OFDM for Broadband Communication

Course Reader (EIT 140)

LUND UNIVERSITY

2013

Edited by Thomas Magesacher
Department of Electrical and Information Technology
Lund University
Box 118, S-221 00 LUND
SWEDEN

ISBN 91-7167-036-X
Second Edition (V2.0.6)

Contents

Preface

This course reader is a compilation of contributions from several authors. Additionally, the following literature is recommended for broadening and deepening the reader's view on the subject:

- J.B. Anderson, *Digital Transmission Engineering*, IEEE Press, ISBN 0-7803-3457-4, 1999.
- E. Biglieri and G. Taricco, *Transmission and Reception with Multiple Antennas: Theoretical Foundations*, now Publishers Inc., ISBN 1933019018, 2004.
- J. M. Cioffi, "Advanced Digital Communication," *class reader EE379C*, Stanford University, Available from http://www.stanford.edu/class/ee379c/
- R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, ISBN 0-7803-3483-3, 1999.
- S. Lin and D. J. Costello, *Error Control Coding*, Prentice Hall, ISBN 0-13-042672-5, second edition, 2004.
- J. L. Massey, *Applied Digital Information Theory—Lecture Notes, ETH Zurich*, Available from www.isi.ee.ethz.ch/education/public/pdfs/aditI.pdf.
- A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge, ISBN 0521826152, 2003.
- J. G. Proakis, *Digital Communications*, Mc Graw Hill, ISBN 0-07-232111-3, fourth edition, 2001.
- T. Starr, J. M. Cioffi, and P. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice Hall, Englewood Cliffs, 1998.
- D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*, Cambridge, ISBN 0521845270, 2005.
- R. Van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House, ISBN 0890065306, 2003.

Part I

The Elegance of OFDM and DMT

1 Sinusoids, multiplexing and... a miracle?

Ever since the early days of radio communications one of the guiding notions has been that the amplitude and the phase of a *sinusoidal* radio wave can carry information from a transmitter to a receiver. Why sinusoids? Sinusoids were easy to generate with existing hardware components, their phase and amplitude could be simply modulated and demodulated using available components, and, moreover, collision of signals from different radio transmitters could be simply avoided by a proper choice of the carrier frequencies – by proper frequency planning. Sinusoids have been the key signal components in radio communications ever since.

The availability of hardware components that could generate sinusoids has played an important role in the popularity of sinusoids. Another reason for their popularity builds on a key appearance of the radio environment: its *linearity*. This linearity is quite remarkable; many phenomena in nature appear to be non-linear. Fluid mechanics, for instance, is governed by non-linear differential equations, the weather forecast cannot be made reliably for more than three days ahead because of the chaotic nature of atmospheric mechanics, and mechanics of materials does not follow linear rules. Non-linear systems have been difficult to address mathematically and consequently have been harder to exploit. Radio waves, however, closely follow Maxwell's linear relations and can be addressed with well-developed mathematical tools from linear algebra.

Mathematically, infinitely long sinusoids have a typical relation to linear environments: only the amplitude and the phase of a sinusoidal radio waveform are affected by the environment, not its frequency. In other words, an infinitely long sinusoidal radio wave of a certain frequency will, after propagation through the linear radio channel, be observed at the receiver as an infinitely long sinusoidal radio wave with the same frequency – only its amplitude and its phase are changed. Because of this property, sinusoids are generally being referred to as *eigenfunctions* of linear systems. A combination of the availability of suitable hardware components, the well-developed mathematical field of linear algebra and the above-discussed eigen-property makes sinusoids particularly suitable for use in radio communications.

In today's *digital* communication systems, sinusoids play an important role, too. Based on a stream of information bits, Quadrature Amplitude Modulation (QAM), which is one of the most fundamental modulation schemes, changes the amplitude and/or the phase of a sinusoid every T_{sym} seconds, which we call the *symbol period.* The information bits are mapped consecutively onto pairs of real numbers $\{x_k^{(\mathrm{i})}, x_k^{(\mathrm{q})}\}$, referred to as transmit symbols. This mapping, implemented for example by means of a look-up table, has a strong impact on the performance of the scheme. A QAM modulator generates a continuous-time signal

$$s(t) = p(t - kT_{\mathrm{sym}})(x_k^{(\mathrm{i})} \cos(2\pi F_{\mathrm{c}} t) - x_k^{(\mathrm{q})} \sin(2\pi F_{\mathrm{c}} t)), \quad kT_{\mathrm{sym}} \leq t < (k+1)T_{\mathrm{sym}},$$

where F_{c} is the so called carrier frequency in Hz—a sinusoidal signal with frequency F_{c} carries the information we want to transmit. The waveform $p(t)$ shapes each transmit symbol in time domain and

Chapter written by P. Ödling, J.J. van de Beek, D. Landström, P.O. Börjesson, and T. Magesacher.

thus determines the frequency characteristic of the $s(t)$. Using Euler's formula, the modulator output $s(t)$ can also be written as

$$s(t) = \mathrm{Re}\left\{p(t - kT_{\mathrm{sym}})\left(x_k^{(\mathrm{i})} + jx_k^{(\mathrm{q})}\right)e^{j2\pi F_{\mathrm{c}}t}\right\}, \qquad kT_{\mathrm{sym}} \le t < (k+1)T_{\mathrm{sym}}, \tag{1}$$

which is a frequently used description. The complex notation is a mathematical convenience and serves as a means to describe two orthogonal real-valued dimensions. Amplitude and phase of $s(t)$ are two independent parameters, which are implicitly controlled by the two independent symbols $\{x_k^{(\mathrm{i})}, x_k^{(\mathrm{q})}\}$.

The left plot of Figure 1 shows an exemplary waveform $s(t)$ of three transmitted symbols. For the sake of simplicity, we choose a rectangular pulse $p(t) = 1, 0 \le t \le T_{\mathrm{sym}}$. Introducing the complex notation $x_k^{(\mathrm{i})} + jx_k^{(\mathrm{q})}$, each symbol can be interpreted as a point in the complex plane. The right plot illustrates the set of valid transmission symbols $\{x_k^{(\mathrm{i})}, x_k^{(\mathrm{q})}\}$ as points in the complex plane, which is commonly referred to as constellation plot. The three symbols modulated onto the waveform are marked by plus signs (+). The term $e^{j2\pi F_{\mathrm{c}}t}$ represents a pointer rotating counter-clockwise in the complex plane and changing its length and its phase (angle between the pointer and the real axis) every T_{sym} seconds to $\sqrt{(x_k^{(\mathrm{i})})^2 + (x_k^{(\mathrm{q})})^2}$ and $\arctan(x_k^{(\mathrm{q})}/x_k^{(\mathrm{i})})$, respectively. The operation $\mathrm{Re}\{\cdot\}$ in (1), i.e., extracting the real part of this rotating pointer, corresponds to the projection of the pointer onto the real axis of the complex plane. A receiver tunes its local oscillator to the same radio frequency and tracks the phase/amplitude of the received sinusoid, an operation commonly referred to as demodulation.

Since amplitude and phase of the modulated sine wave now change with the symbol period, the resulting signal is not mathematically an eigenfunction of the linear channel. After all, only *infinitely long* sinusoids with constant amplitude and phase have the above described invariance property. The fact that the practically used signals are not eigenfunctions of the linear channel has some important practical implications. To illustrate this, we focus now on one of the immediately measurable consequences appearing when we transmit *two* modulated sinusoids in parallel. This concept, known as *frequency-division multiplexing* in the context of accommodating several users communicating over the same channel, is a means to increase the data rate without having to shorten the symbol period—at the cost of increasing bandwidth.

Opposite to how true infinitely long sinusoids would behave, any two such modulated sinusoids now generally *interfere* with each other. In other words, both before and after passing through the linear channel, one sinusoid can actually be measured at other frequencies. This undesirable interference limits the reliable detection of the amplitudes/phases and needs to be avoided. One effective way to avoid this interference is to choose the frequencies of the sinusoids sufficiently far apart. The interference onto other frequencies is generally largest on frequencies close to that of the transmitted carrier and smaller when the two frequencies are chosen far from each other.

We now describe an experiment in order to investigate *how much* interference occurs and answer the question *how far* apart the frequencies need to be chosen. We modulate five sinusoids of different (but equally spaced) frequencies and transmit them in parallel over a linear radio channel. The amplitude in each symbol interval can take two values, the first amplitude level representing a 'one' and the other level a 'zero'. Thus we transmit 5 bits per symbol interval. At the receiver, we demodulate the sinusoids (we

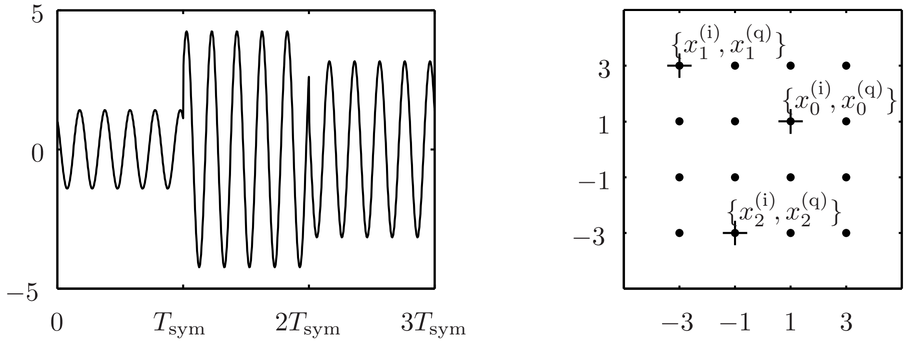


Figure 1: Amplitude/phase modulation of a sinusoid.

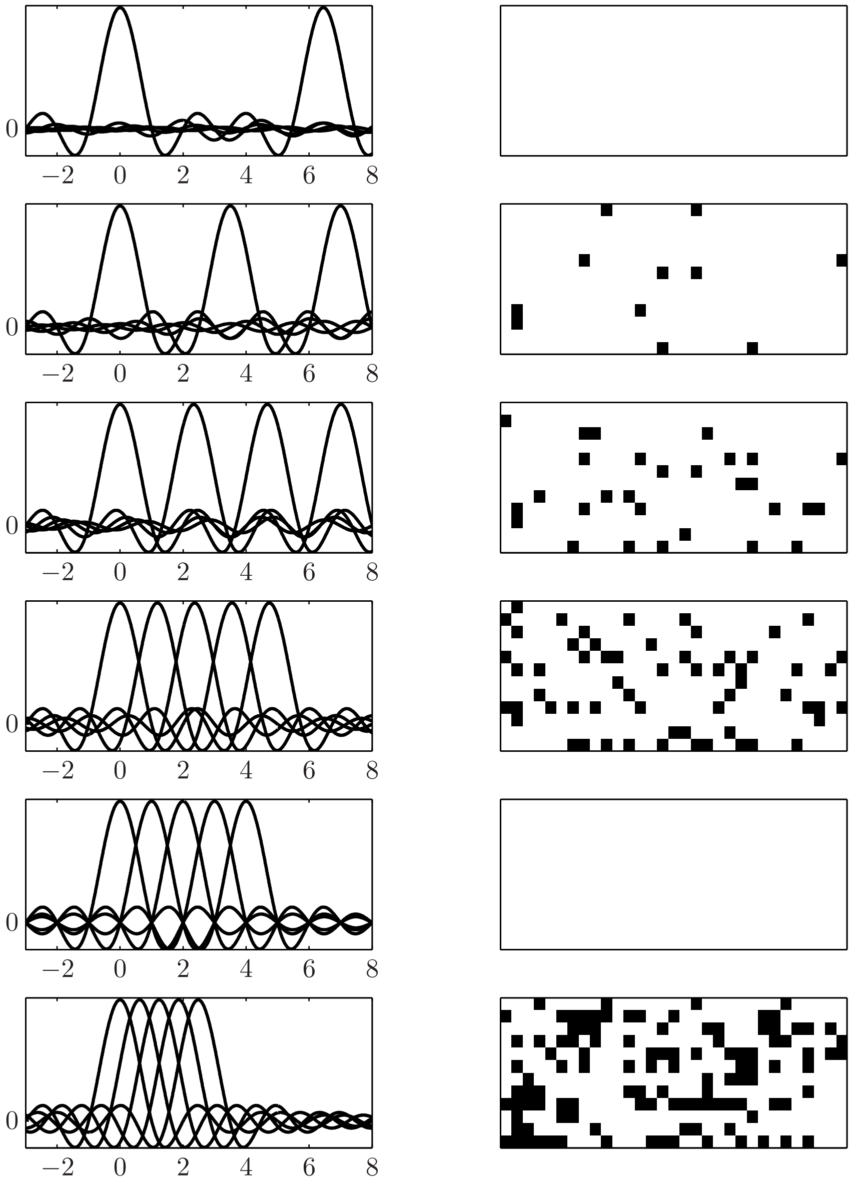


Figure 2: Multicarrier transmission.

extract the amplitude in each of the symbol intervals) and we make decisions for each symbol interval and for each of the modulated sinusoids as to whether the amplitude represented a ‘zero’ or a ‘one’. The heart of the experiment is that we vary the frequency-spacing of the sinusoids. We expect the bit-decisions at the receiver to become more and more erroneous as the frequencies are chosen closer together, because we expect the interference to increase.

Figure 2 illustrates what happens to the reception of the bits (here associated with a black or a white pixel) for various choices of the frequency spacing. The plots on the left show the Fourier transforms, which reveal the frequency characteristics, of the transmitted signals; in the plots on the right, black pixels illustrate erroneous detections at the receiver. As we anticipated, when we choose the frequencies of the two modulated sinusoids far apart, very little interference occurs, and, consequently, the receiver’s detector hardly makes any errors (topmost figures). When we choose other frequencies, closer together, the receiver decisions become less reliable. The number of erroneous decisions gradually increases, again in line with our expectations. We again decrease the frequency spacing of the carriers and suddenly... a miracle occurs! Just when the level of interference seems to prevent any reliable decisions at the receiver we obtain error-free transmission! Is this possible? Is something wrong with experimental set-up? With the equipment? We immediately choose an even smaller frequency difference and see: the interference and the associated bit-errors are back again (lowermost figures). Changing back to the previous setting confirms what we just observed: no bit-errors. Something very special is happening for this particular choice of carrier frequencies. The reason for this seemingly miraculous behaviour will be revealed at the end of this chapter.

