Chapitre 3. Radio Channel Equalization

1. Introduction

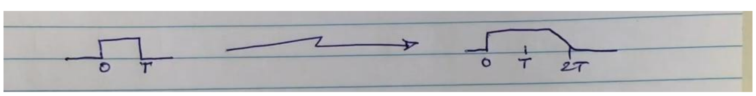
Equalization refers to the process of adjusting the transmitted signal. Indeed, distortions and interference may occur during transmission and reception (multipath, noise,…).

2. Inter-symbol interference (ISI)

Intersymbol Interference (ISI) occurs when symbols transmitted over a communication channel interfere with each other, causing distortion in the received signal.

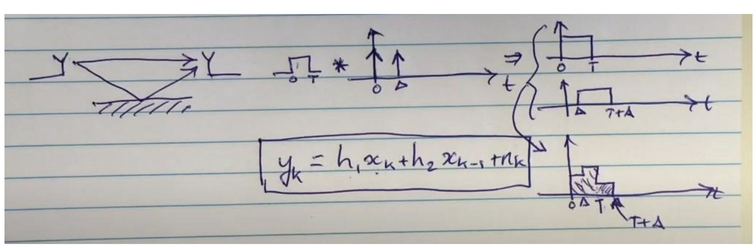
The ISI can be caused by different situations:

a) **Multipath Fading**: In wireless communications, signals often take multiple paths to reach the receiver, causing them to arrive at slightly different times. This can create interference and distortion.



There two situations where this case arises:

- First one in the **wireless communication,** the following example shows a pulse response on the cannel at $t = 0$ for the first path and a pulse response at Δt for the second path, in this case the signal is attenuated.



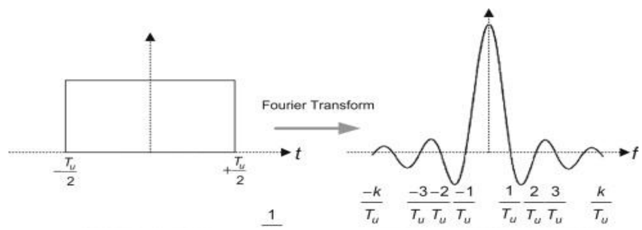
The final result is the addition of two responses

-

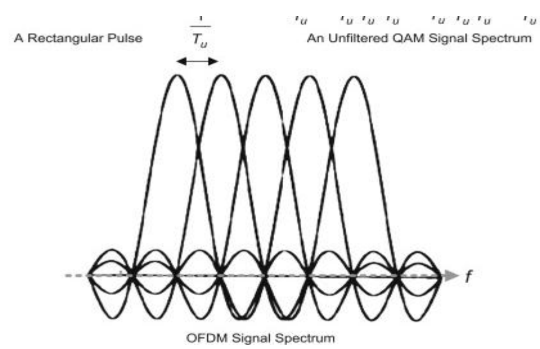
b) Frequency Selective Fading: Different frequency components of a signal can be attenuated differently by the transmission medium (channel).

Bandwidth of the pulse stream.

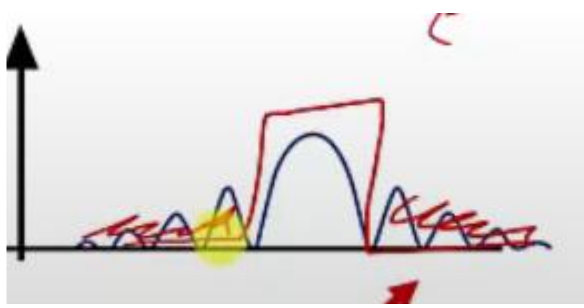
We know that the TF(rect) generate a sinc function.

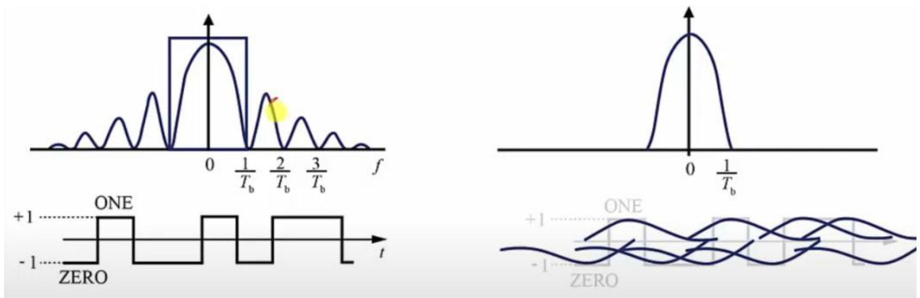


When we have a stream of symbol in the frequency domain we will have and overlapping sincs



One of the obvious solutions is to liter each sinc





When we want to recover our signal in the temporal domain we will get overlapping symbol thus iner-symbol interference

To mitigate ISI, **equalization techniques** can be applied at the receiver to compensate for the channel-induced distortions. Additionally, ***pulse shaping techniques*** can be used to control the bandwidth of the transmitted signal and reduce ISI.

