Modern Digital Communications: A Hands-On Approach

OFDM

Prof. Bixio Rimoldi

Last revision: Sept. 13, 2017

Review and Motivation

When possible, the signaling method of choice for band-limited AWGN channels is one for which the baseband equivalent channel output signal has the form

$$Y(t) = \sum_k a[k]\psi(t - kT) + Z(t),$$

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

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

and $Z(t)$ is the additive Gaussian noise (assumed white for simplicity).

For continuity with PDC, in the above expression we are using capital letters for random variables and random processes and lowercase letters for their realization and for deterministic waveforms. (Later on we will change convention and use capital letters for the Discrete Fourier Transform of an N-tuple.)

The above signaling method is particularly convenient since with a single matched filter we obtain all the projections that constitute sufficient statistics.

The kth projection yields

$$Y[k] = a[k] + Z[k]$$

where $Z[k]$ is the kth element of a sequence of i.i.d. Gaussian random variables. Thus it is as if we had sent the symbols through an ideal discrete-time AWGN channel.

If the channel impulse response h is not known to the sender, the received signal has the form

$$Y(t) = \sum_k a[k]p(t - kT) + Z(t),$$

where $p = \psi \star h$

Because 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

$$Y[k] = \sum_{l=0}^{L-1} a[k-l]h[l] + Z[k] \tag{1}$$

Here $h[k]$, $k = 0, 1, \ldots, L-1$ is the impulse response of the discrete-time channel "seen" by the symbols.

Now each $Y[k]$ depends on $a[k], a[k-1], \ldots, a[k-L+1]$, a phenomenon called inter-symbol interference (ISI).

An optimal thing to do is then to use a Viterbi decoder to make a maximum likelihood decision. Sometimes (indeed often) the complexity of the Viterbi algorithm is prohibitive.

The alternative that we pursue hereafter is to use signals that allow us to avoid ISI.

Fundamental question: how can we construct pulses p_n, $n = 1, 2, \ldots, N$, such that $p_i * h$ is orthogonal to $p_j * h$ whenever $i \neq j$, regardless of the channel impulse response h?

Answer: every linear time invariant (LTI) system has the complex exponentials as eigenfunctions.

If the eigenfunction $p_n(t) = \exp\{j2\pi f_n t\}$ is the input to an LTI of impulse response h, the output is $p_n(t)h_{\mathcal{F}}(f_n)$ where $h_{\mathcal{F}}$ is the Fourier transform of h. Notice that $h_{\mathcal{F}}(f_n)$ is just a scaling factor.

Furthermore, $\exp\{j2\pi f_k t\}$ and $\exp\{j2\pi f_l t\}$ are orthogonal whenever $f_k \neq f_l$.

Hence the output $h_{\mathcal{F}}(f_k)\exp\{j2\pi f_k t\}$ and $h_{\mathcal{F}}(f_l)\exp\{j2\pi f_l t\}$ are orthogonal to each other.

So why not choose distinct frequencies $f_1, \ldots, f_N$ and send signals of the form

$$s(t) = \sum_{k=1}^{N} a[k] \exp\{j2\pi f_k t\}.$$

We will do something similar. Notice that the above information symbols are modulating carriers placed at different frequencies and orthogonal to one another.

Hence it would be quite appropriate if the above technique were called *orthogonal frequency division multiplexing* (OFDM).

The above idea is a good one but its implementation has drawbacks (see next page). The term OFDM is reserved to its discrete-time counterpart which can be implemented efficiently.

The complex exponentials have infinite duration. This is a problem that we can avoid by truncating the complex exponentials to finite duration. They will still be orthogonal to one another if we choose the frequencies as a function of the duration. For instance we could choose

$$s(t) = \sum_{k=1}^{N} a[k] \exp\left\{ j2\pi \frac{k}{T} t \right\}$$

for $t \in [0, T]$ and $s(t) = 0$ elsewhere. There are still two problems:

1. The truncated complex exponentials are no longer eigenfunctions of LTI systems. Hence the channel will modify (not just scale) the truncated complex exponentials and their orthogonality can no longer be guaranteed.
2. The LTI system at hand has eigenfunctions that we could use, but there is no reason why they should be shifted versions of one another. Hence, for each eigenfunction, we may need a separate correlator or matched filter to implement the receiver front end. In practice it means having N oscillators. This is quite unpractical for large values of N, say $N = 1000$.

There is an elegant solution to both problems.

As a preview, the solution to the first problem is to appropriately extend the complex exponentials in such a way that the channel output over the interval $[0, T]$ is indistinguishable from that produced by the un-truncated (infinite-length) complex exponentials. To do so the length of the extension has to be the length of the channel impulse response.

We will take care of the second problem in a clever way by means of a signal processing tool that has a fast implementation. You are already familiar with it: the fast Fourier transform (FFT).

Starting the above discussion in the continuous-time domain has the advantage of letting us develop the intuition in the domain that we know best.

For the details, it is best to start afresh from the discrete-time domain.

Along the way we will learn that there is a discrete-time finite-length counterpart to the fact that infinite-length complex exponentials are eigenfunctions of continuous-time LTI systems.

OFDM: A Discrete-Time Approach

Seen from the “matched-filter” output, the symbol-level channel-model is

$$y[k] = \sum_{l=0}^{L-1} a[k-l]h[l] + z[k], \tag{2}$$

where the noise is i.i.d. and Gaussian and $h[0], h[1], \ldots, h[L-1]$ is the symbol-level-equivalent channel impulse response.

Notice that we are no longer using capital letters for random variables. From now on we reserve capital letters for the DFT domain. (Whether a quantity is deterministic or random should be clear from the context.)

The key about the above model is that there is ISI.

We break down the symbol-level input/output sequences into blocks of length $N > L$ (usually $N \gg L$).

We let the mth input and output blocks be represented by the column vectors $\boldsymbol{a}^{(\boldsymbol{m})} = (a[mN], \ldots, a[mN + N - 1])^T$ and $\boldsymbol{y}^{(\boldsymbol{m})} = (y[mN], \ldots, y[mN + N - 1])^T$, respectively.

Due to ISI, $\boldsymbol{y}^{(\boldsymbol{m})}$ does not only depend on $\boldsymbol{a}^{(\boldsymbol{m})}$ but also on the last $L - 1$ symbols of $\boldsymbol{a}^{(\boldsymbol{m-1})}$. Let those symbols be denoted by $\tilde{\boldsymbol{a}}^{(m-1)}$. Ignoring the noise, we may now write

$$\boldsymbol{y}^{(\boldsymbol{m})} = H \begin{pmatrix} \tilde{\boldsymbol{a}}^{(m-1)} \\ \boldsymbol{a}^{(m)} \end{pmatrix}$$

where H is the $N \times (N + L - 1)$ (Toepliz) matrix defined by

$$H = \begin{pmatrix} h[L-1] & \dots & h[0] & 0 & 0 & \dots & 0 \\ 0 & h[L-1] & \dots & h[0] & 0 & \dots & 0 \\ \vdots & & & & & & \\ 0 & \dots & 0 & h[L-1] & \dots & h[0] & 0 \\ 0 & \dots & 0 & 0 & h[L-1] & \dots & h[0] \end{pmatrix}.$$

(Using a matrix to describe the input/output relationship of an LTI seems an overkill. In fact, the matrix needs as many columns as the number of input symbols and as many rows as the number of output symbols. Moreover, every such row contains the same information, namely the LTI's impulse response. We are doing so since we will do FFTs on blocks of inputs. The FFT is linear but not time-invariant. Hence the resulting behavior can not be described by an impulse response alone.)

An obvious way to prevent the ISI from propagating across block boundaries is to insert $L-1$ zeros between any two adjacent blocks of symbols.

Here is what goes through the discrete-time (symbol-level) channel if we do so.

$(\boldsymbol{a}^{(m-1)})^{\boldsymbol{T}}$	$0, \ldots, 0$	$(\boldsymbol{a}^{(m)})^{\boldsymbol{T}}$	$0, \ldots, 0$

This means loosing $L-1$ out of $N+L-1$ channel uses and this loss can be made as small as we wish by making N large.

