OFDM (Orthogonal Frequency Division Multiplexing)

Motivation: Recall from previous lectures

pulse shaping

$$x(t) = \sum_i A_i \, \psi(t - iT)$$

Symbols produced by the encoder (e.g. QAM)

Transmitted signal (baseband equivalent)

Received signal is $R(t) = x(t) + z(t)$

AWGN with zero mean and variance N_0 $\mathcal{N}(0, N_0)$

The equivalent symbol-level channel:

$$A_i \longrightarrow \oplus \longrightarrow Y_i$$

$$z_i \sim \mathcal{N}(0, \sigma^2)$$

What we like about this signaling method:

- We can write the transmitted signal as an orthonormal expansion
- Waveform former (pulse shaping) and the n-tuple former (MF + sampling at kT) are quite easy to implement.
- The error probability depends only on the A_i's (amplitude) and the noise variance.

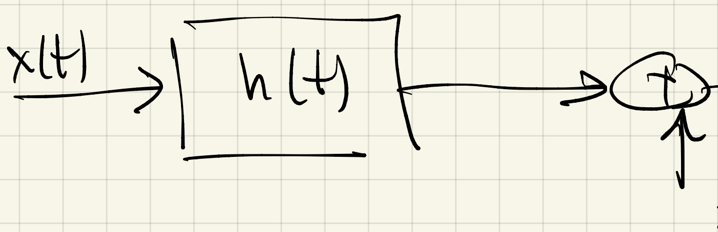
- The Power Spectral Density (PSD)

$$S_x(f) = \frac{|\Psi_F(f)|^2}{T} \sum_k \underbrace{K_A[k]}_{\text{the self-similarity of } \{A_k\}} e^{-j2\pi f T k}$$

$K_A[k] = E|A_k|^2$ if $\{A_k\}$ are uncorrelated.

There are several problems with this signaling method.

If the channel has an impulse response $h(t)$



$$Y(t) = \sum_k A_k p(t-kT) + z(t)$$

where $p(t) = (\psi * h)(t)$.

In general $p(t)$ is not orthogonal to $p(t-iT)$.

From the MF output we see a symbol-level channel

$$Y_i = \sum_{k=0}^{L-1} A_{i-k} h_k + z_i .$$

$$= A_i h_0 + \underbrace{\sum_{k=1}^{L-1} A_{i-k} h_k}_{\text{Inter Symbol Interference (ISI)}} + z_i$$

$h = [h_0, h_1 \ldots h_{L-1}]$ is the symbol-level channel response.

Various methods to combat the ISI:

- Equalization: "compensate" for h ("invert the channel")
  → this leads to increase of the noise

- Viterbi decoder: high complexity $\sim M^L$ where M is the size of the $\{A_i\}$ alphabet (4-QAM, $M=4$ for example)

- OFDM: Elegant approach that prevent ISI from happening.

Key idea: Symbol-by-symbol on eigenfunctions.

Suppose $\varphi_i(t)$, $i=0,\dots N-1$ are the eigenfunctions of $h(t)$.

Then:

$$\sum_{i=0}^{N-1} A_i \varphi_i(t) \longrightarrow \boxed{h(t)} \longrightarrow \sum_{i=0}^{N-1} \underbrace{\lambda_i}_{\substack{\text{eigenvalue} \\ \text{associated to } \varphi_i(t)}} A_i \varphi_i(t)$$

If in addition $\varphi_i(t) \perp \varphi_j(t)$ for $i \neq j$, then the transmitted signal is written again as an orthonormal expansion.

How to find the eigenfunctions?

