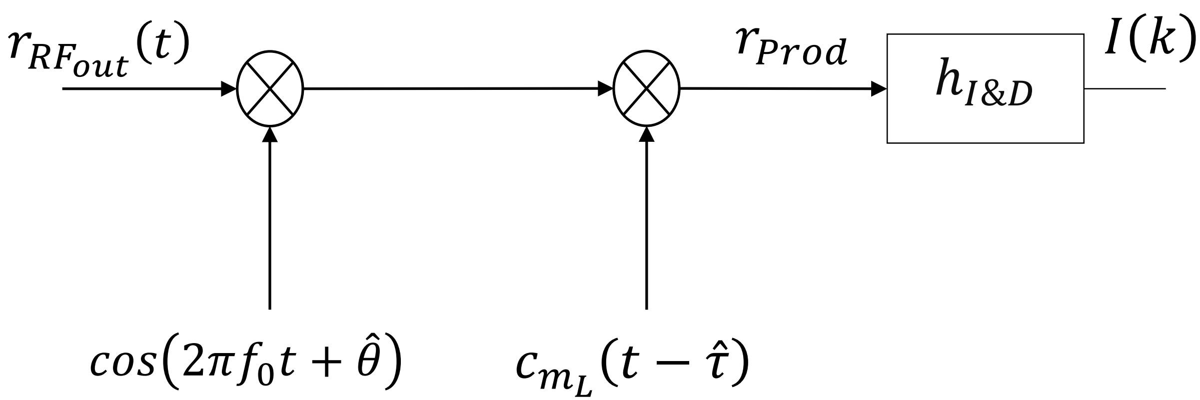
$$I(k) = r_I(\hat{\tau} + (k+1)T_I) = \frac{1}{T_I}\int_{\hat{\tau}+\mathrm{k}T_I}^{\hat{\tau}+(k+1)T_I} r_{Prod}(u)du$$

- The correlator output can thus be seen as the output of a filter that takes $r_{Prod}$ as an input
- It is easy to show that this filter, referred to as the <u>Integrate and Dump (I&D) filter</u>, has an impulse response equal to :

$$h_{I\&D}(t) = \frac{1}{T_I} rect_{T_I}\left(t - \frac{T_I}{2}\right)$$

# 4.2 Modeling the Integrate and Dump Filter

- This I&D filter is thus a low-pass filter with an approximate bandwidth of $\sim\frac{2}{T_I}$ Hz.
- A simplified representation of the correlation process is thus



# 4.3 Useful Signal Model at the Corr. Output

- The signal $r_{Prod}$ at the input of the I&D filter can be modeled as:

$$r_{Prod}(t) = r_{RF_{out}}(t)\cos\left(2\pi f_0 t - \hat{\theta}\right)c_{m_L}(t-\hat{\tau})$$

- This can be written as:

$$r_{Prod}(t) = \frac{A}{2}d(t-\tau)\tilde{c}_m(t-\tau)c_{m_L}(t-\hat{\tau})\left[\cos\left(4\pi f_0 t - \theta - \hat{\theta}\right) + \cos\left(\theta - \hat{\theta}\right)\right] + r_{Prod,n}(t)$$

where $r_{Prod,n}(t) = n_{RF_{out}}(t)\cos\left(2\pi f_0 t - \hat{\theta}\right)c_{m_L}(t-\hat{\tau})$ is the noise component

# 4.3 Useful Signal Model at the Corr. Output

- Then, the signal $r_{Prod}$ goes through the low-pass I&D filter → It can thus be anticipated that this filter will fully remove the high frequency component of the useful signal:

$$I(k) = \frac{A}{2T_I} \int_{\hat{\tau}+kT_I}^{\hat{\tau}+(k+1)T_I} d(u-\tau)\tilde{c}_m(u-\tau)c_{m_L}(u-\hat{\tau})\cos\left(\theta-\hat{\theta}\right) du + n_I(k)$$

with

$$n_I(k) = \frac{1}{T_I} \int_{\hat{\tau}+kT_I}^{\hat{\tau}+(k+1)T_I} r_{Prod,n}(u)du$$

the noise term at the correlator output

# 4.3 Useful Signal Model at the Corr. Output

- To take advantage of the correlation properties of the PRN code, the correlation operation has to be performed over an exact period of the PRN code or a multiple of it
- To be able to read the data at the correlator output, the correlation operation has to be performed within a useful data bit
- In the following, it will be assumed that:
  – the receiver is synchronized with the incoming PRN code ($\hat{\tau} \sim \tau$),
  – the correlation duration $T_I$ is a multiplier of the PRN code duration $T_R$ and a divider of the data bit duration $T_D$.
    - Note that in reality, these conditions are not met during the “acquisition” phase, but only during the synchronisation phase. It entails that correlation degradations can appear during the acquisition phase

# 4.3 Useful Signal Model at the Corr. Output

- In this situation, and assuming that the phase estimation error $\theta - \hat{\theta}$ remains constant over the correlation duration:

$$I(k) = \frac{A}{2} d(k) \cos(\theta - \hat{\theta}) \underbrace{\left( \frac{1}{T_I} \int_{\hat{\tau}+kT_I}^{\hat{\tau}+(k+1)T_I} \tilde{c}_m(u-\tau) c_{m_L}(u-\hat{\tau}) du \right)}_{K_{\tilde{c}_m, c_{m_L}}(\tau-\hat{\tau})} + n_I(k)$$

where

- $d(k)$ is the value of the useful data bit during the integration interval $[\hat{\tau} + kT_I;\ \hat{\tau} + (k+1)T_I]$
- $K_{\tilde{c}_m, c_{m_L}}$ represents the correlation value between $\tilde{c}_m$ and $c_{m_L}$

# 4.3 Useful Signal Model at the Corr. Output

- Thus, the correlator output model is:

$$I(k) = \frac{A}{2} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \, cos(\varepsilon_\theta) + n_I(k)$$

- where
  - $\varepsilon_\tau = \tau - \hat{\tau}$ is the code delay estimation error made by the receiver
  - $\varepsilon_\theta = \theta - \hat{\theta}$ is the carrier phase estimation error made by the receiver

# 4.3 Useful Signal Model at the Corr. Output

- Note that the previous correlator output model assumed that the carrier phase error was constant over the correlation duration
  - If this is not the case (residual Doppler), then a more accurate model is:

$$I(k) = \frac{A}{2} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) sinc\left(\pi \varepsilon_f T_I\right) cos(\varepsilon_\theta) + n_I(k)$$

where

- $\varepsilon_\theta$ is the <u>carrier phase estimation error</u> made by the receiver in the middle of the correlation interval
- $\varepsilon_f = f_0 - \widehat{f_0}$ is the <u>carrier frequency estimation error</u> made by the receiver

# 4.3 Useful Signal Model at the Corr. Output

- Note that $K_{\tilde{c}_m,c_m}$ includes the effect of the RF front-end on the incoming PRN signal.
- Thanks to the the Wiener Lee relation, it is possible to link $K_{\tilde{c}_m,c_m}$ with the correlation function $K_{c_m}$

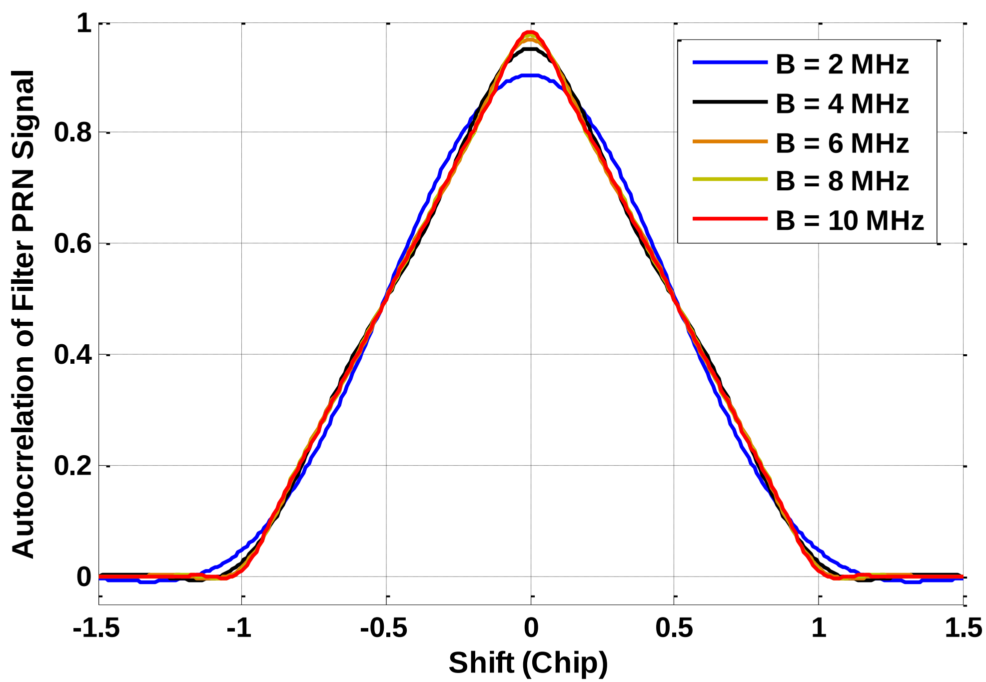
$$K_{\tilde{c}_m,c_m}(x) = \left(K_{c_m} * h_{RF_{BB}}\right)(x) = \int_{-\infty}^{+\infty} H_{RF,BB}(f) S_{c_m}(f) e^{2i\pi f x} df$$

- In our case:

$$K_{\tilde{c}_m,c_m}(x) = \int_{-\mathrm{B}/2}^{\mathrm{B}/2} S_{c_m}(f) e^{2i\pi f x} df$$

# 4.3 Useful Signal Model at the Corr. Output

- Representation of $K_{\tilde{c}_m,c_m}$ assuming
  - an infinite PRN sequence
  - a chipping rate of 1,023 Mcps
  - rectangular shaping waveform



# 4.3 Useful Signal Model at the Corr. Output

- Assuming that the RF front-end is large enough
  - The correlator output model becomes, based on the PRN sequence properties:

$$I(k) = \begin{cases} \frac{A}{2} d(k) K_{c_m}(\varepsilon_\tau)\, cos(\varepsilon_\theta) + n_I(k) & if\, c_m = c_{m_L} \\ n_I(k) & if\, c_m \neq c_{m_L} \end{cases}$$

- This is the basis of CDMA:
  - The PRN sequences are chosen to be as uncorrelated as possible
  - the information is accessible only if the user knows the PRN sequence used.

# 4.3 Useful Signal Model at the Corr. Output

- It can be seen that the useful part of the correlator output will have a significant value only if:
  - The local PRN signal uses the same PRN code as the one used by the incoming signal.
  - The code delay estimation error $\varepsilon_\tau$ is very small (much less than 1 chip), so that as $K_{\tilde{c}_m,c_m}(\varepsilon_\tau)$ is maximum
  - The carrier phase estimation error $\varepsilon_\theta$ is very small (much smaller than 1 rad) so that $cos(\varepsilon_\theta)\sim 1$

# 4.3 Useful Signal Model at the Corr. Output

- Note that a PRN signal synchronization corresponds to the multiplication of the incoming PRN signal with itself.
  - Because the PRN signal is binary, it also corresponds to the wipe-off of the PRN code, also referred to as de-spreading of the PRN signal.
  - The consequence is to bring the incoming signal back to its original narrow band at the output of the multiplier and before the low-pass I&D filter.

# 4.3 Useful Signal Model at the Corr. Output

- Assuming that the correct PRN sequence is used and that the local replica is synchronized in time and phase with the incoming signal, then:

$$I(k) \sim \frac{A}{2} d(k) \underbrace{K_{\tilde{c}_m, c_m}(0)}_{\substack{\sim 1 \\ if\ B\ is\ large}} + n_I(k) \sim \frac{A}{2} d(k) + n_I(k)$$

- It is thus possible to read the data bit at the correlator output as long as the useful signal power is sufficiently greater than that of the noise

# 4.3 Useful Signal Model at the Corr. Output

- The power of the useful signal at the correlator output is:

$$P_u = \frac{C}{2} K_{\tilde{c}_m, c_m}^2(0) \sim \frac{C}{2}$$

# 4.4 Noise Power at the Correlator Output

- The computation of the noise power at the correlator output follows a method related to the structure of the correlation process:
  1. Computation of the PSD of the noise term at the I&D filter input
  2. Computation of the PSD of the noise term at the I&D filter output
  3. Computation of the noise power at the correlator output by integrating its PSD.

# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D filter input

- The expression of the noise at the I&D input is

$$r_{Prod,n}(t) = n_{RF_{out}}(t)\cos\left(2\pi f_0 t - \hat{\theta}\right)c_{m_L}(t-\hat{\tau})$$

- let us assume that:
  - $c_{m_L}$ is a fully random PRN sequence,
  - there is an additional phase term $\vartheta$ in the carrier that has a uniform distribution over $[0; 2\pi]$.
  - $n_{RF_{out}}$, $c_{m_L}$ and $\vartheta$ are independent random variables

# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D filter input

- It can then be shown that the autocorrelation function of $r_{Prod,n}$ can be computed as:

$$K_{r_{Prod,n}}(x) = E\left[r_{Prod,n}(t)r_{Prod,n}(t-x)\right] = \frac{1}{2}\cos(2\pi f_0 x)\, K_{n_{RFout}}(x)K_{c_{m_L}}(x)$$

- The PSD of $r_{Prod,n}$ is then:

$$S_{r_{Prod,n}}(f) = \mathrm{FT}\left[K_{r_{Prod,n}}\right](f)$$

$$S_{r_{Prod,n}}(f) = \frac{1}{4}\left(\delta(f-f_0) + \delta(f+f_0)\right) * S_{n_{RFout}}(f) * S_{c_{m_L}}(f)$$

# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D filter input

- $S_{n_{RFout}}$ is the PSD of the noise at the RF front-end filter output:

$$S_{n_{RFout}}(f) = \frac{N_0}{2} * \left(\delta(f - f_0) + \delta(f + f_0)\right)$$

- Thus:

$$S_{r_{Prod,n}}(f) = \frac{N_0}{8}\left(\left|H_{RF_{BB}}(f - 2f_0)\right|^2 + 2\left|H_{RF_{BB}}(f)\right|^2 + \left|H_{RF_{BB}}(f + 2f_0)\right|^2\right) * S_{c_{m_L}}(f)$$

# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D output

- Using the Wiener-Lee relation, the PSD of the noise at the correlator output (or equivalently at the I&D output) can then be written as:

$$S_{n_I}(f) = |H_{I\&D}(f)|^2 S_{r_{Prod,n}}(f)$$

with $|H_{I\&D}(f)|^2$ is related to the low-pass I&D transfer function already defined

# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D output

- Let us compare the bandwidth of the I&D filter and the bandwidth of the noise term at the I&D input:
  - the I&D filter bandwidth is approximately $\frac{2}{T_I}$ Hz with $T_I \geq T_R$
  - $S_{r_{Prod,n}}$ is the convolution of the PSD of the local PRN signal ($S_{c_{m_L}}$) with another function → the bandwidth of $S_{r_{Prod,n}}$ will thus be at least that of $S_{c_{m_L}}$.

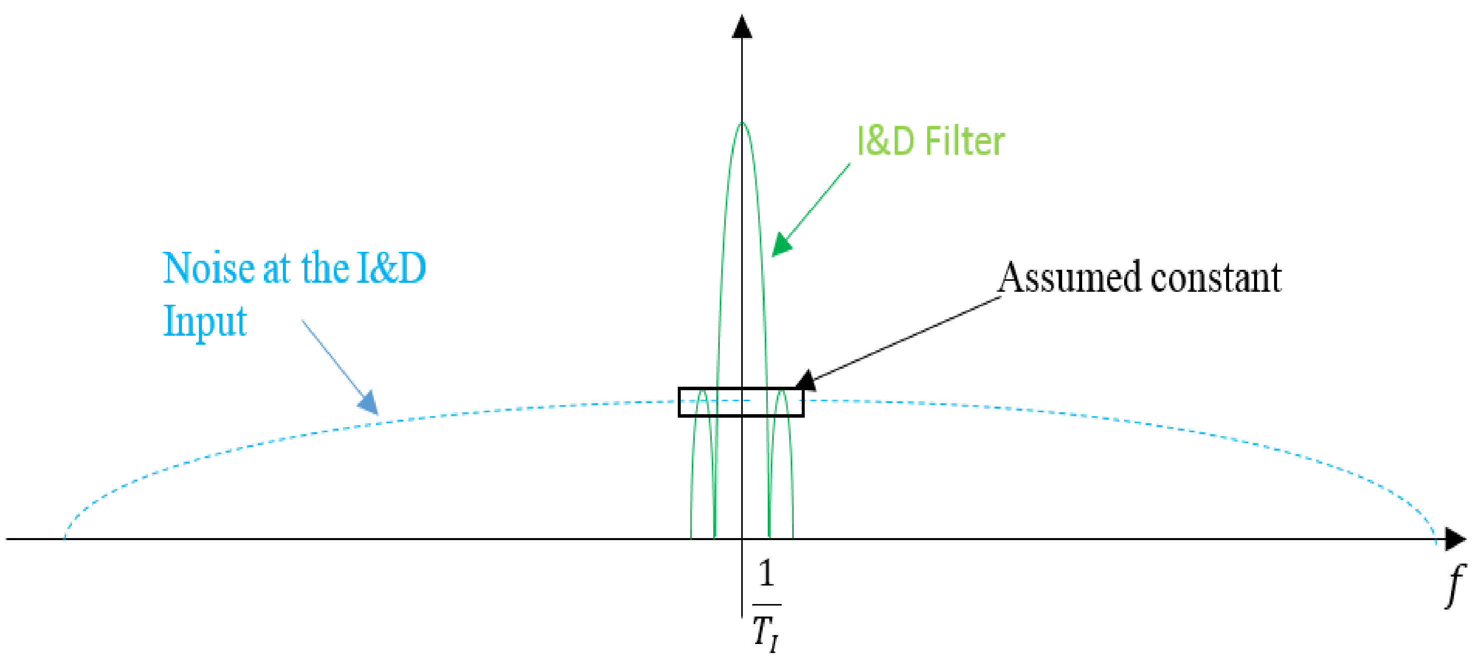
→ $S_{r_{Prod,n}}$ is significantly wider than that of $H_{I\&D}$

# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D output

- It is thus possible to assume that $S_{r_{Prod,n}}$ is constant over the bandwidth of the I&D filter

$$S_{n_I}(f) \sim |H_{I\&D}(f)|^2 S_{r_{Prod,n}}(0)$$



# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D output

- Let us thus compute $S_{r_{Prod,n}}(0)$

$$S_{r_{Prod,n}}(0)$$
$$= \frac{N_0}{8}\left(\left(\left|H_{RF_{BB}}(u-2f_0)\right|^2 + 2\left|H_{RF_{BB}}(u)\right|^2 + \left|H_{RF_{BB}}(u+2f_0)\right|^2\right) * S_{c_{m_L}}(u)\right)(0)$$
$$= \frac{N_0}{8}\int_{-\infty}^{+\infty}\left(\left|H_{RF_{BB}}(u-2f_0)\right|^2 + 2\left|H_{RF_{BB}}(u)\right|^2 + \left|H_{RF_{BB}}(u+2f_0)\right|^2\right) S_{c_{m_L}}(-u)du$$

- Since $S_{c_{m_L}}$ is located at baseband and is symmetric

$$S_{r_{Prod,n}}(0) = \frac{N_0}{4}\int_{-\infty}^{+\infty}\left|H_{RF_{BB}}(u)\right|^2 S_{c_{m_L}}(u)du$$

# 4.4 Noise Power at the Correlator Output

## Computation of the noise PSD at the I&D output

- Finally, the noise PSD at the I&D output (or correlator output) is:

$$S_{n_I}(f) = \frac{N_0}{4} |H_{I\&D}(f)|^2 \int_{-\infty}^{+\infty} \left|H_{RF_{BB}}(f)\right|^2 S_{c_{m_L}}(f) \mathrm{d}f$$

This expression represents the wide-band noise at the I&D filter that has been filtered by the low-pass narrow band I&D filter → <u>strong reduction of the noise power at the correlator output</u>

# 4.4 Noise Power at the Correlator Output

## Computation of the noise power at the correlator output

- The power of the noise at the correlator output is then obtained by integrating $S_{n_I}$ over all frequencies:

$$P_{n_I} = \int_{-\infty}^{+\infty} S_{n_I}(f)df$$

$$P_{n_I} = \frac{N_0}{4}\left(\int_{-\infty}^{+\infty} \left|H_{RF_{BB}}(f)\right|^2 S_{c_{m_L}}(f)df\right)\left(\int_{-\infty}^{+\infty} |H_{I\&D}(f)|^2 df\right)$$

# 4.4 Noise Power at the Correlator Output

## Computation of the noise power at the correlator output

- The Parseval formula provides

$$\int_{-\infty}^{+\infty} |H_{I\&D}(f)|^2 df = \int_{-\infty}^{+\infty} |h_{I\&D}(t)|^2 dt = \int_{0}^{T_I} \frac{1}{T_I^2} dt = \frac{1}{T_I}$$

- Finally, the noise power at the correlator output is:

$$P_{n_I} = \frac{N_0}{4T_I} \int_{-\infty}^{+\infty} \left|H_{RF_{BB}}(f)\right|^2 S_{c_{m_L}}(f) df$$

# 4.4 Noise Power at the Correlator Output

## Computation of the noise power at the correlator output

- Note that:

$$\int_{-\infty}^{+\infty} \left|H_{RF_{BB}}(f)\right|^2 S_{c_{m_L}}(f)df = K_{\tilde{c}_m}(0)$$

- We can thus write:

$$P_{n_I} = \frac{N_0}{4T_I} K_{\tilde{c}_m}(0) \xrightarrow[B\ is\ very\ large]{} \frac{N_0}{4T_I}$$

# 4.4 Noise Power at the Correlator Output

## Computation of the noise power at the correlator output

- Note that the noise power at the correlator output mainly depends on:
  - the noise density level at the RF front-end output,
  - the correlation duration: the longer $T_I$, the lower the noise power at the correlator output,
- Its dependence with the RF front-end filter bandwidth is very small

# 4.5 SNR at the Correlator Output

- Based on the results obtained, the SNR at the correlator output for a very large RF front-end bandwidth is:

$$SNR_{out} = \frac{P_u}{P_{n_I}} = 2T_I \frac{C}{N_0}$$

- This expression only depends upon:
  - the correlation duration → the longer $T_I$, the higher the SNR. It is thus recommended to use $T_I = T_D$ if possible
  - The <u>signal-power-to-noise-density ratio: C/$N_0$</u>

# 4.6 Correlation Gain

- The correlation gain is referred to as the ratio between the SNR at the correlator input and the SNR at the correlator output:

$$G_{corr} = \frac{SNR_{out}}{SNR_{in}} = 2T_I B$$

  – The minimum correlation duration should be chosen equal the PRN code duration $T_R$
  – the minimum RF front-end bandwidth $B$ to let the incoming signal should be at least $2f_c$.

→ the minimum correlation gain brought by the correlation operation is $G_{corr} > 4T_R f_c$.

# 4.6 Correlation Gain

- The PRN code duration can be expressed as:

$$T_R = NT_c = \frac{N}{f_c}$$

where

- $N$ is the number of chips constituting the PRN sequence
- $T_c$ is the PRN chip duration

- As a consequence,

$$G_{corr} > 4N$$

# 4.6 Correlation Gain

- A typical PRN sequence is composed of several hundreds of thousands of chips. The SNR increase due to the correlation operation is thus very large
  - For GPS L1 C/A, $N$=1023 chip, thus the minimum correlation gain is 4096, or 36 dB.
    - This gain would be even higher if the RF front-end filter was larger than $2f_c$ or if the correlation duration is greater than $T_R$. For instance, for an RF front-end filter of 20 MHz and $T_I = T_d$=20 ms, the correlation gain becomes 59 dB.

# 4.6 Correlation Gain

- Such a large correlation gain authorizes to have an incoming signal with a very small SNR at the RF front-end output.
  - In fact, DS-SS signals generally have a SNR at the RF output that is strongly negative (the noise fully dominates the useful signal) → the DS-SS signals are in fact below the thermal noise floor.
- However, this is generally not a problem at the correlator output for the users knowing the PRN sequence used thanks to the correlation operation

  → used for secret communications

# 5. Effect of an Interference on the Correlation Operation

1. Interference Model
2. Interference at the Correlator Output
3. Equivalent $N_0$

# 5.1 Interference Model

- The type of interference considered here is a continuous interference such that its PSD at the RF front-end output can be modeled as:

$$S_{I_{RF_{out}}}(f) = S_I(f)\left(\left|H_{RF_{BB}}(f)\right|^2 * \left(\delta(f-f_0)+\delta(f+f_0)\right)\right)$$

with

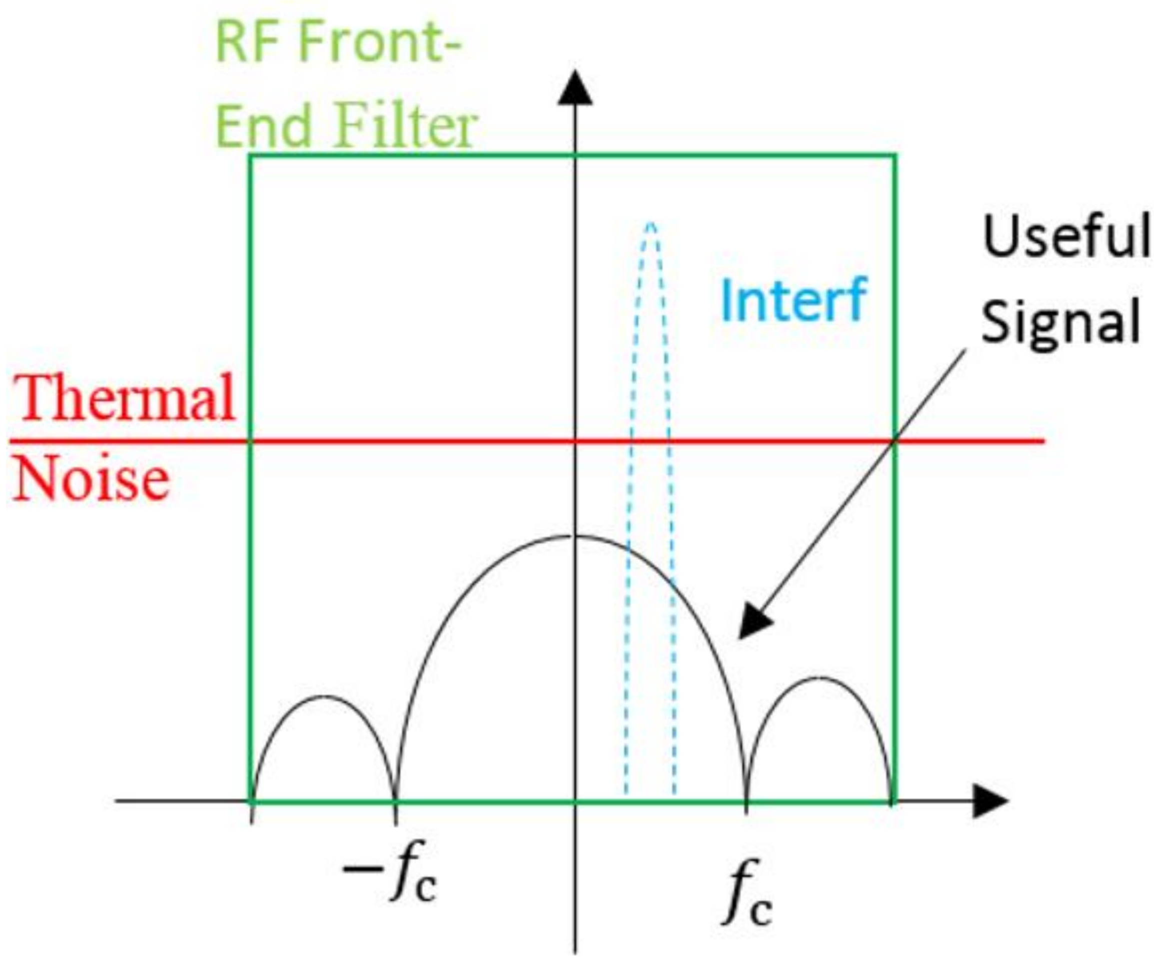
$$S_I(f) = \frac{C_I}{2} S_{I_{BB}}(f) * [\delta(f-f_0)+\delta(f+f_0)]$$

where

- $S_I$ is the interference PSD at the RF front-end input.
- $C_I$ is the power of the interference at the RF front-end output
- $S_{I_{BB}}$ is the unit power PSD of the equivalent baseband interference

# 5.1 Interference Model

- Representation at RF Front-End Output



# 5.1 Interference Model

- As a consequence, the PSD of the interference at the output of the RF front-end modeled as:

$$S_{I_{RF_{out}}}(f) = \frac{C_I}{2}\left(S_{I_{BB}}(f)\left|H_{RF_{BB}}(f)\right|^2\right) * [\delta(f-f_0) + \delta(f+f_0)]$$

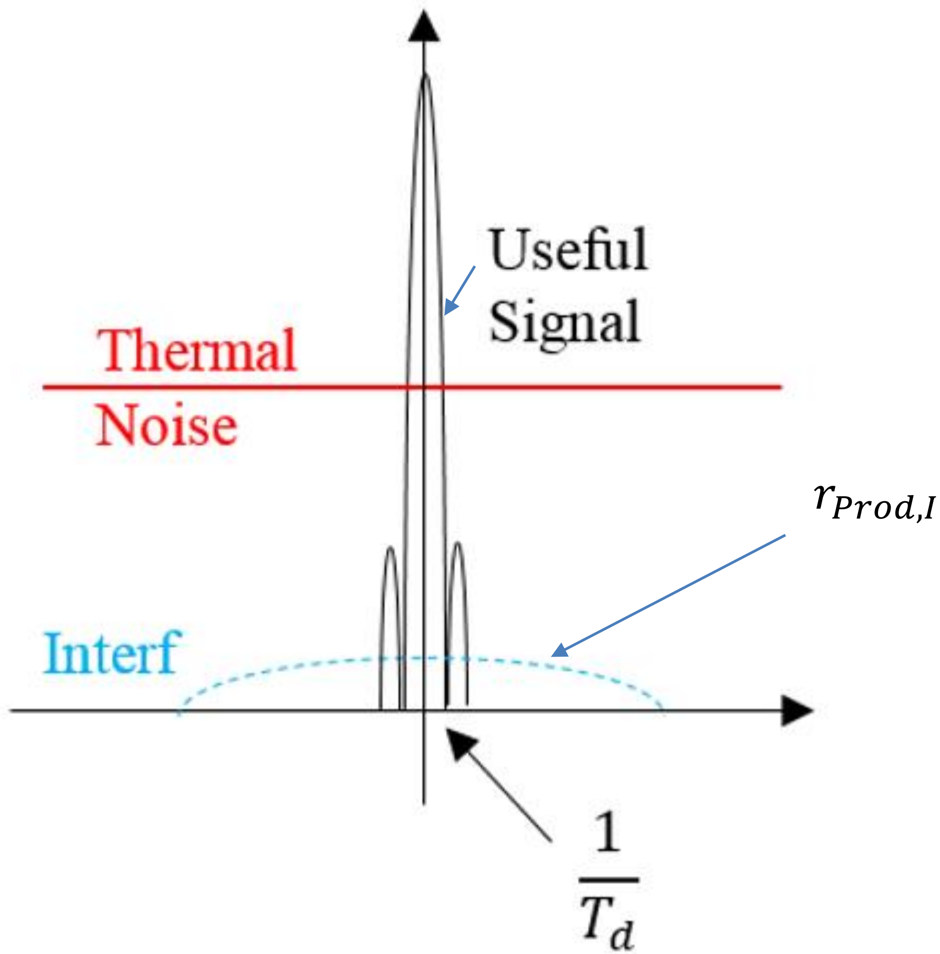
- It is now possible to reproduce exactly the same methodology as the one used for thermal noise to find the power of the interference at the correlator output

# 5.2 Interference at the Correlator Output

- The interference is multiplied by the local carrier and PRN signal replicas to provide $r_{Prod,I}$.
  - The multiplication with the PRN signal creates the spreading of the interference over the bandwidth of the PRN signal
    - This is similar to the spreading of the useful data when the DS-SS signal is transmitted, except that the interference is spread when it is received.

# 5.2 Interference at the Correlator Output

- Representation at I&D Input



# 5.2 Interference at the Correlator Output

- By analogy with the case of an incoming white noise, the PSD of the interference at the I&D input (after multiplication by the local replica) can be expressed as:

$$S_{r_{Prod,I}}(f) = \frac{C_I}{8}\left(S_{I_{BB}}(f)\left|H_{RF_{BB}}(f)\right|^2\right) * \left(\delta(f-2f_0) + 2\delta(f) + \delta(f+2f_0)\right) * S_{c_{m_L}}(f)$$

# 5.2 Interference at the Correlator Output

- It is thus possible to assume that $S_{r_{Prod,I}}$ is constant over the narrow bandwidth of the I&D filter

$$S_{n_I}(f) \sim |H_{I\&D}(f)|^2 S_{r_{Prod,I}}(0)$$

- Thanks to the multiplication by the local PRN signal, any disturbance entering the RF front-end is spread over a wide bandwidth with respect to the I&D filter

→ The effect of these disturbances is thus equivalent to the effect of a White noise with a level $I_0$ to define

# 5.2 Interference at the Correlator Output

- Finally, the noise power at the I&D output (or correlator output) is:

$$P_{n_I} = \frac{C_I}{4T_I} SSC\left(I, c_{m_L}, H_{RF_{BB}}\right)$$

where SSC is the Spectral Separation Coefficient

$$SSC\left(I, c_{m_L}, H_{RF_{BB}}\right) = \int_{-\infty}^{+\infty} \left|H_{RF_{BB}}(f)\right|^2 S_{I_{BB}}(f) S_{c_{m,L}}(f) df$$

- The SSC is representative of the spectral overlap of the interference and the local PRN signal
  - A small spectral overlap creates only a small interference

# 5.2 Interference at the Correlator Output

- By analogy with the power of the thermal noise at the correlator output, it is then possible to consider that an interference has an impact at the correlator output that is equivalent to that of a White noise with:

$$I_0 = C_I \times SSC\left(I, c_{m_L}, H_{RF_{BB}}\right)$$

# 5.3 Equivalent $N_0$

- In presence of thermal noise and interference, it is possible to consider that the impact of these disturbances at the correlator output is equivalent to that of a single White noise with a constant PSD equal to:

$$S_{disturb}(f) = \frac{N_{0,eq}}{2} = \frac{N_0}{2} + \frac{I_0}{2}\ W/Hz$$

- We then speak of an equivalent $N_0$ noted $N_{0,eq}$

# 5.3 Equivalent $N_0$

- As a consequence, it makes sense in DS-SS to only refer to equivalent White noise to assess the signal reception quality since all the processing occurs at the correlator output
  - The measure of reception quality is then generally given through the $C/N_{0,eq}$ rather than the SNR.
    - It is then receiver independent
  - The link between the SNR and the $C/N_0$ was given earlier (it includes the knowledge of $T_I$):

$$SNR_{out} = 2T_I \frac{C}{N_0}$$

# 6. ACQUISITION OF DS-SS SIGNALS

1. Introduction
2. Acquisition Detector
3. Acquisition strategy
4. Improved Detector
5. Acquisition Performance
6. Conclusions

# 6.1 Introduction

- The goal of the acquisition process is to:
  - Detect the presence of the useful signal
  - Give a rough estimate of its main parameters in order to start the tracking process: the code delay and the Doppler frequency

# 6.1 Introduction

- The usual way to detect the presence of a signal is to compute a detection criterion to be compared against a threshold.
  - The detection criterion should be designed to be well above the threshold when the signal is present.

- It has been seen that the use of the correlation operation could allow the detection of a GNSS signal.