In what follows, $\boldsymbol{a}^{(\boldsymbol{m})}$ is exactly as before we inserted the zeros, and $\boldsymbol{y}^{(\boldsymbol{m})}$ is the block that lines up with $\boldsymbol{a}^{(m)}$. Mathematically,

$$\begin{aligned}\boldsymbol{y}^{(\boldsymbol{m})} &= H\begin{pmatrix}\mathbf{0}\\ \boldsymbol{a}^{(m)}\end{pmatrix}\\ &= H_r\boldsymbol{a}^{(m)}\end{aligned}$$

where the subscript r to H (r for remove) denotes that the block consisting of the $L-1$ leftmost columns has been removed. The matrix H_r is $N \times N$.

Now comes a crucial point. For a reason that will soon become clear, in front of $\boldsymbol{a}^{(m)}$ we insert $\tilde{\boldsymbol{a}}^{(m)}$ which has also length $L-1$. (See the figure.)

$(\boldsymbol{a}^{(\boldsymbol{m}-1)})^{\boldsymbol{T}}$	$(\tilde{\boldsymbol{a}}^{(\boldsymbol{m})})^{\boldsymbol{T}}$	$(\boldsymbol{a}^{(\boldsymbol{m})})^{\boldsymbol{T}}$	$(\tilde{\boldsymbol{a}}^{(\boldsymbol{m}+1)})^{\boldsymbol{T}}$
	tail of $\boldsymbol{a}^{(\boldsymbol{m})}$		tail of $\boldsymbol{a}^{(\boldsymbol{m}+1)}$

With this modification,

$$
\begin{aligned}
\boldsymbol{y^{(m)}} &= H \begin{pmatrix} \tilde{\boldsymbol{a}}^{(m)} \\ \boldsymbol{a}^{(m)} \end{pmatrix} \\
&= H_c \boldsymbol{a}^{(m)}
\end{aligned}
$$

where H_c is obtained by removing the first $L-1$ columns of H and adding them to the last $L-1$ columns:

$$
H = \begin{pmatrix}
h[L-1] & \dots & h[0] & 0 & 0 & \dots & 0 \\
0 & h[L-1] & \dots & h[0] & 0 & \dots & 0 \\
\vdots & & & & & & \\
0 & \dots & 0 & h[L-1] & \dots & h[0] & 0 \\
0 & \dots & 0 & 0 & h[L-1] & \dots & h[0]
\end{pmatrix}.
$$

$$
H_c = \begin{pmatrix}
h[0] & 0 & 0 & \dots & 0 & h[L-1] & \dots & h[2] & h[1] \\
h[1] & h[0] & 0 & \dots & 0 & 0 & h[L-1] & \dots & h[2] \\
\vdots & & & & & & & & \vdots \\
0 & 0 & 0 & \dots & h[L-1] & h[L-2] & \dots & h[1] & h[0]
\end{pmatrix}.
$$

H_c is the $N \times N$ *circulant* matrix, i.e., a matrix for which each row is obtained by *circularly* shifting the previous row by one position to the right.

One can easily verify that the input/output relationship of a circulant matrix is described by a *circular convolution*.

Circulant matrices are the matrix equivalent of linear time invariant (LTI) system: They describe linear maps (all matrices do) that are also shift invariant in the sense that a circular shift to the input results in a circular shift in the output.

The equivalence goes rather far. Recall that complex exponentials are eigenfunctions of LTIs. Similarly, for every integer $i = 0, 1, \ldots, N-1$, the complex exponential $(\beta^{i0}, \beta^{i1}, \ldots, \beta^{i(N-1)})^T$, where $\beta = \exp(j\frac{2\pi}{N})$ is a primitive N-th root of unity, is an eigenvector to any $N \times N$ circulant matrix.

For a proof of the above and other interesting facts, you are encouraged to read the appendices.

The input/output relationship of an LTI takes a particularly simple form in the frequency domain: The (Fourier) transform of the output is the transform of the input times the transform of the impulse response.

Similarly, if the output N-tuple relates to the input N-tuple via a circulant matrix H_c then the DFT of the output is the DFT of the input times (componentwise) the DFT of the first column of H_c (the channel impulse response).

The above result should not be new to you: you should know that circular convolution becomes a product in the DFT domain.

The bottom line for us is:

If $\boldsymbol{Y}^{(m)}$ is the DFT of $\boldsymbol{y}^{(m)}$ and $\boldsymbol{A}^{(m)}$ the DFT of $\boldsymbol{a}^{(m)}$ then

$$\boldsymbol{Y}^{(m)} = D\boldsymbol{A}^{(m)} + \boldsymbol{Z}^{(m)}$$

where D is the diagonal matrix that has the components of $\boldsymbol{\lambda}$ on its diagonal and $\boldsymbol{\lambda}$ is the DFT of the first column of H_c (i.e., the channel impulse response).

The vector $\boldsymbol{Z}^{(m)}$ is the DFT of $\boldsymbol{z}^{(m)}$. Both $\boldsymbol{Z}^{(m)}$ and $\boldsymbol{z}^{(m)}$ are i.i.d. and Gaussian. Due to the fact that the DFT is not unitary, in going from $\boldsymbol{z}^{(m)}$ to $\boldsymbol{Z}^{(m)}$ the variance has increased by a factor N.

We Summarize

Take the source bits and make N-length blocks of QAM symbols. Let $\boldsymbol{A}^{(m)}$ be the mth such block.

Take the inverse DFT (IDFT) of $\boldsymbol{A}^{(m)}$. Call it $\boldsymbol{a}^{(m)}$.

Stick the prefix $\tilde{\boldsymbol{a}}^{(m)}$ on top of $\boldsymbol{a}^{(m)}$ to form $\begin{pmatrix} \tilde{\boldsymbol{a}}^{(m)} \\ \boldsymbol{a}^{(m)} \end{pmatrix}^T$.

Send $\ldots \begin{pmatrix} \tilde{\boldsymbol{a}}^{(m-1)} \\ \boldsymbol{a}^{(m-1)} \end{pmatrix}^T, \begin{pmatrix} \tilde{\boldsymbol{a}}^{(m)} \\ \boldsymbol{a}^{(m)} \end{pmatrix}^T \ldots$ over the discrete-time symbol-level channel.

Let $\boldsymbol{y}^{(m)}$ be the N-tuple of channel output symbols that line up with the symbols in $\boldsymbol{a}^{(m)}$.

Take the DFT of $\boldsymbol{y}^{(m)}$ to obtain $\boldsymbol{Y}^{(m)}$.

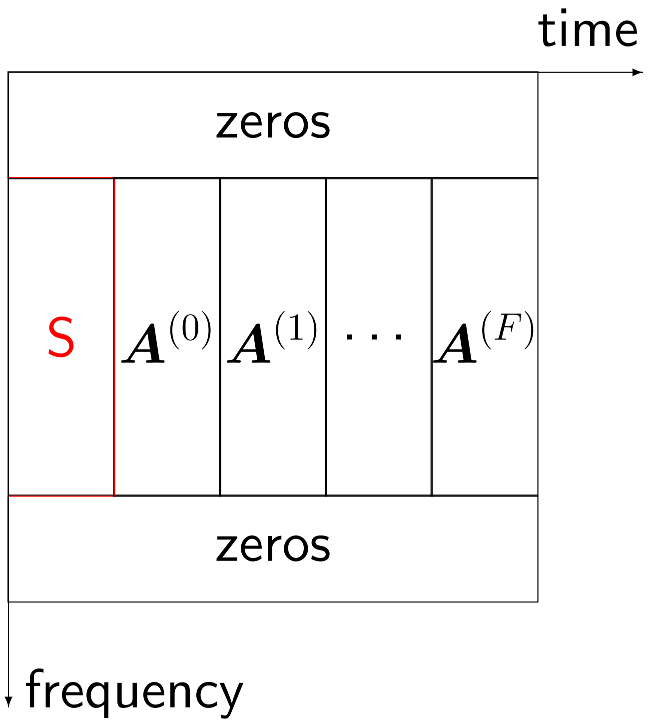
Since $\boldsymbol{Y}^{(m)} = D\boldsymbol{A}^{(m)} + \boldsymbol{Z}^{(m)}$ where D is diagonal, we see a discrete channel which is free of inter-symbol interference.

We still have to show how to estimate the diagonal matrix D. We do this in the next lecture. In a first pass of the implementation, you should test/debug your system using the actual D (as opposed to its estimate).