Complex exponentials are eigenfunctions of linear time-invariant systems:

$$e^{j2\pi f t} \longrightarrow \boxed{h(t)} \longrightarrow \underbrace{\lambda}_{h_F(f)} e^{j2\pi f t}$$

Let $x(t) = e^{j2\pi f t}$

$$y(t) = (x * h)(t) = \int x(t-\alpha)\, h(\alpha)\, d\alpha =$$

$$= \int e^{j2\pi f (t-\alpha)}\, h(\alpha)\, d\alpha =$$

$$= e^{j2\pi f t} \underbrace{\int h(\alpha)\, e^{-j2\pi f \alpha}\, d\alpha}_{FFT(h(t)) \, = \, h_F(f)}$$

$$= h_F(f)\, e^{j2\pi f t} \qquad \square$$

There are a few problems

(i) We need finite-length signals

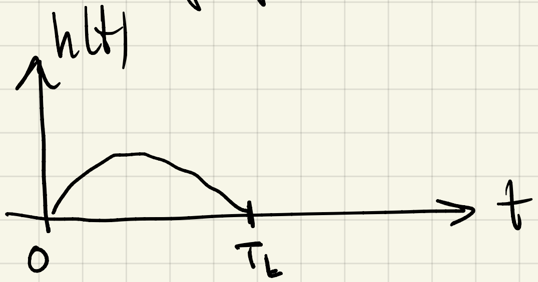
$$\psi_i(t) = \begin{cases} \frac{1}{N} e^{j2\pi f t}, & t \in [0, T] \\ 0, & \text{otherwise} \end{cases}$$

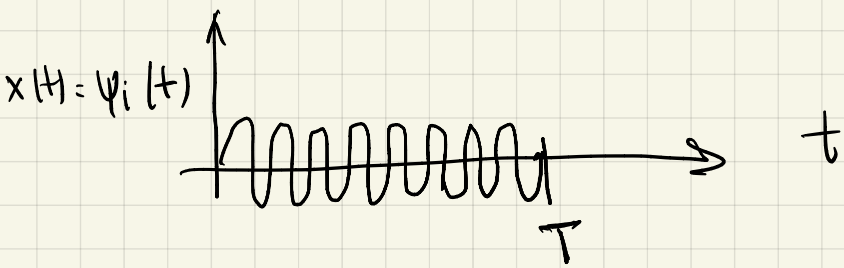
where N is to be defined later.

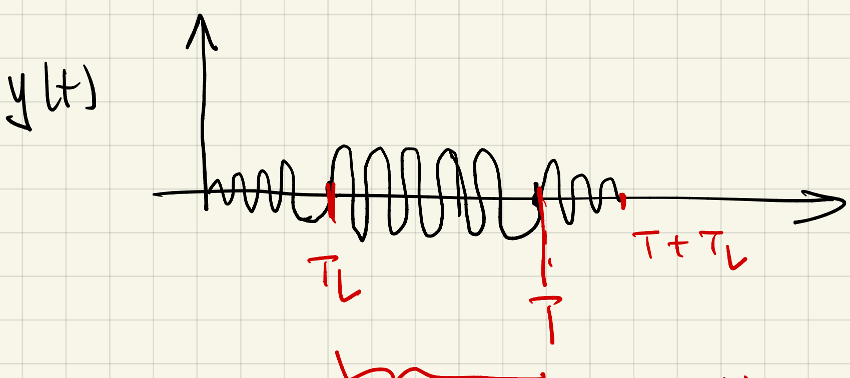
(ii) If we choose $f_i = \frac{i}{T}$ $\quad i = 0, \dots N-1$

then $\varphi_i \perp \varphi_j$ for $i \neq j$

(iii) Truncated exponentials are not anymore eigenfunctions of linear time-invariant systems