Our above experiment illustrates the elegance of Orthogonal Frequency Division Multiplex (OFDM) and Discrete Multi-Tone (DMT) as it was first experienced and recognised in the sixties. In OFDM and DMT the transmitted sinusoids are packed together to such a degree that our intuition does not allow the thought of interference-free transmission. Yet this is exactly what is happening as the result of a subtle choice of the sinusoid frequencies. The way in which OFDM and DMT avoid the inter-sinusoidal interference is both ingenious and extremely efficient.

2 A desirable channel partitioning

One of the key aspects of *digital* communications is that there is no relation between the transmitted waveforms and the signal representing the message to be transmitted. The message is assumed to be a sequence of zeros and ones. How these bits represent the message is not relevant in the context of this book. If we are not limited to any of the characteristics of the message signal, *how* should we choose the transmitted waveforms? We could, for instance, choose waveforms that are simple to transmit, simple to receive, or waveforms that have neat properties in the sense that they do not disturb other users. The choice is enormous. By carefully adapting the waveforms to characteristics of the communications channel, we can accomplish features that are desirable in a certain situation. In this book, we choose a set of waveforms that, on one hand, is particularly simple to receive in linear channels, and, on the other hand, uses the frequency spectrum in an efficient way.

We want to transmit information-carrying real-valued numbers x_k, representing the message. For this purpose, we let each number x_k modulate one finite-length real-valued transmit signal component $s_k(n)$. The word 'modulate' means that x_k scales the amplitude of $s_k(n)$. We then transmit the sum of these modulated signal components, the multiplex

$$s(n) = \sum_k x_k s_k(n). \tag{2}$$

Now we pass the signal multiplex $s(n)$ through a linear time-invariant dispersive channel, characterised by its impulse response $h(n) \neq 0, n = 0, \dots, M$, as schematically depicted in Figure 3. The length of the channel impulse response is $M + 1$. An input sample $s(n_0)$ fed into the channel at an arbitrary time instant n_0 does not just affect the output sample $\tilde{r}(n_0)$ but also influences M subsequent samples $\tilde{r}(n), n = n_0 + 1 \dots n_0 + M$. Hence, we refer to M as the dispersion of the channel. Each transmit signal component $s_k(n)$ produces by itself, after passing through the channel, a receive signal component $\tilde{r}_k(n)$. Due to the linearity property of the channel, we observe that the received signal $\tilde{r}(n)$ is a multiplex of the receive signal components $\tilde{r}_k(n)$,

$$\tilde{r}(n) = h(n) * s(n) = \sum_k x_k \left(h(n) * s_k(n)\right) = \sum_k x_k \tilde{r}_k(n), \tag{3}$$

where $*$ denotes linear convolution.

This equation suggests that the transmit and receive signal components appear in fixed pairs $\{s_k(n), \tilde{r}_k(n)\}$. However, there are two problems that equation (3) does not immediately reveal. The first is that we somehow need to separate the sum of the received waveforms at the receiver in order to retrieve the data. We will return to this problem in a moment. The second problem appears if we transmit subsequent blocks of length N, one after another, which is necessary in order to convey information. Due to the dispersive channel, a fixed relation between $s_k(n)$ and $\tilde{r}_k(n)$ is not quite obvious. After passing through

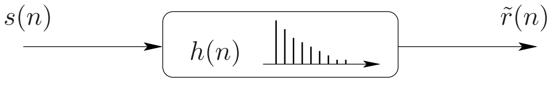


Figure 3: The discrete-time linear channel.

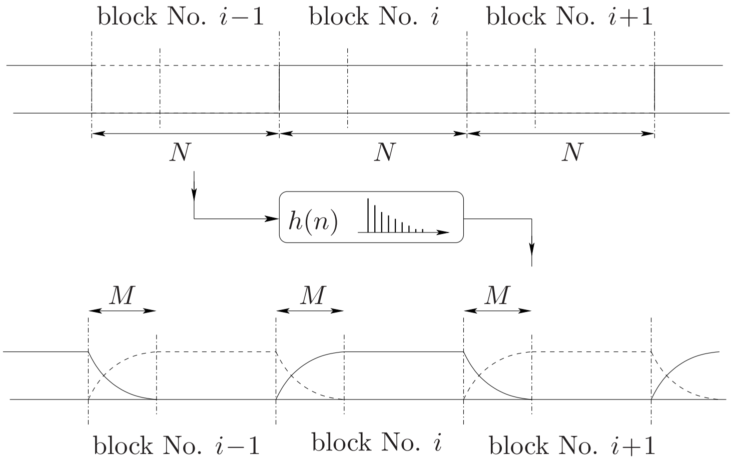


Figure 4: Inter-block interference.

the dispersive channel $h(n)$, the first M samples of any received block will inevitably be corrupted by the *previous* block, as illustrated by Figure 4. In such a communication system, we can *not* find finite-length signals $s_k(n)$ that *always* have the response $\tilde{r}_k(n)$ because the response will depend on the particular preceding waveform.

We focus on the choice of discrete-time waveforms or signals that represent these blocks of real numbers we want to transmit. Central to OFDM is the concept of *subchannels.* By proper arrangements, we seek to address the physical channel in such a way that it appears to the receiver as many, parallel, and independent subchannels. Throughout the book, we will refer to this as *partitioning* the channel into parallel subchannels.

Imagine that we, by some proper choice of transmitter and receiver operations, could make the dispersive channel look like a large number of independent subchannels, through which data symbols would arrive at the receiver only slightly distorted, and in a controllable fashion. Then, we could transmit data symbols in parallel on these subchannels and process the received symbols independently and straightforwardly at the receiver in order to extract the transmitted information, see Figure 5.

In order to overcome the problem of inter-block interference caused by the dispersive channel, we make the following subtle distinction: we ignore the first L samples of each received block of $\tilde{r}(n)$ which yields a new signal $r(n)$. We choose the actual number L such that it is larger or equal than the dispersion M of the channel, which is the length of the channel impulse response minus 1. In order to create parallel subchannels, we transmit signals of length $N+L$, ignore the first received L samples at the receiver and process vectors of length N. We try to find pairs $\{s_k(n), r_k(n)\}$ where the $s_k(n)$ have length $N+L$ and the $r_k(n)$ have length N.

The key aspect of channel partitioning is now to find a set of transmit signals such that the data symbol modulated on one transmit signal component is easily *separable* at the receiver from the other data symbols. We will try to find a set of finite-length signal component pairs $\{s_k(n), r_k(n)\}$ having this

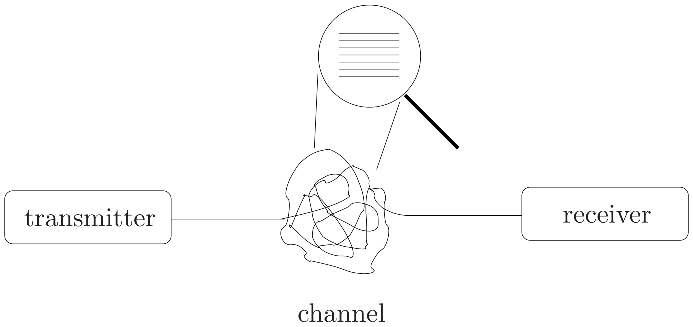


Figure 5: Channel partitioning.

property.

In order to develop the notion of ‘separability’, we introduce the *correlation* of two signals as a separation tool. This measure is the heart of the channel partitioning. We define the correlation of two length-N signals $a(n)$ and $b(n)$ as

$$\langle a, b\rangle = \sum_{n=0}^{N-1} a(n)b^*(n), \tag{4}$$

where * denotes complex conjugation. We say that the signals are uncorrelated, separable, or *orthogonal*, if their correlation is zero. Orthogonality of signals gives us a tool to separate components of a multiplex at the receiver, and thus to retrieve the transmitted data. If the receive signal components $r_k(n)$ are pairwise orthogonal, the symbols are simply recovered from $r(n)$ by

$$\langle r, r_k\rangle = \sum_{n=0}^{N-1} \left(\sum_m x_m r_m(n)\right) r_k^*(n) = \sum_m x_m \langle r_m, r_k\rangle = x_k, \tag{5}$$

provided that the self-correlation $\langle r_k, r_k\rangle$ of each signal is one, i.e., the set consists of orthonormal signals. The receiver correlates the received multiplex $r(n)$ with each of the signal components $r_k(n)$ in order to extract the transmitted symbols x_k.

The importance and practical relevance of this relation must not be underestimated. In general, for arbitrary transmit signals, a dispersive channel distorts a transmitted signal so much that complicated receiver structures are needed to recover the transmitted data. This problem is known as the *equalisation* problem in dispersive channels. For the signals we choose here, the data recovering process is amazingly simple: not only is the signal separation simple (a bank of correlators), but the detection of the data symbols (the processing to retrieve the symbols x_k) is straightforward as well.

A set of transmit signal components and orthogonal receive signal components is by no means unique. There are many orthogonal sets $r_k(n)$, whose elements are the responses of a certain channel $h(n)$ to a certain set of transmit signal components. Furthermore, in general, a channel partitioning only applies to the particular channel $h(n)$ at hand. A signal set partitioning one channel will, in general, not partition another channel.

Imagine therefore that we could find signal pairs $s_k(n)$ and orthogonal $r_k(n)$ that *do not depend* on the particular channel impulse response $h(n)$. Then, the channel partitioning would be valid for *any* dispersive channel. Clearly, such a solution is of particular interest from a practical point of view. The transmitter and the receiver will have a fixed design – the transmit and receive signals can be implemented in hardware and the channel partitioning would offer its simple receiver detection structure in any channel.

Summarising the above, we want to address the channel with finite-length ($N + L$ samples) channel-independent signals $s_k(n)$ such that the receiver, using a bank of length-N channel-independent correlators, perceives the channel as partitioned, as if it were a number of independent subchannels. We try to obtain an orthogonal set of receive signals $r_k(n)$: the data symbols are then separable by the correlation operation (4).

We now formally repeat how we want the channel partitioning to work. As a criterion for the choice of the transmit and receive signals, we present three desired properties. For a given channel $h(n)$, we

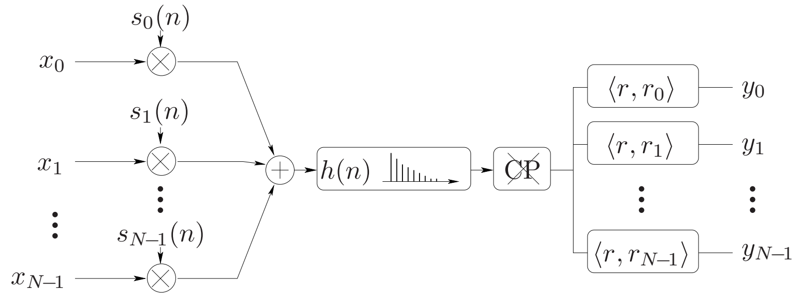


Figure 6: System structure: bank of transmit signal components $s_k(n)$ scaled by the symbols x_k, discrete-time linear time-invariant channel $h(n)$, removal of the cyclic prefix (CP), and bank of correlators $r_k(n)$.

want to find pairs of signals $s_k(n), n = -L, \ldots, N-1$ and $r_k(n), n = 0, \ldots, N-1$ that satisfy

$$1. \quad \langle r_k, r_m \rangle = \begin{cases} 1 & k = m \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$2. \quad r_k(n) = h(n) * s_k(n) \qquad \text{for } n = 0, \ldots, N-1 \tag{7}$$

$$3. \quad r_k(n) \text{ and } s_k(n) \text{ meet condition 2 for all channels } h(n) \tag{8}$$

These three properties allow the simple receiver structure depicted in Figure 6. The orthonormality in (6) allows for the decomposition of the received multiplex. The second property (7) specifies the transmit signals once the received signals have been fixed. Finally, the third property guarantees a fixed and generally applicable implementation of the transmitter multiplexer and the receiver correlator bank.

3 Discrete Multi-Tone

From the theory of linear time-invariant systems, we recall that sinusoids are eigenfunctions of the linear time-invariant channel. That means, if we feed an infinitely long sinusoid through a linear time-invariant channel, only the amplitude and the phase are changed, not its frequency. Although this result only applies to infinitely long signals, it motivates us to explore the following channel partitioning. Consider the set of receive signal components

$$\begin{aligned} r_k^{(\mathrm{i})}(n) &= \frac{1}{\sqrt{N}} \cos(2\pi \frac{k}{N} n), & k &= 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor, \\ r_k^{(\mathrm{q})}(n) &= -\frac{1}{\sqrt{N}} \sin(2\pi \frac{k}{N} n), & k &= 1, \ldots, \left\lfloor \frac{N-1}{2} \right\rfloor, \end{aligned} \qquad n = 0, \ldots, N-1. \tag{9}$$

These N signals are discrete-time, finite-length sinusoids with frequencies $\frac{k}{N}$. Figure 7 shows the signals $r_k^{(\mathrm{i})}(n)$ and $r_k^{(\mathrm{q})}(n)$ for $N = 8$, $k = 0, \ldots N-1$. The receive signal components given by (9), marked in Figure 7 by a grey background, are orthogonal, as shown in Memo I.1. Thus requirement (6) is almost satisfied. The choice of the frequencies of the sinusoids, k/N, guarantees that the N-sample signals contain an integer number of periods. This is the crucial choice for the orthogonality property (6) to hold.