OFDM: From A MATLAB Perspective

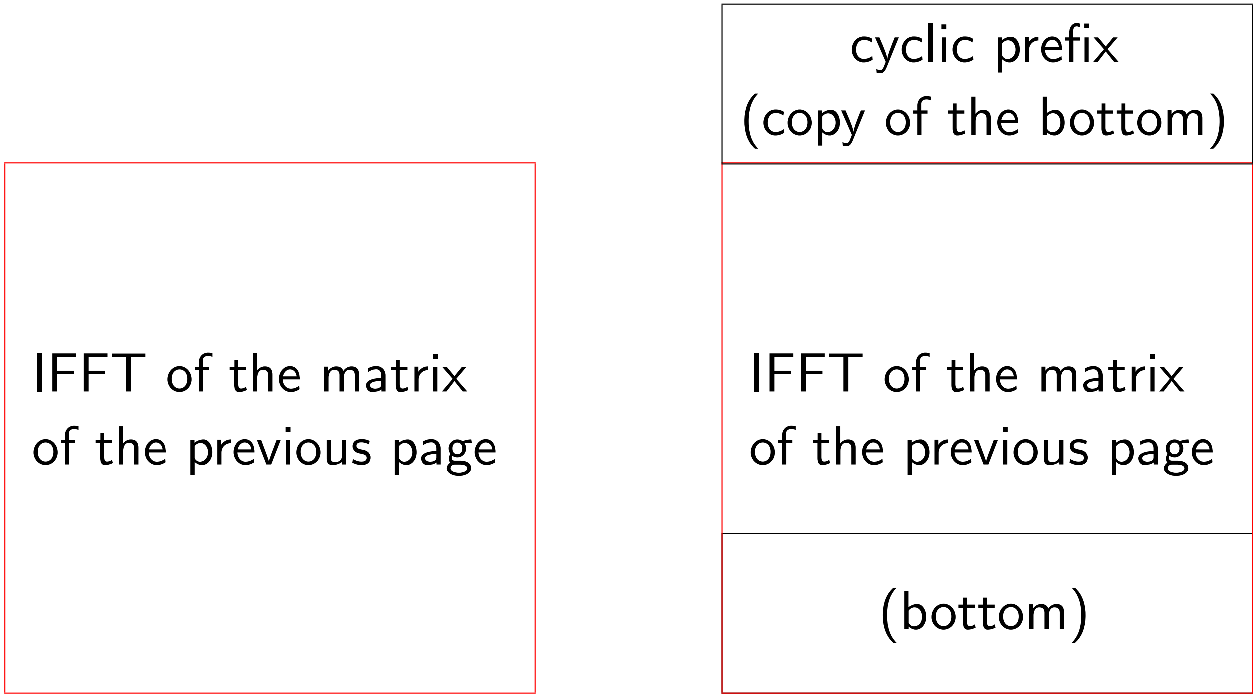
OFDM can be implemented very efficiently with MATLAB.

For some frame length F and for some N (power of 2), form an N by $F+1$ matrix of symbols from the desired constellation (e.g. QAM). The matrix shall have the following form



The first column consists of the training block, and the top and bottom may (or may not) have one or more rows of zeros. Zeros may be used to switch off the carriers at the two extremes of the spectrum, thereby controlling the amount of energy that spills out of the assigned frequency interval (see Appendix D).

Next take the IFFT and then add on top the cyclic prefix by copying the L bottom rows:



By means of the MATLAB command `(:)` we finally obtain the column vector of channel symbols $a[k]$.

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

Note that at the sender (and at the receiver) we have created F columns just because MATLAB allows us to take their IDFT in one shot. So we do it for pure convenience. When we developed the concept behind OFDM we were thinking of processing one column at a time.

Channel Coefficients Estimation

We have seen that by using OFDM we 'forge' parallel channels described in matrix form as

$$\boldsymbol{Y}^{(m)} = D\boldsymbol{A}^{(m)} + \boldsymbol{Z}^{(m)}$$

where D is the diagonal matrix of channel coefficients. The question is how to estimate the matrix D?

If, for certain values of m, we substitute $\boldsymbol{A}^{(m)}$ with an N-tuple $\boldsymbol{S}$ known to the receiver then

$$\boldsymbol{Y}^{(m)} = D\boldsymbol{S} + \boldsymbol{Z}^{(m)}$$

where D is diagonal. The same result is obtained by

$$\boldsymbol{Y}^{(m)} = S\boldsymbol{D} + \boldsymbol{Z}^{(m)}$$

where S is the diagonal matrix that has $\boldsymbol{S}$ as its diagonal elements and $\boldsymbol{D}$ is the N-tuple consisting of the diagonal elements of D.

Assume that $\boldsymbol{D}$ is zero-mean, Gaussian, and independent of $\boldsymbol{Z}^{(m)}$.

Hence $\boldsymbol{Y}^{(m)}$ and $\boldsymbol{D}$ are jointly Gaussian and, the minimum mean squared error (MMSE) estimate[1] of $\boldsymbol{D}$ based on the observable $\boldsymbol{Y}^{(m)}$ is

$$\hat{\boldsymbol{D}} = K_{\boldsymbol{DY}} K_{\boldsymbol{Y}}^{-1} \boldsymbol{Y}^{(m)},$$

where

$$\begin{aligned} K_{\boldsymbol{DY}} &\coloneqq E[\boldsymbol{D}\boldsymbol{Y}^{\dagger}] = K_{\boldsymbol{D}} S^{\dagger} \quad \text{and} \\ K_{\boldsymbol{Y}} &\coloneqq E[\boldsymbol{Y}\boldsymbol{Y}^{\dagger}] = S K_{\boldsymbol{D}} S^{\dagger} + K_{\boldsymbol{Z}} \end{aligned}$$

are covariance matrices. ($\boldsymbol{D}$ and $\boldsymbol{Y}$ are zero-mean and we have dropped the superscript m to $\boldsymbol{Y}^{(m)}$ since we are assuming that the statistic is the same for all blocks.) Recall that S is the diagonal matrix that consists of the elements of $\boldsymbol{S}$.

[1] See the lecture notes on MMSE estimation

Both $K_{\boldsymbol{Y}}$ and $K_{\boldsymbol{DY}}$ depend on $K_{\boldsymbol{D}}$. The question is how to find $K_{\boldsymbol{D}}$.

The most rigorous way is to determine $K_{\boldsymbol{D}}$ from the channel model. See Appendix C for an example.

Another possibility (that does not require modeling) is to estimate $K_{\boldsymbol{Y}}$ from the data and then use it in the following expression

$$
\begin{aligned}
\hat{\boldsymbol{D}} &= K_{\boldsymbol{DY}} K_{\boldsymbol{Y}}^{-1} \boldsymbol{Y}^{(m)} \\
&= E[\boldsymbol{D}\boldsymbol{Y}^{\dagger}] K_{\boldsymbol{Y}}^{-1} \boldsymbol{Y}^{(m)} \\
&= E[S^{-1}(\boldsymbol{Y} - \boldsymbol{Z})\boldsymbol{Y}^{\dagger}] K_{\boldsymbol{Y}}^{-1} \boldsymbol{Y}^{(m)} \\
&= S^{-1}(K_{\boldsymbol{Y}} - K_{\boldsymbol{Z}}) K_{\boldsymbol{Y}}^{-1} \boldsymbol{Y}^{(m)} \\
&= S^{-1}(I - K_{\boldsymbol{Z}} K_{\boldsymbol{Y}}^{-1}) \boldsymbol{Y}^{(m)},
\end{aligned}
$$

where we have used twice the relationship $\boldsymbol{Y}^{(m)} = S\boldsymbol{D} + \boldsymbol{Z}^{(m)}$.

A few comments are in order.

The matrix S is diagonal. Hence S^{-1} is obtained by substituting each diagonal element with its reciprocal.

To determine $K_{\boldsymbol{Y}}^{-1}\boldsymbol{Y}^{(m)}$ there is no need to invert the matrix. Instead you may rely on the following general idea.

Let say that we want to find $B^{-1}\boldsymbol{y}$ for some general matrix B and vector $\boldsymbol{y}$. The solution is the $\boldsymbol{x}$ that solves the equation $\boldsymbol{y} = B\boldsymbol{x}$. Rather than inverting B, which works only if B is non-singular, Matlab is capable of finding the $\boldsymbol{x}$ that minimizes the norm of the difference $\boldsymbol{y} - B\boldsymbol{x}$.

This is done with the MATLAB command `B\y` (notice the backslash).

If the equation $\boldsymbol{y} = B\boldsymbol{x}$ has many solutions, Matlab will find the $\boldsymbol{x}$ that has the fewest nonzero components. If it has no solution, Matlab will find the best approximation in the sense mentioned above.

It should be said that MATLAB finds the answer more efficiently than through the inverse of B even when it exists. See `doc mldivide` for more.