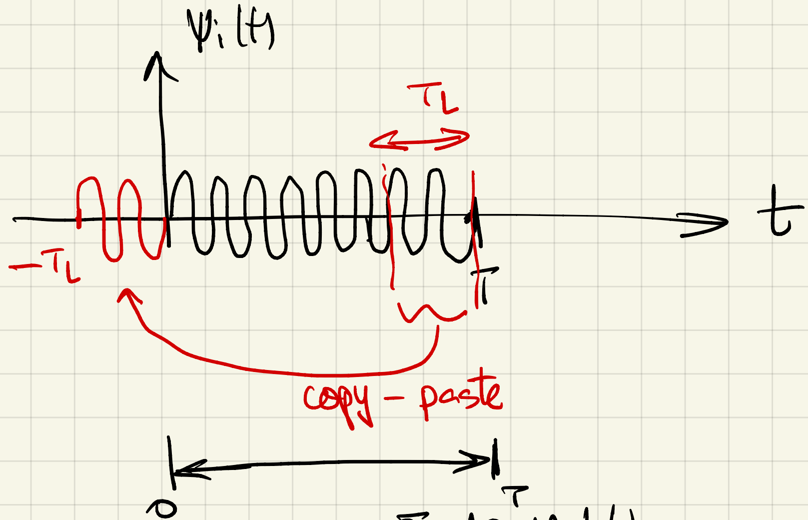
$$y(t) = x(t) * h(t) =$$

$$= \int x(t-\alpha) \underbrace{h(\alpha)}_{\substack{\text{Support on} \\ [0, T_L]}} d\alpha$$

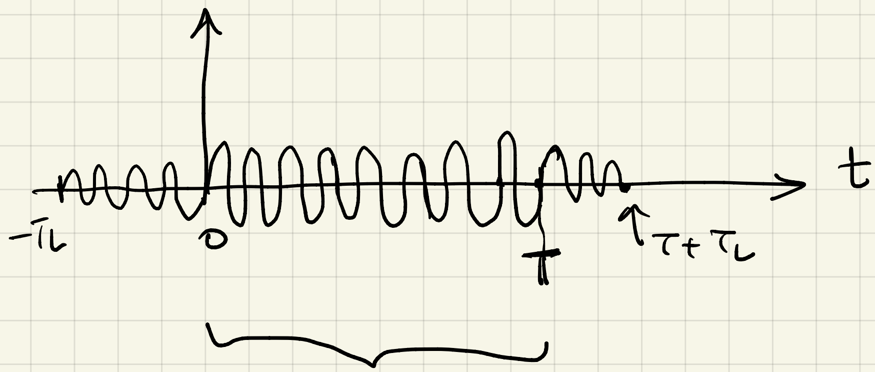
In this portion $y(t) = \lambda_i \varphi_i(t)$

We want to use only the portion of the output signal that behaves as an eigenfunction.

To extend that part to $[0, T]$, we add a cyclic extension to the input signal.



$$\sum A_i \psi_i(t)$$



$$\sum A_i \lambda_i \psi_i(t)$$

(iv) At first it looks we need to have N oscillators at the sender and at the receiver to implement

$$\psi_i(t) = \frac{1}{N} e^{j2\pi f_i t} \qquad i = 0, \dots N-1$$

$\Rightarrow$ <u>Huge complexity</u>. $\qquad f_i = \frac{i}{T}$

We actually need <u>only</u> the samples of

$$x(t) = \frac{1}{N} \sum_{i=0}^{N-1} A_i e^{j2\pi \frac{i}{T} t} \quad , t \in [0, T]$$

Let's define $\underline{x} = (x_0, x_1, \dots x_{N-1}) \qquad x_k = x(kT_s)$

$$T_s = \frac{T}{N}$$

$$x_k = x(kT_s) = \frac{1}{N}\sum_{i=0}^{N-1} A_i e^{j\frac{2\pi}{N}ik}$$

Notice that $\underline{x} = DFT^{-1}(A)$ where $A = (A_0, A_1 \dots A_{N-1})$

Define: $\beta = e^{j\frac{2\pi}{N}}$

$$F = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \beta & \beta^2 & & \beta^{N-1} \\ \vdots & \beta^2 & (\beta^2)^2 & & (\beta^2)^{N-1} \\ & & \vdots & & \vdots \\ 1 & \beta^{N-1} & (\beta^{N-1})^2 & & (\beta^{N-1})^{N-1} \end{pmatrix}$$