Note that since unique reconstruction of the sine signals $r_0^{(\mathrm{q})}(n)$ and $r_{N/2}^{(\mathrm{q})}(n)$ for even N is not possible, these signals do not carry information. Observe also that the number of oscillations per N samples increases with increasing k as long as $k \in \{0, \ldots, \lfloor N/2 \rfloor\}$, i.e., up to half the sampling frequency.
For $k \in \{\lfloor N/2 \rfloor + 1, \ldots, N-1\}$, the number of oscillations per N samples decreases with increasing k. In fact, it is easy to verify that

$$\begin{aligned} \cos(2\pi \frac{k}{N} n) &= \cos(2\pi \frac{(N-k)}{N} n), & k &= 0, \ldots, N, \\ \sin(2\pi \frac{k}{N} n) &= -\sin(2\pi \frac{(N-k)}{N} n), & k &= 0, \ldots, N, \end{aligned} \tag{10}$$

which is also illustrated in Figure 7. To summarise, the number of mutually orthogonal real-valued length-N discrete-time sinusoidal waveforms with an integer number of periods is equal to N.

The question is now whether we can find the signals we need to transmit in order to get the responses (9). Remembering the eigen-property of a linear time-invariant channel, we choose the N components

$$\begin{aligned} s_k^{(\mathrm{i})}(n) &= \frac{1}{\sqrt{N}} \cos(2\pi \frac{k}{N} n), & k &= 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor, \\ s_k^{(\mathrm{q})}(n) &= -\frac{1}{\sqrt{N}} \sin(2\pi \frac{k}{N} n), & k &= 1, \ldots, \left\lfloor \frac{N-1}{2} \right\rfloor, \end{aligned} \qquad n = -L, \ldots, N-1, \tag{11}$$

which are *identical* to the receive signal components except that they are longer ($N + L$ samples). For convenience, we use the time-indexes $-L, \ldots, N-1$ to refer to the values of these signals. The receive

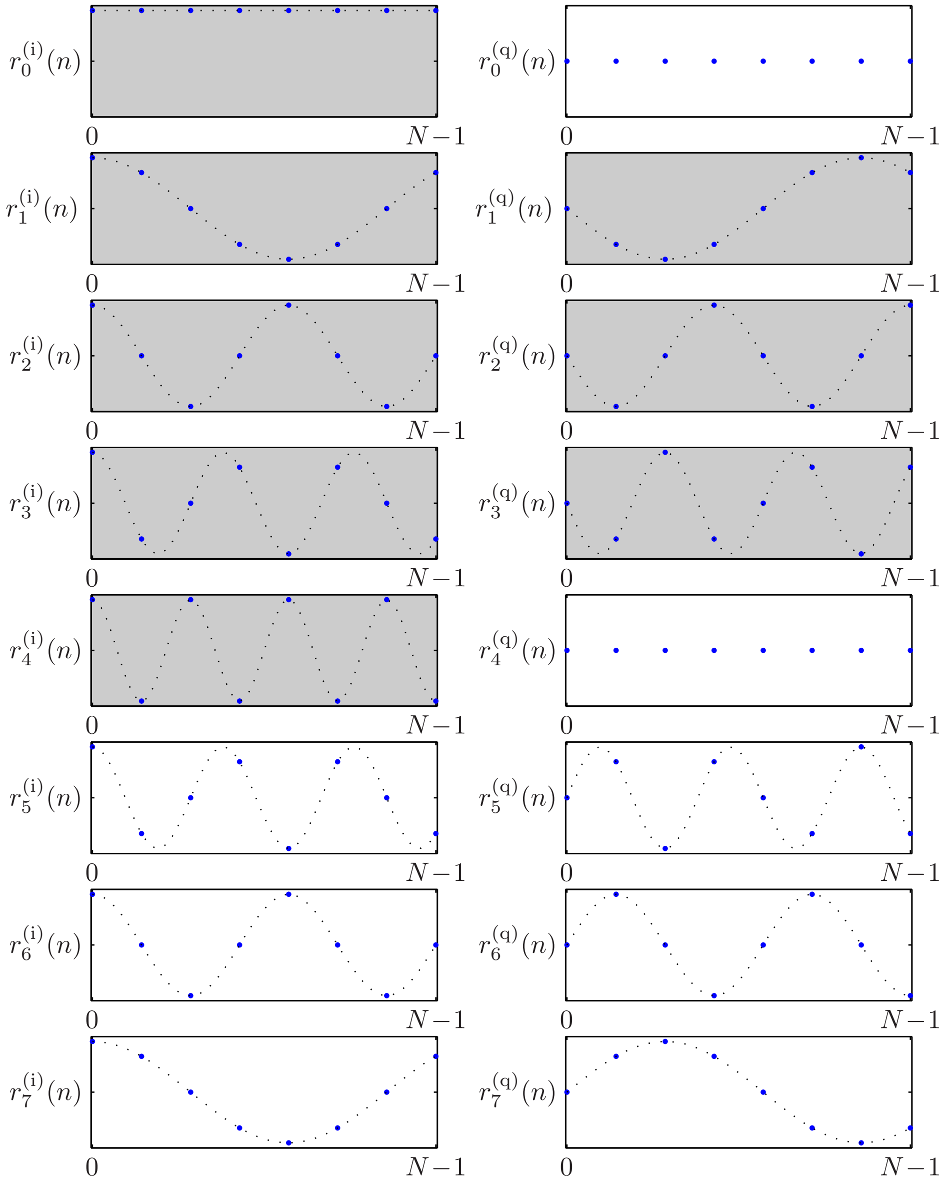


Figure 7: The receive signal components $r_k^{(\mathrm{i})}(n)$ (cosine signals, left column) and $r_k^{(\mathrm{q})}(n)$ (sine signals, right column) for $N = 8$. The receive signal components given by (9), marked with a grey background, are orthogonal, as shown in Memo I.1. We exploit N real-valued dimensions. The thin dotted lines indicate the corresponding continuous-time signals assuming perfect reconstruction within the Nyquist band (up to half the sampling rate).

signals are truncated transmit signals (with indexes $0, \ldots, N-1$). Figure 8 depicts an exemplary pair $\{s_k^{(\mathrm{q})}(n), r_k^{(\mathrm{q})}(n)\}$ of transmit and receive signal components. We will see that this set of signal-pairs $s_k^{(\mathrm{i})}(n), s_k^{(\mathrm{q})}(n), r_k^{(\mathrm{i})}(n), r_k^{(\mathrm{q})}(n)$ has properties similar to the desired properties 1-3, equations (6) to (8), we formulated above.

Memo I.2 shows how the transmit signals (11) behave in a dispersive channel. Remember that we wish to have a fixed and invariant relation between the transmit signals and the receive signals. For the sake of simple notation, we will restrict our considerations to even N hereinafter. As we will see, even values of N are of greater practical importance than odd values. Note, however, that all the concepts

We need to prove the orthogonality for all the pairs in the set of receive signals given by (9). Using the trigonometric facts that

$$\sum_{n=0}^{N-1} \cos(2\pi \frac{k}{N} n) = \begin{cases} N & \text{if } k = 0, N, 2N, \ldots, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \sum_{n=0}^{N-1} \sin(2\pi \frac{k}{N} n) = 0,$$

we find, for $k, m \in \{0, \ldots, \lfloor \frac{N}{2} \rfloor\}$, that

$$\begin{aligned} \langle r_k, r_m \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} \cos(2\pi \frac{k}{N} n) \cos(2\pi \frac{m}{N} n) = \\ &= \frac{1}{2N} \sum_{n=0}^{N-1} \left[\cos(2\pi \frac{k-m}{N} n) + \cos(2\pi \frac{k+m}{N} n) \right] = \\ &= \begin{cases} 1 & \begin{cases} \text{if } k = m \in \{0, N/2\}, & N \text{ even,} \\ \text{if } k = m = 0, & N \text{ odd,} \end{cases} \\ \frac{1}{2} & \begin{cases} \text{if } k = m \in \{1, \ldots, N/2 - 1\}, & N \text{ even,} \\ \text{if } k = m \in \{1, \ldots, \lfloor \frac{N}{2} \rfloor\}, & N \text{ odd,} \end{cases} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Similarly, we obtain for $k, m \in \{1, \ldots, \lfloor \frac{N-1}{2} \rfloor\}$

$$\frac{1}{N} \sum_{n=0}^{N-1} \sin(2\pi \frac{k}{N} n) \sin(2\pi \frac{m}{N} n) = \begin{cases} \frac{1}{2} & \text{if } k = m, \\ 0 & \text{otherwise} \end{cases}$$

and for $k \in \{1, \ldots, \lfloor \frac{N-1}{2} \rfloor\}, m \in \{0, \ldots, \lfloor \frac{N}{2} \rfloor\}$

$$\frac{1}{N} \sum_{n=0}^{N-1} (-\sin(2\pi \frac{k}{N} n)) \cos(2\pi \frac{m}{N} n) = 0.$$

Memo I.1: Proof of the orthogonality of the vectors (9).

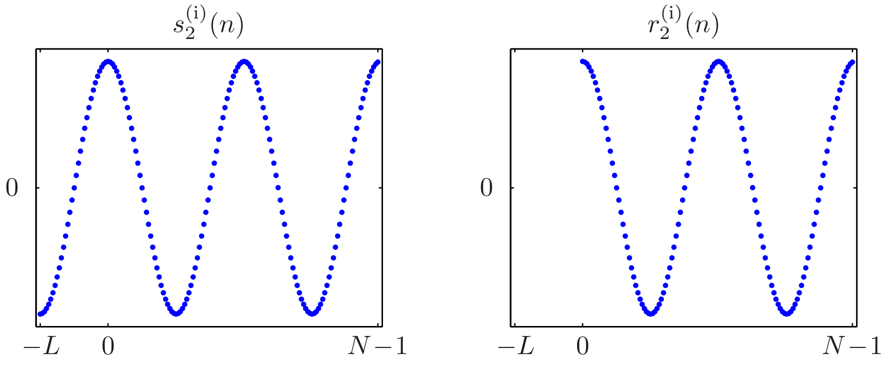


Figure 8: Exemplary transmit and receive signal components for $k = 2$, $N = 128$, $L = 32$.

can be straightforwardly formulated for odd N. At the transmitter, we construct the following multiplex using the real-valued data symbols $x_k^{(\mathrm{i})}$ and $x_k^{(\mathrm{q})}$ and the transmit signal components (11):

$$s(n) = \frac{1}{\sqrt{N}} \left(x_0^{(\mathrm{i})} + 2 \sum_{k=1}^{N/2-1} \left(x_k^{(\mathrm{i})} \cos(2\pi \frac{k}{N} n) - x_k^{(\mathrm{q})} \sin(2\pi \frac{k}{N} n) \right) + x_{N/2}^{(\mathrm{i})} \cos(\pi n) \right). \tag{12}$$

We have introduced a scaling factor 2 for the transmit components $k = 1, \ldots, \frac{N}{2} - 1$, which may appear like an arbitrary choice at this point. The purpose of this scaling factor will become clear soon. Using the results from Memo I.2, we find that

$$r(n) = \frac{1}{\sqrt{N}} \left(A_0 x_0^{(\mathrm{i})} + 2 \sum_{k=1}^{N/2-1} \left((A_k x_k^{(\mathrm{i})} - B_k x_k^{(\mathrm{q})}) \cos(2\pi \frac{k}{N} n) - \right. \right. \\ \left. \left. (B_k x_k^{(\mathrm{i})} + A_k x_k^{(\mathrm{q})}) \sin(2\pi \frac{k}{N} n) \right) + A_{N/2} x_{N/2}^{(\mathrm{i})} \cos(\pi n) \right), \tag{13}$$

which shows two things. First, we observe that the receive signal components are scaled by coefficients A_k and B_k, which depend on the channel. Secondly, we note that one transmit signal component $s_k^{(\mathrm{i})}(n), k = 1, \ldots, N-1$ does not only map onto its corresponding receive signal component $r_k^{(\mathrm{i})}(n)$ but also on $r_k^{(\mathrm{q})}(n)$. The transmit signal components $s_k^{(\mathrm{q})}(n)$ exhibit the same behaviour, they do not only map onto $r_k^{(\mathrm{q})}(n)$ but also on $r_k^{(\mathrm{i})}(n)$. One transmitted symbol modulated onto one transmit component will affect *two* correlator outputs. Apparently, the second property (7) is not completely satisfied. Figure 9 depicts exemplary transmit and receive multiplexes.

Finally, the third desired property (8) is not satisfied either, since the length L of the Cyclic Prefix (CP) required to avoid the aforementioned inter-block interference, depends on the channel. Apparently, the choice (9) and (11) does not fully satisfy any of the desired properties. The set of signals (9) and (11) is, however, as close as it gets to fulfilling (6)–(8) and we will continue exploring this choice. An important part of Memo I.2 is that it gives an explicit expression for the attenuations A_k and B_k: they are the N-point cosine- and sine-transform of the channel impulse response.

At the receiver, we implement a bank of the corresponding correlators. Using the results from Memo I.2 and (13), we recover the data symbols as

$$\begin{aligned}
\langle r, r_0^{(\mathrm{i})} \rangle &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} r(n) = A_0 x_0^{(\mathrm{i})}, \\
\langle r, r_k^{(\mathrm{i})} \rangle &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} r(n) \cos(2\pi \frac{k}{N} n) = A_k x_k^{(\mathrm{i})} - B_k x_k^{(\mathrm{q})}, \qquad k = 1, \ldots, N/2-1, \\
\langle r, r_{N/2}^{(\mathrm{i})} \rangle &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} r(n) \cos(\pi n) = A_{N/2} x_{N/2}^{(\mathrm{i})}, \\
\langle r, r_k^{(\mathrm{q})} \rangle &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} r(n)(-\sin(2\pi \frac{k}{N} n)) = B_k x_k^{(\mathrm{i})} + A_k x_k^{(\mathrm{q})}, \qquad k = 1, \ldots, N/2-1.
\end{aligned} \tag{14}$$

We obtain a channel partitioning in which the data symbols $x_k^{(\mathrm{i})}$ and $x_k^{(\mathrm{q})}$ are scaled and spread at the receiver over two of the correlator outputs matched to the receive components $r_k^{(\mathrm{i})}(n)$ and $r_k^{(\mathrm{q})}(n)$ –

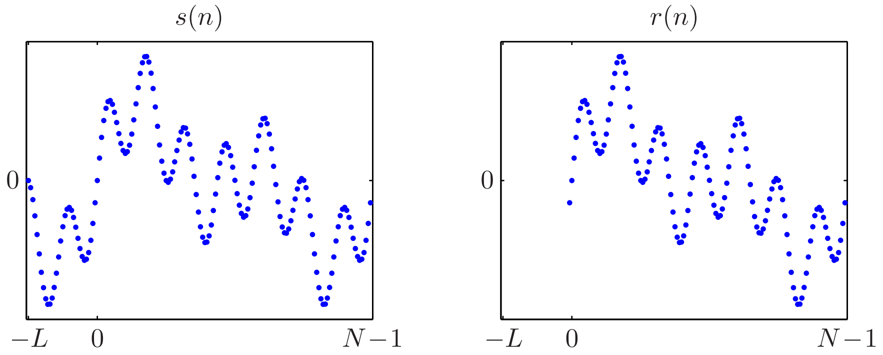


Figure 9: Exemplary transmitted multiplex $s(n)$ and received multiplex $r(n)$ ($N = 128$, $L = 32$).