Bottom line: use `B\y` instead of `inv(B)*y`.

Appendix A: Toepliz, and Circulant Matrices

A linear transformation from $\mathbb{C}^N$ to $\mathbb{C}^N$ can be described by an $N \times N$ matrix H. If the matrix is $Toepliz$, meaning that $H_{ij} = h_{i-j}$, then the transformation which sends $\boldsymbol{u} \in \mathbb{C}^N$ to $\boldsymbol{v} = H\boldsymbol{u}$ can be described by the $convolution$ sum

$$v_i = \sum_k h_{i-k} u_k.$$

A Toepliz matrix is a matrix which is constant along its diagonals.

A circulant matrix is a special kind of Toepliz matrix.

A matrix H is *circulant* if $H_{ij} = h_{[i-j]}$ where here and hereafter the operator $[\cdot]$ applied to an index denotes the index taken modulo N.

When H is circulant, the operation that maps $\boldsymbol{u}$ to $\boldsymbol{v} = H\boldsymbol{u}$ may be described by the *circulant convolution*

$$v_i = \sum_k h_{[i-k]} u_k.$$

A circulant matrix H is completely described by its first column $\boldsymbol{h} = (h_0, h_1, \ldots, h_{(N-1)})^T$, or by its first row $\boldsymbol{r} = (h_0, h_{[-1]}, \ldots, h_{[-(N-1)]})$, or by any other row or column.

The DFT as a Matrix

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

$$
\begin{aligned}
\boldsymbol{U} &= F^{\dagger}\boldsymbol{u} \\
F &= (\boldsymbol{f}_0, \boldsymbol{f}_1, \ldots, \boldsymbol{f}_{N-1}) \\
\boldsymbol{f}_i &= \begin{pmatrix} \beta^{i0} \\ \beta^{i1} \\ \vdots \\ \beta^{i(N-1)} \end{pmatrix} \qquad i = 0, 1, \ldots, N-1,
\end{aligned}
\tag{3}
$$

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

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

This definition of DFT corresponds to the FFT defined in Matlab. Notice that $FF^{\dagger} = NI$ where I is the identity matrix (and $F{=}F^T$).

Eigenvectors of Circulant Matrices

Theorem. *Every circulant matrix* $H \in \mathbb{C}^{N \times N}$ *has exactly* N *eigenvectors which may be taken as the columns of the matrix* F *described above. Moreover* $\boldsymbol{\lambda} = F\boldsymbol{r}^T = F^\dagger \boldsymbol{h}$ *is the vector of eigenvalues where* $\boldsymbol{r}$ *and* $\boldsymbol{h}$ *are the fist row and column of* H*, respectively.*

Proof. The key is the fact that $\boldsymbol{f}_i\beta^i$ is $\boldsymbol{f}_i$ with each component shifted up cyclically by one position. Now let $\boldsymbol{r}_i$ $(i = 0, 1, \dots, N-1)$ be the ith row of H.

$$H\boldsymbol{f}_i = \begin{pmatrix} \boldsymbol{r}_0\boldsymbol{f}_i \\ \boldsymbol{r}_1\boldsymbol{f}_i \\ \vdots \\ \boldsymbol{r}_{N-1}\boldsymbol{f}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{r}\boldsymbol{f}_i \\ \boldsymbol{r}\boldsymbol{f}_i\beta^i \\ \vdots \\ \boldsymbol{r}\boldsymbol{f}_i\beta^{(N-1)i} \end{pmatrix} = (\boldsymbol{r}\boldsymbol{f}_i)\boldsymbol{f}_i = \lambda_i\boldsymbol{f}_i.$$

The above shows that each column of F is an eigenvector and since all columns are orthogonal of each other it implies that the number of eigenvectors is indeed N.

It also shows that the eigenvector $\boldsymbol{f}_i$ has $\lambda_i = \boldsymbol{r}\boldsymbol{f}_i$ as eigenvalue. Hence $\boldsymbol{\lambda}^T = \boldsymbol{r}F$ or, equivalently, $\boldsymbol{\lambda} = F^T\boldsymbol{r}^T = F\boldsymbol{r}^T$.

As already observed, the first row $\boldsymbol{r}$ of H and the first column $\boldsymbol{h}$ are related by $r_l = h_{[-l]}$, $l = 0, 1, \dots, N-1$. Hence

$$\boldsymbol{r}\boldsymbol{f}_i = \sum_l r_l\beta^{il} = \sum_l h_{[-l]}\beta^{il} = \sum_l h_l\beta^{-il} = \boldsymbol{f}_i^\dagger\boldsymbol{h}. \qquad \square$$

Example:

$$H = \begin{pmatrix} h_0 & h_1 \\ h_1 & h_0 \end{pmatrix} \in \mathbb{C}^{2\times 2}.$$

This is a circulant matrix. Hence

$$f_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad f_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

are eigenvectors and the corresponding eigenvalues are

$$\begin{pmatrix} \lambda_0 \\ \lambda_1 \end{pmatrix} = F^\dagger \boldsymbol{h} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \end{pmatrix} = \begin{pmatrix} h_0 + h_1 \\ h_0 - h_1 \end{pmatrix}.$$

Indeed

$$H\boldsymbol{f}_0 = \begin{pmatrix} h_0 + h_1 \\ h_1 + h_0 \end{pmatrix} = (h_0 + h_1) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \lambda_0 \boldsymbol{f}_0$$

and

$$H\boldsymbol{f}_1 = \begin{pmatrix} h_0 - h_1 \\ h_1 - h_0 \end{pmatrix} = (h_0 - h_1) \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \lambda_1 \boldsymbol{f}_1.$$

The following result, which should already be familiar to you (perhaps with different notation), states that in the DFT domain a cyclic convolution becomes a component-wise product.

We prove it for completeness and to see a straightforward application of the fact that the columns of F are eigenvectors of any circulant matrix.

Corollary. *Let $\boldsymbol{y} = H\boldsymbol{a}$ for some circulant matrix H, i.e, $\boldsymbol{y}$ is the result of circularly convolving the first column of H with $\boldsymbol{a}$. Then*

$$\boldsymbol{Y} = D\boldsymbol{A}$$

where $\boldsymbol{Y} = F^\dagger \boldsymbol{y}$ and $\boldsymbol{A} = F^\dagger \boldsymbol{a}$ are the DFT of $\boldsymbol{y}$ and $\boldsymbol{a}$, respectively, and D is the diagonal matrix that has the eigenvalue λ_i of the eigenvector $\boldsymbol{f}_i$ as its ith diagonal element.

Proof.

$$\begin{aligned}
\boldsymbol{y} &= H\boldsymbol{a} \\
&= H\left(\frac{1}{N}F\boldsymbol{A}\right) \\
&= \frac{1}{N}(HF)\boldsymbol{A} \\
&= \frac{1}{N}(FD)\boldsymbol{A}
\end{aligned}$$

Where for the last equality we used the fact that $H\boldsymbol{f}_i = \lambda_i \boldsymbol{f}_i$ and the definition of D. This shows that $F^\dagger \boldsymbol{y} = \frac{1}{N}F^\dagger F D\boldsymbol{A} = D\boldsymbol{A}$. $\square$

Appendix B: OFDM = Communication Using Eigenvectors

In this appendix we develop the idea of using eigenvectors to communicate and underline the fact that the idea is very much in line with the idea of using Nyquist pulses as a vehicle for the information symbols.

As a start, recall how we communicate across the AWGN channel: we send signals of the form $s(t) = \sum_{k=0}^{N-1} s_k \psi_k(t)$, where $\{\psi_k(t) : k = 0, 1, \ldots, N-1\}$ forms an orthonormal family, and at the receiver we obtain the sufficient statistic by forming the inner product of the received signal $y(t) = s(t) + z(t)$ with each of the $\psi_k(t)$.