$\underline{A} = F^\dagger \underline{a}$ is the DFT of $\underline{a}$

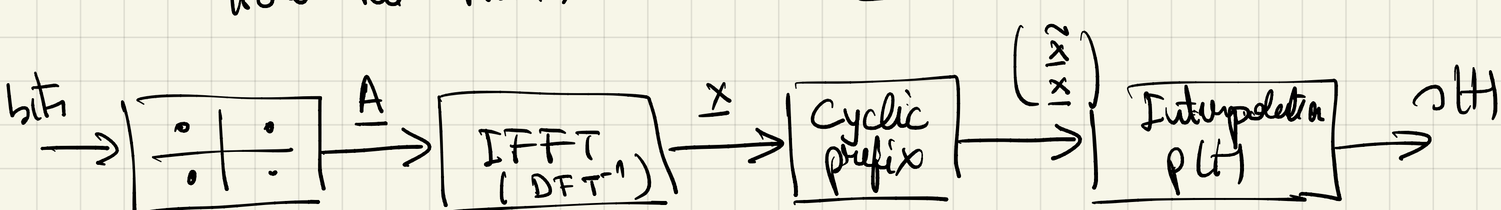
$\underline{a} = \frac{1}{N} F \underline{A}$ is the DFT^{-1} of $\underline{A}$

$$\underline{a} = \frac{1}{N} F \underline{A} = \frac{1}{N} F F^\dagger \underline{a} \Rightarrow \frac{1}{N} F F^\dagger = I_N \text{ (identity matrix)}$$

$\Rightarrow$ If we take $\frac{1}{\sqrt{N}} F$, this is an unitary transformation

---

How the transmitter looks like:



Constellation mapper

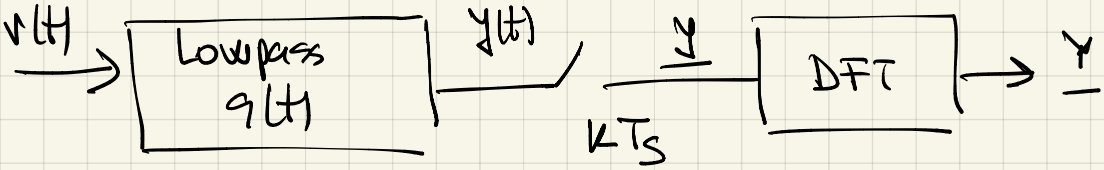
$p(t) = \text{sinc}\left(\frac{t}{T_s}\right)$

$$s(t) = \underbrace{[\quad\quad]}_{\substack{t \in [-T_c, 0] \\ \text{cyclic extension}}} + \underbrace{\frac{1}{N}\sum_{i=0}^{N-1} A_i e^{j\frac{2\pi}{T}it}}_{x(t),\ t \in [0, T]}$$

For $t \in [0, T]$ the received signal will be

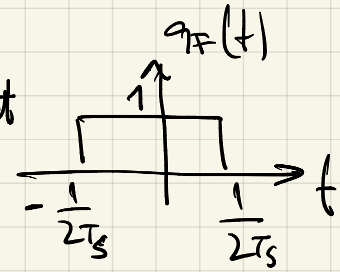
$$r(t) = \frac{1}{M} \sum_{i=0}^{M-1} \lambda_i A_i e^{j \frac{2\pi}{T} i t} + \underbrace{z(t)}_{\text{complex AWGN with PSD } N_0}$$

<u>Receiver</u> (Maximum likelihood)



$$q(t) = \frac{1}{T_s} \operatorname{sinc}\left(\frac{t}{T_s}\right) = \frac{1}{T_s} p^*(-t)$$

