

Modern Digital Communications: A Hands-On Approach

OFDM

Dr. Nicolae Chiurtu

- Course material of Prof. Bixio Rimoldi -

Last revision: Nov. 19, 2024

Review and Motivation

The (baseband-equivalent) signaling method of choice for band-limited AWGN channels is¹

$$x(t) = \sum_k a_k \psi(t - kT),$$

where a_k is a sequence of symbols, typically from a QAM constellation, T is the symbol time, and the pulse $\psi(t)$ is unit-norm and such that it is orthogonal to $\psi(t - kT)$ for any nonzero integer k , i.e.,

$$\langle \psi(t), \psi(t - kT) \rangle = \delta_k.$$

If the channel only adds noise, then the channel output is

$$y(t) = x(t) + z(t),$$

where we are assuming that the noise $z(t)$ has independent real and imaginary parts which are both Gaussian, zero-mean, and white.

¹By default, we use capital letter for random variables and random processes, but not in these notes. Here we reserve capital letters for quantities (variable and functions) that are in the Fourier domain.

We like the above signalling method for various reasons, including the following:

- The signal is written as an orthonormal expansion.
- The fact that the orthonormal basis consists of shifted versions of a single pulse, simplifies the implementation. At the sender, we obtain $a_k\psi(t - kT)$ by sending $a_k\delta(t - kT)$ into a filter of impulse response $\psi(t)$. At the receiver, we obtain all the sufficient statistics by sending the received baseband-equivalent signal into the matched filter of impulse response $\psi^*(-t)$ and by sampling the matched filter output at integer multiples of T .
- For a fixed power spectral density of the noise, the error probability is completely determined by the encoder.

- If the symbols produced by the encoder are uncorrelated, which is the case here, then the power spectral density (PSD) of the transmitted signal is, up to scaling, completely determined by the pulse $\psi(t)$ according to

$$S_x(f) = \mathcal{E} \frac{|\psi_{\mathcal{F}}(f)|^2}{T},$$

where $\mathcal{E} = E[|a_k|^2]$ is the symbol energy.

- From the encoder output to the sampled matched filter output, the channel is the symbol-level discrete-time channel

$$y_k = a_k + z_k,$$

where z_k is the k th element of a sequence of i.i.d. circularly symmetric Gaussian random variables.

If the channel is a linear time-invariant (LTI) system of impulse response h , the received signal has the form

$$y(t) = \sum_k a_k p(t - kT) + z(t),$$

where $p = \psi \star h$.

Because typically we do not have control over h , we cannot choose p to be orthogonal to its shifts by integer multiples of T .

The T -spaced samples of the matched filter output at the receiver (matched to ψ) are still a sufficient statistic. (The filtered signal is in the space spanned by the basis that created the original signal.)

The general form of those samples is

$$y_k = \sum_{l=0}^{L-1} a_{k-l} h_l + z_k. \quad (1)$$

Here h_l , $l = 0, 1, \dots, L - 1$ is the impulse response of the discrete-time channel “seen” by the symbols.

Now each y_k depends on $a_k, a_{k-1}, \dots, a_{k-L+1}$, a phenomenon called inter-symbol interference (ISI).

There are several methods to deal with ISI. The most common ones are:

- We undo the ISI by means of an equalizer. A linear equalizer is a linear filter that “undoes” the ISI at the expense of boosting the variance of the noise.

- By means of a maximum-likelihood sequence decoder, implemented with the Viterbi algorithm. This achieves optimal performance in terms of minimizing the block-error probability, but it can be quite complex. Its complexity is proportional to M^L , where M is the size of the symbol alphabet and L the length of the sampled channel-impulse-response.
- We do OFDM, which is a signaling method that prevents ISI from happening. This is an elegant and efficient signaling method, which is adopted in many modern communication standards, including the 4G and 5G wireless standards.

The OFDM concept is developed in the remainder of these notes.

The Fundamental Question Leading to OFDM

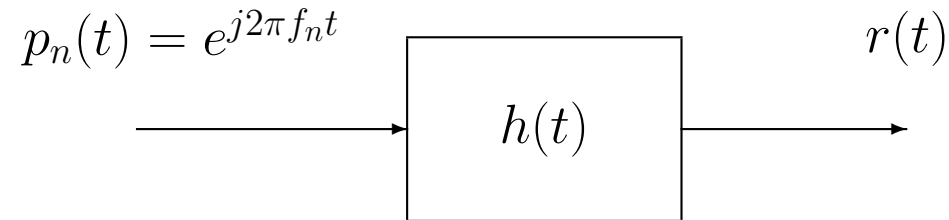
Given that any signaling method can be written as an orthonormal expansion such as

$$x(t) = \sum_{n=0}^{N-1} a_n p_n(t),$$

can we construct a set of pulses p_n , $n = 0, 1, \dots, N - 1$, such that $p_i \star h$ is orthogonal to $p_j \star h$ whenever $i \neq j$? And if yes, can we do it so that it works regardless of the channel impulse response h ?