3. Equalization

An equalizer is a filter that is applied at the receiver to counteract the effects of the channel. The goal of equalization is to recover the original transmitted signal by undoing the spreading that causes ISI. By estimating the characteristics of the channel and applying an inverse filter, the equalizer can correct for the distortions that occur during transmission

There are two type of equalizations, linear and non-linear equalizer.

3.1 Linear equalization

Linearity implies that the equalizers produce symbol estimates y[m] based on simple linear operations on the matched filter output. It mitigate ISI without enhancing noise but not very efficient in sever ISI. Includes Zero Forcing Equalizaer (ZFE) and Minimum Mean Square Error (MMSE) Equalizer.

a) Zero Forcing Equalizaer (ZFE)

The ZFE uses the inverse of the channel's frequency response to remove the distortions introduced by the channel. If the channel's frequency response H(f) is known, the ZFE applies a filter with transfer function 1/H(f) to undo the channel's effects. By doing this, the equalizer aims to restore the transmitted signal to its original form.

Let the channel's impulse response be h(t), and the transmitted signal be x(t). The received signal r(t) is given by: r(t)=x(t)∗h(t)

The ZFE applies a filter with impulse response g(t), where g(t) is the inverse of h(t), i.e., g(t)∗h(t)=δ(t), where δ(t) is the Dirac delta function. This filter "undoes" the convolution, removing ISI:

$$\boldsymbol{r_{equalized}(t) = r(t) * g(t) = x(t) * h(t) * g(t)}$$

In the frequency domain, the received signal can be expressed as:

$$\boldsymbol{R(f) = X(f) \cdot H(f)}$$

The ZFE applies an inverse filter $\boldsymbol{G(f) = \frac{1}{H(f)}}$ so that:

$$\boldsymbol{R_{equalized}(f) = R(f) \cdot G(f) \; = \; X(f) \cdot H(f) \cdot \frac{1}{H(f)} = X(f)}$$

This means that the effect of the channel is completely removed, restoring the original transmitted signal X(f).

Disadvantage:

- Noise Amplification: If the channel's frequency response H(f) has very small values for certain frequencies, the inverse filter 1/H(f) will have very large values, which can significantly amplify noise at those frequencies.
- Requires Perfect Channel Knowledge: The ZFE assumes that the channel response H(f) is perfectly known at the receiver. In practical scenarios, estimating the channel accurately can be challenging.

b) Minimum Mean Square Error (MMSE) Equalizer

The MMSE equalizer balances **ISI cancellation** with **noise minimization**, offering better performance in practical scenarios where noise is present.

The received signal r(t) is the convolution of the transmitted signal and the channel response, with added noise:

r(t)=x(t)∗h(t)+n(t)

In the frequency domain, this becomes:

R(f)=X(f)H(f)+N(f)

MMSE Equalizer Formula in matrix form:

$$\hat{a} = (H^H H + \sigma^2 I)^{-1} H^H r$$

Where:

- $\hat{a}$ is the **estimated transmitted signal** (a vector of transmitted symbols).
- H is the **channel matrix**, representing the relationship between the transmitted and received signals (for a MIMO system, this describes the path gain between each transmitter and receiver).
- H^H is the **Hermitian transpose** (conjugate transpose) of the channel matrix H.
- σ^2 is the **noise variance** (power of the additive white Gaussian noise, AWGN).
- I is the **identity matrix**, with the same dimensions as $H^H H$.
- r is the **received signal vector**, which includes the transmitted signal affected by the channel and noise.

3.2 Non-linear equalizers

In this category, the equalizers generate the symbol outputs through more complicated mathematical operations than simple linear operations. It performs well in sever ISI.

Includes Decision Feedback Equalizer (DEF), Maximum-Likelihood (ML) and Maximum Likelihood Sequence Estimation (MLSE).

A) Maximum Likelihood Sequence Estimation

Definition: MLSE is a method used to detect the transmitted data sequence that maximizes the likelihood of the observed received sequence, considering the presence of ISI and noise.

- Consider a digital communication system where the transmitted signal s(t) passes through a channel with impulse response h(t). The received signal r(t) can be expressed as:

$$r(t) = s(t) * h(t) + n(t)$$

where $*$ denotes convolution and n(t) is the additive noise.

- If the transmitted signal is represented as a sequence of symbols a_k with pulse shape p(t):

$$s(t) = \sum_k a_k \, p(t - kT)$$

Where, T is the symbol period.