(Matched filter, normalized such that



$\underline{y}$ represents a sufficient statistic.
So does $\underline{Y}$ (DFT is an invertible transformation)

$$\underline{y} = (y_0, y_1 \dots y_{M-1})$$

y_k: how does it look

$$y_k = \frac{1}{M} \sum_{i=0}^{M-1} \lambda_i A_i e^{j \frac{2\pi}{T} i k T_s} + z_k$$

$$= \frac{1}{M} \sum_{i=0}^{M-1} \underbrace{\lambda_i A_i}_{} e^{j \frac{2\pi}{M} i k} + z_k$$

we notice that

$$\underline{y} = (y_0, y_1 \dots y_{M-1}) = DFT^{-1}(\lambda_0 A_0, \lambda_1 A_1, \dots \lambda_{M-1} A_{M-1})$$

$$\underline{Y} = DFT(\underline{y}) = (\lambda_0 A_0, \lambda_1 A_1, \dots \lambda_{M-1} A_{M-1}) + \underline{Z}$$

$$y(t) = r(t) + \underbrace{z_{LP}(t)}$$

↳ AWGN with variance $\frac{N_0}{T_s} = \sigma^2$

$$\underline{Z} = DFT\left(z_{LP}(kT_s)\right)$$

$$= \sqrt{N}\ \underbrace{\frac{1}{\sqrt{N}} F^{+}}_{\text{unitary transformation}} z$$

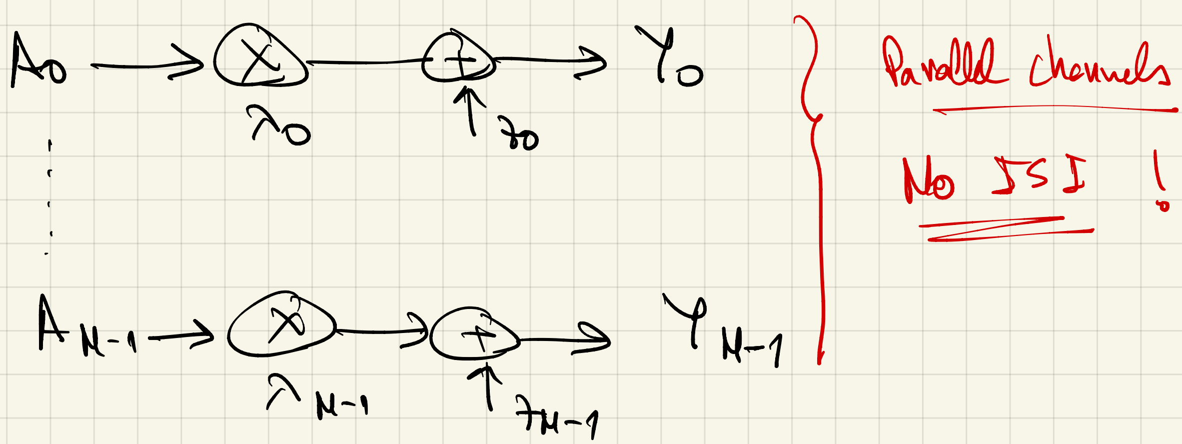
$\Rightarrow$ $\underline{Z}$ has iid zero-mean Gaussian components with variance $\underline{N\sigma^2}$.

From $\underline{A}$ to $\underline{Y}$ the channel is:

$$\underline{Y} = \Delta\,\underline{A} + \underline{Z}$$

Δ = diag $(\lambda_0, \lambda_1 \ldots \lambda_{N-1})$

Pictorially:



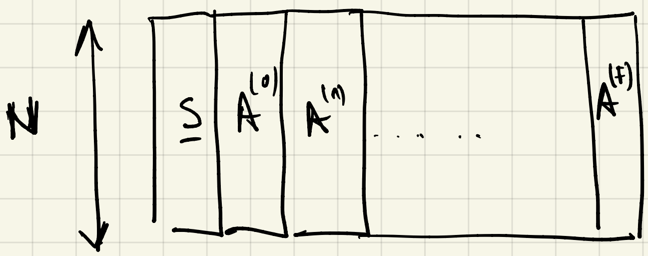
We will see that $\underline{\lambda} = (\lambda_0, \lambda_1 \ldots \lambda_{N-1})$

$$\underline{\lambda} = DFT(\underline{h_0}) \quad \text{where}$$

$\underline{h_0} = (h_{0,0}, h_{0,1} \ldots h_{0,N-1})$ is the symbol-level channel