For $n = 0, \ldots, N-1$:

$$\begin{aligned}
h(n) * \cos(2\pi\frac{k}{N}n) &= \sum_{m=0}^{M} h(m)\cos(2\pi\frac{k}{N}(n-m)) \\
&= \sum_{m=0}^{M} h(m)\Big(\cos(2\pi\frac{k}{N}n)\cos(2\pi\frac{k}{N}m) \\
&\qquad\qquad + \sin(2\pi\frac{k}{N}n)\sin(2\pi\frac{k}{N}m)\Big) \\
&= A_k\cos(2\pi\frac{k}{N}n) - B_k\sin(2\pi\frac{k}{N}n)
\end{aligned}$$

and

$$\begin{aligned}
h(n) * (-\sin(2\pi\frac{k}{N}n)) &= -\sum_{m=0}^{M} h(m)\sin(2\pi\frac{k}{N}(n-m)) \\
&= -\sum_{m=0}^{M} h(m)\Big(\sin(2\pi\frac{k}{N}n)\cos(2\pi\frac{k}{N}m) \\
&\qquad\qquad - \cos(2\pi\frac{k}{N}n)\sin(2\pi\frac{k}{N}m)\Big) \\
&= -B_k\cos(2\pi\frac{k}{N}n) - A_k\sin(2\pi\frac{k}{N}n)
\end{aligned}$$

where

$$\begin{aligned}
A_k &= \sum_{m=0}^{M} h(m)\cos(2\pi\frac{k}{N}m), \\
B_k &= -\sum_{m=0}^{M} h(m)\sin(2\pi\frac{k}{N}m),
\end{aligned}$$

Memo I.2: The response of a channel $h(n)$ to the transmit signal components (11).

they are *almost* separated. However, we can talk about parallel channels in the sense that pairs of two subchannels, characterised by a unique centre frequency, carry *two* data symbols. These subchannels are completely separated at the receiver by means of the orthogonal receiver filters. Furthermore, if the receiver *knows* the channel attenuations A_k and B_k, it can easily separate the two real-valued data symbols from the two correlator outputs on each of the subchannels (the correlator outputs (14) are a linear combination of two unknowns).

Let us now introduce an interpretation of the transmitter and receiver operations which will both aid our manipulation and at the same time increase our understanding of the system. As a notational convention, we collect the N real-valued symbols $x_k^{(\mathrm{i})}$, $x_k^{(\mathrm{q})}$ into $N/2$ complex-valued data symbols $\underline{x}_k$. Similarly, at the receiver, we collect the N real-valued correlator outputs $\langle r, r_k^{(\mathrm{i})}\rangle$, $\langle r, r_k^{(\mathrm{q})}\rangle$ into $N/2$ complex-valued correlator outputs $\underline{y}_k$. We define

$$\underline{x}_k \triangleq x_k^{(\mathrm{i})} + jx_k^{(\mathrm{q})} \qquad \underline{y}_k \triangleq \langle r, r_k^{(\mathrm{i})}\rangle + j\langle r, r_k^{(\mathrm{q})}\rangle. \tag{15}$$

With this notation and using (14), we get the compact relation $\underline{y}_k = \underline{H}_k\underline{x}_k$ through

$$\underline{y}_k = (A_k x_k^{(\mathrm{i})} - B_k x_k^{(\mathrm{q})}) + j(B_k x_k^{(\mathrm{i})} + A_k x_k^{(\mathrm{q})}) = (A_k + jB_k)(x_k^{(\mathrm{i})} + jx_k^{(\mathrm{q})}) = \underline{H}_k\underline{x}_k \tag{16}$$

where

$$\underline{H}_k \triangleq A_k + jB_k. \tag{17}$$

In the definitions (15) and (17) we have been somewhat sloppy ignoring that the elements $x_0^{(q)}$, $x_{N/2}^{(q)}$, B_0, $B_{N/2}$ have not been defined previously. However, since these components will appear later when we develop the complex baseband multiplex, we choose to formulate the definitions more generally. For now, these definitions should be interpreted such that the yet undefined elements are simply equal to zero, i.e., $\underline{x}_0 \triangleq x_0^{(i)}$. Note that in our mathematical notation all complex-valued quantities are underlined. Complex notation is a convenient way to compactly denote orthogonal but still related quantities. In a practical implementation, a complex-valued signal is implemented as two parallel signal paths.

With Memo I.2 we rewrite the complex attenuation factors $\underline{H}_k$ as

$$\underline{H}_k = \sum_{m=0}^{M} h(m) \left(\cos(2\pi \frac{k}{N} m) - j \sin(2\pi \frac{k}{N} m) \right) = \sum_{m=0}^{M} h(m) e^{-j2\pi \frac{k}{N} m}. \tag{18}$$

and we recognise that $\underline{H}_k$ are the discrete Fourier transform (DFT) coefficients of the discrete-time channel $h(n)$. In other words, the subchannel attenuations are samples of the Fourier transform of the discrete-time channel $h(n)$.

Summarising, with the choice of signals made in (9) and (11), we obtain the channel partitioning

$$\underline{y}_k = \underline{H}_k \underline{x}_k \qquad \text{where} \quad \underline{H}_k = \sum_{n=0}^{M} h(n) e^{-j2\pi \frac{nk}{N}}. \tag{19}$$

This relation shows that the signals (9) and (11) are actually closer to satisfying property (7) than we thought. With a fixed transmit and receive structure (the third requirement) we can, for any dispersive channel, truly talk about complex-valued subchannels – one for each complex-valued data symbol. The DFT has also relevance for the implementation in the transmitter and the receiver as we will see in the next section.

Let us now have a closer look at the bank of correlators at the receiver described by (14). We process the received signal $r(n)$ according to

$$\underline{y}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} r(n) e^{-j2\pi \frac{k}{N} n}, \qquad k = 0, \ldots, \frac{N}{2}, \tag{20}$$

which is nothing else than the scaled N-point Discrete Fourier Transform (DFT) evaluated at $k = 0, \ldots, \frac{N}{2}$. So, not only have we managed to reduce the receiver complexity by channel partitioning (turning the channel dispersion into a multiplication acting on the subchannels), we have now also access to a low-complexity tool at the receiver: the Fast Fourier Transform (FFT). This algorithm, first published in 1965, efficiently computes the DFT if the number of points N can be factored. The FFT is particularly suitable for hardware implementation. The result obtained in (20) explains the factor 2 introduced in (12).

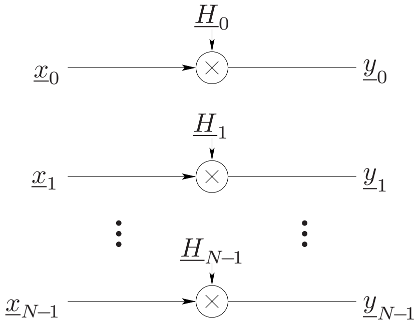


Figure 10: Parallel channel model for the OFDM/DMT system.

The transmit multiplex $s(n)$ given by (12) can be written as

$$\begin{aligned} s(n) = \frac{1}{\sqrt{N}} \Bigg(& x_0^{(\mathrm{i})} + \sum_{k=1}^{N/2-1} \left(x_k^{(\mathrm{i})} \cos(2\pi \frac{k}{N} n) - x_k^{(\mathrm{q})} \sin(2\pi \frac{k}{N} n) \right) + \\ & + x_{N/2}^{(\mathrm{i})} \cos(\pi n) + \\ & + \sum_{k=N/2+1}^{N-1} \left(x_{N-k}^{(\mathrm{i})} \cos(2\pi \frac{N-k}{N} n) - x_{N-k}^{(\mathrm{q})} \sin(2\pi \frac{N-k}{N} n) \right) \Bigg) . \end{aligned} \tag{21}$$

We make the following choice for the transmit symbols

$$x_k^{(\mathrm{i})} = x_{N-k}^{(\mathrm{i})} \quad \text{and} \quad x_k^{(\mathrm{q})} = -x_{N-k}^{(\mathrm{q})}, \tag{22}$$

which implies $x_0^{(\mathrm{q})} = x_{N/2}^{(\mathrm{q})} = 0$. Using the identities (10), we obtain for $n = -L, \ldots, N-1$

$$\begin{aligned} s(n) = & \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(x_k^{(\mathrm{i})} \cos(2\pi \frac{k}{N} n) - x_k^{(\mathrm{q})} \sin(2\pi \frac{k}{N} n) \right) \\ & + j \frac{1}{\sqrt{N}} \underbrace{\sum_{k=0}^{N-1} \left(x_k^{(\mathrm{i})} \sin(2\pi \frac{k}{N} n) + x_k^{(\mathrm{q})} \cos(2\pi \frac{k}{N} n) \right)}_{=0} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \underline{x}_k e^{j2\pi \frac{k}{N} n} . \end{aligned} \tag{23}$$

On the interval $0, \ldots, N-1$, this is the scaled N-point Inverse Discrete Fourier Transform (IDFT) of the data symbols we want to transmit. The corresponding fast version of the IDFT, the Inverse Fast Fourier Transform (IFFT), is used in the transmitter to generate the signal $s(n)$. Note that the choice made in (22) ensures Hermitian symmetry in the DFT domain and yields a real-valued time-domain signal $s(n)$.

The signal samples $s(n)$ for $-L, \ldots, -1$ can be obtained from the other signal values: they are copies of $s(n)$ for $N-L, \ldots, N-1$. This prefix extends the transmitted signal in a cyclic manner and is therefore often referred to as the *cyclic prefix*. Figure 11 illustrates the cyclic prefix.

At this point, the following equations summarise the multicarrier system we have been looking at so far:

$$\begin{aligned} \underline{x}_{N-k} &= \underline{x}_k^* && k = 0, \ldots, N/2 \\ s(n) &= \mathrm{IDFT}_N \{\underline{x}_k\}_{k=0}^{N-1} && n = 0, \ldots, N-1 \\ s(n) &= s(n+N), && n = -L, \ldots, -1 \\ r(n) &= h(n) * s(n), && n = 0, \ldots, N-1 \\ \underline{y}_k &= \mathrm{DFT}_N \{r(n)\}_{n=0}^{N-1} && k = 0, \ldots, N/2 \end{aligned} \tag{24}$$

We will refer to this system, schematically depicted in Figure 12, as *'DMT' system*. Exponential base functions and a cyclic prefix are the key characteristics. We refer to one multiplex of data symbols as a 'DMT symbol', to a sequence of such DMT symbols as a 'DMT signal' and to one pair of transmit/receive signals as a 'subcarrier' (addressing its exponential characteristics) or a 'tone'. Note that we distinguish a *DMT symbol* $s(n)$ from a *data symbol* $\underline{x}_k$. One of the distinguishing elements between what is understood by most people under DMT and OFDM is that the multiplex $s(n)$ of DMT is real-valued.

4 Orthogonal Frequency Division Multiplex

Up to now we have tacitly assumed that the discrete-time channel $h(n)$ actually passes all the tones $k = 0, \ldots, N-1$. If one of the channel attenuations $\underline{H}_k$ would be zero for a particular subchannel, the data symbol $\underline{y}_k$ modulated on this subchannel could never be recovered at the receiver.

A potential zero-response, or a very severe attenuation of any of the subchannels is not unusual. Moreover, in many situations we know the usable frequencies of the system in advance, and, furthermore,

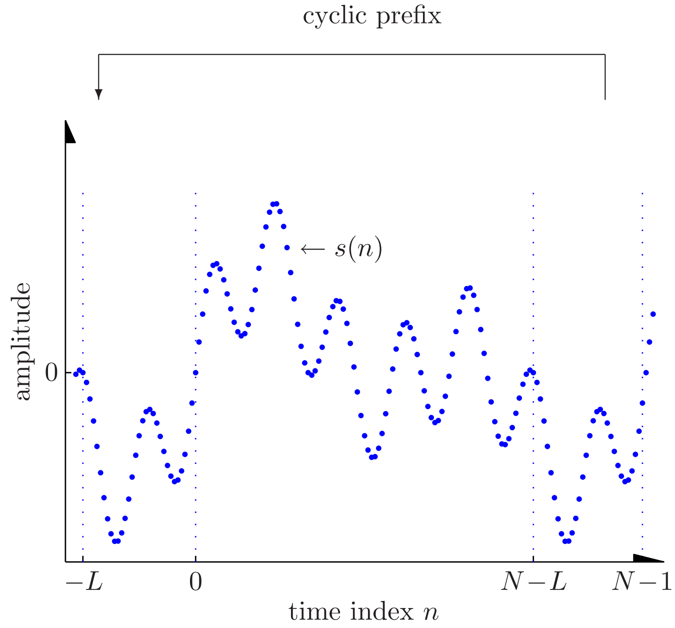


Figure 11: An interpretation of the cyclic prefix: it is a copy of the last L samples of a symbol.