# 6.1 Introduction

- Each DS-SS useful signal is taken care of separately, as each signal requires a correlation with its own spreading code locally generated
  - This processing unit dedicated to a specific DS-SS signal is referred to as a receiver channel

# 6.2 Acquisition Detector

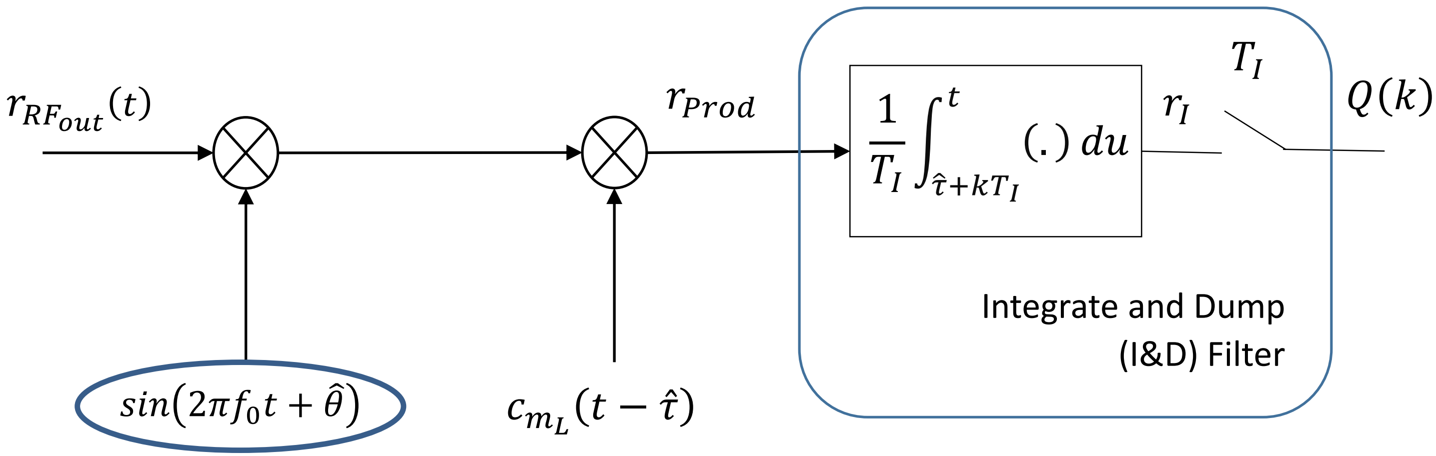
- It has been seen that the correlator output can be modelled as:

$$I(k) \sim \frac{A}{2} d(k) K_{\tilde{c}_m, c_m}(\varepsilon_\tau) sinc\left(\pi \varepsilon_f T_I\right) cos(\varepsilon_\theta) + n_I(k)$$

- If the uncertainty on the Doppler, code delay and phase is small, then I gathers all the signal power
  - However, it is difficult to ensure phase synchronization during the entire acquisition stage

# 6.2 Acquisition Detector

- Another correlator is typically used




Local carrier is in quadrature phase

- A derivation similar to what was done to obtain I provides the following model:

$$Q(k)\sim\frac{A}{2}d(k)K_{\tilde{c}_m,c_m}(\varepsilon_\tau)sinc\left(\pi\varepsilon_f T_I\right)cos(\varepsilon_\theta)+n_Q(k)$$

# 6.2 Acquisition Detector

- Terminology:
  - I is called <u>in-phase</u> correlator output
  - Q is called <u>quadrature-phase</u> correlator output

- I and Q have a very similar expression. The only difference is:
  - Relation to the phase error that is in quadrature
  - $n_I$ and $n_Q$ are both Gaussian, have the same power, but are independent

# 6.2 Acquisition Detector

- To gather the whole signal energy, the following detector is typically used

$$T(k) = I^2(k) + Q^2(k) \xrightarrow[Low\ Noise]{} \frac{A^2}{4} K^2_{\tilde{c}_m, c_m}(\varepsilon_\tau) sinc^2\left(\pi \varepsilon_f T_I\right)$$

→This detector depends upon

→the Doppler

→code delay errors

→It does not depend upon

→the data bit value

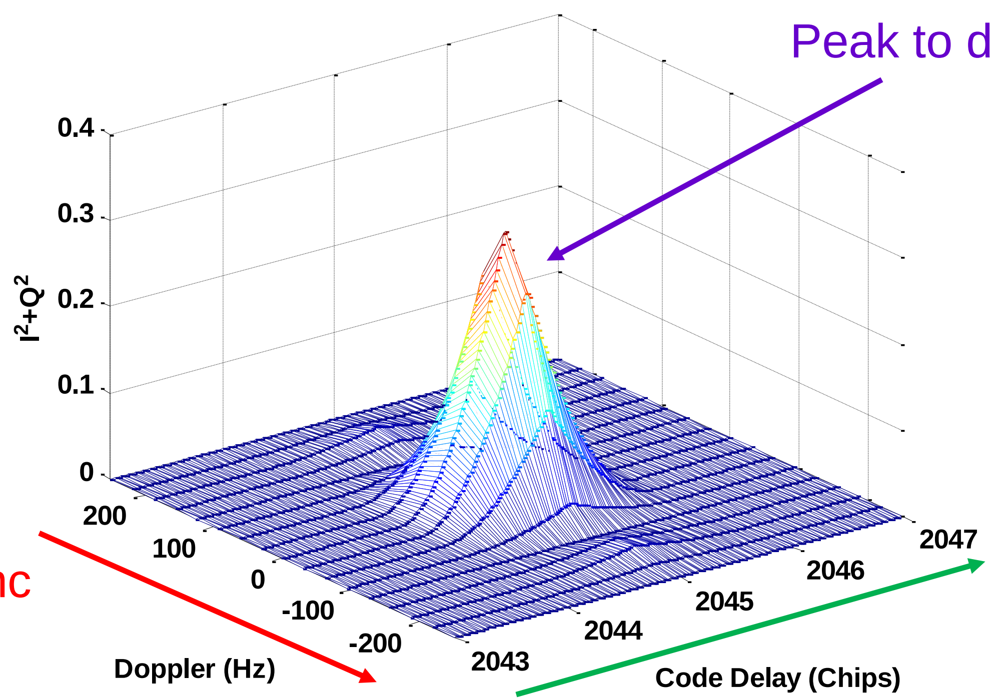
→the carrier phase error

# 6.2 Acquisition Detector

## Example: Rectangular shaping

The detector is maximized only when both the Doppler and the code delay are roughly estimated

Peak to detect and track




Impact of the sinc function → the longer $T_I$, the narrower the sinc

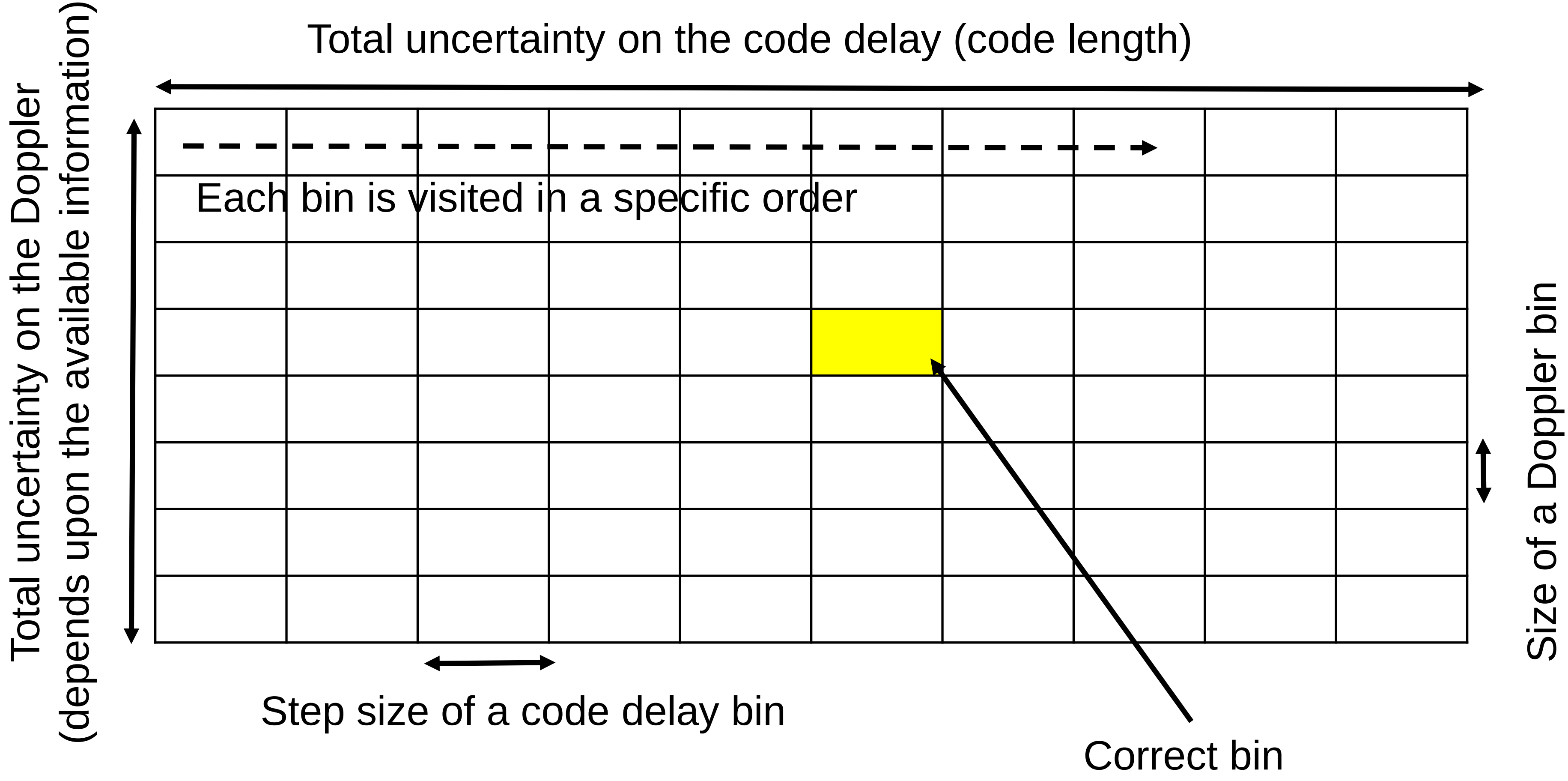
Impact of the correlation function → low side-lobes are preferable

# 6.3 Acquisition Strategy

- The detector is maximized when the incoming and local Doppler and code delay are very close
  - The traditional acquisition technique consists in computing the detector with local replicas that are based on all the Doppler and code delay that the incoming signal can have.
  - Detection should occur when the ‘right’ couple (Doppler, code delay) has been used to generate the local replica

# 6.3 Acquisition Strategy

## Acquisition Matrix



# 6.3 Acquisition Strategy

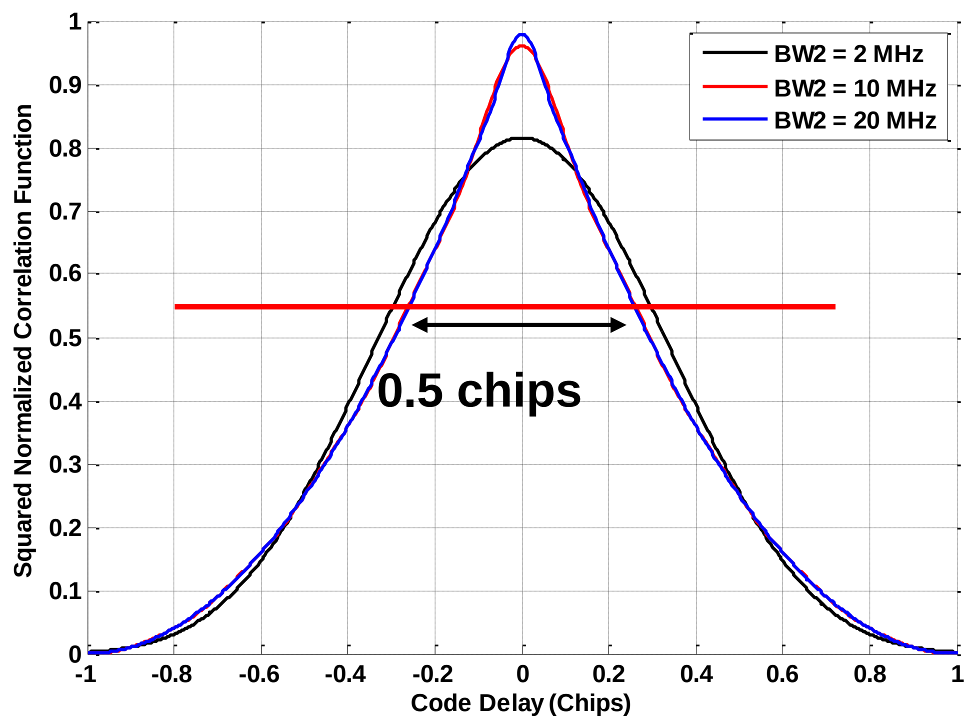
- It is then necessary to define the step size of the bins for both the code and Doppler bins
  - It is important to try to minimize the number of bins to visit in order to limit the acquisition time
  - It is however more important to make sure that the step size of the bins is small enough not to miss the detection of the signal

# 6.3 Acquisition Strategy

## Code Delay Uncertainty

For a code delay bin of size $S_{bin}$, the worst case scenario gives a degradation of the detector of $K^2_{\tilde{c}_m,c_m}(S_{bin}/2)$

For a rectangular shaping, the usual code bin has a width of 0.5 chips, which means a potential degradation of 2.5 dBs



# 6.3 Acquisition Strategy

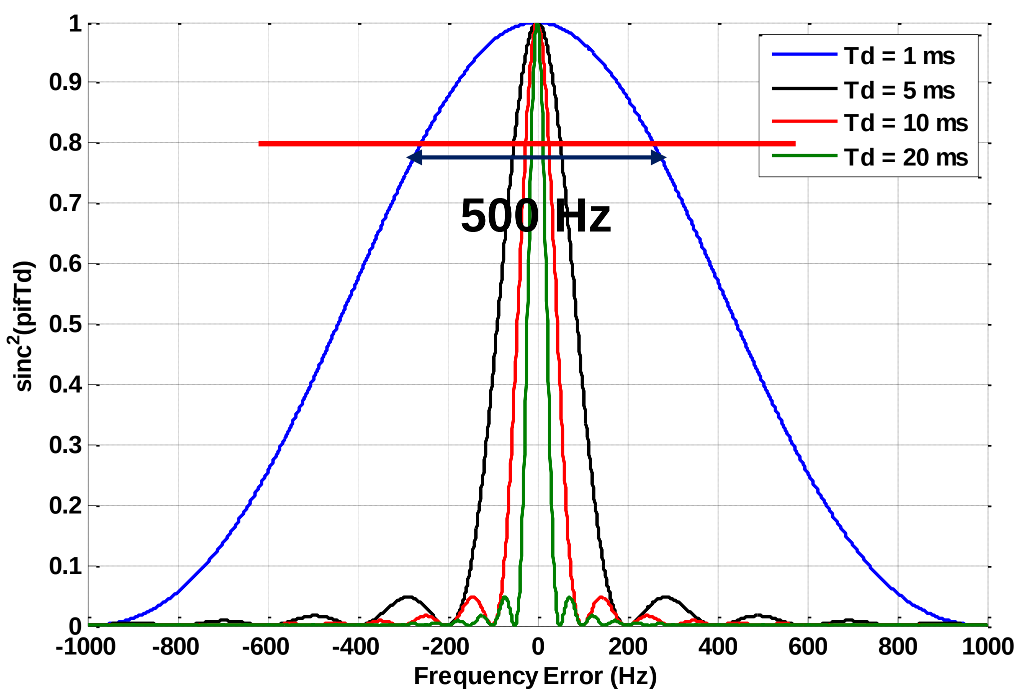
## Doppler Uncertainty

For a Doppler bin of size $S_{bin}$, the worst case scenario gives a degradation of the received signal power at the correlator output of $sinc^2(\pi S_{bin} T_I/2)$

The 'sharpness' of the sinc function depends upon the integration time $T_I$

A rule-of-thumb is to use a Doppler bin of $1/2T_I$ or $2/3T_I$

This leads to a worst case degradation of 0.4-0.9 dBs



# 6.4 Improved Detector

## Increasing the Detection Capability

- Increasing the integration time
  - results in a proportional increase of the correlator output SNR → the peak will be more visible above the noise
  - However, there are 2 limitations:
    - The integration should not be done across a data bit transition.
      - This is difficult since the receiver does not know the location of this transition at this stage
    - Increasing the integration time requires a narrower Doppler bin size (the sinc function is narrower). This will increase the number of bins to visit

# 6.4 Improved Detector

## Increasing the Detection Capability

- Using several consecutive correlator outputs:
  - This allows gather more energy over time
  - The associated acquisition detector is

$$T(k) = \sum_{q=1}^{M} \left( I^2(q) + Q^2(q) \right)$$

where

- All $I^2$ and $Q^2$ are computed with a local replica using the same couple (Doppler, code delay)
- M is referred to as the number of non-coherent summations. The time spent on one bin is called dwell time ($MT_I$)