After passing through the channel, the received signal becomes:

$$r(t) = \sum_k a_k \, [p(t - kT) * h(t)] + n(t)$$

Let g(t)=p(t)∗h(t) be the effective pulse shape after the channel. The received signal can be written as:

$$r(t) = \sum_k a_k \, [g(t - kT)] + n(t)$$

The sampled signal at t=nT

$$r(nT) = \sum_k a_k \, [g(nT - kT)] + n(nT)$$

Problem Formulation

Given the received samples r(nT), the objective of Maximum Likelihood Sequence Estimation (MLSE) is to estimate the transmitted sequence {ak} that maximizes the likelihood function. Assuming additive white Gaussian noise n(nT), **the likelihood function probability is proportional to the difference of the estimated and transmitted signal** :

$$P(r|a) \propto \exp\left(-\frac{1}{2\sigma^2}\sum_n \left(r(nT) - \sum_k a_k g(nT - kT)\right)^2\right)$$

Maximizing this likelihood function is equivalent to minimizing the following metric:

$$\hat{a} = \arg\min_a \sum_n \left(r(nT) - \sum_k a_k g(nT - kT)\right)^2$$

This come down to a least-squares problem, where we seek to minimize the squared error between the received samples and the samples estimated based on the transmitted symbols.

Viterbi Algorithm:

The **Viterbi Algorithm (VA)** is applied in **MLSE** using a recursively defined **metric**.

Recursive Metric Definition:

The function $\Lambda[T+1]$ can be written as:

$$\Lambda[T+1] = \Lambda[T] + \lambda[T],$$

Where:

- $\Lambda[T]$: metric up to time T,
- $\lambda[T]$: branch metric at time T, given by:

$$\lambda[T] = \left| r[T] - \sum_{l=1}^{L} h[l]\tilde{a}[T-l] \right|^2$$

The branch metric $\lambda[k]$ evaluates the mismatch (squared error) between the received signals.

Example

We want to transmit a sequence of:$\{ak\} = [1,0,1,0,1]$. The impulse response of a channel is $h(t) = \delta(t) + 0.5\delta(t-T)$, this means that the signal has an ISI coefficient of 0.5. The signal waveform: Rectangular pulse of width T. Assuming the presence of an additive white Gaussian noise (AWGN) with σ=0.1

1. Simulation of Received Signal

The **received signal** is formed as:

$$r[k] - \sum_{l=1}^{L} h[l]a[k-l] + n[k]$$

where:

- h[l]=[1, 0.5]: impulse response samples,
- L=2: channel memory (current and previous symbol contribute),
- n[m] AWGN noise.

The impulse response of channel is $h(t) = \delta(t) + 0.5\delta(t - T)$.

0.5: coefficient of ISI, meaning part of the current symbol affects the next one with this weight.

1. **Transmit the first bit**: a1=1

   r(T)= 1.a1+n(T) =1+n(T)

2. **Transmit the second bit**: a2=0

   r[2T]=h[0]a[2]+h[1]a[1]+n[2T]

   r[2T]=1. a[2]+0.5· a[1]+ n[2T]= 0+0.5·1+[2T] = 0.5+n[2T]

3. **Transmit the third bit**: a3=1

   r[3T]=h[0]a[3]+h[1]a[2]+n[3T]

   r[3T]=1. a[3]+0.5· a[2]+n[3T]=1+0.5·0+n[3T]=1+n[3T]

4. **Transmit the fourth bit**: a4=0

   r[4T]=h[0]a[4]+h[1]a[3]+n[4T]

   r(4T)=1. a[4]+0.5· a[3]+ n[4T]=0+0.5·1+ n[4T]=0.5+n[4T]

5. **Transmit the fifth bit**:a5=1

   r(5T)=a5·1+0.5·a4+n(5T)= 1+0.5·0+n(5T)=1+n(5T)

Sequence of the received signal

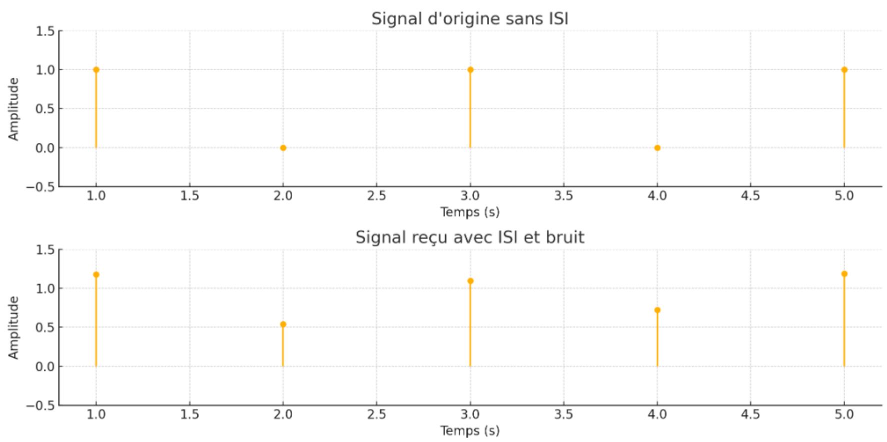
The samples of noise n[T] are generated randomly using m=0, σ=0.1:

1. n[1T]≈ 0.05
2. n[2T] ≈− 0.03

3. n[3T] ≈ 0.02
4. n[4T] ≈− 0.04
5. n[5T] ≈ 0.01

Les valeurs reçues r(t) deviennent :

1. r(T) ≈ 1+0.05 = 1.05
2. r(2T) ≈ 0.5−0.03 = 0.47
3. r(3T) ≈ 1+0.02 = 1.02
4. r(4T) ≈ 0.5−0.04 = 0.46
5. r(5T) ≈ 1+0.01 = 1.01



The graphs clearly show the effect of inter-symbol interference (ISI) on the received signal, compared to the original signal without ISI.

Find original sequence using Viterbi Algorithme

At each time point in the Viterbi algorithm, we only keep the states whose minimum metrics were determined at the previous time point.
If at time T, the state with the minimum metric is state 1, then at time 2T, we only need to calculate the metrics for the states that can be reached from state 1. Let's look at this in more detail:

$\Lambda[k+1] = \Lambda[k] + \lambda[k]$,

Step 1: Initialization at T:

There are two possible states, 0 and 1. Metrics are calculated based on the received value r(T)=1.05

- For each state:
  - **State 0** (assuming transmission of 0): Metric0(T)=(r(T)−0)2=$(1.05-0)^2$=1.1025
  - **State 1** (assuming transmission of 1): Metric1(T)=$(r(T)-1)^2$=$(1.05-1)^2$=0.0025

At T, the state with the minimum metric is state 1, with a metric of 0.0025

Step 2: Calculation at 2T:

Only the state reached with the minimum metric at T (state 1) is considered. We calculate the metrics for possible transitions from state 1 at time 2T based on the received value r(2T)=0.47

- **From state 1 to 0**:
  Metric10(2T)=Metric1(T)+$(r(2T)-0)^2$=0.0025+$(0.47-0)^2$=0.0025+0.2209=0.2234

- **From state 1 to 1**:
  Metric11(2T)=Metric1(T)+(r(2T)−1)2=0.0025+(0.47−1)2=0.0025+0.2809=0.2834

At 2T, the state with the minimum metric is state 10, with a metric of 0.2234.

Step 3: Calculation at 3T

At this step, we only consider the state 10 reached at 2T with the minimum metric. We then calculate metrics for possible transitions from state 10 based on the received value r(3T)=1.02.

- **From state 10 to 0**:
  Metric100(3T)=Metric10(2T)+$(r(3T)-0)^2$=0.2234+$(1.02-0)^2$=0.2234+1.0404=1.2638
- **From state 10 to 1**:
  Metric101(3T)=Metric10(2T)+$(r(3T)-1)^2$=0.2234+$(1.02-1)^2$=0.2234+0.0004=0.2238

At 3T, the state with the minimum metric is state 101 with a metric of 0.2238

Step 4: Calculation at 4T

Now, we only consider state 101, which has the minimum metric at 3T. We calculate metrics for the transitions from state 101 based on the received value r(4T)=0.46

- **From state 101 to 0**:

$$\text{Metric1010}(4T)=\text{Metric101}(3T)+(r(4T)-0)^2=0.2238+(0.46-0)^2=0.2238+0.2116=0.4354$$

- **From state 101 to 1**:

$$\text{Metric1011}(4T)=\text{Metric101}(3T)+(r(4T)-1)^2=0.2238+(0.46-1)^2=0.2238+0.2916=0.5154$$

At 4T, the state with the minimum metric is state 1010 with a metric of 0.4354.

Step 5: Calculation at 5T

We only consider state 1010, the state with the minimum metric at 4T. We calculate the metrics for transitions from state 1010 based on the received value r(5T)=1.01

- **From state 1010 to 00**:

$$\text{Metric10100}(5T)=\text{Metric1010}(4T)+(r(5T)-0)^2=0.4354+(1.01-0)^2=0.4354+1.0201=1.4555$$

- **From state 1010 to 1**:

$$\text{Metric10101}(5T)=\text{Metric1010}(4T)+(r(5T)-1)^2=0.4354+(1.01-1)^2=0.4354+0.0001=0.4355$$

At 5T, the state with the minimum metric is state 10101 with a metric of 0.4355.

By tracking the minimal states at each time kT, we reconstructed the original sequence as {1, 0, 1, 0, 1}, which is the transmitted sequence.