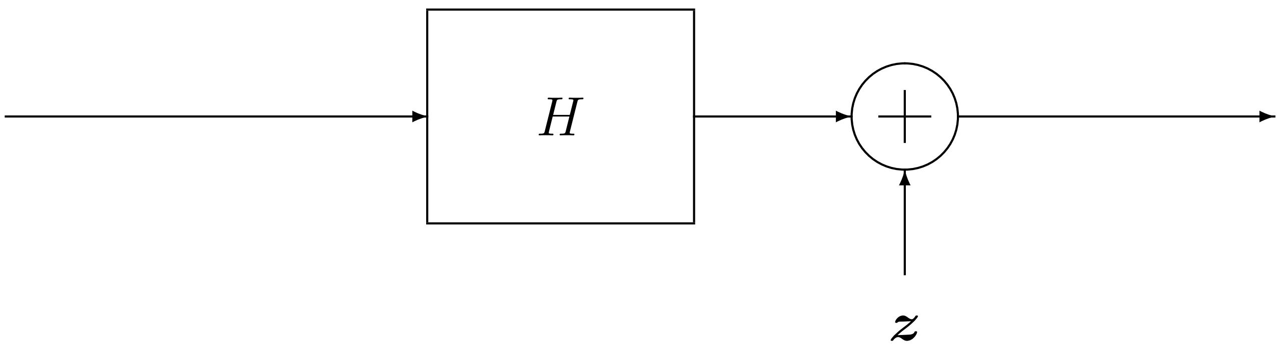
By using a Nyquist pulse $\psi(t)$ we can choose $\psi_k(t) = \psi(t - kT)$ and do all the inner products with a single (matched) filter and output sampled at integer multiples of T. The channel seen from the sampled matched filter output is referred to as the symbol level channel. It takes symbols as input and outputs symbols plus white Gaussian noise.

Now assume we do the same but the physical channel has an unknown impulse response. For the receiver, it is as if we were using a new non-Nyquist pulse. There result is the so-called inter-symbol interference.

The linear input-output relationship may be described by a filter represented by some matrix H (see the figure below).

One can use the eigenvectors $\boldsymbol{\psi}_0$, $\boldsymbol{\psi}_1$, $\ldots \boldsymbol{\psi}_{N-1}$ of $H \in \mathbb{C}^{N\times N}$ to form the inputs to the linear transformation (see again the figure).

$$\boldsymbol{s} = U_0\boldsymbol{\psi}_0 + \ldots + U_{N-1}\boldsymbol{\psi}_{N-1} \qquad \boldsymbol{y} = U_0\lambda_0\boldsymbol{\psi}_0 + \ldots + U_{N-1}\lambda_{n-1}\boldsymbol{\psi}_{N-1} + \boldsymbol{z}$$



As shown in the above figure, if we use eigenvectors as information carriers, the linear transformation will only scale them.

If the eigenvectors are orthogonal to one another, the orthogonality is preserved after the eigenvectors go through the channel.

In this case, the inner product of the channel output with ψ_i yields $U_i\lambda_i + Z_i$ where all the Z_i are iid and Gaussian and λ_i is the ith eigenvalue (supposed to be known for the moment).

If the matrix H is circulant, then the columns of the DFT matrix F are eigenvectors (and are orthogonal to one another).

In this case, the channel input $\frac{1}{N}(U_0\boldsymbol{f}_0 + \ldots + U_{N-1}\boldsymbol{f}_{N-1})$ is simply $\frac{1}{N}F\boldsymbol{U}$. This can be computed very efficiently since it is the IFFT of $\boldsymbol{U}$.

At the receiver we can do the N projections in one shot and very efficiently by taking the FFT of $\boldsymbol{y}$ which yields

$$\boldsymbol{Y} = F^{\dagger}\boldsymbol{y} = \begin{pmatrix} \langle \boldsymbol{y}, \boldsymbol{f}_0 \rangle \\ \langle \boldsymbol{y}, \boldsymbol{f}_1 \rangle \\ \vdots \\ \langle \boldsymbol{y}, \boldsymbol{f}_{N-1} \rangle \end{pmatrix} = \begin{pmatrix} U_0\lambda_0 + Z_0 \\ U_1\lambda_1 + Z_1 \\ \vdots \\ U_{N-1}\lambda_{N-1} + Z_{N-1} \end{pmatrix}$$

Let us summarize what we do:

We use the components of $\boldsymbol{U}$ as the coefficients of a linear combination of orthogonal eigenvectors. We do this very efficiently by taking the IFFT of $\boldsymbol{U}$.

At the receiver we make all the projections in one shot and very efficiently by taking the FFT.

The result is a sufficient statistic and in fact it is the same as if we had sent $\boldsymbol{U}$ through a channel that scales each component and adds independent Gaussian noise but does not introduce ISI. The output of this imaginary channel is $\boldsymbol{Y} = D\boldsymbol{U} + \boldsymbol{Z}$ where D is diagonal.

A maximum likelihood decision on the components of $\boldsymbol{U}$ is obtained by means of a slicer applied independently to each component of $\boldsymbol{Y}$.

Pretty neat, no?

Notice that we have modulated a pulse train at two levels: the first level is the one that led to the symbol level channel. The second level is the one used to communicate across the symbol level channel by modulating eigenvectors.

Recall that the DFT matrix F is not unitary. (To make it into a unitary transformation we need to scale it by $\frac{1}{\sqrt{N}}$.)

Hence if $\boldsymbol{z} \sim \mathcal{CN}(0, \sigma^2 I_N)$ is the additive noise prior to the DFT, where I_N is the $N \times N$ identity matrix, then after the DFT the additive noise becomes

$$\boldsymbol{Z} = F^{\dagger}\boldsymbol{z} \sim \mathcal{CN}(0, N\sigma^2 I_N).$$

Appendix C: The Discrete-Time Channel Impulse Response

The signal used for wireless communication often propagates from the sender to the receiver along multiple paths. If there are M paths and the ith path has strength α_i and propagation delay τ_i then the channel impulse response is

$$h(t) = \sum_{l=0}^{M-1} \alpha_l \delta(t - \tau_l).$$

It is realistic to assume that $\alpha_0, \ldots, \alpha_{M-1}$ are realizations of zero-mean jointly Gaussian random variables.

We are interested in the discrete-time equivalent of $h(t)$.

There are two discrete-time channels of interest. One is the *symbol-level channel impulse response* already discussed.

The other is the channel "seen" by the samples of the transmitted signal. We call this the *sample-level channel impulse response.*

We take a look at both.

Symbol-Level Channel Impulse Response

We assume that the sender uses the pulse $p(t)$ and the inner products at the receiver are done between the received signal and $q(t - kT)$, $k \in \mathbb{Z}$.

Often we assume that p is a Nyquist pulse and $q = p$ but we can be more general than that. As long as the sequence $\{q(t - kT) : k \in \mathbb{Z}\}$ is an orthogonal one and forms a basis for the inner product space spanned by the channel-filtered-pulse $p * h$, the sampler output sequence forms a sufficient statistic and the noise samples are realization from independent zero-mean Gaussian random variables.

There is little motivation in requiring that the pulse p be Nyquist if we assume anyway that the channel has a non-trivial impulse response. You may want to use a rectangle for p.

The symbol-level channel impulse response is obtained by sampling $h_s(t) = p(t) * h(t) * q^*(-t)$ at the symbol rate. If we let $f(t) = p(t) * q^*(-t)$ then we have

$$
\begin{aligned}
h_s(t) &= (h * f)(t) \\
&= \sum_{l=0}^{M-1} \alpha_l f(t - \tau_l),
\end{aligned}
$$

$$
h[n] = \sum_{l=0}^{M-1} \alpha_l f(nT - \tau_l).
$$

As an important special case we mention that when $q = p$ we obtain

$$h[n] = \sum_{l=0}^{M-1} \alpha_l R_p(nT - \tau_l)$$

where

$$R_p(\xi) = \int_{-\infty}^{\infty} p(t+\xi)p^*(t)dt.$$

The Covariance $K_{\boldsymbol{D}}$

$K_{\boldsymbol{D}}$ is the covariance of $\boldsymbol{C} = F^\dagger \boldsymbol{c}$ where $\boldsymbol{c}$ is the first column of H_c, namely

$$\boldsymbol{c} = (h[0], h[1], \ldots, h[L-1], 0, \ldots, 0)^T.$$

Hence

$$K_{\boldsymbol{D}} = E[F^\dagger \boldsymbol{c}\boldsymbol{c}^\dagger F] = F^\dagger K_{\boldsymbol{c}} F.$$

When $q = p$, we can also write $K_{\boldsymbol{D}}$ in terms of the pulse autocorrelation $R_p(\xi)$.