# 6.4 Improved Detector

## Increasing the Detection Capability

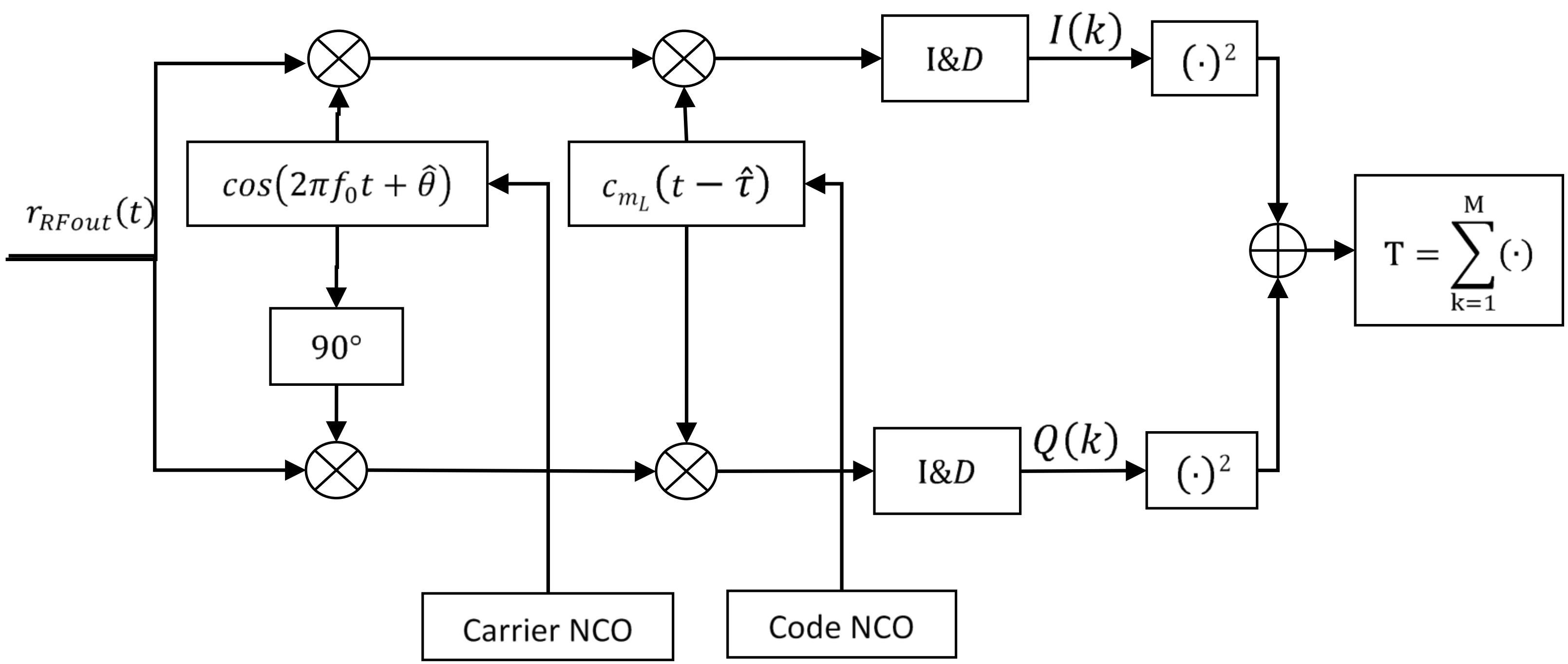
- Using several non-coherent summations:
  - The new detector also implies squaring losses.

$$\begin{cases} I = A\cos(\varphi) + n_I \\ Q = A\sin(\varphi) + n_Q \end{cases} \Rightarrow I^2 + Q^2 = A^2 + \underbrace{A\left(\cos(\varphi)n_I + \sin(\varphi)n_Q\right) + n_I^2 + n_Q^2}_{\text{Extra Noise}}$$

  - This means that the SNR will not increase as fast as for a longer coherent integration
  - In particular, low SNR will lead to a very limited gain

# 6.4 Improved Detector

## Acquisition Architecture



# 6.5 Acquisition Performance

- The sensitivity performance of the acquisition is based on the success of the comparison of the acquisition detector with a predefined threshold
- The acquisition performance is generally based on 2 parameters:
  - Detection probability: what is the probability to acquire a given signal
    - Can be turned into acquisition sensitivity: at which signal strength can I acquire a given signal with a given probability
  - Time to acquire the signal: for practical reasons

# 6.5 Acquisition Performance

- The theoretical performance is based on the following hypothesis test (Neyman-Pearson test)
  - Hypothesis $H_0$: the useful signal is not present → there is only noise or the detector is on the wrong bin
    - What should be the value of the threshold in order to avoid false acquisition with a given Probability of False Alarm ($P_{fa}$)?
  - Hypothesis $H_1$: The useful signal is present (detector is on the right bin)
    - Using the threshold defined with Hypothesis $H_0$, what is the probability of detection of the incoming signal?

# 6.5 Acquisition Performance

- Hypothesis $H_0$
  - $H_0$: The useful signal is not present. The detector is then:

$$T_0 = \sum_{k=1}^{M} \left( n_i^2(k) + n_q^2(k) \right)$$

  - This detector can be normalized by the correlator output noise variance:

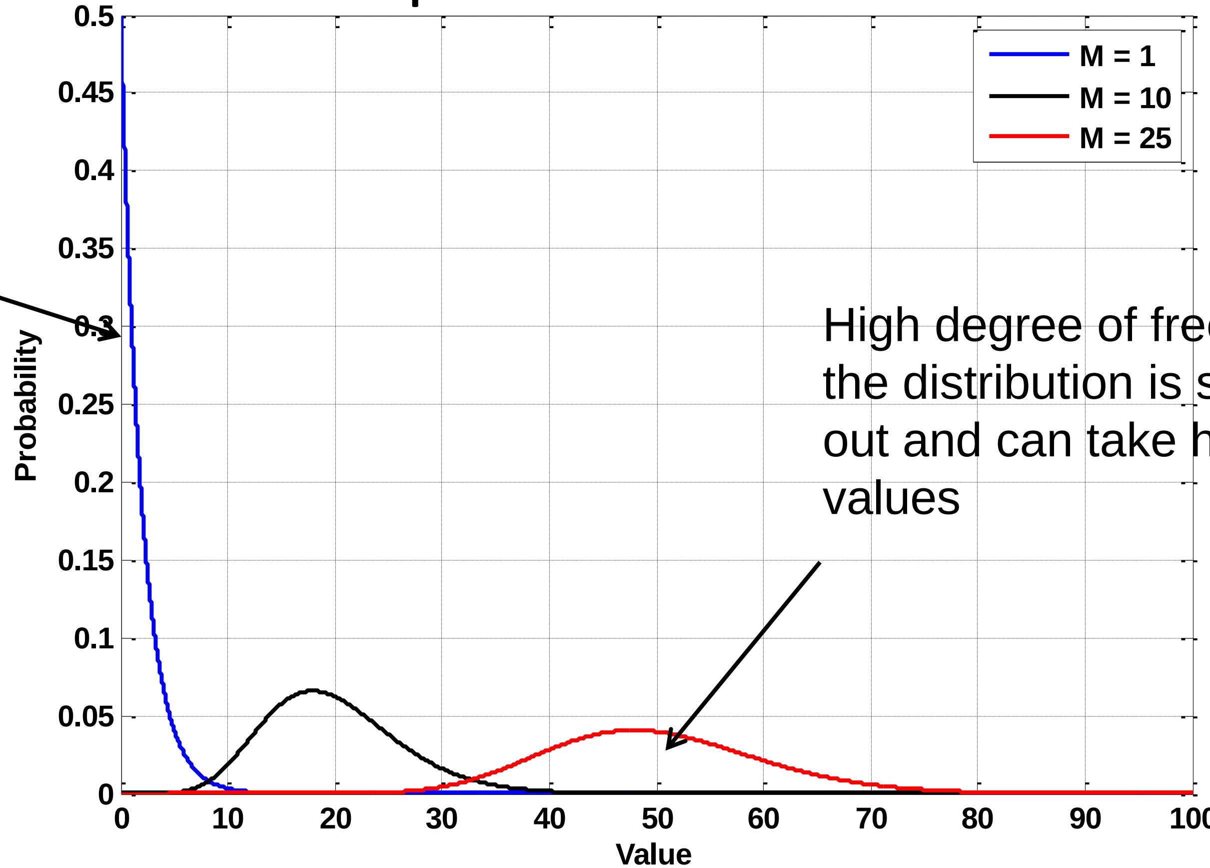
$$\bar{T}_0 = \frac{T_0}{P_{n_i}} = \sum_{k=1}^{M} \left( \bar{n}_i^2(k) + \bar{n}_q^2(k) \right)$$

→ The sum of N normalized independent squared Gaussian random variables has a Chi-square distribution with N degrees of freedom → 2M degrees of freedom

# 6.5 Acquisition Performance

- Hypothesis $H_0$
  - Overview of a Chi-square distribution

Low degree of freedom, the distribution is dense and has mostly small values

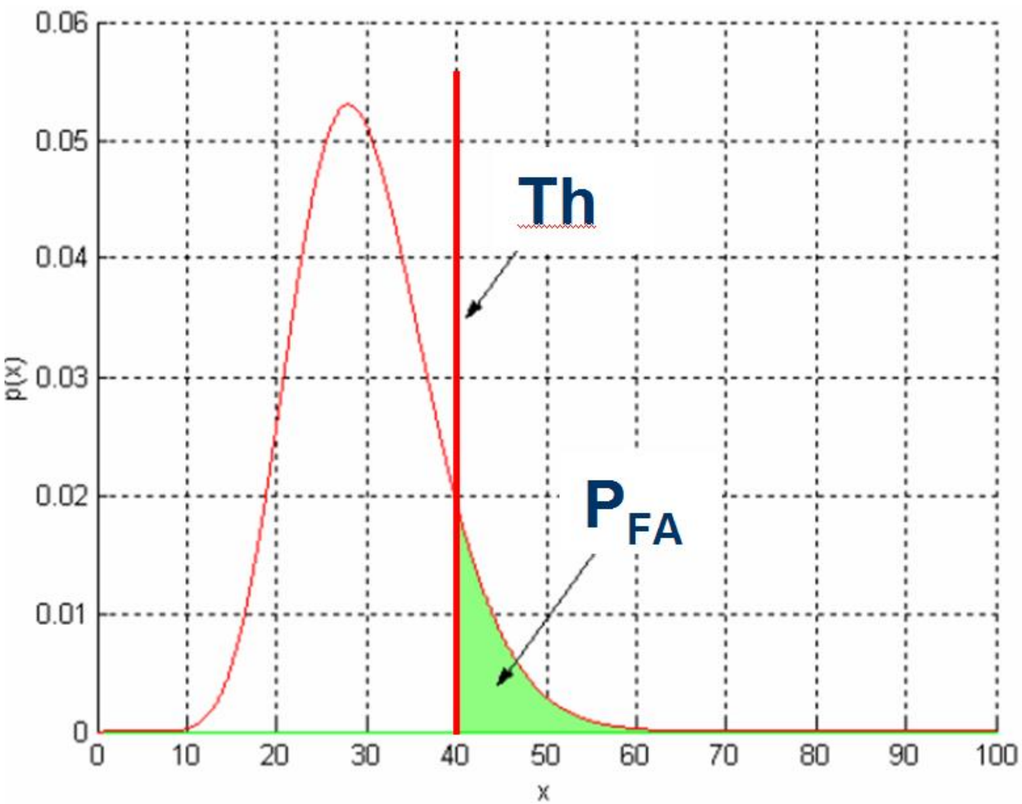



High degree of freedom, the distribution is spread out and can take high values

# 6.5 Acquisition Performance

- Hypothesis $H_0$

For a given probability of false alarm $P_{FA}$, it is easy to




$$P\left(\bar{T}_0 > \overline{Th}\right) = P_{fa}$$

$$\Downarrow$$

$$P\left(\bar{T}_0 > \overline{Th}\right) = P\left(T_0 > \underbrace{P_{n_i} \times \overline{Th}}_{Th}\right) = P_{fa}$$

Value of the actual test threshold

# 6.5 Acquisition Performance

- Hypothesis $H_1$
  - $H_1$: The useful signal is present. The detector becomes:

$$T_1 = \sum_{k=1}^{M} \left( \begin{array}{c} \left( \frac{A}{2} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \operatorname{sinc}\left(\pi \varepsilon_f T_I\right) \cos(\varepsilon_\theta) + n_i(k) \right)^2 \\ + \left( \frac{A}{2} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \operatorname{sinc}\left(\pi \varepsilon_f T_I\right) \sin(\varepsilon_\theta) + n_q(k) \right)^2 \end{array} \right)$$

  - After normalization by the correlator noise power:

$$\bar{T}_1 = \frac{T_1}{P_{n_i}} = \sum_{k=1}^{M} \left( \begin{array}{c} \left( A \sqrt{\frac{T_I}{N_0}} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \operatorname{sinc}\left(\pi \varepsilon_f T_I\right) \cos(\varepsilon_\theta) + \bar{n}_i\ (k) \right)^2 \\ + \left( A \sqrt{\frac{T_I}{N_0}} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \operatorname{sinc}\left(\pi \varepsilon_f T_I\right) \sin(\varepsilon_\theta) + \bar{n}_q\ (k) \right)^2 \end{array} \right)$$

# 6.5 Acquisition Performance

- Test Hypothesis $H_1$
  - As seen previously, the signal amplitude is related to the signal power via:

$$C = \frac{A^2}{2}$$

  - Thus:

$$\bar{T}_1 = \sum_{k=1}^{M} \left( \left( \sqrt{2\frac{CT_I}{N_0}} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \operatorname{sinc}\left(\pi \varepsilon_f T_I\right) \cos(\varepsilon_\theta) + \bar{n}_i\ (k) \right)^2 + \left( \sqrt{2\frac{CT_I}{N_0}} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \operatorname{sinc}\left(\pi \varepsilon_f T_I\right) \sin(\varepsilon_\theta) + \bar{n}_q\ (k) \right)^2 \right)$$

# 6.5 Acquisition Performance

- Test Hypothesis $H_1$
  - The non-centrality parameter, assuming that the code delay and Doppler errors remain constant, equals:

$$\lambda = 2\frac{C}{N_0} T_I d(k) K^2_{\tilde{c}_m, c_m}(\varepsilon_\tau)\,\mathrm{sinc}^2\left(\pi \varepsilon_f T_I\right)$$

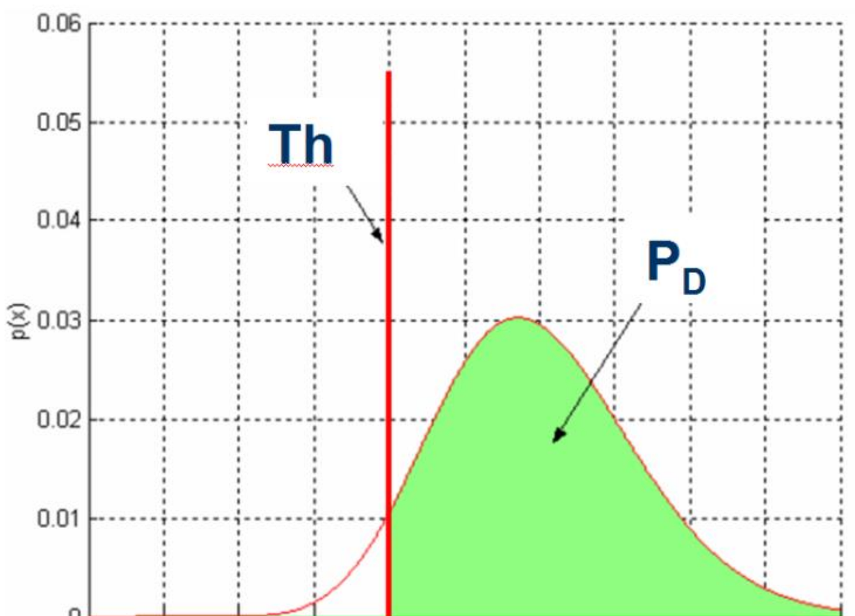
→The non-centrality parameter depends upon

- the signal $C/N_0$,
- the number of non-coherent summation,
- the correlation time, and
- the Doppler and code delay errors

# 6.5 Acquisition Performance

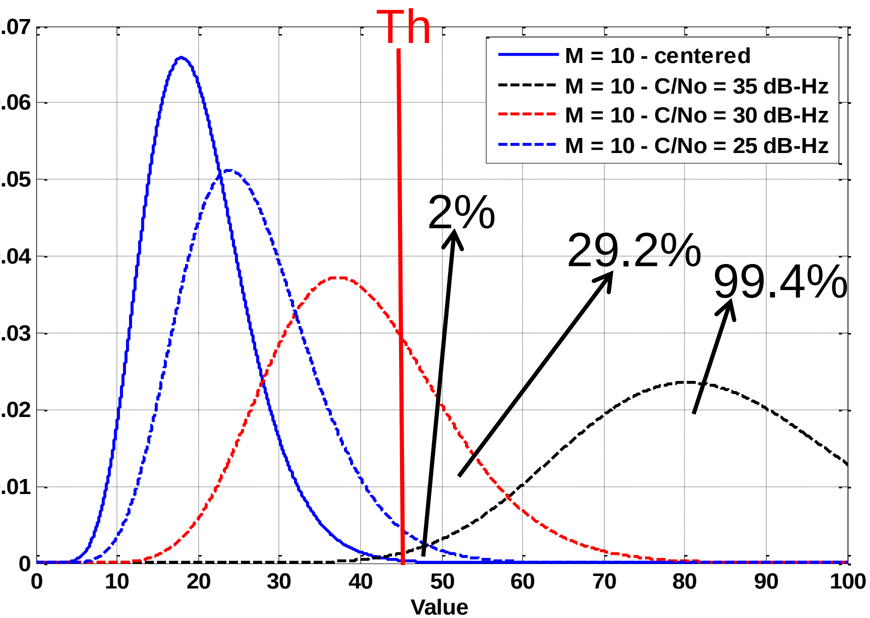
- Test Hypothesis $H_1$
  - Using the previously defined threshold Th it is easy to find the value of $P_d$ for a given $C/N_0$:

$$P_d = P(\overline{T}_1 > \overline{Th}) = P\left(T_1 > \underbrace{P_{n_i} \times \overline{Th}}_{Th}\right)$$