We estimate $\lambda_0, \lambda_1 \ldots \lambda_{N-1}$ by transmitting known symbols (to the receiver) (training symbols)

Implementation in Matlab | Python

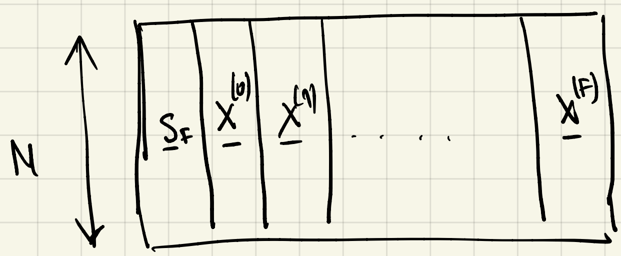


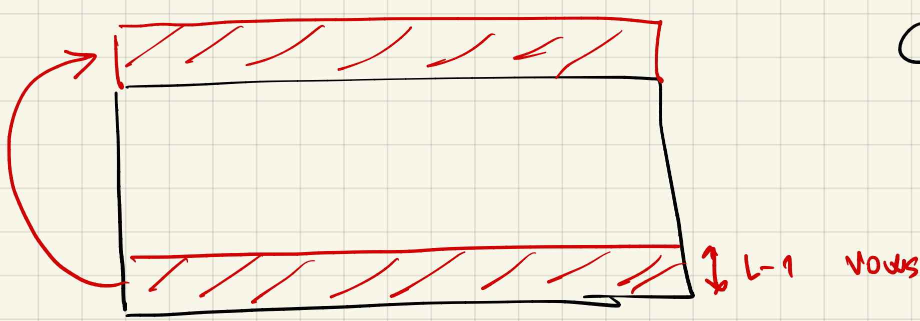
Create a $N \times (F+2)$ matrix of symbols

First column ($\underline{S}$) is the vector of "training" symbols, known to the receiver.

$\Downarrow$ $\underline{\mathrm{DFT}^{-1}}$

For a matrix, can be done column-wise, in one shot, as we need.





Copy-paste cyclic prefix

We serialize the columns and send them to the channel.

At the receiver you do the obvious 'reverse' operations.

Let $\underline{A} \in \mathbb{C}^N$ and $\underline{a} = \frac{1}{N} F \underline{A}$ $\quad \underline{A} = (A_0, A_1, \dots A_{N-1})$

It is convenient to write $\underline{a}$ as follows:

$$\underline{a} = \frac{1}{N}\left[\underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{\substack{\text{samples of} \\ e^{j2\pi f_0 t} \\ \text{with } f_0 = 0}} \times A_0 + \underbrace{\begin{pmatrix} 1 \\ e^{j\frac{2\pi}{N}} \\ \vdots \\ e^{j\frac{2\pi}{N}(N-1)} \end{pmatrix}}_{\substack{\text{samples of} \\ e^{j2\pi f_1 t} \\ \text{with } f_1 = \frac{1}{T} \\ (T = NT_S)}} \times A_1 + \underbrace{\begin{pmatrix} 1 \\ e^{j\frac{2\pi}{N}\cdot 2} \\ \vdots \\ e^{j\frac{2\pi}{N}\cdot 2(N-1)} \end{pmatrix}}_{\substack{\text{samples of} \\ e^{j2\pi f_2 t} \\ \text{with } f_2 = \frac{2}{T}}} \times A_2 + \dots \right]$$

Power Spectral Density (PSD) of the transmitted signal.