To see this, define the $N \times M$ matrix

$$R = \begin{pmatrix} R_p(-\tau_0) & R_p(-\tau_1) & \dots & R_p(-\tau_{M-1}) \\ R_p(T-\tau_0) & R_p(T-\tau_1) & \dots & R_p(T-\tau_{M-1}) \\ \vdots & & & \vdots \\ R_p((L-1)T-\tau_0) & R_p((L-1)T-\tau_1) & \dots & R_p((L-1)T-\tau_{M-1}) \\ 0 & 0 & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}.$$

and the random vector of channel path strengths

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{M-1} \end{pmatrix}.$$

Now

$$
\begin{aligned}
\boldsymbol{c} &= R\boldsymbol{\alpha} \\
K_{\boldsymbol{c}} &= RK_{\boldsymbol{\alpha}}R^{\dagger} \quad \text{where } K_{\boldsymbol{\alpha}} \text{ is the covariance of } \boldsymbol{\alpha} \\
K_{\boldsymbol{D}} &= F^{\dagger}K_{\boldsymbol{c}}F = (F^{\dagger}R)K_{\boldsymbol{\alpha}}(F^{\dagger}R)^{\dagger}
\end{aligned}
$$

where $F^{\dagger}R$ is the DFT (column by column) of R.

A Simple Example

Let us assume that p is the unit-energy rectangle with support $[-T/2, T/2]$ and $q = p$. Then R_p is the triangle $R_p(\xi) = 1 - \frac{|\xi|}{T}$ for $\xi \in [-T, T]$ and zero otherwise.

Assume that the multipath channel has three paths ($M = 3$) with delays $\tau_0 = \frac{T}{2}$, $\tau_1 = \frac{3T}{4}$, $\tau_2 = 2T$. This implies that $h[n] = 0$ for $n > 2$, i.e., $L = 3$. Hence

$$R = \begin{pmatrix} 0.5 & .25 & 0 \\ 0.5 & 0.75 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{N \times M}.$$

The top three rows of the matrix have the form

$$(R_p(\xi - \tau_0), R_p(\xi - \tau_1), R_p(\xi - \tau_2))$$

for $\xi = 0$ (first row), $\xi = T$ (second row), and $\xi = 2T$ (third row), respectively.

Let the path strengths be jointly Gaussian, zero-mean, with covariance matrix

$$K_{\boldsymbol{\alpha}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.125 \end{pmatrix}.$$

We have all what we need to let MATLAB compute

$$K_{\boldsymbol{D}} = (F^\dagger R) K_{\boldsymbol{\alpha}} (F^\dagger R)^\dagger$$

where, once again, $F^\dagger R$ is the DFT of R.

Sample-Level Channel Impulse Response

If the samples of the transmitted signal of interest $s(t)$ are taken every T_s seconds it means that the support of $s_{\mathcal{F}}(f)$ lies in the interval $[-\frac{1}{2T_s}, \frac{1}{2T_s}]$. (We are assuming that the conditions of the sampling theorem are met.) Hence for this signal it does not matter if we change $h_{\mathcal{F}}(f)$ outside $[-\frac{1}{2T_s}, \frac{1}{2T_s}]$.

Let the lowpass-filtered channel impulse response $h_{LP}(t)$ be $h_{\mathcal{F}}(f)$ for $f \in [-\frac{1}{2T_s}, \frac{1}{2T_s}]$ and zero outside. Hence

$$h_{LP}(t) = h(t) * \frac{1}{T_s} \operatorname{sinc}\left(\frac{t}{T_s}\right) = \sum_{l=0}^{M-1} \frac{\alpha_i}{T_s} \operatorname{sinc}\left(\frac{t-\tau_i}{T_s}\right).$$

The discrete-time equivalent of $h(t)$ for a signal sampled at multiples of T_s seconds is thus

$$h_{LP}[k] = T_s \, h_{LP}(kT_s), \quad k \in \mathbb{Z}.$$

Now we could proceed as for the symbol-level channel and find $K_{\boldsymbol{D}}$.

Appendix D: The Power Spectral Density of OFDM

We compute the power spectral density (PSD) of

$$s(t) = \sum_k a[k]\psi(t - kT)$$

where

$$a[k] = \frac{1}{N}\sum_{n=0}^{N-1} A^{(\lfloor \frac{k}{N} \rfloor)}[n] \exp\left(j\frac{2\pi}{N}nk\right).$$

We assume that all $A^{(m)}[n]$ are independent and zero-mean and that their statistic may depend on n (the carrier frequency index) but not on m (the time index).

For the moment we neglect the cyclic prefix. At the end we will argue that its effect is negligible.

If we insert and swap the sums we see that $s(t)$ is the sum of N signals, one for each subcarrier. Since those signals carry uncorrelated information, the signals themselves are uncorrelated and their spectra add.

Hence we may compute the spectrum for each n separately. For this reason we define the signal that corresponds to the nth carrier

$$
\begin{aligned}
s_n(t) &= \sum_k \frac{A^{(\lfloor \frac{k}{N} \rfloor)}[n]}{N} \exp\left(j\frac{2\pi}{N}nk\right)\psi(t-kT) \\
&= \sum_k a_n[k]\psi(t-kT)
\end{aligned}
$$

where we have defined

$$
a_n[k] = \frac{A^{(\lfloor \frac{k}{N} \rfloor)}[n]}{N} \exp\left(j\frac{2\pi}{N}nk\right).
$$

The signal $s_n(t)$ is not wide sense stationary (WSS) for two reasons.

First, because the pulse is placed at fixed locations on the time axis. The usual trick to circumvent this problem is to insert a delay τ which is independent of anything else and is uniformly distributed in the interval $[0, T]$.

The other similar reason is that the sequence $\{a_n[k] : k \in \mathbb{Z}\}$ contains the term $A^{(\lfloor \frac{k}{N} \rfloor)}[n]$ which changes only at values of k that are multiples of N. For this reason the sequence is not WSS but we can make it to be WSS by adding to k a random index U uniformly distributed in $\{0, 1, \ldots, N-1\}$.

With these modifications the signal $s_n(t)$ is WSS and we can apply the standard formula to determine its PSD, namely (see e.g., *Principles of Digital Communications*)

$$S_n(f) = \frac{|\psi_{\mathcal{F}}(f)|^2(f)}{T} \sum_l R_n[l] \exp\Big(- j2\pi l f T\Big),$$

where $R_n[l] = E[a_n[k+l]a_n^*[k]]$ is the autocorrelation of $\{a_n[k] : k \in \mathbb{Z}\}$. (The summation is the discrete-time Fourier transform (DTFT) of $R_n[l]$.)

When $|l| \geq N$, $R_n[l] = 0$ since the two symbols $a_n[k+l]$ and $a_n[k]$ belong to different blocks hence are independent.

When $|l| < N$, due to the random shift U they belong to the same block with probability $1 - \frac{|l|}{N}$. When they are in the same block, $A^{(\lfloor \frac{k}{N} \rfloor)}[n]$ and $A^{(\lfloor \frac{k+l}{N} \rfloor)}[n]$ are the same. Thus

$$R_n[l] = \frac{1}{N^2}\Big(1 - \frac{|l|}{N}\Big) E_n \exp\Big(j\frac{2\pi}{N} l n\Big), \quad |l| < N,$$

where $E_n = E[|A^{(m)}[n]|^2]$ is the energy of the constellation used for the nth carrier.

Now

$$\sum_l R_n[l]\exp(-j2\pi lfT) = \sum_{l=-N+1}^{N-1} \frac{E_n}{N^2}\Big(1-\frac{|l|}{N}\Big)\exp\Big(j2\pi l\big(\frac{n}{N}-fT\big)\Big).$$

This is the discrete-time Fourier transform of a triangle, evaluated at $\frac{n}{N} - fT$. After a few steps we obtain (see Appendix E)

$$\sum_l R_n[l]\exp(-j2\pi lfT) = \frac{E_n}{N^3}\frac{\sin^2\Big(\pi N\big(\frac{n}{N}-fT\big)\Big)}{\sin^2\Big(\pi\big(\frac{n}{N}-fT\big)\Big)}.$$