The answer to both questions is yes, and the functions to be used are the eigenfunctions of the LTI.

Every LTI system has the complex exponentials as eigenfunctions.



$$\begin{aligned} r(t) &= \int_{-\infty}^{\infty} p_n(t - \alpha) h(\alpha) d\alpha \\ &= \int_{-\infty}^{\infty} e^{j2\pi f_n(t - \alpha)} h(\alpha) d\alpha \\ &= e^{j2\pi f_n t} \int_{-\infty}^{\infty} h(\alpha) e^{-j2\pi f_n \alpha} d\alpha \\ &= p_n(t) h_{\mathcal{F}}(f_n). \end{aligned}$$

If the eigenfunction $p_n(t) = \exp\{j2\pi f_n t\}$ is the input to an LTI of impulse response $h(t)$, the output is $p_n(t)\lambda_n$, where $\lambda_n = h_{\mathcal{F}}(f_n)$ is the Fourier transform of $h(t)$ evaluated at f_n .

Hence if we send the sum of weighted pulses²

$$\sum_{n=0}^{N-1} A_n \exp\{j2\pi f_n t\},$$

where the weight is the information-carrying symbol, we receive

$$\sum_{n=0}^{N-1} A_n \lambda_n \exp\{j2\pi f_n t\} + z(t).$$

We see that the symbols have been modified from A_n to $A_n \lambda_n$.

Furthermore, using Parseval's identity, it is clear that two complex exponentials of distinct frequencies are orthogonal to one another. Hence the channel output is still a sequence of weighted pulses that are orthogonal to one another, where the weight is $A_n \lambda_n$.

²For reasons that will become clear, we are now using A_n instead of a_n .

If we can estimate λ_n , we are back to the basic PDC setting that we know so well.

Complex exponentials are only a tentative answer since there are two technical problems: Complex exponentials are neither finite-energy nor of finite support.

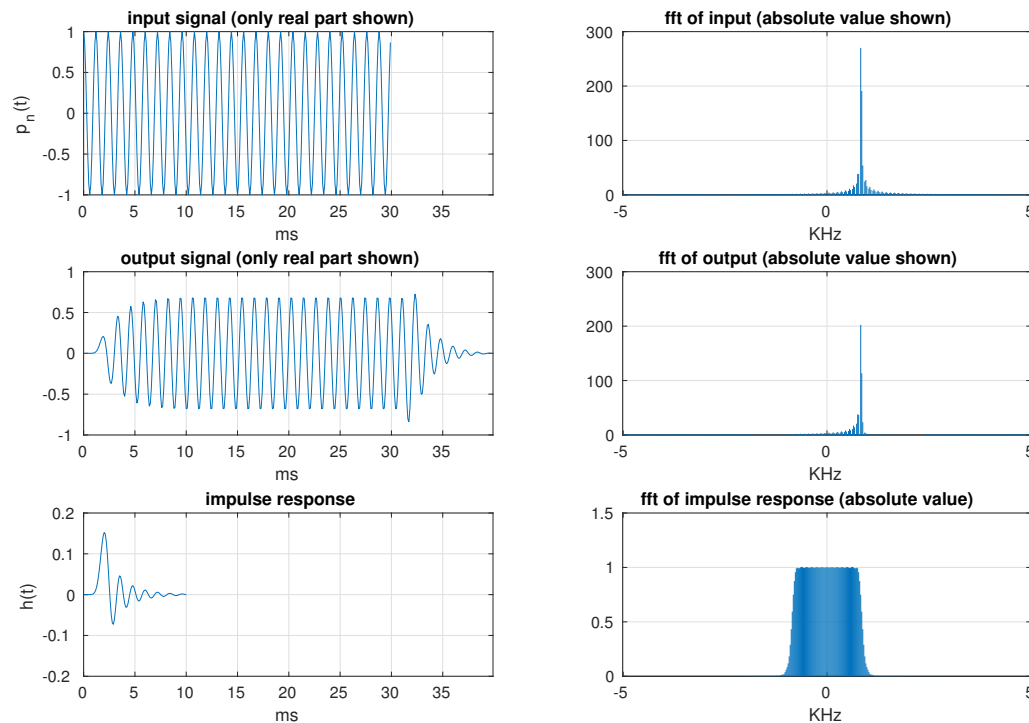
To fix these problems, we choose a fixed time $T = NT_s$, where T_s is the sampling interval and N a positive integer, we truncate the complex exponentials so that their support is $[0, T]$, and we choose the frequencies to be integer multiples of $\frac{1}{T}$, i.e.,