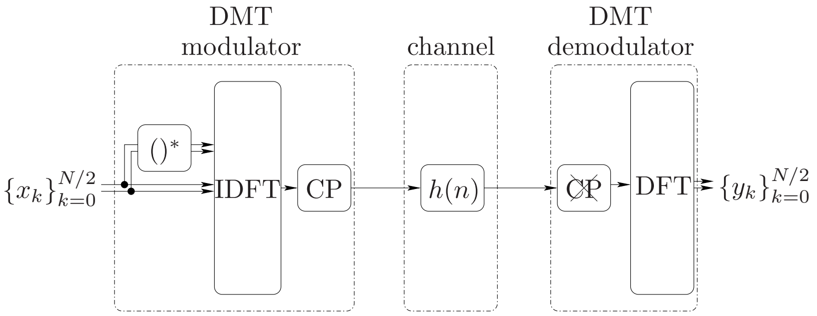


Figure 12: Elementary structure of the DMT system (24).

in many situations the baseband frequencies do *not* belong to these useful frequencies. In radio channels, for instance, the useful frequencies are grouped into a narrow frequency region near a carrier frequency and the channel is said to have a *passband* characteristic.

Which multiplex do we transmit if $h(n)$ is a passband channel? Perhaps the most obvious idea at this point is to simply increase the frequencies of the sinusoidal signals we used in the previous section. Obviously, in order to resolve higher frequencies we need to represent the signals with more samples. Therefore, let us for this purpose explore the signals of length Np samples (p is an integer)

$$\begin{aligned} v_k^{(\mathrm{i})}(m) &= \frac{1}{\sqrt{N}}\cos(2\pi\left(f_{\mathrm{c}}+\frac{k}{Np}\right)m), \\ v_k^{(\mathrm{q})}(m) &= -\frac{1}{\sqrt{N}}\sin(2\pi\left(f_{\mathrm{c}}+\frac{k}{Np}\right)m), \end{aligned} \qquad m = 0,\ldots,Np-1, \tag{25}$$

where $k = -N/2+1,\ldots,N/2$, f_{c} is a positive *carrier frequency*, and the oversampling factor p is sufficiently large to ensure that $f_{\mathrm{c}} + \frac{k}{pN} < \frac{1}{2}$, $k = -N/2+1,\ldots,N/2$ holds for all the N normalised frequencies. Exemplary receive signal components for $N = 8, p = 32, f_{\mathrm{c}} = 1/32$ are shown in Figure 13.

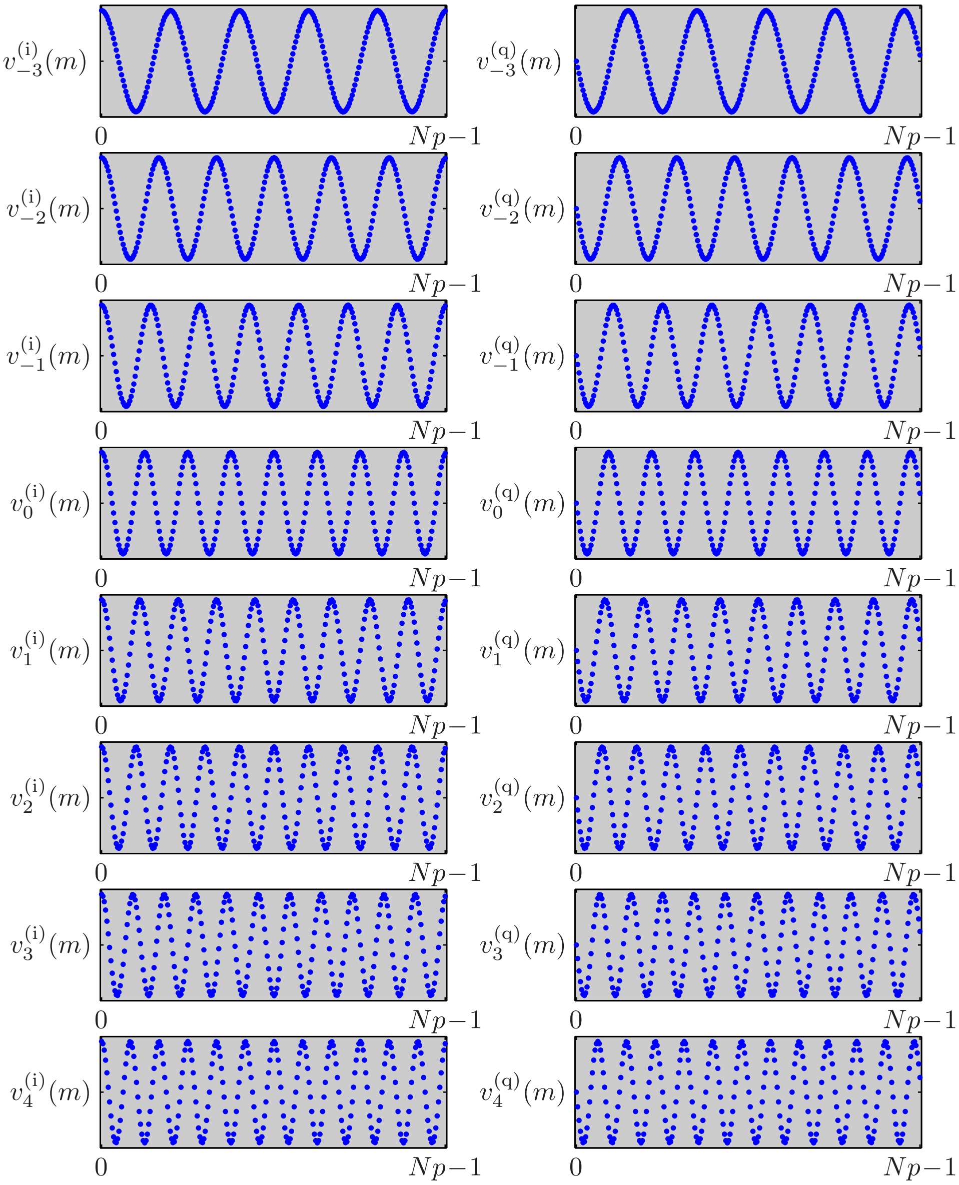


Figure 13: The passband receive signal components $v_k^{(\mathrm{i})}(m)$ (cosine signals, left column) and $v_k^{(\mathrm{q})}(m)$ (sine signals, right column) according to (25) for $N = 8, p = 32, f_\mathrm{c} = 1/32,\ k = -3, \dots, 4$. We exploit $2N$ real-valued dimensions.

Similarly, we consider the transmit signals of length $(N + L)p$ samples

$$\begin{aligned} u_k^{(\mathrm{i})}(m) &= \frac{1}{\sqrt{N}} \cos(2\pi \left(f_\mathrm{c} + \frac{k}{Np} \right) m), \\ u_k^{(\mathrm{q})}(m) &= -\frac{1}{\sqrt{N}} \sin(2\pi \left(f_\mathrm{c} + \frac{k}{Np} \right) m), \end{aligned} \qquad m = -Lp, \dots, Np - 1, \tag{26}$$

where $k = -N/2 + 1, \dots, N/2$. The signal properties illustrated in Memo I.1 and Memo I.2 can straightforwardly be extended to prove that the signals (25) and (26) have the same desirable properties as the baseband signals (9) and (11) and therefore they *partition* the passband channel.

With the transmit signal components (26), we can write the passband transmit signal as

$$\begin{aligned} u(m) &= \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \left(x^{(\mathrm{i})}_{[k]_N} \cos(2\pi \left(f_\mathrm{c}+\frac{k}{Np}\right) m) - x^{(\mathrm{q})}_{[k]_N} \sin(2\pi \left(f_\mathrm{c}+\frac{k}{Np}\right) m)\right) \\ &= \mathrm{Re}\left\{ \underbrace{\frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \underline{x}_{[k]_N} e^{j2\pi \frac{k}{Np} m}}_{\underline{s}(m)} e^{j2\pi f_\mathrm{c} m} \right\}, \end{aligned} \tag{27}$$

where $[k]_N$ denotes the modulo-N operation applied to k. We use the modulo operator to compactly denote that negative k values are replaced by their complement $N+k$. We recognise the term $e^{j2\pi f_\mathrm{c} m}$, which represents the sine-cosine modulation introduced earlier along with the idea of QAM. The input signal $\underline{s}(m)$ to this sine-cosine modulator is nothing else than an oversampled version of the baseband multiplex we already encountered before.

Let us for now assume that the oversampled multiplex $\underline{s}(m)$ is generated by a perfect factor-p oversampling operation applied to the baseband multiplex, which then can be described by

$$\begin{aligned} \underline{s}(n) &= \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \underline{x}_{[k]_N} e^{j2\pi \frac{k}{N} n} \\ &= \frac{1}{\sqrt{N}} \left(\sum_{k=0}^{N/2} \underline{x}_k e^{j2\pi \frac{k}{N} n} + \sum_{k=-N/2+1}^{-1} \underline{x}_{\underbrace{(N+k)}_{m}} e^{j2\pi \frac{k}{N} n} \right) \\ &= \frac{1}{\sqrt{N}} \left(\sum_{k=0}^{N/2} \underline{x}_k e^{j2\pi \frac{k}{N} n} + \sum_{m=N/2+1}^{N-1} \underline{x}_m e^{j2\pi \frac{m-N}{N} n} \right) \\ &= \frac{1}{\sqrt{N}} \left(\sum_{k=0}^{N/2} \underline{x}_k e^{j2\pi \frac{k}{N} n} + \sum_{k=N/2+1}^{N-1} \underline{x}_k \underbrace{e^{-j2\pi \frac{N-k}{N} n}}_{e^{j2\pi \frac{k}{N} n}} \right) \\ &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \underline{x}_k e^{j2\pi \frac{k}{N} n}, \end{aligned} \tag{28}$$

which is again just the scaled N-point IDFT of the transmit symbols $\underline{x}_k$. However, note that the baseband multiplex described by (28) is complex-valued. We can interpret (28) as a multiplex using the transmit signal components

$$\underline{s}_k(n) = \frac{1}{\sqrt{N}} e^{j2\pi \frac{kn}{N}}, \qquad n = -L, \ldots, N-1 \tag{29}$$

Using Memo I.2, the passband transmit signal $u(m)$ yields the channel output

$$\begin{aligned} v(m) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \Bigg(&(A_{[k]_N} x^{(\mathrm{i})}_{[k]_N} - B_{[k]_N} x^{(\mathrm{q})}_{[k]_N}) \cos(2\pi \left(f_\mathrm{c}+\frac{k}{pN}\right) m) - \\ &(B_{[k]_N} x^{(\mathrm{i})}_{[k]_N} + A_{[k]_N} x^{(\mathrm{q})}_{[k]_N}) \sin(2\pi \left(f_\mathrm{c}+\frac{k}{pN}\right) m) \Bigg) \end{aligned} \tag{30}$$

The receiver performs the cosine-sine demodulation, low pass filtering and downsampling by factor p of the receive signal, which yields

$$\underline{r}(n) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \left((A_{[k]_N} x^{(\mathrm{i})}_{[k]_N} - B_{[k]_N} x^{(\mathrm{q})}_{[k]_N}) + j(B_{[k]_N} x^{(\mathrm{i})}_{[k]_N} + A_{[k]_N} x^{(\mathrm{q})}_{[k]_N}) \right) e^{j2\pi \frac{k}{N} n}, \tag{31}$$

which can be written as

$$\underline{r}(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left((A_k x^{(\mathrm{i})}_k - B_k x^{(\mathrm{q})}_k) + j(B_k x^{(\mathrm{i})}_k + A_k x^{(\mathrm{q})}_k) \right) e^{j2\pi \frac{k}{N} n}, \tag{32}$$

following the steps of (28). We can interpret (32) as receive multiplex using the receive signal components

$$\underline{r}_k(n) = \frac{1}{\sqrt{N}} e^{j2\pi\frac{kn}{N}}, \qquad n = 0, \ldots, N-1 \tag{33}$$

Finally, we pass the complex-valued receive multiplex $\underline{r}(n)$ through a bank of correlators, which yields

$$\langle \underline{r}, \underline{r}_k \rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \underline{r}(n) e^{-j2\pi\frac{k}{N}n} = (A_k x_k^{(\mathrm{i})} - B_k x_k^{(\mathrm{q})}) + j(B_k x_k^{(\mathrm{i})} + A_k x_k^{(\mathrm{q})}) \tag{34}$$

where $k = 0, \ldots, N-1$. The operation performed by these correlators is equivalent to the scaled N-point DFT of the complex-valued baseband receive multiplex $\underline{r}(n)$

$$\underline{y}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \underline{r}(n) e^{-j2\pi\frac{k}{N}n}. \tag{35}$$

By collecting sine and cosine waveform components into a complex notation similar to our earlier approach, we obtain as the complex output of the correlator for frequency k

$$\underline{y}_k = \underline{H}_k \underline{x}_k, \qquad k = 0, \ldots, N-1 \tag{36}$$

because the received signals (25) are orthogonal. The channel attenuations $\underline{H}_k$ are now, similar to the result in Memo I.2,

$$\underline{H}_{[k]_N} = \sum_{m=0}^{(M+1)p-1} h(m) e^{-j2\pi(f_\mathrm{c}+\frac{k}{pN})m}, \qquad k = -N/2+1, \ldots, N/2 \tag{37}$$

Let us again summarise the system we have been looking at:

$$\begin{aligned}
\underline{s}(n) &= \mathrm{IDFT}_N \{\underline{x}_k\}_{k=0}^{N-1} && n = 0, \ldots, N-1 \\
\underline{s}(n) &= \underline{s}(n+N), && n = -L, \ldots, -1 \\
\underline{s}(m) &= \{\underline{s}(n)\}_{\uparrow p}, && m = -Lp, \ldots, Np-1 \\
u(m) &= \mathrm{Re}\left\{\underline{s}(m) e^{j2\pi f_\mathrm{c} m}\right\} && \\
v(m) &= u(m) * h(m) && \\
\underline{r}(m) &= \left(v(m)\, 2e^{-j2\pi f_\mathrm{c} m}\right) * h_\mathrm{LP}(m) && \\
\underline{r}(n) &= \{\underline{r}(m)\}_{\downarrow p}, && n = 0, \ldots, N-1 \\
\underline{y}_k &= \mathrm{DFT}_N \{\underline{r}(n)\}_{n=0}^{N-1} && k = 0, \ldots, N-1
\end{aligned} \tag{38}$$

We will refer to this system, depicted in Figure 14, as *'OFDM' system.* The 'O' refers to the orthogonality of the vectors r_k at the receiver, 'FD' refers to the fact that the partitioning in fact is

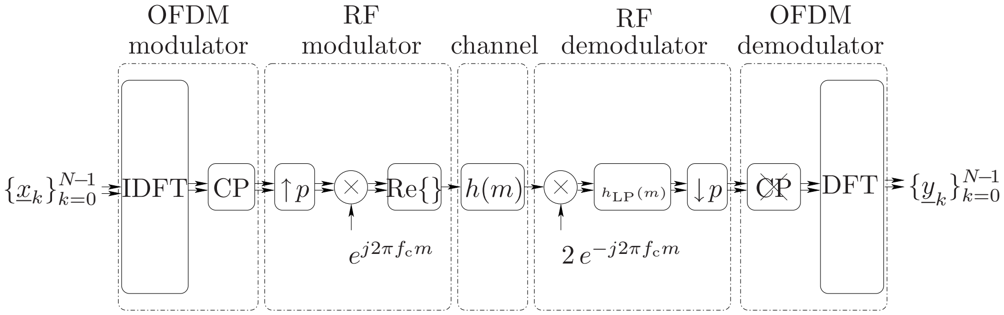


Figure 14: Elementary structure of the OFDM system (38).

a frequency-partitioning, and the 'M' finally expresses the fact that we transmit a multiplex of data symbols. OFDM is then an abbreviation for *Orthogonal Frequency Division Multiplexing.*

We will find it useful to make a *model* of the above passband system. We have seen already in the previous section that we can interpret the data symbols as complex numbers by grouping them in pairs modulating sine and cosine waves with the same frequency. At the receiver we interpret the output of the associated correlators again as complex numbers.