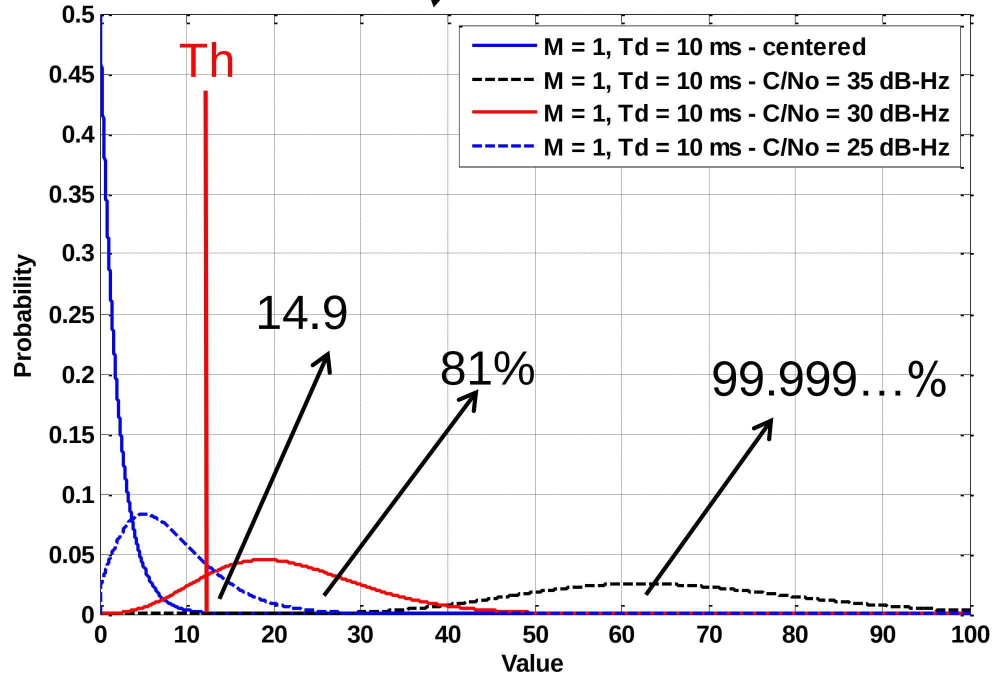
# 6.5 Acquisition Performance

- Example of Sensitivity Performance



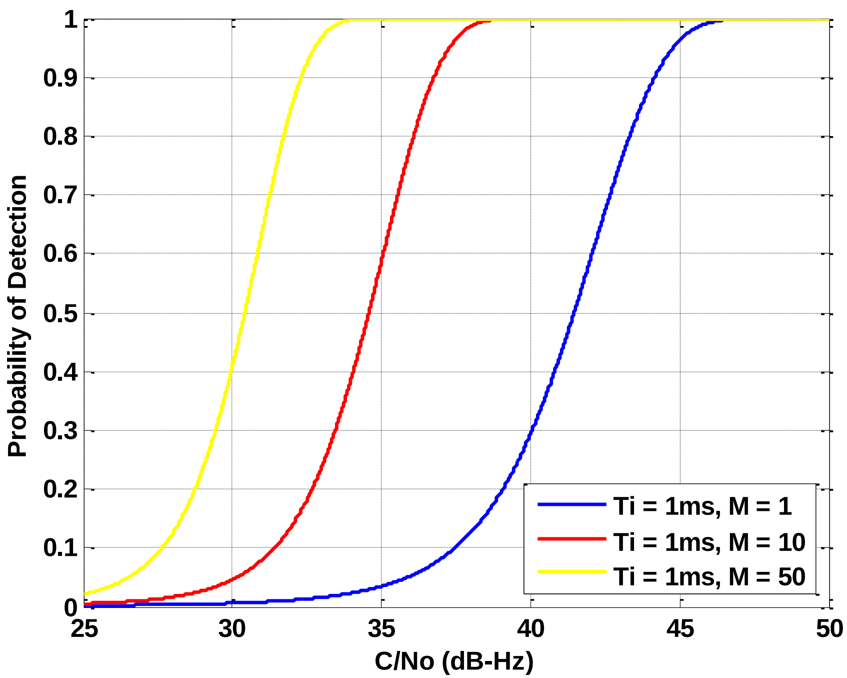

$T_D$ = 1 ms
M = 10
$C/N_0$ = 25, 30, 35 dB-Hz
No uncertainty considered

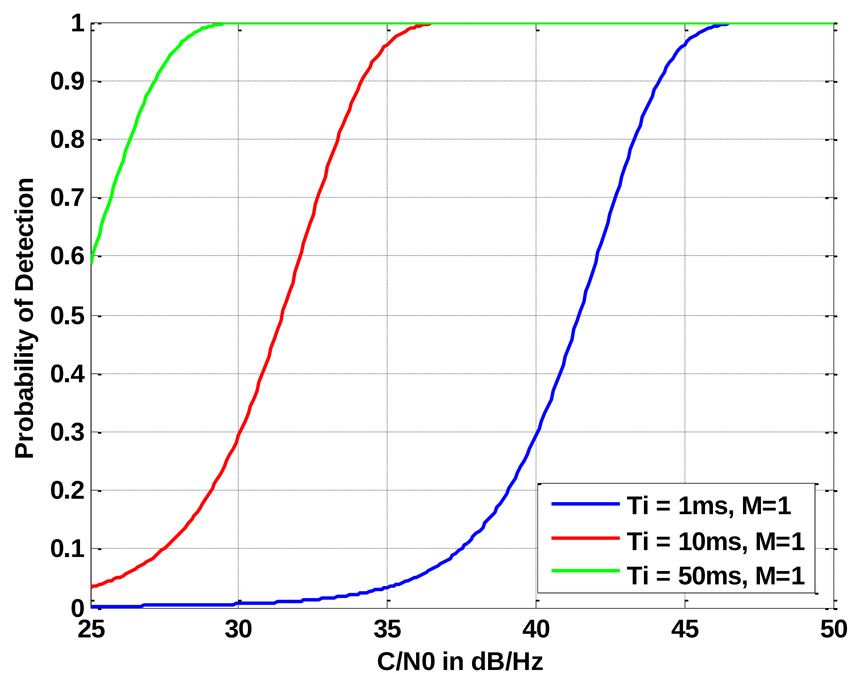
$T_D$ = 10 ms
M = 1
$C/N_0$ = 25, 30, 35 dB-Hz
No uncertainty considered



# 6.5 Acquisition Performance

Probability of detection as a function of the incoming $C/N_0$




3.4 dBs worst case losses assumed $P_{FA} = 1e^{-3}$

# 6.5 Acquisition Performance

- Mean Time-To-Acquire (MTTA)
  - It is also fundamental to know how long it will take to acquire the signal for a signal at a given $C/N_0$, since a too long acquisition time might not be practical
  - The MTTA will depend upon the acquisition strategy:
    - Sequential vs parallel: bins are visited one after the other vs several bins are visited at the same time
    - Single dwell vs multiple dwell: decision is taken after one test or after several tests (for confirmation)

# 6.5 Acquisition Performance

- Mean Time-To-Acquire (MTTA)
  - For a sequential single dwell acquisition strategy, the MTTA 1 satellite can be written as :

$$T_{acq} = \frac{2 + (2 - P_d)(q-1)\left(1 + KP_{fa}\right)}{2P_d} MT_I \ (sec)$$

where

- $q$ represents the number of bins to visit
- $K$ represents the penalty factor caused by a false alarm (could be a second test or a failed initiation of the tracking loops – it is usually taken around 1 sec)

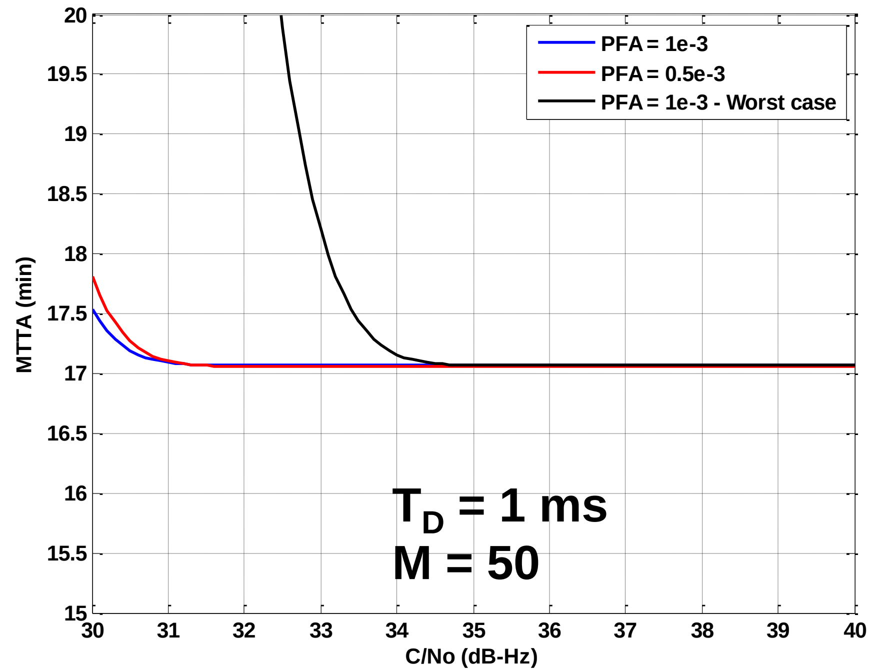
# 6.5 Acquisition Performance

- Mean Time-To-Acquire (MTTA)
  - 1023 chips to search with 0.5 chip step = 2046 bins
  - 10 kHz to search with 500 Hz step = 20 bins

q=40920 search bins

$$\bar{T} \xrightarrow[\substack{P_D \to 1 \\ P_{FA} << 1 \\ q >> 1}]{} \approx \frac{q}{2} M T_I \,(\text{sec})$$

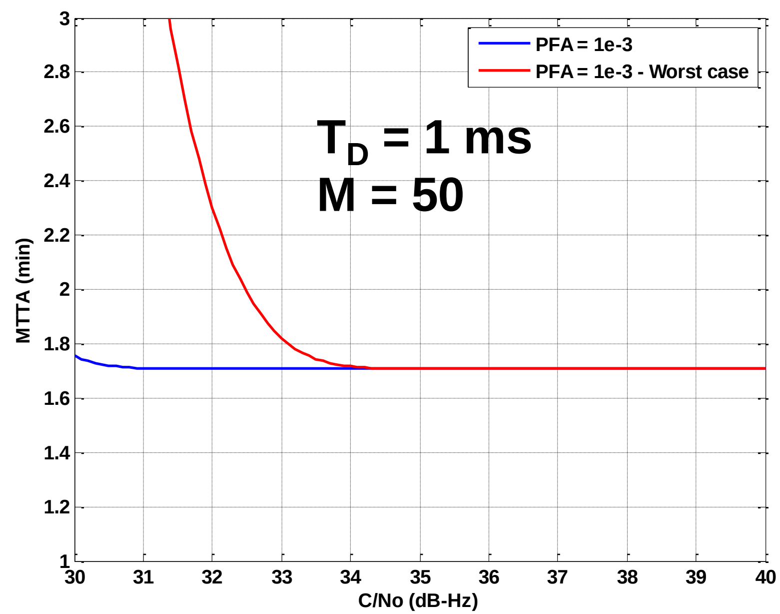
With large uncertainties and sequential search, it will take a long time to acquire the first satellite.



# 6.5 Acquisition Performance

- Mean Time-To-Acquire (MTTA)
  - Assuming a Doppler uncertainty of 1 KHz → 3 Doppler bins to search

The MTTA is reduced by 10. Using paralle search (many correlators are used in parallel), it is easy to reach sub-min acquisition.



# 6.6 Conclusion

- The output of the acquisition block provides:
  - An estimation of the code delay:
    - The uncertainty depends upon the size of the code bin. The size of the code bin depends upon the size of the correlation peak, and thus on the modulation used by the signal
  - An estimation of the Doppler frequency:
    - the uncertainty depends only upon the coherent integration time. Usually, the integration time is chosen equal to the duration of one code period.
- The acquisition performance will depend upon
  - The detection performance
  - The MTTA

# 7. DS-SS Carrier Phase Tracking

# 7.1 Introduction

- It is required to track the phase of the incoming signal. Once the phase is tracked, then:

$$I(k) = \frac{A}{2} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \cos(\varepsilon_\theta) + n_I(k) \xrightarrow[\substack{\varepsilon_\theta \sim 0 \\ \varepsilon_\tau \sim 0}]{} \frac{A}{2} d(k) + n_I(k)$$

  - Phase tracking is thus necessary to be able to decode the navigation message

- Note that is phase tracking is successful, the Q correlator output is only composed of noise:

$$Q(k) = \frac{A}{2} d(k) K_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau) \sin(\varepsilon_\theta) + n_Q(k) \xrightarrow[\varepsilon_\theta \sim 0]{} n_Q(k)$$

# 7.1 Introduction

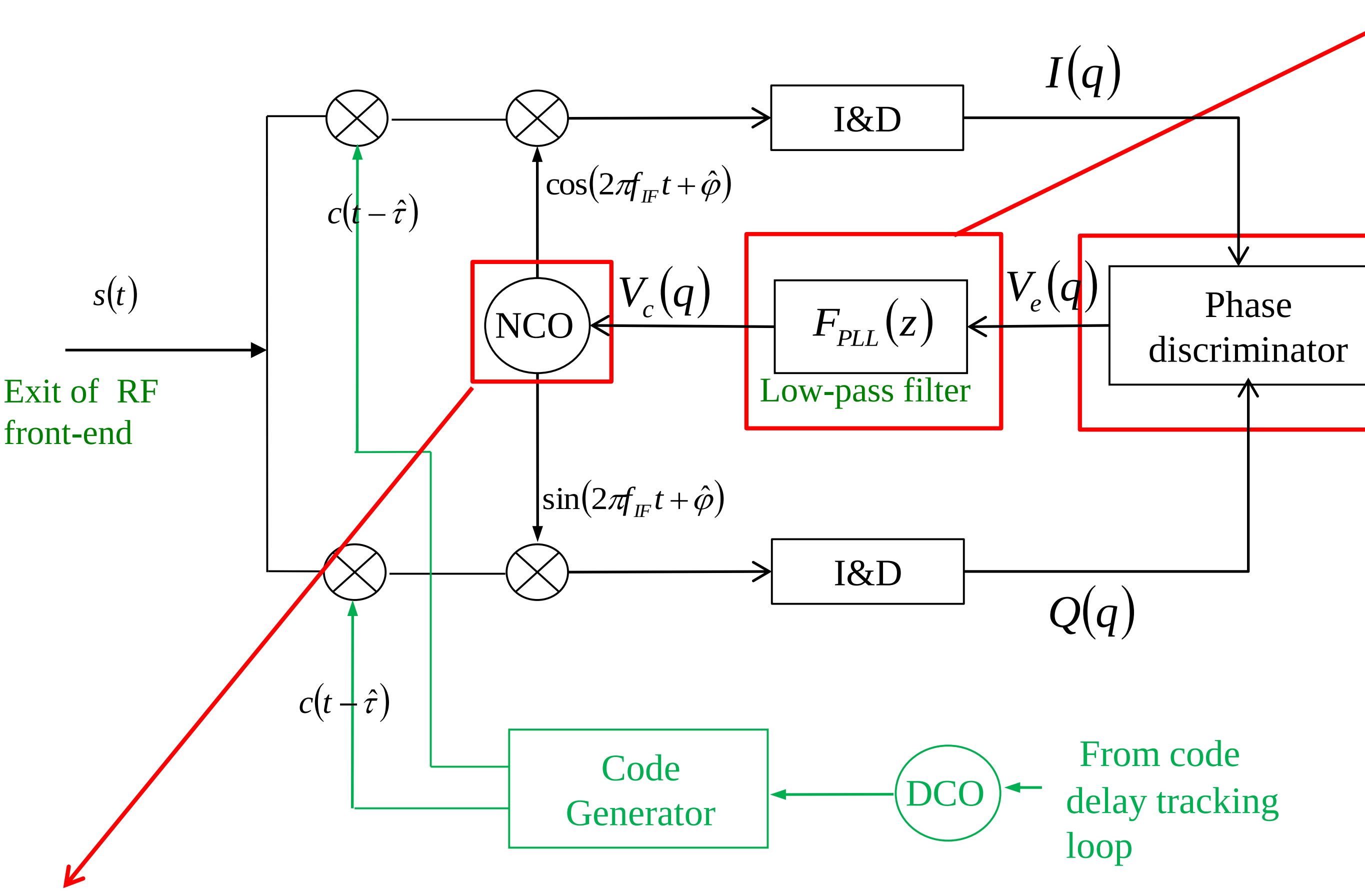
- The goal of a Phase Lock Loop (PLL) is to generate a local carrier whose phase is perfectly synchronized with the carrier phase of the incoming signal
  - The loop is thus said to track the incoming signal carrier phase

- A pure PLL is designed to track the carrier phase of a pure carrier (not modulated by an unknown useful data stream).
  - A modified version of the PLL is generally used to track a carrier modulated by data: Costas PLL, or Costas loop.

# 7.1 Introduction

- The basic blocks of a PLL used by DS-SS receivers are:
  - Controlable local carrier generator in the form of a Numerically-Controlled Oscillator (NCO), which is able to generate a local replica with a specific frequency.
  - Correlators to have access to correlator outputs with confortable SNR
  - Discriminator, which is an estimator of the synchronisation error between the local carrier and the incoming signal carrier
  - Filter to refine the discriminator output and control the NCO

# 7.2 PLL General Architecture




The filter is a low-pass filter that sets the characteristics of the loop:
- Loop response
- noise filtering

The phase discriminator is meant to provide a first estimate of the current phase error

Generates a local carrier with a frequency controlled by the output of the low-pass filter (or a phase which is the integral of its input)

# 7.2 PLL General Architecture

- Note that within the loop structure:
  - some signals (incoming signal, local replicas) are available at the sampling rate (every sampling period $T_s$),
  - other signals (correlator outputs, discriminator outputs, low-pass filter outputs) are available at the correlation rate (every correlator output period $T_I$).

# 7.3 Phase Discriminator

- A phase discriminator is a function that estimates the carrier phase difference between the incoming and local carrier, which is also referred to as phase tracking error.

- In the case of a DS-SS signal, the phase discriminator uses the correlator outputs as input.