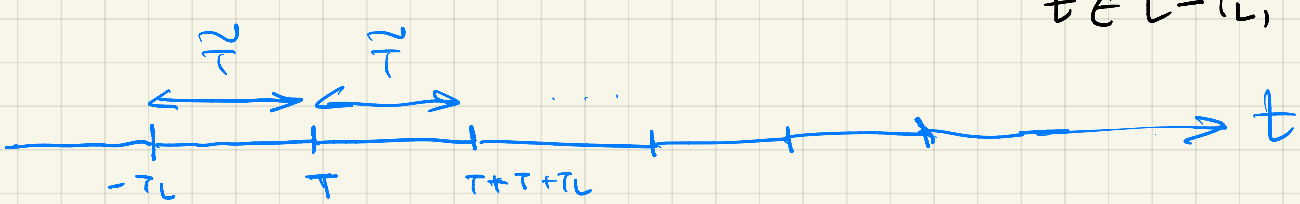
The transmitted signal (baseband equivalent) can be written as follows:

$$s(t) = \sum_{k=0}^{N-1} s_k(t) \quad \text{with}$$

$$s_k(t) = \sum_{i=-\infty}^{\infty} A_k^{(i)} \phi_k(t - i\tilde{T})$$

where $\tilde{T} = T + T_L$ and $\phi_k = \frac{1}{N} e^{j\frac{2\pi}{T}kt}$

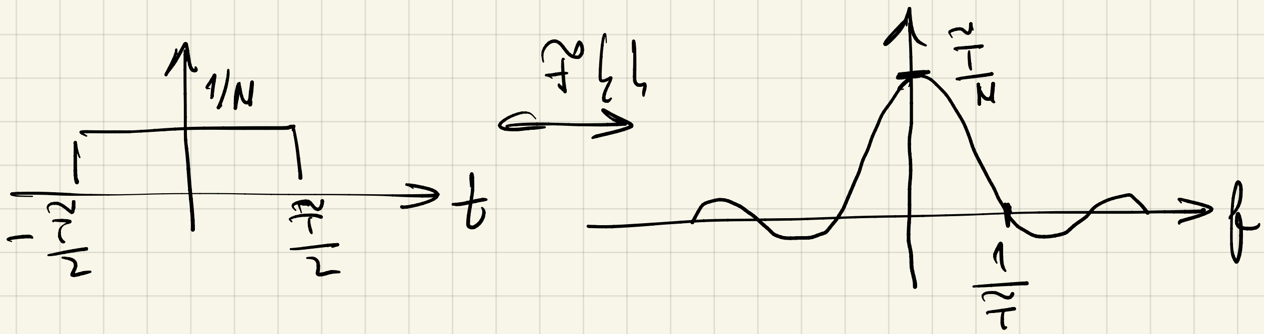
$t \in [-T_L, T]$



The power spectral density of $x_k(t)$ is

$$E|A_k|^2 \, \frac{|\Phi_{k,F}(f)|^2}{\tilde{T}}$$

Recall:



$$\frac{\tilde{T}}{N}\,\mathrm{sinc}(f\tilde{T})$$

For an arbitrary $u(t)$ and $u(t-\tilde{T})$

$$u(t) \overset{\mathcal{F}}{\longleftrightarrow} u_F(f) \qquad u(t-\tilde{T}) \overset{\mathcal{F}}{\longleftrightarrow} u_F(f)\, e^{-j2\pi f \tilde{T}}$$

$$|u_F(f)| = |u_F(f)\, e^{-j2\pi f \tilde{T}}|$$

Hence:

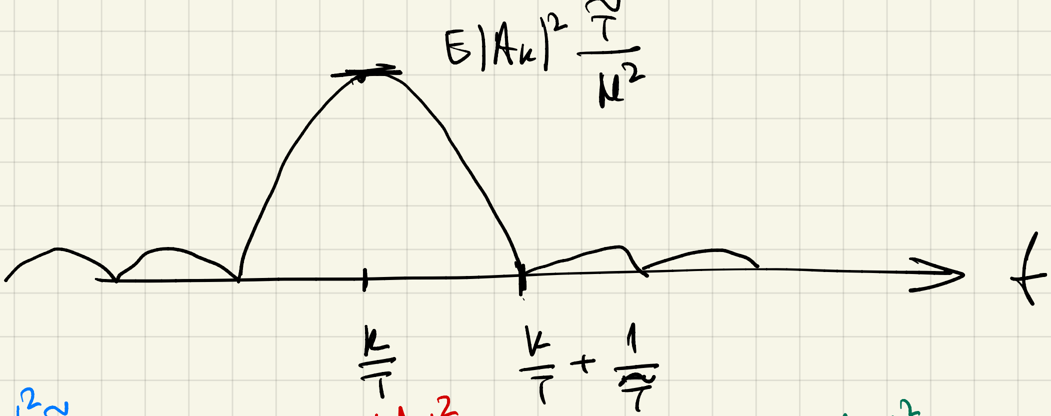
$$|\Phi_{0,F}(f)|^2 = \frac{\tilde{T}^2}{N^2}\,\mathrm{sinc}^2(f\tilde{T})$$