We now introduce a model of a fictitious system which has the same *behaviour* as the real-world, implementable passband system above. The received *complex* signal can be written as

$$\underline{r}(n) = \underline{h}(n) * \underline{s}(n), \tag{39}$$

where the *complex* channel impulse response $\underline{h}(n)$ is given by

$$\underline{h}(n) = \mathrm{IDFT}_N \left\{ \underline{H}_k \right\} = \sum_{k=0}^{N-1} \underline{H}_k e^{j2\pi \frac{kn}{N}}. \tag{40}$$

The received values become

$$\underline{y}_k = \sum_{n=0}^{N-1} \underline{r}(n) e^{-j2\pi \frac{kn}{N}} = \mathrm{DFT}_N \left\{ \underline{r}(n) \right\} \tag{41}$$

This model system *behaves* as the passband system in that $\underline{y}_k = \underline{H}_k \underline{x}_k$ but the complex signals have low-pass character.

5 Signal Spectra and Block Tone Grid

Let us now again look at the frequency characteristics of the transmit and the receive signal components using the Fourier transform. The explicit expressions are derived in Memo I.3. The Fourier transforms of a set of signal components are shown in Figure 15. Both the transmitted and the received OFDM/DMT multiplex consist of a number of discrete signals, each resonating at a single frequency, while the signal lasts. However, the frequency content of the multiplexes is infinite, since the multiplexes

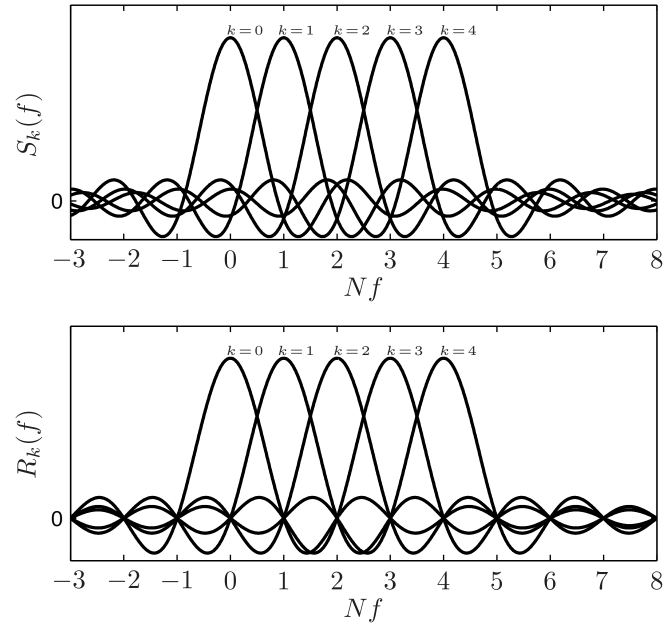


Figure 15: Fourier transforms $S_k(f)$ and $R_k(f)$ of transmit signal components $s_k(n)$ and receive signal components $r_k(n)$, respectively ($k = 0, 1, 2, 3, 4,\ N = 32,\ L = 4$). The subcarriers are orthogonal at the receiver but not at the transmitter.

are of finite length. Observe that the Fourier transform $R_k(f)$ of the receive signal component $r_k(n)$ is exactly zero for $f = \frac{m}{N}$ for $m \neq k$. Consequently, at $f = \frac{k}{N}$ only $r_k(n)$ contributes to the frequency content of the multiplex. This property is the frequency-domain appearance of orthogonality and finally reveals the seemingly miraculous outcome of the experiment discussed in the beginning of this chapter. The choice of the set of N sine and cosine signals of length N with exclusively integer number of periods ensures orthogonality. The separability of the components, which has been established in Memo I.1 in time domain, is obvious in frequency domain as depicted in Figure 15. The DFT at the receiver evaluates the frequency response exactly at k/N for each received block length. We will call these frequencies the *tones* of the OFDM/DMT system. Orthogonality is accomplished for the tones of the system – and only at the receiver.

The Fourier transform of the receive signal component $\underline{r}_k(n)$ is a function of the normalised frequency f:

$$R_k(f) = \mathcal{F}\left\{r_k(n)\right\} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-j2\pi\frac{k}{N}n} e^{-j2\pi f n} =$$

$$= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-j2\pi\left(f+\frac{k}{N}\right)n} = \frac{1}{\sqrt{N}} \frac{1 - e^{-j2\pi(f+\frac{k}{N})N}}{1 - e^{-j2\pi(f+\frac{k}{N})}} =$$

$$= \frac{1}{\sqrt{N}} \frac{\sin(\pi(f+\frac{k}{N})N)}{\sin(\pi(f+\frac{k}{N}))} e^{-j\pi(f+\frac{k}{N})(N-1)},$$

where we use the geometric series expression and the Euler formulae in the last two equalities, respectively. Similarly, the Fourier transform of the transmit signal component $\underline{s}_k(n)$ becomes

$$S_k(f) = \frac{1}{\sqrt{N}} \frac{\sin(\pi(f+\frac{k}{N})(N+L))}{\sin(\pi(f+\frac{k}{N}))} e^{-j\pi(f+\frac{k}{N})(N-L-1)}$$

Memo I.3: The Fourier transforms of $\underline{s}_k(n)$ and $\underline{r}_k(n)$.

For any tone at the receiver, the contribution of the other tones vanishes. We will use the term *tone-domain* to refer to the discrete N-dimensional space of distinct frequency samples of the signal's DFT. We use this term merely to distinguish it from the term *frequency-domain* which we reserve for the frequency characteristics of the continuous-time signals in an OFDM/DMT system. The tones in the OFDM/DMT system are indexed by $k = 0, \ldots, N-1$ according to the frequency of the associated discrete-time exponential.

Now recall that we transmit a series of OFDM/DMT symbols, where each symbol is a multiplex of N modulated transmit signal components, and that each data symbol is affected multiplicatively by the channel. This allows us to look at the OFDM/DMT signal by means of a grid, with a block index on one axis (indexing the OFDM/DMT symbols) and a subchannel index on the other (indexing the subchannels). For general solutions of the channel partition problem, there may be little reason to *order* the subchannels. For our particular solution with complex exponentials, or tones, we number the tones after their frequencies in increasing order. Therefore, we will refer to the block-tone grid of OFDM/DMT. Figure 16 illustrates the block-tone grid.

A price we pay for the simple receiver structure in OFDM/DMT is that we explicitly ignore the first L received samples at the receiver and do not use these for data communication. Prefix-based OFDM/DMT is thus inherently suboptimal in the sense of making maximum use of the channel. Communication systems that use all parts of the signal at the receiver may potentially perform better than our system. A measure of the potential gain of such other transmission schemes compared to OFDM/DMT is

$$\mathsf{R}_{\text{gain}} = \frac{N+L}{N}. \tag{42}$$

This gain also applies to the transmit power. Potentially, a communication system could obtain the same performance with $\frac{N+L}{N}$ better power efficiency compared to OFDM/DMT. This is a deliberate choice in

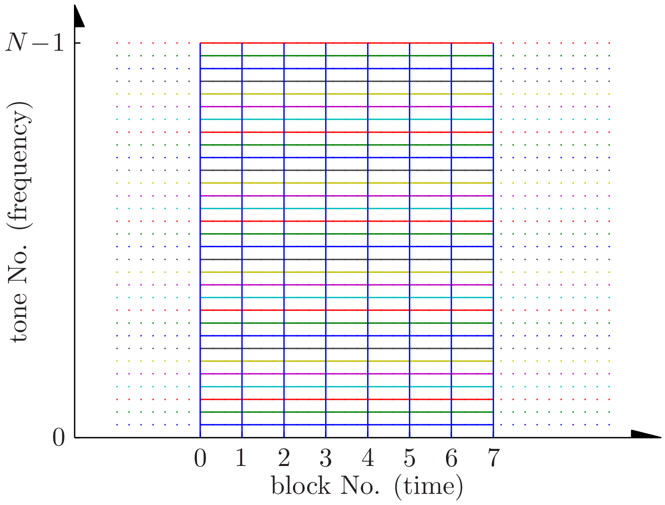


Figure 16: The block-tone grid.

an OFDM/DMT system-design in order to achieve a simple receiver structure. For large N, this potential gain in power efficiency and data rate becomes very small.

6 Understanding OFDM/DMT by matrix notation

In this section, we express the results from the previous section in matrix notation. This is useful for two reasons. First, it offers more insight and deeper understanding. Secondly, the matrix formulation becomes useful in subsequent chapters as a *tool* to compactly describe and analyse the system. We stress that this section does not contain any concepts that were not already presented earlier. Only the notation and the representation is different.

First, we describe the channel dispersion using a convolution matrix $\mathbf{H}$, where we assume $L = M$:

$$\begin{bmatrix} r(0) \\ r(1) \\ \vdots \\ r(N-1) \end{bmatrix} = \begin{bmatrix} h(L) & \cdots & h(1) & h(0) & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & & h(1) & h(0) & \ddots & & & & \vdots \\ \vdots & & h(L) & \vdots & h(1) & \ddots & & & & \vdots \\ \vdots & & & h(L) & \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & 0 & h(L) & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \ddots & \ddots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \cdots & 0 & h(L) & \cdots & h(1) & h(0) \end{bmatrix} \begin{bmatrix} s(-L) \\ \vdots \\ s(-1) \\ s(0) \\ s(1) \\ \vdots \\ s(N-1) \end{bmatrix}. \tag{43}$$

or

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{s} \tag{44}$$

This matrix can represent the real-valued convolution in the baseband system in Section 3 – then the matrix entries of $\boldsymbol{H}$ are real-valued. Alternatively, $\boldsymbol{H}$ can represent the complex channel $\underline{h}(n)$ in (40), in which case its entries are complex. Note how the first L receiver samples $r(n), n = -L, \dots, -1$ are implicitly ignored in this equation. The linear channel matrix $\boldsymbol{H}$ maps the $(N + L) \times 1$ vector $\boldsymbol{s}$ onto the $N \times 1$ vector $\boldsymbol{r}$).

The correlation operation at the receiver can be defined in vector-notation as

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle \triangleq \boldsymbol{b}^{\mathrm{H}} \boldsymbol{a} \tag{45}$$

where $\cdot^{\mathrm{H}}$ denotes the Hermitian transpose. The two properties (6) and (7) that we expressed in the previous section now become: Find transmit vectors $\boldsymbol{s}_m$ and receive vectors $\boldsymbol{r}_k$ such that

$$\boldsymbol{r}_m^{\mathrm{H}}\boldsymbol{r}_k = \begin{cases} 1 & \text{if } k = m, \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

$$\boldsymbol{H}\boldsymbol{s}_k = \boldsymbol{r}_k \tag{47}$$

or, even more compact, collecting all the column vectors $\boldsymbol{s}_k$ in the matrix $\boldsymbol{S}$ and the row vectors $\boldsymbol{r}_k^{\mathrm{H}}$ in the matrix $\boldsymbol{R}$: Find matrices $\boldsymbol{S}$ and $\boldsymbol{R}$ such that

$$\boldsymbol{R}\boldsymbol{R}^{\mathrm{H}} = \boldsymbol{I}_N \tag{48}$$

$$\boldsymbol{H}\boldsymbol{S} = \boldsymbol{R}^{\mathrm{H}} \tag{49}$$

where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix. Based on our experience gained in this chapter, we weaken the requirement (47) and allow a scaling of the receive signals

$$\boldsymbol{H}\boldsymbol{s}_k = \lambda_k \boldsymbol{r}_k, \tag{50}$$

or equivalently

$$\boldsymbol{H}\boldsymbol{S} = \boldsymbol{R}^{\mathrm{H}}\boldsymbol{\Lambda}, \tag{51}$$

where $\boldsymbol{\Lambda}$ is an arbitrary diagonal matrix.