# 7.3 Phase Discriminator

- Typical phase discriminators insensitive to data bits
  - Product → <u>used as an example in the following</u>

$$D_P = I(k)\times Q(k) = \frac{A^2}{8} K^2_{\tilde{c}_m,c_m}(\varepsilon_\tau)\sin(2\varepsilon_\theta) \underset{\varepsilon_\theta \to 0}{\to} \frac{A^2}{4} K^2_{\tilde{c}_m,c_m}(\varepsilon_\tau)\cdot\varepsilon_\theta$$

This discriminator requires a normalization to provide the phase tracking error

  - Arctangent

$$D_{\text{Atan}} = \text{Atan}\left(\frac{Q(k)}{I(k)}\right) = \text{Atan}\left(\frac{\sin(\varepsilon_\theta)}{\cos(\varepsilon_\theta)}\right) = \varepsilon_\theta$$

This discriminator does not require a normalization, but the receiver needs to implement an atan function

# 7.3 Phase Discriminator

- Using the correlator output model given earlier:

$$V_d(k) = \frac{A^2}{8} K^2_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau)\sin(2\varepsilon_\theta) + n_d(k)$$

where

- $n_d$ is the resulting noise at the discriminator output
- $\varepsilon_\theta$, as a reminder, represents the difference between the local and incoming carrier phases in the middle of the correlation interval.

# 7.3 Phase Discriminator

- $V_d$ is independent upon the useful data bit value (Costas loop!) but is dependent upon the incoming signal amplitude.
  - Normalization factor shall be used to have access to $\varepsilon_\theta$

For instance: $C_{inst}(k) = I^2(k) + Q^2(k) \xrightarrow[low\ Noise]{} \frac{A^2}{4} sinc^2\left(\pi\varepsilon_f T_I\right) K^2_{\tilde{c}_m, c_{m_L}}(\varepsilon_\tau)$

- Assuming low noise, then the normalized phase discriminator is:

$$\overline{V_d}(k) = \frac{V_d(k)}{I^2(k) + Q^2(k)} \sim \frac{1}{2}\sin(2\varepsilon_\theta) + \overline{n_d}(k)$$

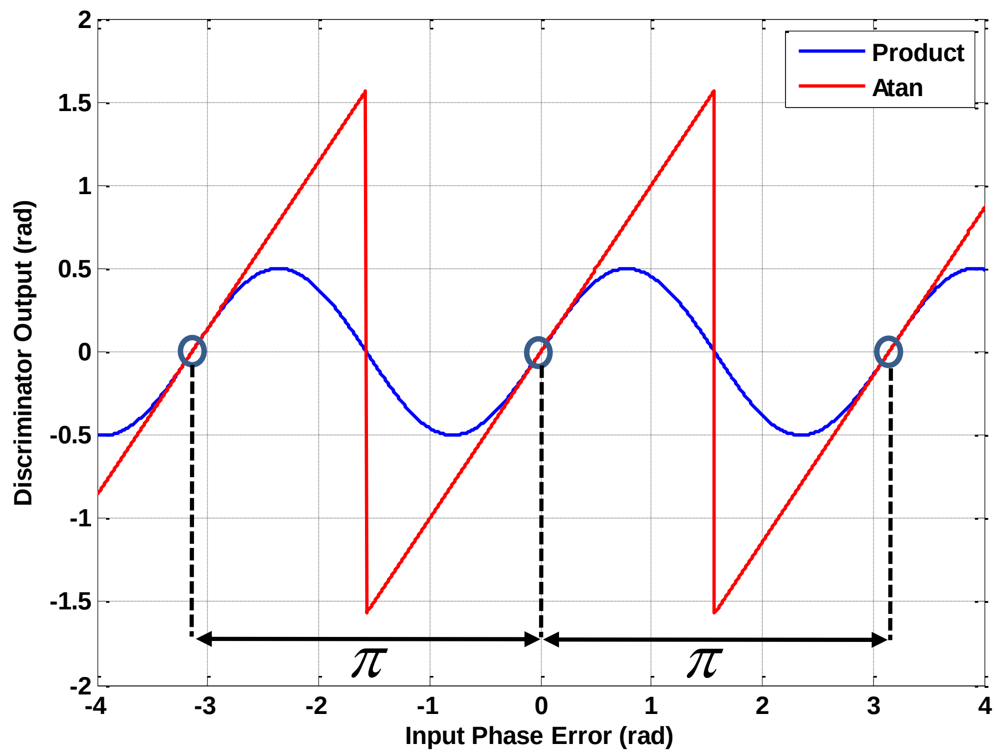
where $\overline{n_d}$ is the normalized discriminator noise

# 7.3 Phase Discriminator

- A stable lock point is a point s that is such that:

$$\begin{cases} \varepsilon_\varphi > 0 \Rightarrow D(s+\varepsilon_\varphi) > 0 \\ \varepsilon_\varphi < 0 \Rightarrow D(s+\varepsilon_\varphi) < 0 \end{cases}$$

The 2 considered discriminators have an infinite number of stable lock points separated by $\pi$.
→ This means that phase tracking is ambiguous



# 7.3 Phase Discriminator

- Because the discriminator is ambiguous, it means that there might be a phase tracking error of $k\pi$
  - Consequently, the data demodulated from the I channel might have the wrong sign

$$I(k) = \frac{A}{2} d(k) K_{c_m}(\varepsilon_\tau) \, cos(\varepsilon_\theta + \pi) + n_I(k) \xrightarrow[\substack{\varepsilon_\theta \sim 0 \\ \varepsilon_\tau \sim 0}]{} -\frac{A}{2} d(k) + n_I(k)$$

  - This offset is a priori unknown and is referred to as the <u>carrier phase ambiguity</u>
  - It is then necessary to solve for the sign of the demodulated navigation data
    - Preamble, CRC, etc…

# 7.3 Phase Discriminator

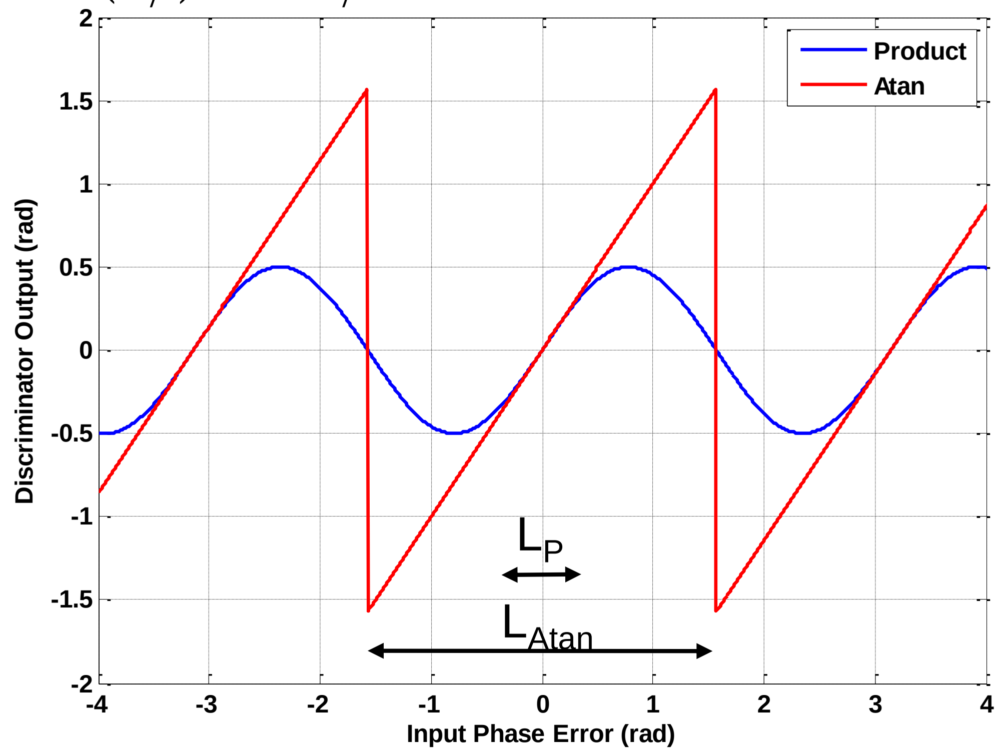
- Once the loop is locked, the phase error will remain with the same ambiguity (the lock point remains the same).
  - If the loop loses lock and re-acquires on a different stability point, the loop is said to have undergone a cycle slip

# 7.3 Phase Discriminator

- The discriminator linearity region $L_D$ is the region around $\varepsilon_\varphi = 0$ in which $D(\varepsilon_\varphi) = \alpha\varepsilon_\varphi$ with $\alpha$ a constant

Consequently, the normalized discriminator should provide the correct (unbiased) information to the tracking loop in its linearity region.

Outside the linearity region, this will endanger the behavior of the loop as it will under- or over-react.



# 7.4 Tracking Loop Model

- Let us assume that the PLL is tracking
  - The tracking error is assumed within the discriminator linear region. Then:

$$\overline{V_d}(k) \sim \frac{1}{2}\sin(2\varepsilon_\theta) + \overline{n_d}(k) \xrightarrow[\varepsilon_\theta \sim 0]{} \varepsilon_\theta + \overline{n_d}(k)$$

- The normalized discriminator output is then sent to a low-pass filter.
  - Using a z-transform gives:

$$V_c(z) = F_{LP}(z)\overline{V_d}(z)$$

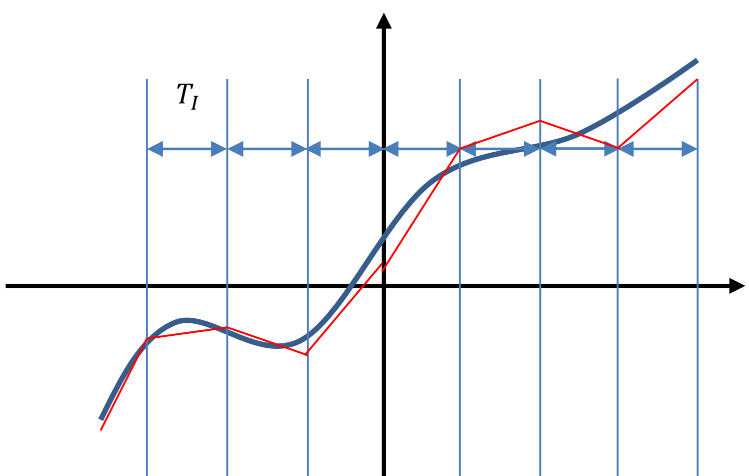
where $F_{LP}$ is the transfer function of the low-pass filter

# 7.4 Tracking Loop Model

- The output of the low-pass filter is then input to a Numerically Controlled Oscillator:
  - A NCO is an electronic device that is able, from a digital input signal $V_c$, to generate a carrier with an instantaneous frequency equal to:

$$f_{NCO}(k) = f_0 + V_c(k)$$

  - The NCO frequency will thus not change between 2 input updates



# 7.4 Tracking Loop Model

- The generated local carrier will thus have a phase varying as a function of the NCO frequency
  - the NCO can be seen as an <u>integrator</u> from the carrier phase point of view:

$$F_{NCO}(z) = \frac{z^{-1}}{1 - z^{-1}}$$

  - It is then possible to represent the whole loop using a z-transform as follows:

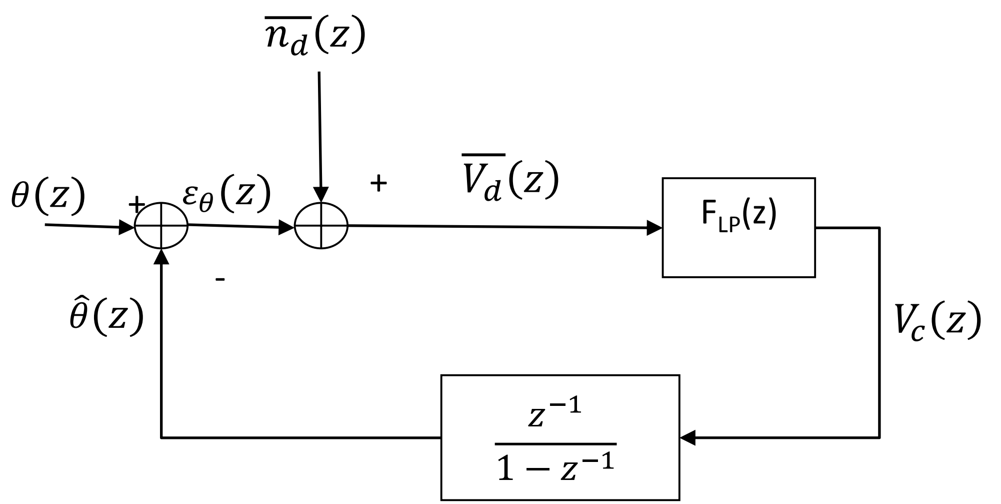
$$\hat{\theta}(z) = \frac{z^{-1}}{1 - z^{-1}} F_{LP}(z)\overline{V_d}(z)$$

# 7.4 Tracking Loop Model

- Assuming the Costas loop is tracking ($\varepsilon_\theta$ is small):

$$\hat{\theta}(z) = \frac{z^{-1}F_{LP}(z)}{1 - z^{-1}}\left(\varepsilon_\theta(z) + \overline{n_d}(z)\right)$$

- This leads to the equivalent linear model of the Costas loop during successful tracking:



# 7.4 Tracking Loop Model

- Thanks to this equivalent linear model, it is possible to directly characterize the response of the Costas loop in tracking mode to any type of incoming phase $\theta$:
  - note that the phase – without the IF component - is now directly represented rather than the full carrier phase

# 7.4 Tracking Loop Model

- Since $\varepsilon_\theta(z) = \theta(z) - \hat{\theta}(z)$, Slide 188 can be used to link the phases of the incoming and local carriers:

$$\hat{\theta}(z) = \underbrace{\frac{\frac{z^{-1}F_{LP}(z)}{1-z^{-1}}}{1+\frac{z^{-1}F_{LP}(z)}{1-z^{-1}}}}_{H_{PLL}(z)}\theta(z) + \underbrace{\frac{\frac{z^{-1}F_{LP}(z)}{1-z^{-1}}}{1+\frac{z^{-1}F_{LP}(z)}{1-z^{-1}}}}_{H_{PLL}(z)}\overline{n_d}(k)$$

- $H_{PLL}$ is referred to as the **equivalent loop filter** (or the loop transfer function in closed loop) since it links the local carrier phase to the phase of the incoming carrier

# 7.4 Tracking Loop Model

- $H_{PLL}$ is a low-pass filter. This means that:
  - the locally-generated phase $\hat{\theta}$ will only be able to follow the slowly varying components of the incoming phase $\theta$ (low frequency terms).
    - If the incoming phase dynamics are high the Costas loop could lose lock depending upon the bandwidth of $H_{PLL}$.
  - $H_{PLL}$ filters the discriminator output noise
    - the narrower the bandwidth of $H_{PLL}$ the less the loop will be affected by noise.

→ Need for a <u>compromise</u> in the design of $H_{PLL}$ between a good rejection of the noise and a good tracking of the incoming phase dynamics

# 7.4 Tracking Loop Model

- A way to characterize the loop is through its **equivalent loop filter bandwidth** $B_L$
  - It is defined as:

$$B_L = \int_0^{1/2T_I} \left| H_{PLL}\left(e^{2j\pi f T_I}\right) \right|^2 df \ \ (Hz)$$

  - It gives a good indication on the level of filtering that the PLL equivalent loop filter provides with respect to thermal noise
    - Equivalently, it controls the loop reaction time: the lower $B_L$, the slowest the loop reaction time

# 7.4 Tracking Loop Model

- Slide 187 can also be used to link the tracking error to the incoming signal phase

$$\varepsilon_\theta(z) = \underbrace{1 - H_{PLL}(z)}_{G_{PLL}(z)}\theta(z) - H_{PLL}(z)\overline{n_d}(z)$$

$G_{PLL}$ is referred to as the **open loop transfer function**.

- It can be noted that $H_{PLL}(z) + G_{PLL}(z) = 1$ → $G_{PLL}$ is a high-pass filter.

→ slow variations of the incoming phase should be well tracked and the resulting phase tracking error should thus be low

# 7.5 Tracking Loop Implementation

- The order of the loop corresponds to the order of $F_{LP}+1$ (the NCO is an integrator filter)
- Implementation of $F_{LP}$
  - 1st order loop design:

$$V_c(k) = \frac{K_1}{2\pi T_I}\overline{V_d}(k)$$

  - 2nd order loop design:

$$V_c(k) = V_c(k-1) + \frac{K_1+K_2}{2\pi T_I}\overline{V_d}(k) - \frac{K_1}{2\pi T_I}\overline{V_d}(k-1)$$

  - 3rd order loop design:

$$V_c(k) = \begin{pmatrix} 2V_c(k-1) - V_c(k-2) + \frac{(K_1+K_2+K_3)}{2\pi T_I}\overline{V_d}(k) \\ -\left(\frac{2K_1+K_2}{2\pi T_I}\right)\overline{V_d}(k-1) + \frac{K_1}{2\pi T_I}\overline{V_d}(k-2) \end{pmatrix}$$

# 7.5 Tracking Loop Implementation

- It is possible to relate the coefficient $K$ to the desired choice of $B_L$ and the correlation duration $T_I$ through:
  - 1st order :

$$K_1 = 4 B_L T_I$$

  - 2nd order:

$$K_1 = \frac{8}{3} B_L T_I; \quad K_2 = \frac{K_1^2}{2}$$

  - 3rd order:

$$K_1 = \frac{60}{23} B_L T_I; \; K_2 = \frac{4}{9} K_1^2; \quad K_3 = \frac{2}{27} K_1^3$$

# 7.6 PLL Performance

- There are usually two figures of merit to characterize the PLL performance:
  - Its tracking accuracy
  - Its sensitivity: this is defined as the minimum SNR that ensures that the loop remains lock
    - The way to analyze the loop sensitivity in a theoretical way is to assess at which $C/N_0$ the tracking error goes too often outside of the discriminator linear region (region in which the discriminator estimates the tracking error without bias), for instance 1% of the time → in this case the probability of losing lock eventually is high.