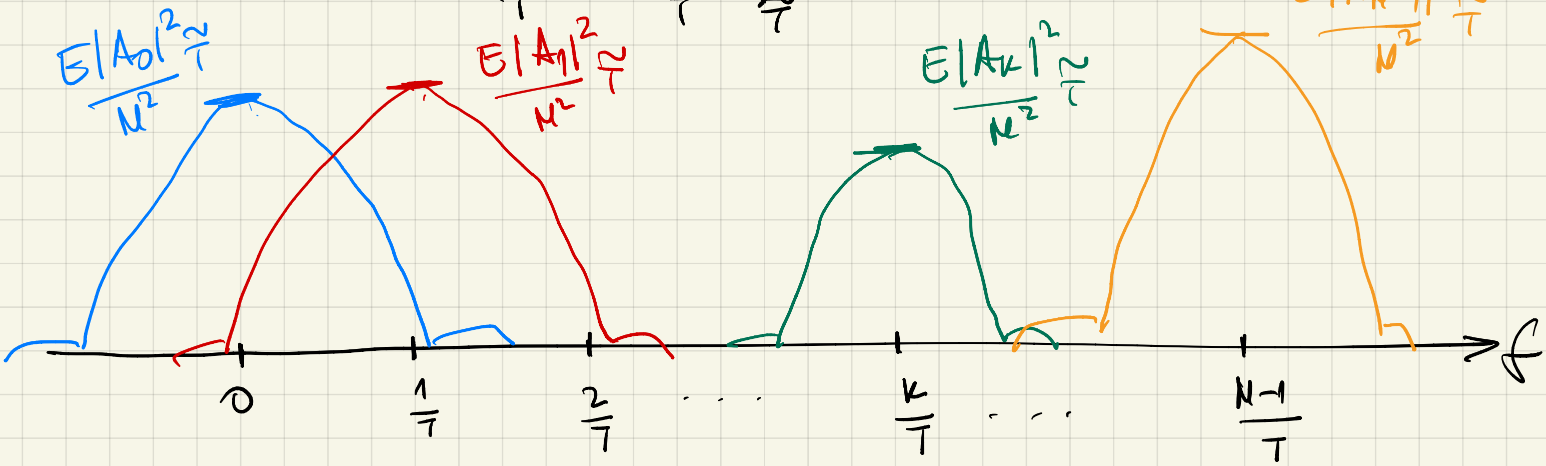
$$|\Phi_{1,F}(f)|^2 = \frac{\tilde{T}^2}{N^2}\,\mathrm{sinc}^2\left(\left(f-\frac{1}{\tilde{T}}\right)\tilde{T}\right)$$

$$\vdots$$

$$|\Phi_{k,F}(f)|^2 = \frac{\tilde{T}^2}{N^2}\,\mathrm{sinc}^2\left(\left(f-\frac{k}{\tilde{T}}\right)\tilde{T}\right)$$

$\Rightarrow$ PSD of $x_k(t) = E|A_k|^2 \frac{\tilde{T}}{N^2}\,\mathrm{sinc}^2\left(\left(f-\frac{k}{\tilde{T}}\right)\tilde{T}\right)$





We can choose the shape of PSD of $s(t)$ by choosing the constellation of A_i, $i=0,\dots M-1$.