If we transmit a data vector $\boldsymbol{x}$ using the columns of $\boldsymbol{S}$, as in

$$\boldsymbol{s} = \boldsymbol{S}\boldsymbol{x}, \tag{52}$$

and we process the received signal $\boldsymbol{r}$ using a bank of correlators $\boldsymbol{R}$, the channel partitioning (19) follows readily from

$$\boldsymbol{y} \triangleq \boldsymbol{R}\boldsymbol{r} \overset{(44)}{=} \boldsymbol{R}\boldsymbol{H}\boldsymbol{s} \overset{(52)}{=} \boldsymbol{R}\boldsymbol{H}\boldsymbol{S}\boldsymbol{x} \overset{(51)}{=} \boldsymbol{R}\boldsymbol{R}^{\mathrm{H}}\boldsymbol{\Lambda}\boldsymbol{x} \overset{(48)}{=} \boldsymbol{\Lambda}\boldsymbol{x}. \tag{53}$$

Having expressed the requirements on the signals in matrix notation, we now also express the solution (9) and (11) in matrix notation. The following choice of matrices satisfies (48) and (49).

$$\boldsymbol{R} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & \dots & w^{N-1} \\ 1 & w^2 & w^4 & \dots & w^{2(N-1)} \\ & \vdots & \vdots & & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \dots & w^{(N-1)^2} \end{bmatrix}, \qquad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{R}^{\mathrm{H}}_{\text{last } L \text{ rows}} \\ \cdots\cdots\cdots\cdots \\ \boldsymbol{R}^{\mathrm{H}} \end{bmatrix}, \tag{54}$$

where $w = e^{-j2\pi\frac{1}{N}}$. The matrices contain the frequencies in the exponentials. The matrix $\boldsymbol{R}$ is known as the normalised DFT-matrix. Observe that the first L rows in $\boldsymbol{S}$ are the same as the last L rows – the cyclic prefix we have identified earlier. It is now obvious that both the transmitter and the receiver can be equipped with a DFT in order to partition the channel. The diagonal of $\boldsymbol{\Lambda}$ becomes

$$\operatorname{diag}(\boldsymbol{\Lambda}) = \sqrt{N}\,\boldsymbol{R} \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(L-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{55}$$

This choice of $\boldsymbol{R}$ and $\boldsymbol{S}$ satisfies the requirements in (48) and (49).

Exploiting the fact that $s(n) = s(N-n), n = -L, \dots, -1$, we can rewrite (43) as

$$\boldsymbol{r} = \tilde{\boldsymbol{H}}\boldsymbol{s}, \tag{56}$$

or

$$\begin{bmatrix} r(0) \\ r(1) \\ \vdots \\ r(N-1) \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \cdots & 0 & h(L) & \cdots & h(1) \\ h(1) & h(0) & \ddots & & 0 & \ddots & \vdots \\ \vdots & h(1) & \ddots & & \ddots & \ddots & h(L) \\ h(L) & \vdots & \ddots & \ddots & & & 0 \\ 0 & h(L) & & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h(L) & & h(1) & h(0) \end{bmatrix} \begin{bmatrix} s(0) \\ s(1) \\ \vdots \\ s(N-1) \end{bmatrix}. \tag{57}$$

Compare this representation with (43) and note how the cyclic construction of the transmit signal component's prefix changes the channel matrix as perceived by the receiver. The channel matrix is a circular matrix performing circular convolution. The normalised DFT-matrix $\boldsymbol{R}$ diagonalises any circular matrix, and in particular, the matrix $\tilde{\boldsymbol{H}}$:

$$\tilde{\boldsymbol{H}} = \boldsymbol{R}^{\mathrm{H}} \boldsymbol{\Lambda} \boldsymbol{R}. \tag{58}$$

Thus, the transmit vectors and the receive vectors are of size $N \times 1$, and they are eigenvectors of the circular channel $\tilde{\boldsymbol{H}}$. The cyclic prefix makes the linear channel $\boldsymbol{H}$ appear as the circular channel $\tilde{\boldsymbol{H}}$ for which the DFT creates a channel partitioning.

7 Digression: where Shannon and Fourier meet

Increasing the *rate* at which information can be transferred from one point and reliably recovered at another, has for decades been one of the prime driving forces of communication technology. In 1948, Claude Shannon introduced his mathematical definition of information. Since then researchers and engineers have been exploring Shannon's limits for reliable communication.

One of Shannon's results is that in an additive white Gaussian noise channel, it is possible to achieve *error-free* communication as long as the data rate remains below a certain level. With W as the bandwidth of the channel, P as the signal power, and V as the noise power, Shannon showed that if the rate of information transmission is lower than

$$C = W \,\mathrm{ld}\,(1 + P/V) \qquad \text{bits/second}, \tag{59}$$

reliable communication is possible. And, if more information is transmitted per second, reliable communication is not possible. Shannon calls this limit C the *channel capacity.* As long as our transmission rate is below C, in order to achieve an arbitrarily low bit-error rate we do *not* have to slow down the transmission rate. Instead, we need to use more involved *encoding* of the message and allow longer delays.

The corresponding result for *coloured* Gaussian noise is summarised below. For a given transmit power P and a noise power spectral density $V(f)$, the channel capacity is given by

$$C_1 = \int\limits_0^W \mathrm{ld}\left(1 + \frac{P(f)}{V(f)}\right) \mathrm{d}f, \tag{60}$$

where the signal power spectral density $P(f)$ is chosen according to

$$P(f) = (\rho - V(f))^+ \tag{61}$$

with

$$(x)^+ = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

and the constant ρ chosen such that

$$\int\limits_0^W P(f) \mathrm{d}f = P. \tag{62}$$

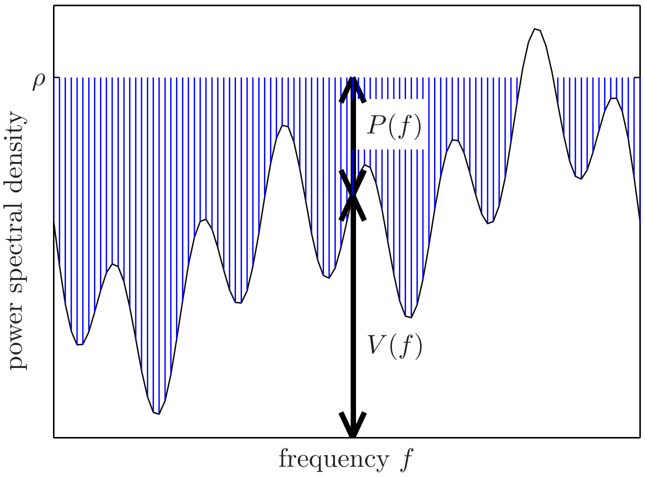


Figure 17: How to choose the transmitted power distribution for a Gaussian additive noise channel: waterfilling.

This choice of $P(f)$, illustrated in Figure 17, is referred to as *waterfilling*. We have a bowl whose bottom has a profile shaped according to $V(f)$. Now we fill this bowl with water, where the amount of water corresponds to P. The level of the water corresponds to ρ and the resulting filling profile corresponds to $P(f)$.

Shannon's engineering partitioning of the channel's frequency band into a set of narrowband subchannels closely relates to the OFDM/DMT systems we will explore in this book. In OFDM/DMT, as in Shannon's elementary bands, the subchannels behave as simple additive white Gaussian noise channels each with its own noise level. In OFDM/DMT, too, the waterfilling technique applies in order to maximise the throughput. However, unlike Shannon's partitioning, OFDM/DMT systems do not partition the channel in *bandlimited* subchannels. Instead, the subchannels overlap spectrally because the partitioning is not strictly performed in frequency. The partitioning is performed with a discrete-time Fourier transform in a tone-domain, as we have demonstrated in detail earlier in this chapter. Nonetheless, the discrete Fourier transform and its fast implementations have added to the practical relevance of Shannon's spectral waterfilling technique.

8 Digression: Other channel partitionings

The properties (6) and (7) we require in the previous section are closely related to the singular value decomposition of the channel $\boldsymbol{H}$:

$$\boldsymbol{H} = \boldsymbol{F}\,[\boldsymbol{\Lambda}|\boldsymbol{0}]\,\boldsymbol{M}^{\mathrm{H}}, \tag{63}$$

where the $N \times N$-matrix $\boldsymbol{F}$ and the $(N+L) \times (N+L)$-matrix $\boldsymbol{M}$ are both unitary. The singular-value decomposition of the channel is unique and generates a solution to the channel partitioning problem.

In general the singular vectors depend on the particular choice of the channel matrix $\boldsymbol{H}$, and the signal does not have the cyclic property which our previous solutions have. In the OFDM/DMT solution of this book we relax the orthogonality requirement on the *transmit* signals. Then there are more solutions $\boldsymbol{F}$ and $\boldsymbol{M}$ to the problem. The one solution we use in this book generating OFDM/DMT has the exponentials as vectors and are independent of $\boldsymbol{H}$.

Part II

Channels and Channel Models

1 Introduction

A communication channel[1] is the physical medium through which information transmission from a source to a sink is carried out. Channels can be classified by a variety of criteria. Depending on the topology of a communication system, we distinguish

1. Single-user communication (point to point)
2. Multi-user communication
   (a) Multiple access channel: several transmitters, single receiver (multi-point to point)
   (b) Broadcast channel: single transmitter, several receivers (point to multi-point)
   (c) Interference channel: several transmitters, several receivers (multi-point to multi-point)

Figure 1 depicts these topologies. In single-user communication, the main focus is on answering the question what is the highest data rate up to which reliable transmission from the source to the sink is possible and how to approach and ultimately reach this limit. In multi-user communication, the limit for each user depends on the data rates of the other users. This leads to pairs (for two users) or tuples (for more than two users) of achievable data rates which are depicted graphically as rate regions (or hyper-regions for more than two users).

A communication channel (user-to-user link) can have one or several input ports and/or one or several output ports. We distinguish

1. Single-input single-output (SISO) channels
2. Multi-input multi-output (MIMO) channels and degenerated cases (single-input multi-output (SIMO) channels, multi-input single-output (MISO) channels)

An example of a MIMO channel is a radio link with several transmit antennae and several receive antennae (each antenna represents a port). Another example is a multi-pair copper cable in the access network, where the two ends of every wire-pair represent a transmit and a receive port, respectively.

A signal is transmitted through a channel in the form of an electric current, an electromagnetic wave or beam, a ray of light, etc.. These physical quantities are subject to various physical phenomena along the way such as

- reflection
- absorption

Chapter written by T. Magesacher and M. Lončar.

[1] Storage channels, on the other hand, transfer information through time instead space. Examples include magnetic tape or disc, optic disc, etc. Although, apart from causality all concepts can be ported to storage channels, they are not considered hereinafter.

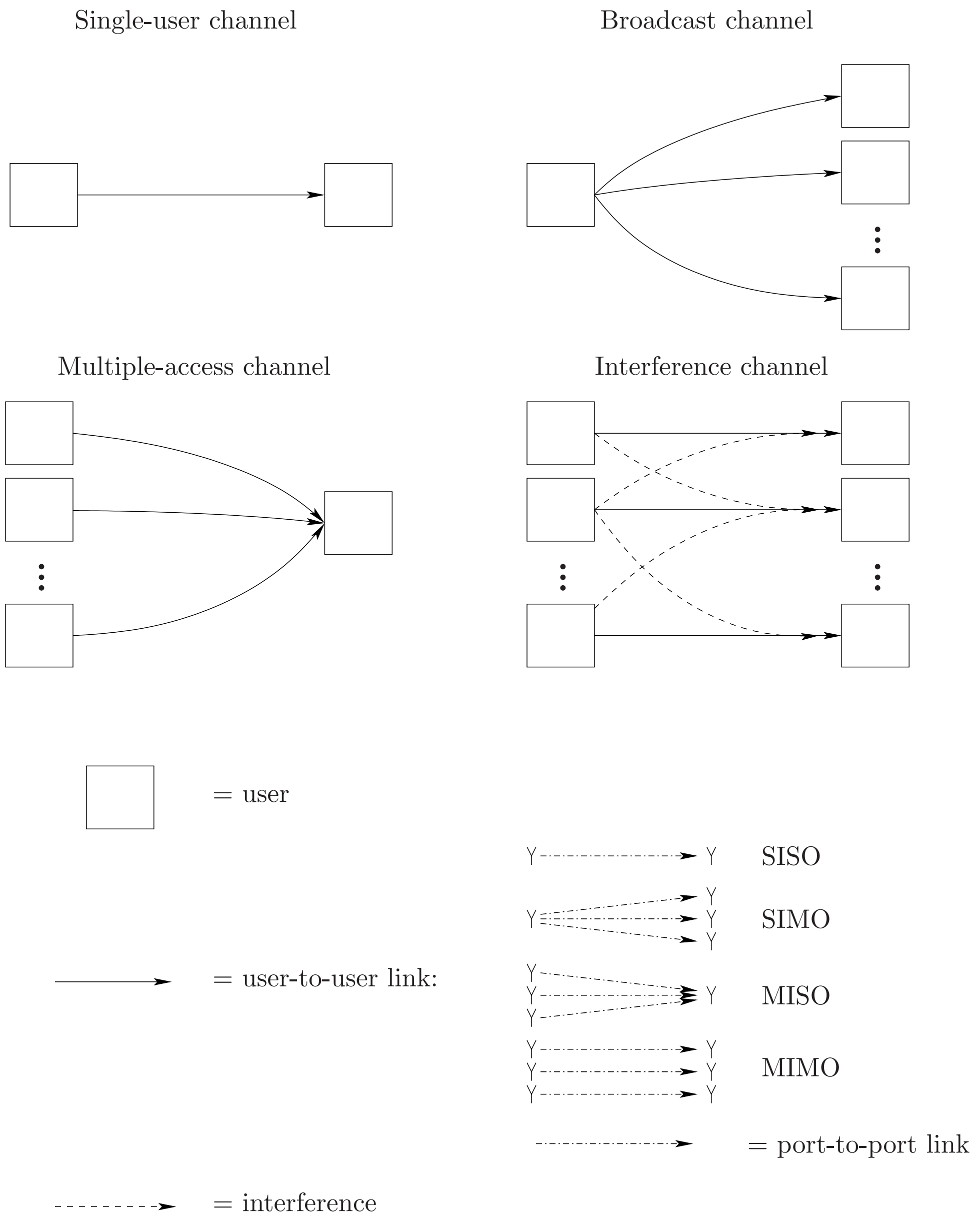


Figure 1: Channel topologies: each square represents a user. Every link between two individual users can be of SISO, SIMO, MISO or MIMO type.