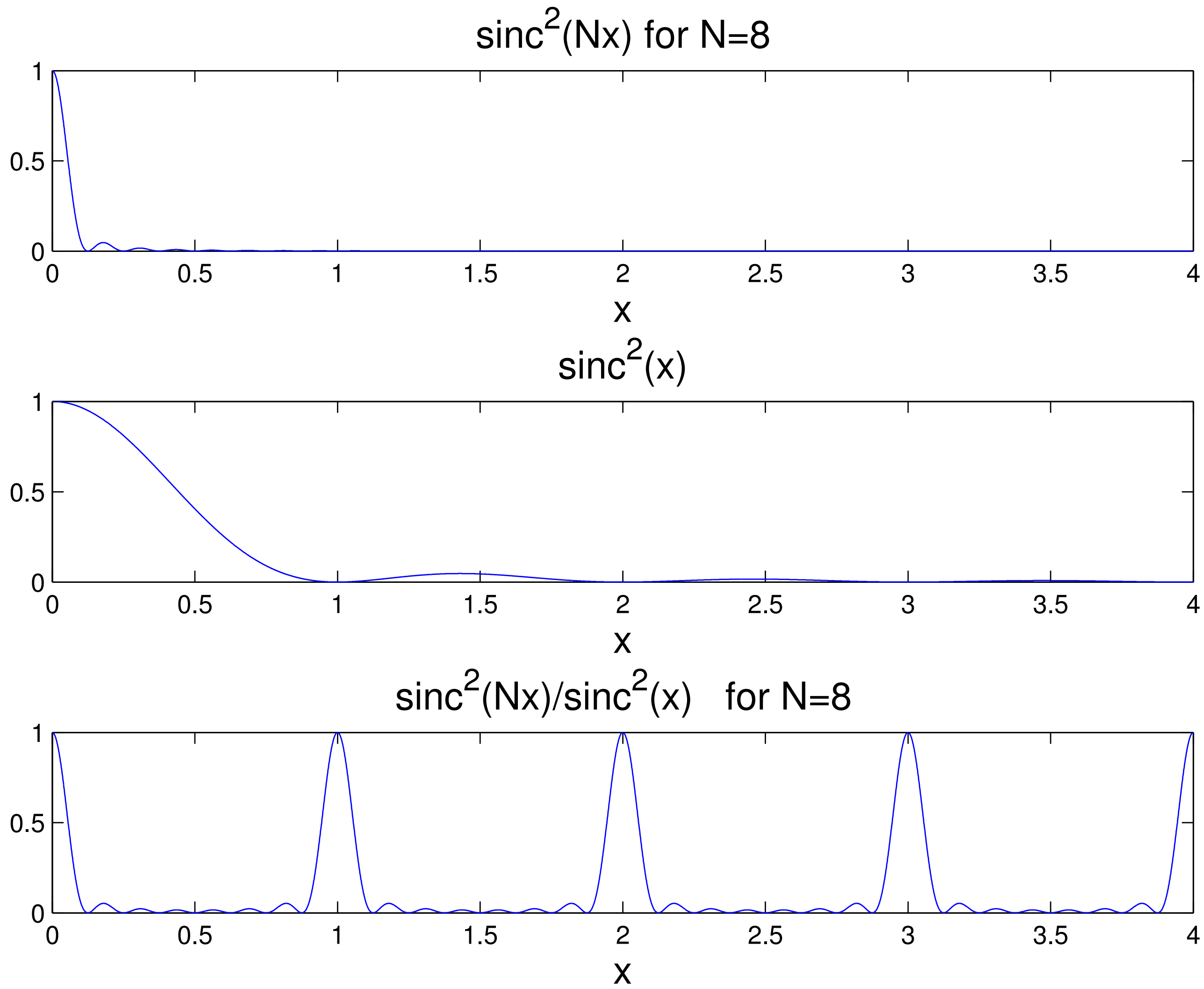
Putting everything together yields the desired result:

$$S_n(f) = \frac{|\psi_{\mathcal{F}}(f)|^2(f)}{T} E_n \frac{\operatorname{sinc}^2\left(N\left(\frac{n}{N} - fT\right)\right)}{N \operatorname{sinc}^2\left(\frac{n}{N} - fT\right)}.$$

Notice that the first term is the contribution due to the pulse, the second due to the symbol sequence that excites the nth carrier, and the third due to the correlation introduced by the IFFT (or, more precisely, by the eigenvector excited by the symbol sequence $\{A^{(m)}[n] : m \in \mathbb{Z}\}$).

The third term needs some clarification. To see what it does, consider the function $\frac{\operatorname{sinc}^2(Nx)}{\operatorname{sinc}^2(x)}$. At integer values of x, the denominator vanishes and so does the numerator. It turns out that the limit is 1. Hence, we get a spike train at integer values of x that becomes narrower as N increases.

The following plot shows the numerator, denominator, and quotient for $N = 8$.



We infer that

$$\frac{\operatorname{sinc}^2\left(N\left(\frac{n}{N}-fT\right)\right)}{N\operatorname{sinc}^2\left(\frac{n}{N}-fT\right)}$$

is a spike train of height $1/N$, centered at values of f for which $fT-\frac{n}{N}=0,\pm1,\pm2,\ldots$.

We are now ready to interpret the expression for $S_n(f)$ that we repeat here for convenience

$$S_n(f)=\frac{|\psi_{\mathcal{F}}(f)|^2(f)}{T}E_n\frac{\operatorname{sinc}^2\left(N\left(\frac{n}{N}-fT\right)\right)}{N\operatorname{sinc}^2\left(\frac{n}{N}-fT\right)}.$$

The power spectral density is composed of narrow pulse trains, one for each n. For $f\geq 0$, the first pulse for each n is at $f=0,\frac{1}{TN},\frac{2}{TN},\ldots,\frac{N-1}{TN}$. Each pulse train is shaped by the envelope $\frac{|\psi_{\mathcal{F}}(f)|^2(f)}{T}E_n$.

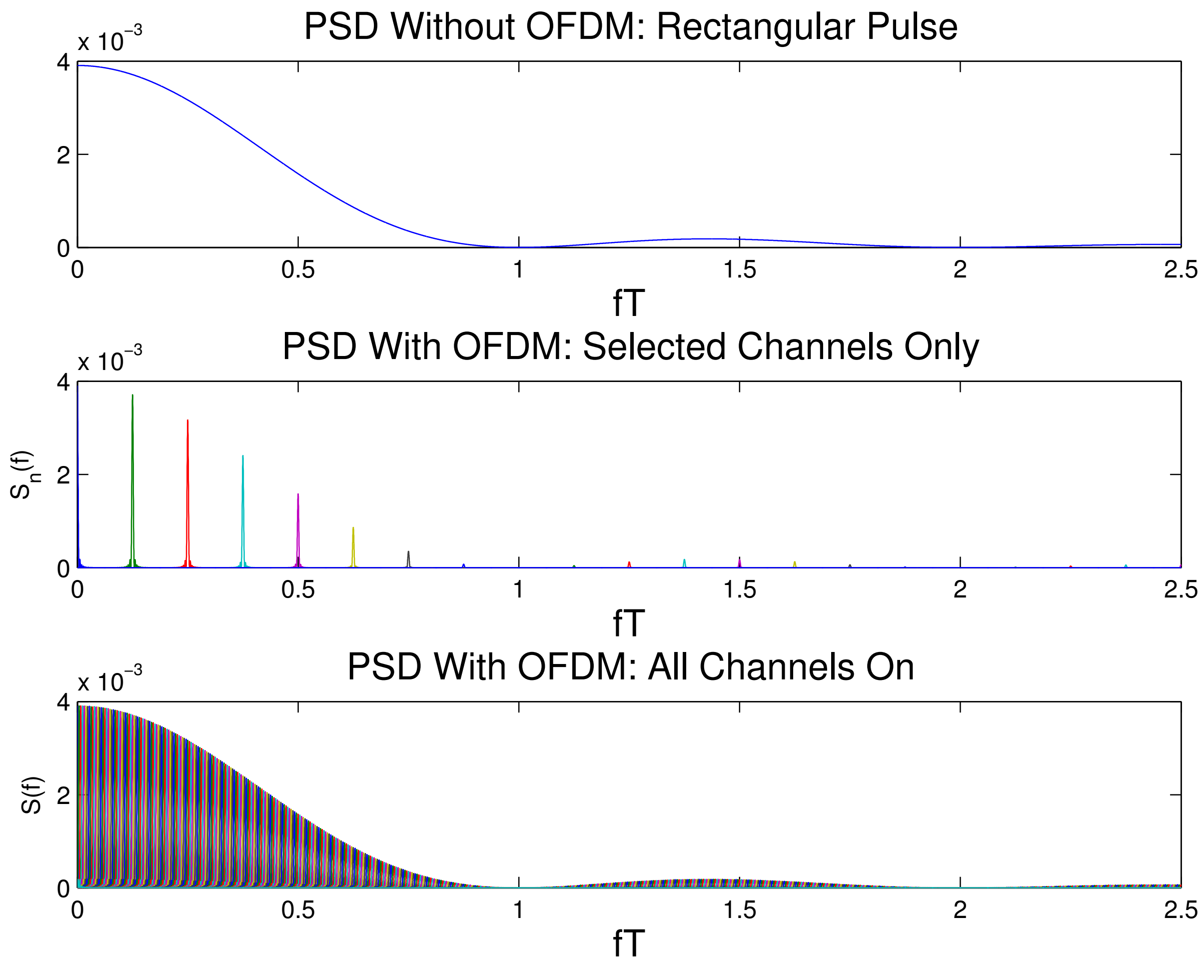
Example: When the pulse is rectangular, i.e.,

$$\psi(t) = \begin{cases} \sqrt{\frac{1}{T}}, & 0 \leq t < T \\ 0, & \text{otherwise} \end{cases}$$