# 7.6 PLL Performance

- Impact of Thermal Noise
  - The PLL tracking error variance in presence of thermal noise can be approximated by

A small $B_L$ will improve tracking in thermal noise

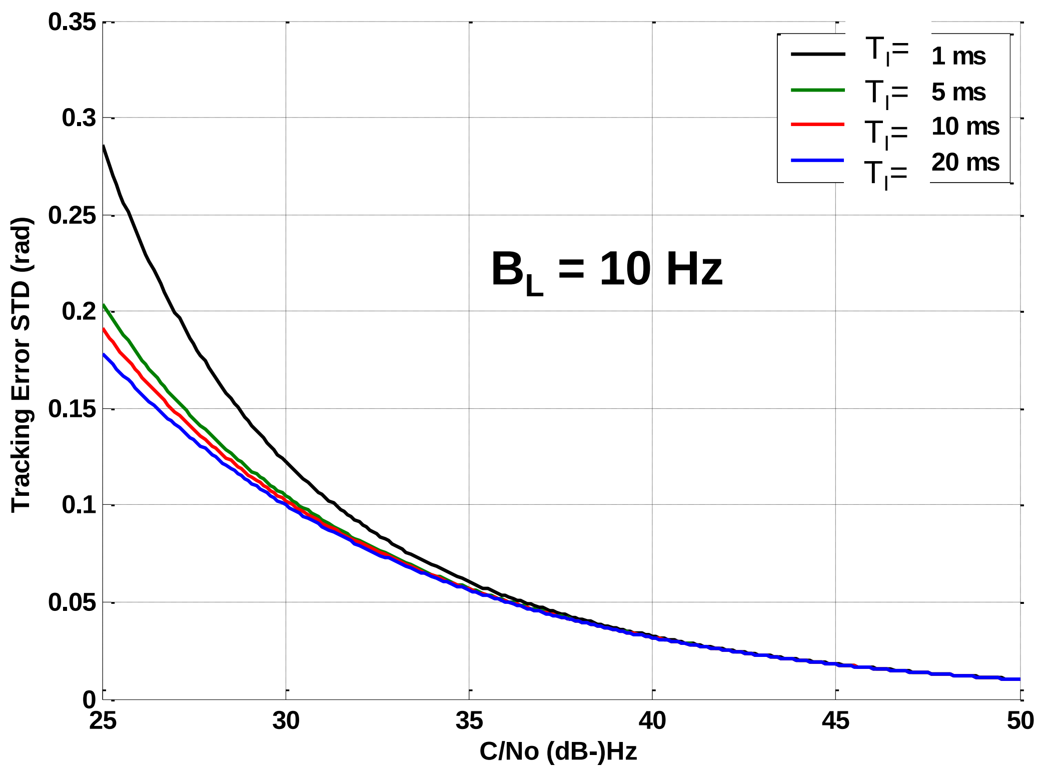
Squaring losses due to the product or division of correlator outputs

$$\sigma^2_{\varepsilon_\varphi, Noise} \approx \frac{B_L}{\frac{C}{N_0}}\left(1+\frac{1}{2\frac{C}{N_0}T_I}\right) \quad \left(rad^2\right)$$

Increasing $T_I$ improves tracking for low $C/N_0$

# 7.6 PLL Performance

- Impact of Thermal Noise
  - Phase tracking error due to thermal noise



# 7.6 PLL Performance

- Impact of Dynamics
  - A PLL tracking loop of order n can track
    - without bias the phase of a signal that has a dynamic of order n-1 (phase delay modeled as a time polynomial of order n-1)
    - with a bias the phase of a signal that has a dynamic of order n (phase delay modeled as a time polynomial of order n)

→ For instance, a 2nd order PLL will be able to track with a bias a variation of the phase corresponding to a constant acceleration.

# 7.6 PLL Performance

- Impact of Dynamics
  - The tracking bias due to constant dynamics is:

| Equivalent Loop Filter Order | Type of Dynamics | Tracking bias (rad) |
|---|---|---|
| 1st Order | Constant velocity $v$ (cycle/sec) | $\theta_b = 2\pi \frac{v}{4B_L}$ |
| 2nd Order | Constant acceleration $a$ (cycle/sec²) | $\theta_b = 2\pi \frac{a}{(1.885 \times B_L)^2}$ |
| 3rd Order | Constant jerk $j$ (cycle/sec³) | $\theta_b = 2\pi \frac{j}{(1.2 \times B_L)^3}$ |

→ The tracking bias increased with $B_L$

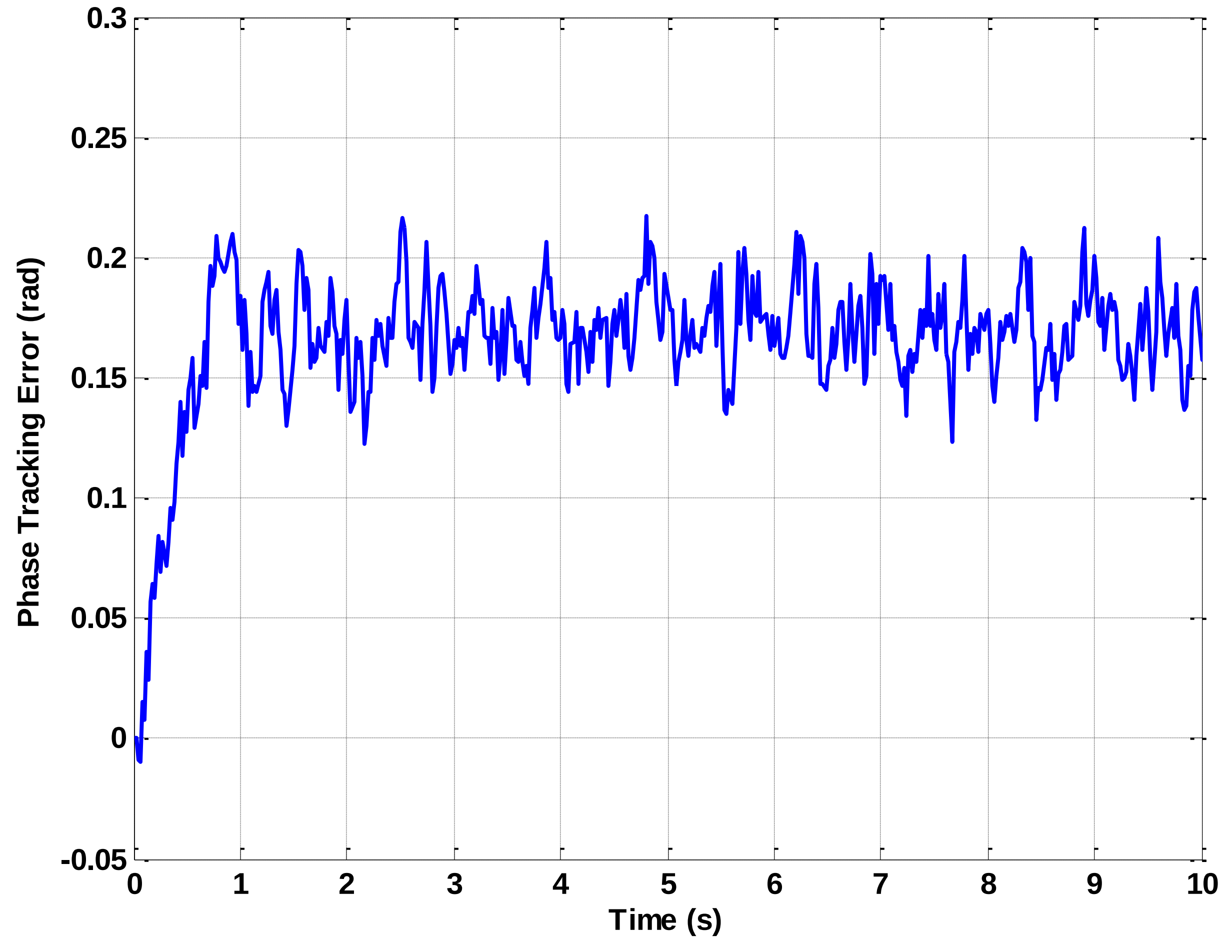
# 7.6 PLL Performance

- Example

3rd order PLL
$B_L$ = 10 Hz
$T_I$ = 20 ms
Jerk = 0.25 g/s

Theory:

$$\varepsilon_{\varphi}(n) \approx 0.171\,\mathrm{rad}$$



# 7.6 PLL Performance

- The PLL loop is always designed in order to:
  - Limit the impact of thermal noise
  - Make sure that the loop is robust to dynamics

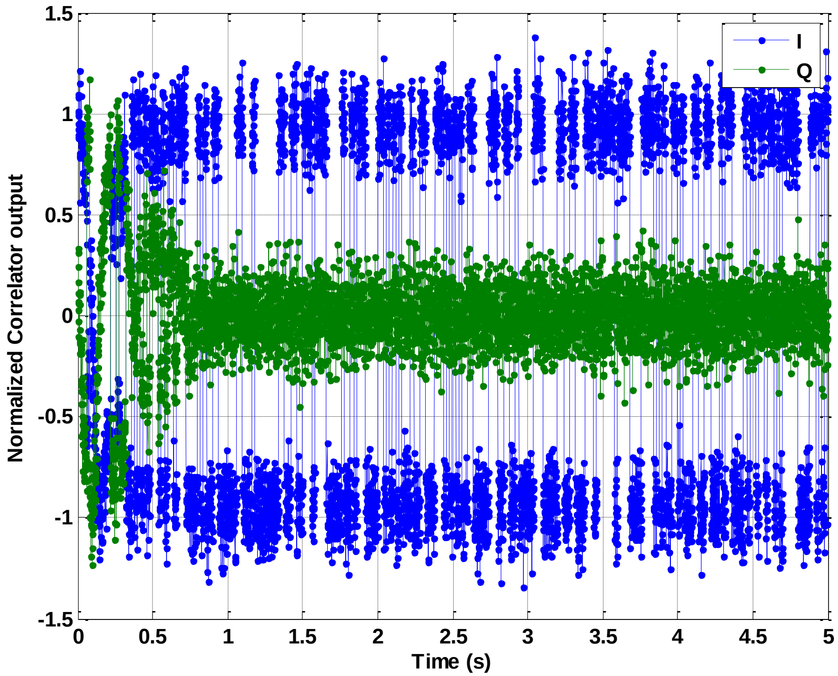
→ Need of a compromise regarding the choice of $B_L$

→ This choice is application dependent

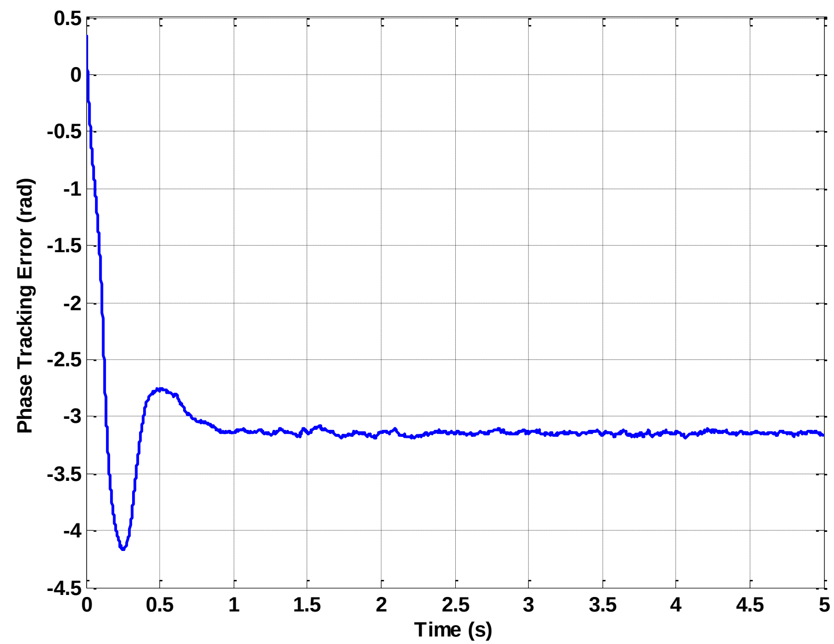
# 7.6 PLL Performance

$C/N_0$ = 45 dB-Hz,
$T_I$ = 1 ms,
Initial Freq. Error of 5 Hz,
$B_L$= 10 Hz

When PLL is locked, the energy is on the I channel and only noise on the Q channel → Data is visible on I




The PLL converges on a stable lock point

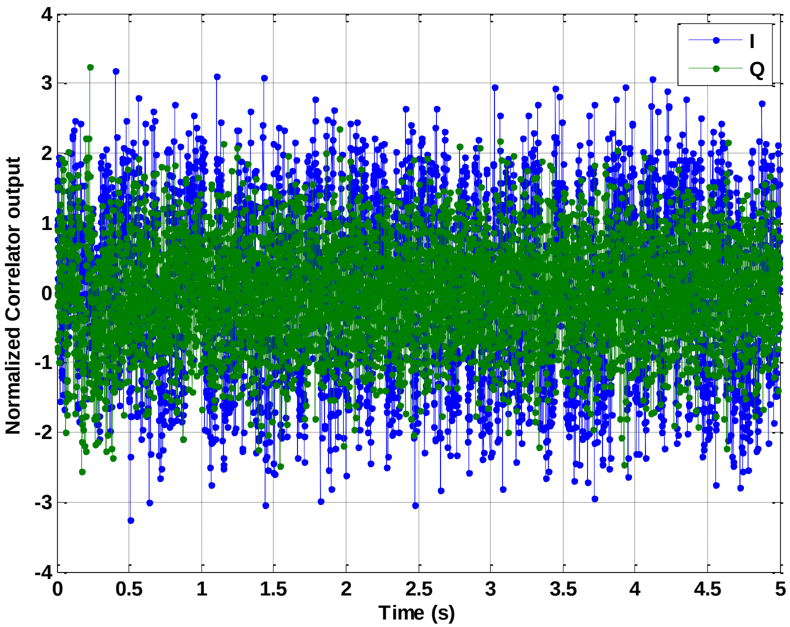


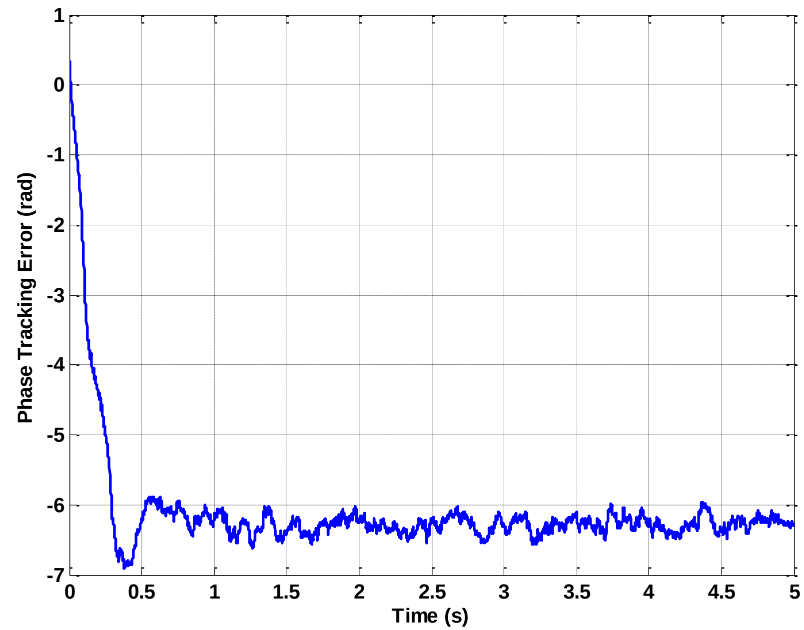
# 7.6 PLL Performance

$C/N_0$ = 30 dB-Hz,
$T_I$ = 1 ms,
Initial Freq. Error of 5 Hz,
$B_L$ = 10 Hz

At lower $C/N_0$, the data is less visible and more difficult to demodulate

The PLL converges on a different stable lock point



# 7.7 Conclusions

- It has been seen that:
  - Carrier phase tracking allows generating a local carrier that is phase-locked with the incoming carrier phase
  - It relies on a discriminator, a loop filter and a NCO
  - The design of the loop is greatly dependent upon a compromise between resistance to thermal noise and resistance to dynamics
  - The loop performance is described by:
    - Its accuracy
    - Its sensitivity

# 8. DS-SS CODE DELAY TRACKING

# 8.1 Introduction

- The goal of a Delay Lock Loop (DLL) is to generate a local replica of the PRN code that is perfectly synchronized with the PRN code in the incoming signal
- To do so, it estimates the synchronisation error between the local replica and the incoming signal using a code delay discriminator
- This error is then filtered and used to control a DCO that commands the shift register generating the local replica

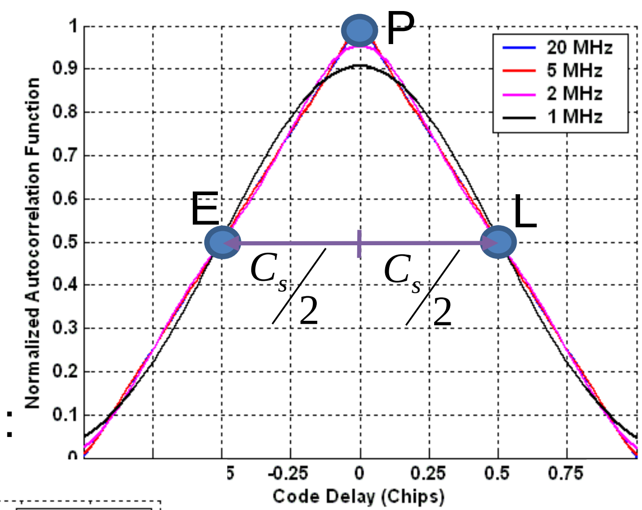
# 8.1 Introduction

- The fundamental principles of a DLL are the same as those of a PLL.