$$p_n(t) = \frac{1}{N} \exp \left\{ j2\pi \frac{n}{T} t \right\} \mathbb{1}_{[0, T]}(t),$$

where $\mathbb{1}_{[0, T]}(t)$ is the indicator function of the interval $[0, T]$, i.e., it is 0 everywhere except for $t \in [0, T]$ where it is 1.

It is straightforward to verify that p_k is orthogonal to p_l when $k \neq l$.

Unfortunately, by truncating the complex exponential we have lost the property that it is an eigenfunction of any LTI system. The following figure is an example of the output of an LTI system when $p_n(t)$ is the input and the impulse response is h . (The frequency-domain figures on the right can be ignored.)



What is important here is that the convolutional integral

$$y(t) = \int_0^{LT_s} x(t - \alpha)h(\alpha)d\alpha,$$

where LT_s is the length of the impulse response h , tells us that the output at some t can be affected only by the input at $t \in [t - LT_s, t]$.

This means that for $t \in [LT_s, T]$, the output $y(t)$ is the same whether the input is the original complex exponential or its truncated counterpart.

So if we send

$$x(t) = \sum_{n=0}^{N-1} A_n p_n(t),$$

for $t \in [LT_s, NT_s]$ we receive

$$y(t) = \sum_{n=0}^{N-1} A_n \lambda_n p_n(t) + z(t).$$

Instead of sending $x(t)$ we send

$$x_C(t) = \tilde{x}(t) + x(t),$$

where

$$\tilde{x}(t) = x(t + NT_s)\mathbb{1}_{[-LT_s, 0]}(t)$$

is obtained by copying the tail of $x(t)$ and pasting it in front of $x(t)$.

Since $x(t)$ is a periodic signal of period T (or an integer fraction thereof), copying its tail and pasting it in front amounts to extending the domain of the truncated complex exponentials from $[0, NT_s]$ to $[-LT_s, NT_s]$. (As we will see, copy-pasting turns out to be more advantageous from an implementation point of view.)

In so doing, the part of the output which is unaffected by the truncation of the complex exponentials is extended to $t \in [0, NT_s] = [0, T]$.

Next, we need to find an efficient way to construct $x_C(t)$. Since we are doing SDR, we just need its samples. In fact, we only need the samples of $x(t)$, since the samples of $\tilde{x}(t)$ are the last $L - 1$ samples of $x(t)$.

Hence we need an efficient way to construct $\mathbf{x} = [x_0, \dots, x_{N-1}]$, where

$$\begin{aligned} x_k = x(kT_s) &= \sum_{n=0}^{N-1} A_n p_n(kT_s) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} A_n \exp \left\{ j2\pi \frac{n}{T} kT_s \right\} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} A_n \exp \left\{ j2\pi \frac{n}{N} k \right\}. \end{aligned}$$

We see that

$$\mathbf{x} = DFT^{-1}(\mathbf{A}),$$

where $\mathbf{A} = (A_0, \dots, A_{N-1})^T$.

Notice that $\mathbf{x} = DFT^{-1}(\mathbf{A})$ can be obtained very efficiently via MATLAB/Python function `ifft`.

So we have an efficient way to construct

$$\begin{aligned}
 x_C(t) &= \tilde{x}(t) + x(t) \\
 &= \underbrace{\text{cyclic extension}}_{t \in [-LT_s, 0]} + \underbrace{\sum_{n=0}^{N-1} A_n p_n(t)}_{t \in [0, T]}.
 \end{aligned}$$

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

$$r(t) = \sum_{n=0}^{N-1} A_n \lambda_n p_n(t) + z(t).$$

Maximum Likelihood Receiver

The waveform former of the sender (PDC terminology) maps the samples into the signal using the interpolation pulse $p(t) = \text{sinc}(\frac{t}{T_s})$, which is a Nyquist pulse.

Hence the receiver front end obtains sufficient statistics by filtering the received signal with the matched filter of impulse response $p^*(-t) = \text{sinc}(\frac{t}{T_s})$ and output sampled at integer multiples of T_s .