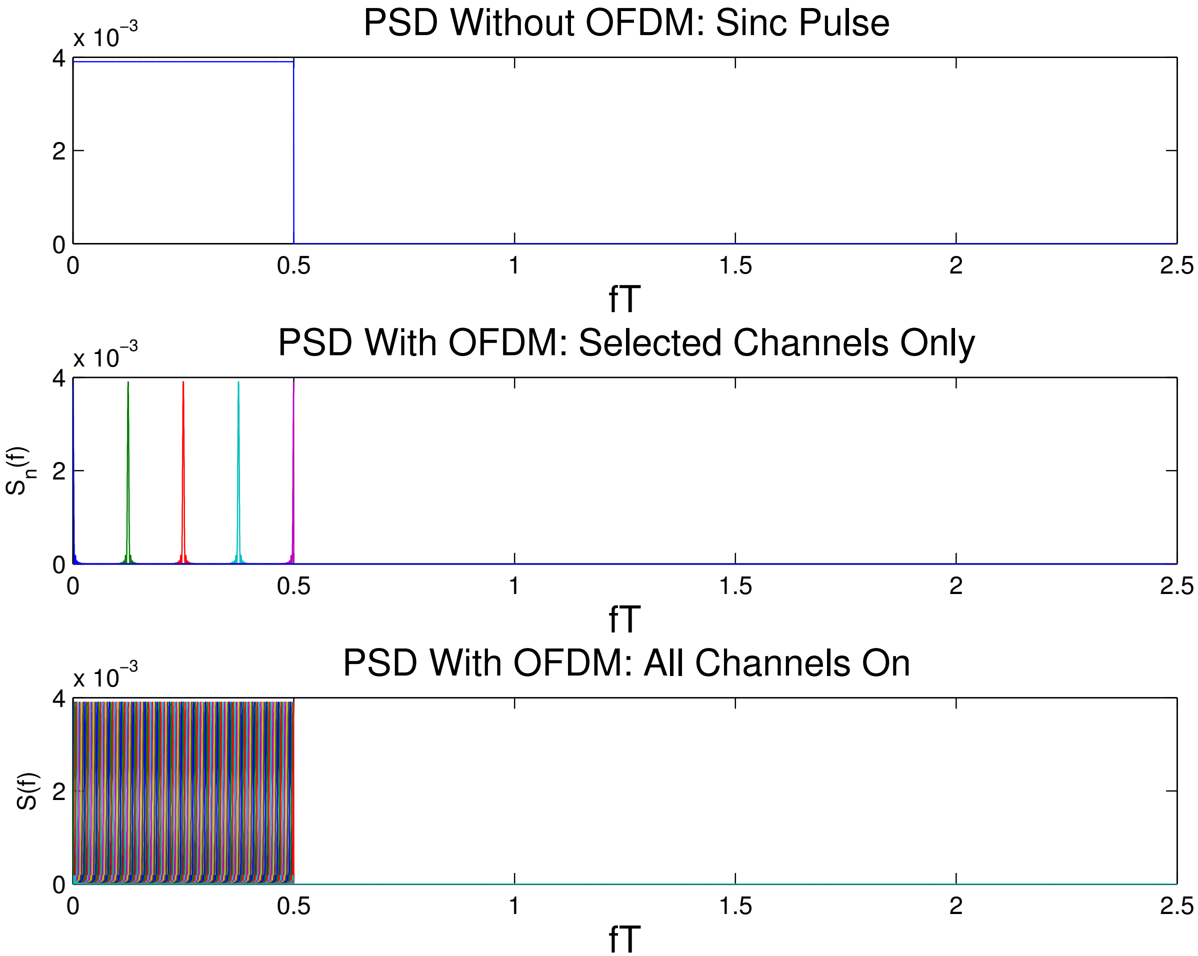
we obtain

$$S_n(f) = \frac{E_n}{N} \operatorname{sinc}^2(Tf) \frac{\operatorname{sinc}^2(N(\frac{n}{N} - fT))}{\operatorname{sinc}^2(\frac{n}{N} - fT)}. \tag{4}$$

The following plots illustrate the PSD of OFDM for the rectangular pulse $\psi(t)$ [Expression (??)] with $N = 256$. The first figure plots $\frac{|\psi_{\mathcal{F}}(f)|^2(f)}{T}$, which is also what we would obtain without OFDM; The second plots $S_n(f)$ for $n = 1,\ 33,\ 65,\ 97,\ 129,\ 161,\ 193,\ 225$. The third plots $S(f)$. The constellation energy is $E_n = 1$ for all n.



The following plots are analogous to the previous ones but for $\psi(t) = \frac{1}{\sqrt{T}} \operatorname{sinc}(\frac{t}{T})$.



In this appendix, we have neglected the effect of the cyclic prefix.

A valid concern is whether or not it has a negative effect on the PSD of the transmitted signal. The worry is that the cyclic prefix inserts an additional discontinuity that affects the spectrum. We argue that it is not the case.

An OFDM block is the superposition of signals of the form $(\beta^{i0}, \beta^{i1}, \ldots, \beta^{i(N-1)})$, where $\beta = \exp(j\frac{2\pi}{N})$ is a primitive N-th root of unity.

The cyclic prefix substitutes the above signal with $(\beta^{i(N-L+1)}, \ldots, \beta^{i(N-1)}, \beta^{i0}, \beta^{i1}, \ldots, \beta^{i(N-1)})$.

Because β is a primitive N-th root of unity, there is no discontinuity in going from $\beta^{i(N-1)}$ toβ^{i0}.

Hence the effect of the cyclic prefix on the PSD is the same as an increase of the blocklength from N to $N+L-1$, i.e., to reduce the spectral lines' width.

Appendix E: A Side Derivation

We show that

$$\sum_l R_n[l] \exp(-j2\pi l f T) = \frac{E_n}{N^3} \frac{\sin^2\left(\pi N\left(\frac{n}{N} - fT\right)\right)}{\sin^2\left(\pi\left(\frac{n}{N} - fT\right)\right)},$$

where

$$R_n[l] = \frac{1}{N^2}\left(1 - \frac{|l|}{N}\right) E_n \exp\left(j\frac{2\pi}{N} ln\right) \text{ for } |l| < N \text{ and 0 otherwise.}$$

Notice that the above is the discrete-time Fourier transform (DTFT) of $R_n[l]$ evaluated at $\omega = fT$.

Let $g[n] = 1$ for $n = 0, 1, \ldots, N-1$ and 0 otherwise. Its DTFT is

$$G_1(\omega) = \sum_{n=0}^{N-1} \exp\{-j2\pi n\omega\} = \exp\{-j\pi(N-1)\omega\} \frac{\sin(\pi N\omega)}{\sin(\pi\omega)}.$$

Let $g_2 = g_1 * g_1$. (For later on notice that this is a triangle of height N

centered at $N-1$.) Its DTFT is

$$G_2(\omega) = G_1(\omega)^2 = \exp\{-j2\pi(N-1)\omega\}\frac{\sin^2(\pi N\omega)}{\sin^2(\pi\omega)}.$$

Let $g_3[n] = g_2[n+N-1]$. (This is a triangle of height N centered at the origin.) Its DTFT is

$$G_3(\omega) = \exp\{j2\pi(N-1)\omega\}G_2(\omega) = \frac{\sin^2(\pi N\omega)}{\sin^2(\pi\omega)}.$$

The desired result follows by observing that $R_n[l] = \frac{E_n}{N^3}g_3[l]\exp\left(j\frac{2\pi}{N}ln\right)$. Its DTFT is $\frac{E_n}{N^3}G_3(\omega - \frac{n}{N})$. Evaluating at $\omega = fT$ gives the desired expression.