- The only difference is that a different discriminator has to be chosen
  - It has to be able to estimate the code delay tracking error

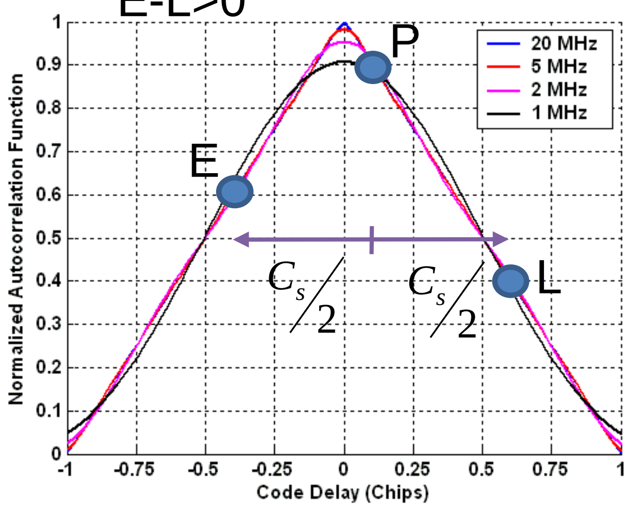
# 8.2 DLL Principles

- The DLL discriminator uses additional correlator outputs:

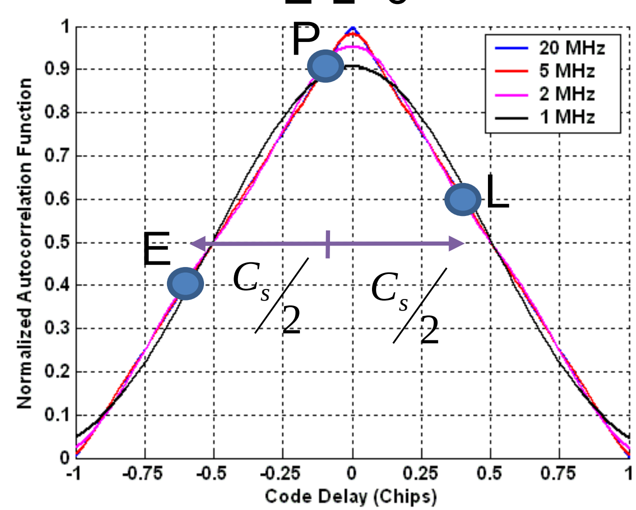



Good synchronization: E-L=0

Advanced synchronization : E-L>0




Late synchronization : E-L<0



# 8.2 DLL Principles

- Specificities of the additional correlators:
  - One that correlates the incoming signal with an early local replica of the PRN. We will consider that the delay is $^{C_s}/_2$
  - One that correlates the incoming signal with a late local replica of the PRN. We will consider that the delay is $-{}^{C_s}/_2$
- Terminology:
  - The new ocrrelators are called Early and Late correlators
  - $C_s$ is referred to as correlator spacing

# 8.2 DLL Principles

- The model for the additional correlators is:

$$I_E(k) = \frac{A}{2} d(k) \mathrm{sinc}\left(\pi \varepsilon_f T_I\right) K_{\tilde{c}_m, c_m}\left(\varepsilon_\tau + \frac{C_s}{2}\right) \cos(\varepsilon_\theta) + n_{Ei}(k)$$

$$Q_E(k) = \frac{A}{2} d(\mathrm{k}) \mathrm{sinc}\left(\pi \varepsilon_f T_I\right) K_{\tilde{c}_m, c_m}\left(\varepsilon_\tau + \frac{C_s}{2}\right) \sin(\varepsilon_\theta) + n_{Eq}(k)$$

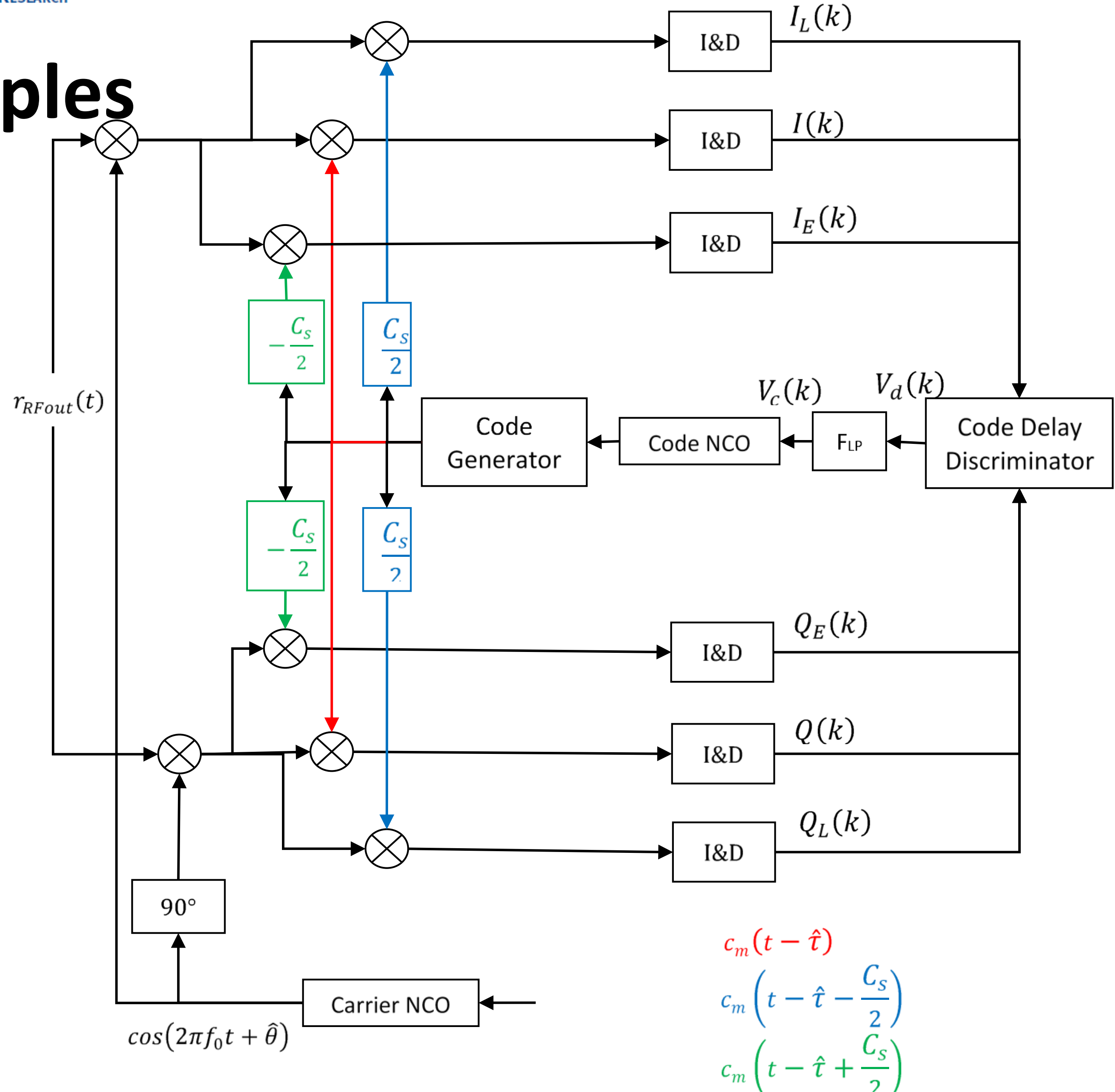
$$I_L(k) = \frac{A}{2} d(\mathrm{k}) \mathrm{sinc}\left(\pi \varepsilon_f T_I\right) K_{\tilde{c}_m, c_m}\left(\varepsilon_\tau - \frac{C_s}{2}\right) \cos(\varepsilon_\theta) + n_{Li}(k)$$

$$Q_L(k) = \frac{A}{2} d(\mathrm{k}) \mathrm{sinc}\left(\pi \varepsilon_f T_I\right) K_{\tilde{c}_m, c_m}\left(\varepsilon_\tau - \frac{C_s}{2}\right) \sin(\varepsilon_\theta) + n_{Lq}(k)$$

- Note that the noise components are:
  - All Gaussian with equal power $\frac{N_0}{4T_I}$
  - I and Q noise components are independent
  - E and L noise components

# 8.2 DLL Principles

- Architecture



# 8.3 Code Delay Discriminators

- There are 3 common DLL discriminators :
  - The coherent Early-Minus-Late (EML)

$$V_e(k) = I_E(k) - I_L(k)$$

  - The non-coherent Early-Minus-Late Power (EMLP) → taken as an example after

$$V_e(k) = \left(I_E^2(k) + Q_E^2(k)\right) - \left(I_L^2(k) + Q_L^2(k)\right)$$

  - The non-coherent Dot-Product (DP):

$$V_e(k) = I_P(k)\left(I_E(k) - I_L(k)\right) + Q_P(k)\left(Q_E(k) - Q_L(k)\right)$$

# 8.3 Code Discriminators

- Terminology:
  - A coherent discriminator assumes that phase tracking is done properly. This means that $\cos(\varepsilon_\varphi) \approx 1$.
    - This means that if phase tracking is not correctly done, then code tracking will likely lose lock
  - A non-coherent discriminator can work independently from the PLL (still requiring frequency lock). It is thus more robust for signals with low $C/N_0$ values, high dynamics, etc…

# 8.3 Code Delay Discriminators

- Let us take the case of the EMLP discriminator
  - The mathematical derivation of the discriminator output is:

$$V_d(k) = \frac{A^2}{4}\mathrm{sinc}^2(\pi\varepsilon_f T_I)\left(K^2_{\tilde{c}_m,c_m}\left(\varepsilon_\tau + \frac{C_s}{2}\right) - K^2_{\tilde{c}_m,c_m}\left(\varepsilon_\tau - \frac{C_s}{2}\right)\right) + n_d(k)$$

where $n_d$ represents the discriminator output noise term

- This discriminator can be normalized by:

$$\overline{V_d}(k) = \frac{(C_s - 2)}{2\left[\left(I_E^2(k) + Q_E^2(k)\right) + \left(I_L^2(k) + Q_L^2(k)\right)\right]}V_d(k) \xrightarrow[\varepsilon_\tau \ll 1]{} \varepsilon_\tau + \overline{n_d}(k)$$

where $\overline{n_d}$ is the normalized discriminator output noise

# 8.3 Code Delay Discriminators

- Assuming a rectangular shaping, it has been seen that $K_{\tilde{c}_m,c_m}$ can be modeled as (large RF front-end bandwidth):

$$K_{\tilde{c}_m,c_m}(x) \sim \begin{cases} 1-|x| & if \quad |x|<1\, chips \\ 0 & else \end{cases}$$

- Let us now assume that, during tracking:
  - The estimated code delay error is smaller than half of the Early-Late spacing
  - The Early and Late correlators are both on the main peak of the PRN code autocorrelation function.

# 8.3 Code Delay Discriminators

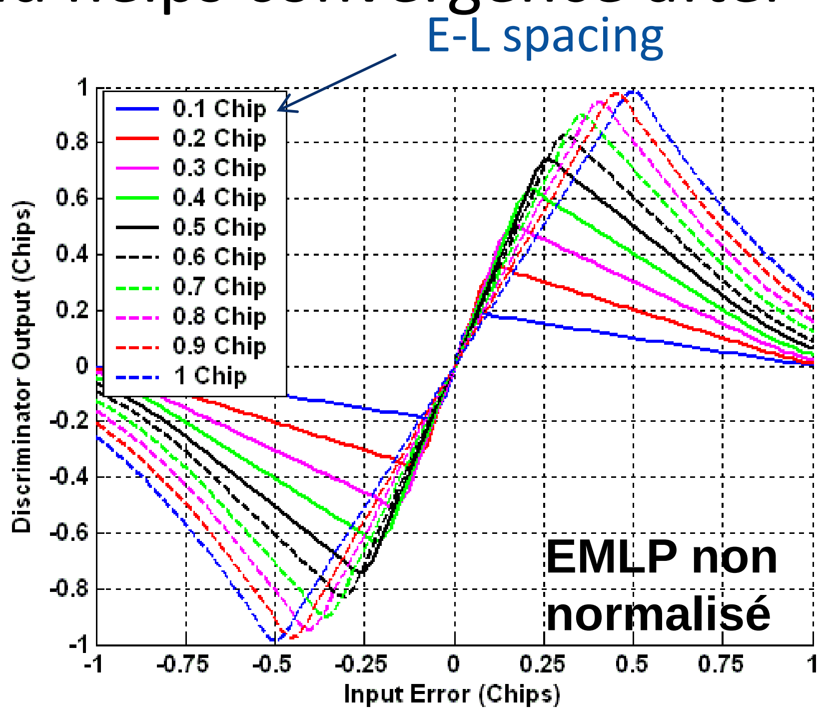
- With the previous assumptions, the EMLP discriminator output can be written as:

$$V_d(k) = \frac{A^2}{2}\operatorname{sinc}^2\left(\pi\varepsilon_f T_I\right)(C_s - 2)\varepsilon_\tau + n_d(k)$$

- It can thus be seen that there is a linear relation between the discriminator output and the code delay tracking error

# 8.3 Code Delay Discriminators

- Impact of the correlator spacing
  - Increasing the E-L spacing extends the linear region of the discriminator, thus its robustness to dynamic stress and large errors, and helps convergence after acquisition.



# 8.4 DLL Performance

- Impact of Thermal Noise
  - The tracking error for the a DLL with the different discriminators is:
    - EML:

$$\sigma^2_{\varepsilon_{\tau_{noise,EML}}} \sim \frac{B_L C_s}{2\frac{C}{N_0}} \quad (chip^2)$$

    - DP:

$$\sigma^2_{\varepsilon_{\tau_{noise,DP}}} \sim \frac{B_L C_s}{2\frac{C}{N_0}}\left(1+\frac{1}{T_I\frac{C}{N_0}}\right) \ (\mathrm{chip}^2)$$

    - EMLP:

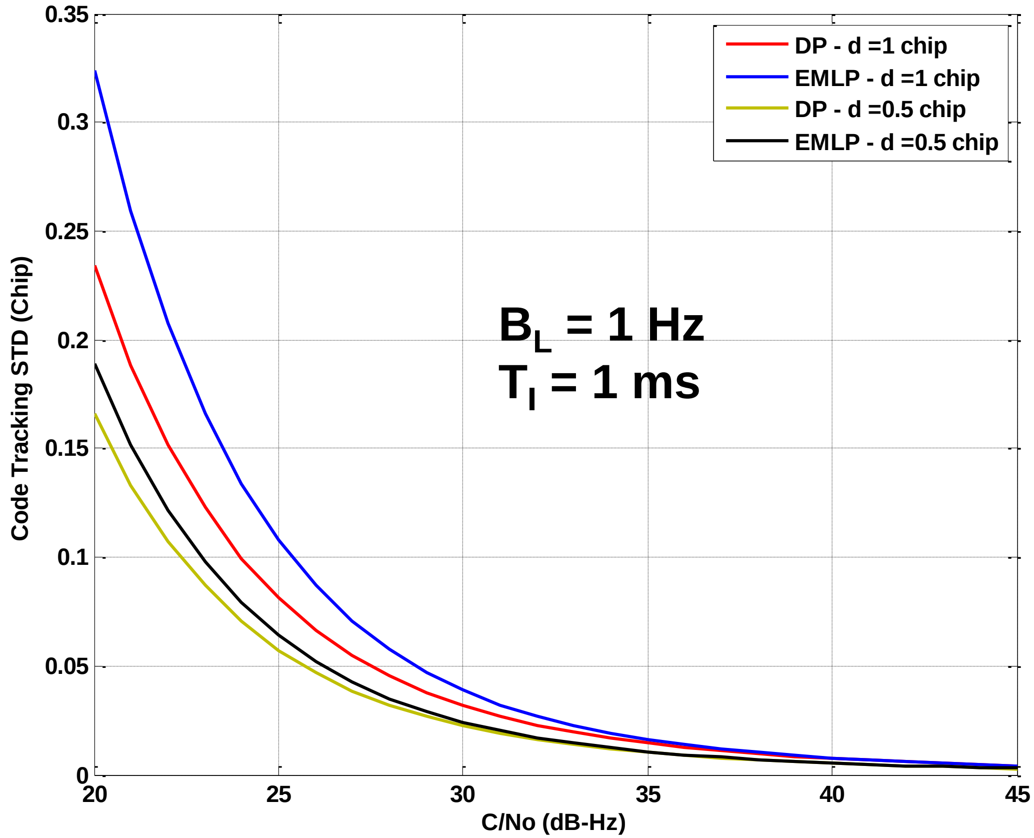
$$\sigma^2_{\varepsilon_{\tau_{noise,EMLP}}} \sim \frac{B_L C_s}{2\frac{C}{N_0}}\left(1+\frac{2}{T_I\frac{C}{N_0}(2-C_s)}\right) \ (\mathrm{chip}^2)$$

# 8.4 DLL Performance

- Impact of Thermal Noise
  - As for the PLL, choosing a narrow $B_L$ reduces the noise of the DLL tracking error
  - Having a narrow correlator spacing also help reduces the noise of the DLL tracking error

# 8.4 DLL Performance

- Impact of Thermal Noise

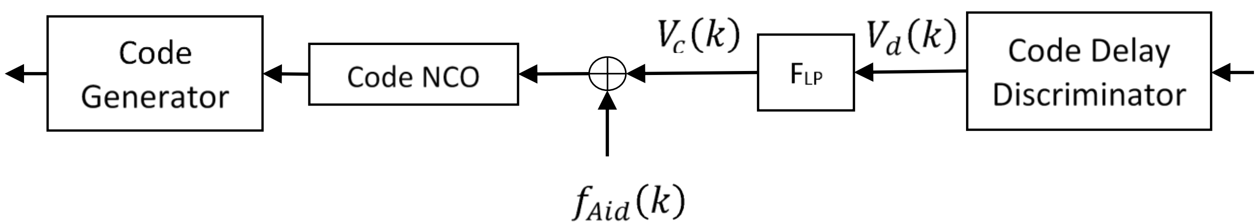


# 8.4 DLL Performance

- Impact of Dynamics
  - Same effect on the DLL as on the PLL
  - However,
    - for the DLL to function, it is required that there is phase and/or frequency lock → signal dynamics are already accurately estimated by the PLL/FLL .
    - It is thus usual to aid the DLL with the output of the PLL/FLL
      - By doing so, most of the signal dynamics are already tracked by the PLL/FLL and there is no need for it to be tracked by the DLL.
      - Typical values of the DLL loop bandwidth when aided by the PLL/FLL can be much lower.

# 8.4 DLL Performance

- Impact of Dynamics
  - Typical aiding scheme:



# 8.5 Conclusions

- It has been shown that:
  - A DLL requires additional correlators to form its discriminator
  - The tracking accuracy is dependent upon
    - the correlator spacing,
    - the loop bandwidth
    - the integration time
  - It is possible to use the PLL output to aid the DLL and thus reduce the loop bandwidth

# 8.5 Conclusions

- DLL+PLL combination