Without loss of optimality, we can use a scaled version thereof, namely

$$q(t) = \frac{1}{T_s} \text{sinc}\left(\frac{t}{T_s}\right),$$

which is an ideal lowpass filter of edge-to-edge bandwidth $\frac{1}{T_s}$.

We are going to use only the samples taken at $t = 0, T_s, 2T_s, \dots, (N-1)T_s$, which means that we are discarding the part of the signal which is affected by the truncation of the complex exponential.

Of course this results in a suboptimal receiver, but if L is small compared to N , then we expect the loss to be negligible.

Let the resulting vector of samples be

$$\mathbf{y} = (y_0, y_1, \dots, y_{N-1})^T.$$

Let $\mathbf{Y}$ be the DFT of $\mathbf{y}$. Since the DFT is an invertible transformation, there is no loss of information in this step.

Recall that y_k is the k -th sample of

$$r(t) = \sum_{n=0}^{N-1} A_n \lambda_n p_n(t) + z(t)$$

Hence

$$y_k = \frac{1}{N} \sum_{n=0}^{N-1} A_n \lambda_n \exp \left\{ j \frac{2\pi}{N} n k \right\} + z_k,$$

where z_k is iid, zero-mean, complex Gaussian of variance $\sigma^2 = \frac{N_0}{T_s}$.

In vector form,

$$\mathbf{y} = DFT^{-1}\left((\lambda_0 A_0, \dots, \lambda_{N-1} A_{N-1})^T\right) + \mathbf{z}.$$

Hence

$$\mathbf{Y} = DFT(\mathbf{y}) = (\lambda_0 A_0, \dots, \lambda_{N-1} A_{N-1})^T + \mathbf{Z},$$

where we have defined

$$\mathbf{Z} = DFT(\mathbf{z})$$

as an iid, zero-mean, complex Gaussian vector of variance $\sigma^2 N = \frac{N_0 N}{T_s}$ in each component. The scaling by N follows from the fact that $\frac{1}{\sqrt{N}} DFT(\mathbf{z})$ is a unitary transformation.

To summarize, from $\mathbf{A}$ to $\mathbf{Y}$ the channel is

$$Y_n = A_n \lambda_n + Z_n, \quad n = 0, \dots, N-1.$$

These are N parallel channels that are completely independent. Each channel is an AWGN channel except that the input to the n th channel is scaled by λ_n . The key is that there is no ISI.

In matrix form,

$$\mathbf{Y} = D\mathbf{A} + \mathbf{Z},$$

where

$$D = \text{diag}(\boldsymbol{\lambda}), \quad \boldsymbol{\lambda} = (\lambda_0, \dots, \lambda_{N-1})^T.$$

We will see that

$$\boldsymbol{\lambda} = DFT(\mathbf{h}_0),$$

where $\mathbf{h}_0$ is the symbol-level channel. This information is not essential since we will estimate $\boldsymbol{\lambda}$ from measurements.

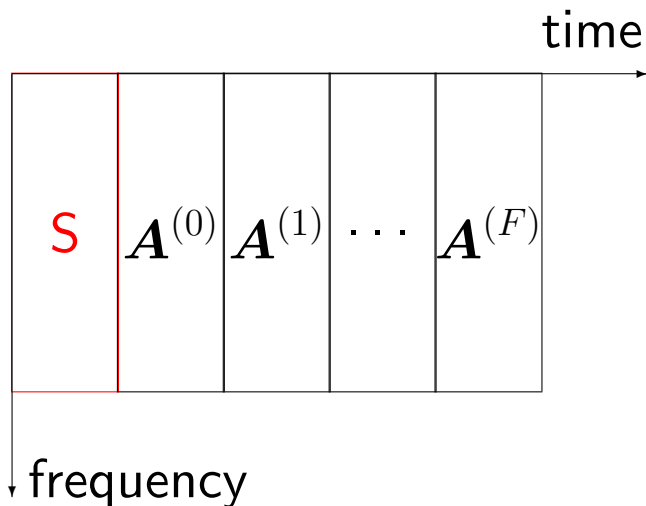
It is time to say that we have focussed on sending one single OFDM block $\mathbf{A}$ of length N by sending a signal $x_C(t)$ that vanishes outside the time interval $[-LT_s, NT_s]$. The procedure can of course be repeated indefinitely by sending the next block of N symbols over the next interval of $\tilde{T} = (L + N)T_s$ seconds.

To help the receiver estimate the vector $\boldsymbol{\lambda}$ of channel coefficients, once in a while we send a block of symbols that are known to the receiver, called training symbols.

OFDM: From A MATLAB/Python Perspective

OFDM can be implemented very efficiently with MATLAB/Python.

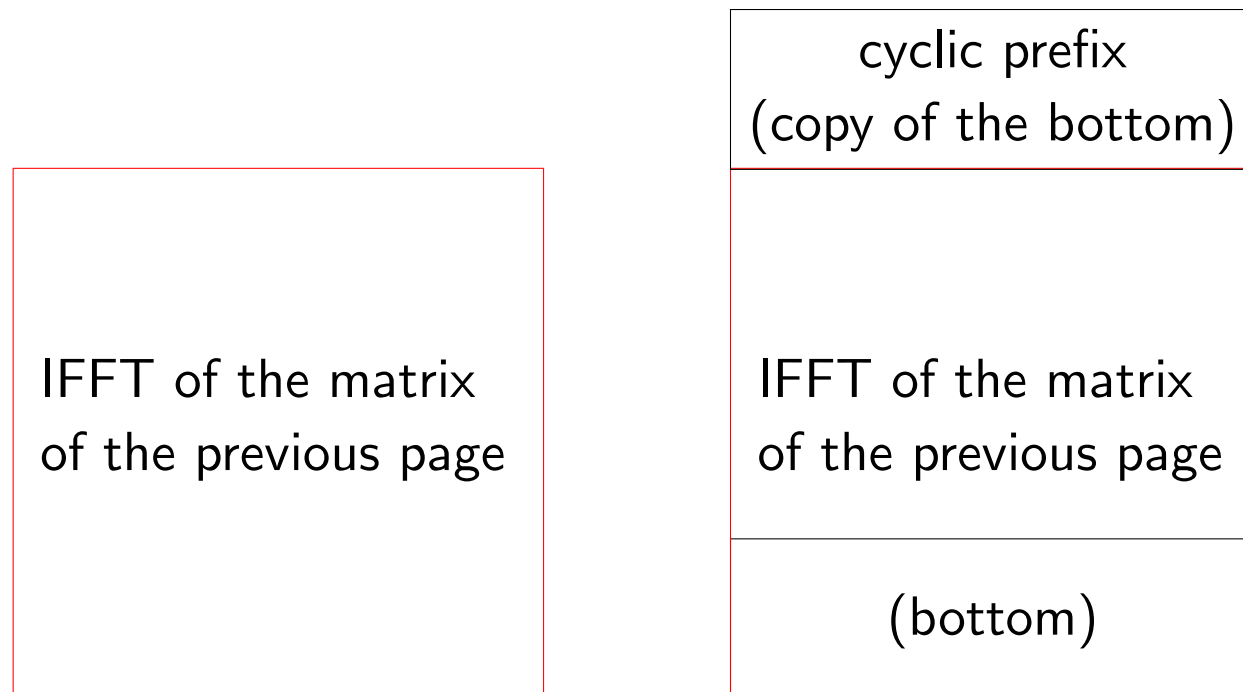
For some frame length F and for some N (power of 2), form an N by $F + 2$ matrix of symbols from the desired constellation (e.g. QAM). The matrix shall have the following form



The first column consists of the training symbols, one symbol per channel.

Next take the inverse DFT of each column. MATLAB/Python do this with one command.

At this point, each column contains the samples needed to generate the corresponding signal without the cyclic extension. The samples of the cyclic extension are obtained by copying the bottom $L - 1$ rows of the matrix and pasting them on top.



By means of the MATLAB command `(:)` (`.flatten('F')` for Python) we finally obtain the vector that contains all the samples of the transmitted signal.

It should now be clear how to organize the corresponding receiver operations.

Appendix A: The DFT as a Matrix

The DFT (discrete Fourier transform) of a vector $\mathbf{u} \in \mathbb{C}^N$ is the vector $\mathbf{U} \in \mathbb{C}^N$ defined by

$$\begin{aligned}\mathbf{U} &= F^\dagger \mathbf{u} \\ F &= (\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_{N-1}) \\ \mathbf{f}_i &= \begin{pmatrix} \beta^{i0} \\ \beta^{i1} \\ \vdots \\ \beta^{i(N-1)} \end{pmatrix} \quad i = 0, 1, \dots, N-1,\end{aligned} \tag{2}$$

where $\beta = \exp \left\{ j \frac{2\pi}{N} \right\}$ is the primitive N -th root of unity in $\mathbb{C}$.

The IDFT (inverse discrete Fourier transform) of $\mathbf{U}$ is $\mathbf{u} = \frac{1}{N} F \mathbf{U}$.

This definition of DFT corresponds to the FFT defined in Matlab/Python. Notice that $FF^\dagger = NI$ where I is the identity matrix (and $F = F^T$). Hence $\frac{1}{\sqrt{N}}F$ is a unitary transformation.

Appendix B: A Convenient Interpretation

Let $\mathbf{U} \in \mathbb{C}^N$ and $\mathbf{u} = \frac{1}{N}F\mathbf{U}$ be its inverse DFT, which is what we send to the reconstruction device (the waveform former). It is convenient to picture $\mathbf{u}$ as follows³

$$\mathbf{u} = \frac{1}{N} \left[\underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{\substack{\text{samples} \\ \text{of} \\ e^{j2\pi ft} \\ \text{with} \\ f = 0}} \times U_0 + \underbrace{\begin{pmatrix} 1 \\ e^{j\frac{2\pi}{N}} \\ \vdots \\ e^{j\frac{2\pi}{N}(N-1)} \end{pmatrix}}_{\substack{\text{samples} \\ \text{of} \\ e^{j2\pi ft} \\ \text{with} \\ f = \frac{1}{T}}} \times U_1 + \cdots + \underbrace{\begin{pmatrix} 1 \\ e^{j\frac{2\pi}{N}(N-1)} \\ \vdots \\ e^{j\frac{2\pi}{N}(N-1)(N-1)} \end{pmatrix}}_{\substack{\text{samples} \\ \text{of} \\ e^{j2\pi ft} \\ \text{with} \\ f = \frac{N-1}{T}}} \times U_{N-1} \right]$$

³For convenience, here we use the form e^{jx} instead of $\exp\{jx\}$.

By sending $\mathbf{u}$ to the waveform former (reconstruction device), we are producing a linear combination of complex exponentials, where the n -th complex exponential, $n = 0, \dots, N - 1$, has frequency $\frac{n}{T}$ and its coefficient is U_n .

Appendix C: The Power Spectral Density

The transmitted signal (baseband-equivalent) can be written as

$$x(t) = \sum_{k=0}^{N-1} x_k(t)$$

with

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

where

$$\tilde{T} = T + T_s L$$

and

$$\phi_k(t) = \frac{1}{N} \exp \left\{ j \frac{2\pi}{T} k t \right\} \mathbb{1}_{[-LT_s, T]}(t).$$

The power spectral density (PSD) of the random process $X_k(t)$ is (see e.g. PDC)

$$E [|A_k|^2] \frac{|\Phi_k(f)|^2}{\tilde{T}},$$

where $\Phi_k(f)$ is the Fourier transform of $\phi_k(t)$.

Recall the following Fourier transform pairs:

$$\begin{aligned} \frac{1}{N} \mathbb{1}_{\left[-\frac{\tilde{T}}{2}, \frac{\tilde{T}}{2}\right]}(t) &\stackrel{\mathcal{F}}{\longleftrightarrow} \frac{\tilde{T}}{N} \text{sinc}(f\tilde{T}) \\ u(t - \tilde{T}) &\stackrel{\mathcal{F}}{\longleftrightarrow} U(f)e^{-j2\pi\tilde{T}f}, \end{aligned}$$

where $U(f)$ is the Fourier transform of $u(t)$.

Hence

$$\begin{aligned} |\Phi_0(f)|^2 &= \left(\frac{\tilde{T}}{N}\right)^2 \text{sinc}^2(f\tilde{T}) \\ |\Phi_1(f)|^2 &= \left(\frac{\tilde{T}}{N}\right)^2 \text{sinc}^2\left(\left(f - \frac{1}{T}\right)\tilde{T}\right) \\ &\vdots \\ |\Phi_k(f)|^2 &= \left(\frac{\tilde{T}}{N}\right)^2 \text{sinc}^2\left(\left(f - \frac{k}{T}\right)\tilde{T}\right). \end{aligned}$$

The PSD of $x_k(t)$ is thus

$$E[|A_k|^2] \frac{\tilde{T}}{N^2} \text{sinc}^2\left(\left(f - \frac{k}{T}\right)\tilde{T}\right)$$

By summing over all k , we obtain the PSD of $x(t)$.

Notice that the PSD of $x(t)$ can be shaped by choosing an appropriate constellation for A_n , $n = 0, \dots, N - 1$.