- attenuation (scaling in amplitude)
- dispersion (spreading in time)
- refraction (bending of a wave or a ray due to variation of the media's diffraction index)
- diffraction (scattered re-radiation, caused by an edge or an object whose size is in the order of the wave length)

The net effect of the channel on the transmit signal can be described as modification of the signal (distortion) and addition of noise.

Depending on its properties, a channel can be

- linear or non-linear

- time-invariant or time-variant (fading)
- frequency-flat or frequency-selective (time-dispersive)

The additive noise can be

- Gaussian or non-Gaussian
- correlated in time/frequency, space (in MIMO systems), over users (in multi-user systems)

Communication channels are usually classified according to the type of physical medium as

1. guided channels
   - wire (e.g.: copper twisted-pairs in the access network, VDSL, ADSL)
   - cable (e.g.: coax cables used in cable networks, DVB-C)
   - fiber (e.g.: optical fibers in backbone networks)
   - microwave guide (e.g.: feeder "pipes" for high-power RF transmitters, radar)
2. unguided channels
   - electromagnetic wireless channel
     - static terminals (geostationary satellite link, deep space communication, terrestrial radio links (DVB-T))
     - mobile terminals (GSM, UMTS, DAB, DVB-H)
   - underwater acoustic channel

Guided channels can usually be assumed time-invariant or very slowly changing with time due to environmental influences like temperature, humidity, etc. The properties of unguided channels may vary continuously and quickly.

Sometimes (e.g. in information theory) several components of the transmit-receive chain (modulation, digital-to-analogue conversion, up-conversion, channel, down-conversion, analogue-to-digital conversion and demodulation/detection) are modelled by a single entity referred to as equivalent *digital channel.* "Digital" refers to the quantisation in amplitude—the set of symbols is finite. One of the most important models of this type is the discrete memoryless channel (DMC). The DMC can have two or more output symbols depending on whether the decision device produces hard or soft decisions. The DMC is completely defined by transition probabilities $p(y_k|x_l)$, i.e., the conditional probabilities that y_k is detected given that x_l was transmitted. These channel models are useful for finding bounds and investigating basic properties of communication schemes.

2 Guided channels

As the name indicates, the physical quantity representing the transmit signal is spatially bounded and guided by the medium. Thus the transmission remains, to a large extent, untouched by events happening outside the medium, which leads to fairly time-invariant propagation conditions. The most important guided communication channels are discussed in the sequel.

2.1 Wireline (copper cable) channel

The "oldest channel" employs a conductor for information transmission. Roughly hundred years ago, Graham Bell patented the idea of twisting a pair of wires in order to increase the immunity against time-varying electromagnetic fields in the vicinity of the pair. Physically, communication is carried out by transmission of voltages and currents. Since the length of the wires can be in the same order of magnitude as the wavelengths (frequencies 300 Hz to 30 MHz, which correspond to wavelengths of roughly 1000km to 10m), voltage and current distribution over length has to be considered. It is then more convenient to think of waves. Theory of wave propagation through conductors is well understood. The frequency band 300 Hz to 3300 Hz is often referred to as *telephone channel* for voice communications, also called POTS

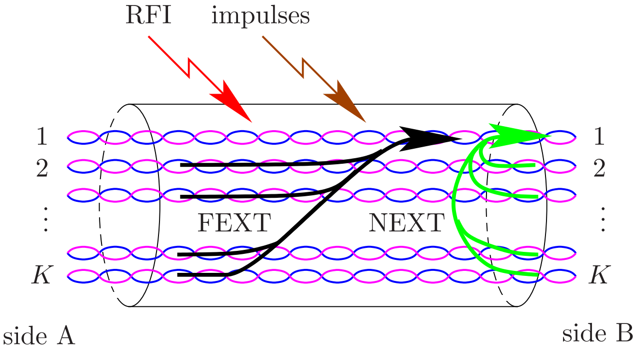


Figure 2: Wireline channel: different types of interference in a wireline communication system (from the perspective of the receiver connected to line No. 1 on side B): near-end crosstalk (NEXT), far-end crosstalk (FEXT), radio frequency interference (RFI), and impulses.

(plain old telephone service). Telephone wires can have lengths of up to 30 km and more. Lately, the exploited frequency range of shorter loops (few hundred meters up to 5 km) has been extended for data transmission. The *digital subscriber line (DSL) channel* occupies the band form 30 kHz up to 30 MHz. The supported bandwidth depends strongly on the length.

Figure 2 illustrates the impairments present in a wireline channel. Electromagnetic coupling among wires in a cable causes crosstalk (in analogue telephone systems it was occasionally possible to listen to other people's conversations). The performance of today's high-speed modem technology is most often limited by crosstalk. Depending on the location of the receiver with respect to the crosstalk-causing transmitter we distinguish between near-end crosstalk (NEXT) and far-end crosstalk (FEXT). While crosstalk results in a quasi-stationary noise source, impulse noise, caused for e.g. by switching events of electro-mechanical devices, is of short duration and occurs at random instants in time. Radio frequency interference (RFI) ingress is another phenomenon impairing wireline transmission: the wires act like antennae and pick up radio signals emitted by wireless sources, which can result in received interference strong enough to overload the receiver front-end.

Modelling the wireline channel is based on multi-port theory. A single wire pair, which constitutes the simplest possible case, is modelled as a two-port. Modelling two wire pairs and their mutual crosstalk behaviour requires a four-port model. Modelling multi-pair cables is a rather complicated task. The inherent problem is that the wire pairs can not be viewed as independent entities. For example, changing the termination impedance on one wire affects the crosstalk properties of the whole cable. Modelling the "inner life" of a multi-port is based on finding an equivalent electrical circuit, whose parameters are often determined by measurements of certain cable types.

Given a certain termination, the effect of a wire pair on a transmitted signal can be described by a linear time-invariant (LTI) system (cf. Section 5.1). The wire simply acts like a filter with a certain transfer function, or equivalently, with a corresponding impulse response. A frequently used model for the transfer function is

$$H_{\text{loop}}(f,d) = \mathrm{e}^{-\frac{d}{d_0}(k_1\sqrt{f}+k_2 f)}\mathrm{e}^{-j\frac{d}{d_0}k_3 f}, \quad k_1 = 3.8\cdot 10^{-3},\ k_2 = -0.541\cdot 10^{-8},\ k_3 = 4.883\cdot 10^{-5}, \qquad (1)$$

where f is the frequency in Hz, d is the length of the loop in m, $d_0 = 1609.3\,\text{m}$ (1 mile) is a reference length and k_1, k_2, k_3 are constants which depend on the wire gauge (the values given above model a 0.5 mm loop). The main effect of the wireline channel on a transmit signal is frequency-dependent attenuation, or equivalently, dispersion in time, which follows directly from the properties of the Fourier transform. Consequently, also the attenuation per length unit is strongly frequency dependent. A rough reference value at 1 MHz is several dB/km. The effect of crosstalk can also be modelled as an LTI system. The European standardisation institute for communications (ETSI) defines the FEXT coupling function

$$H_{\text{FEXT}}(f,d) = k_{\text{f}}\frac{f}{f_0}\sqrt{\frac{d}{d_0}}|H_{\text{loop}}(f,d)|, \qquad k_{\text{f}} = 10^{-45/20}, \quad f_0 = 1\,\text{MHz}, \qquad d_0 = 1\,\text{km} \qquad (2)$$

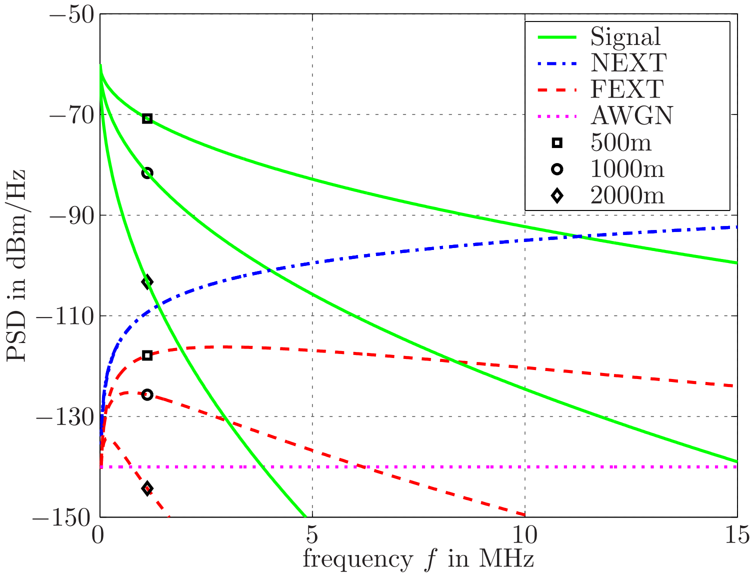


Figure 3: Wireline channel: exemplary power spectral densities of signal, interference and noise at the receive side (transmit PSD: −60 dBm/Hz from 0 to 15 MHz).

and the NEXT coupling function

$$H_{\mathrm{NEXT}}(f,d) = k_{\mathrm{n}} \left(\frac{f}{f_0}\right)^{\frac{3}{4}} \sqrt{1 - |H_{\mathrm{loop}}(f,d)|^4}, \qquad k_{\mathrm{n}} = 10^{-50/20}, \qquad f_0 = 1\,\mathrm{MHz}. \tag{3}$$

Based on these system functions and the transmit power spectral densities (PSDs) of the systems involved, wireline deployment and transceiver equipment can be designed. Since the wireline channel is quasi-stationary, many aspects can be investigated well enough by looking only at PSDs. Figure 3 shows such PSDs at the receive side. While the loop transfer function and the FEXT coupling are strongly length-dependent, the NEXT coupling is virtually independent of the loop length.

Essentially, the copper cable has the worst properties of all communication channels that have an electric conductor as basic medium. However, wires have been installed during the last 100 years, they are in place and thus particularly valuable when it comes to bringing Internet access to people: the copper wires bridge the “last mile”, i.e., the last couple of 100 meters between central offices or street cabinets and peoples’ homes.

2.2 (Coax) cable channel

Coax cable networks are often in place in highly populated residential areas. Originally, the main application was distribution of television. Nowadays, cable is one of the main media for Internet access. The exploited frequency band is in the range from 1 MHz to 1 GHz (wavelength range 300 m to 30 cm). The available frequency band is much larger compared to the wireline channel. Consequently, also the supported data rates are higher.

The main impairment is frequency-selective attenuation. A rough reference value for the length-dependent attenuation at 10 MHz is a few tens of dB/km. Since the useful frequency band in coax cables is much wider compared to the wireline channel, it is often divided among users and/or transmit sources (a TV channel has a bandwidth of roughly 7 MHz). Thus the frequency-dependence per subchannel or per user is by far not as severe as in the wireline channel.

2.3 Waveguide channel

Information is carried by electromagnetic waves travelling inside hollow metallic conductors of various profiles. The main application is transmission of high-power signals from amplifiers to antennae (feeder pipes), e.g. radar or broadcast stations. The power amplifier should be in a dry and air-conditioned location, while the antenna should be located, e.g., on top of a pole. The frequency range that can be exploited when using a wave guide is 3 GHz to 30 GHz (wavelength range 10 cm to 10 mm).

2.4 Optic fiber channel

Information is carried by light rays in the infrared band using wavelengths from 1.3 μm to 1.6 μm. In contrast to copper wires and cable networks, new fibers are installed wherever possible. Research on materials and fibers aims at low attenuation (0.2 dB/km).

Time-dispersion occurs in fibers due to Rayleigh scattering and a wavelength-dependent dielectric. Time dispersion grows with distance. The useful bandwidth can be as large as several hundred thousands of GHz.

Fiber has by far the most desirable physical properties for data transmission and is the preferred medium for the backbone network. The penetration of fiber in the access network will grow. However, although already a decade ago people were predicting that fiber will replace copper and cable within years, quite some time may pass until every household is supplied with fiber.

3 Unguided channels

The (electromagnetic or acoustic) waves carrying the information are spatially unbounded as they travel from the source to the sink. Consequently, any event near or inside the medium may influence the propagation conditions, which results in time-variant channel conditions. The underwater acoustic channel is not treated in this tutorial. Hereinafter, we focus on the wireless channel.

3.1 Wireless channel with fixed terminals

The main effect in obstacle-free propagation (e.g. (geostationary) satellite - earth link, deep-space communication) is signal attenuation, known as *path loss*. Calculation of the total path loss, which is usually assumed to be static (i.e., time-invariant), is referred to as link budget analysis of a communication link. Besides the losses caused by implementation (feed loss, circuit loss, branching loss, pointing loss) and the atmospheric loss, the total attenuation is usually governed by the free-space path loss L_{p}, which determines the receive signal power

$$P_{\mathrm{r}} = P_{\mathrm{t}} G_{\mathrm{t}} G_{\mathrm{r}} L_{\mathrm{p}}, \tag{4}$$

where P_{t} is the transmit signal power and G_{t} and G_{r} are the antenna gains of transmit and receive antenna, respectively. The gain G of a given antenna is the ratio of the signal power received by that antenna and the signal power received by an isotropic antenna. The antenna gain depends on the design and directional properties of an antenna. Due to reciprocity, an antenna exhibits the same gain for transmission and reception. For a dish antenna, the gain is roughly given by

$$G \approx 4\pi A/\lambda^2, \tag{5}$$

where A is the effective area (geometric area multiplied by a factor smaller than 1 to account for imperfections). The wavelength λ is given by $\lambda = c/f$, where c is the propagation speed, which is usually close to the speed of light ($3 \cdot 10^8$ m/s), depending on the environmental conditions. The free-space path loss is given by

$$L_{\mathrm{p}} = \left(\frac{\lambda}{4\pi d}\right)^2, \tag{6}$$

where d is the distance between the two terminals. From (6) it follows that the received signal power in free space decreases with the square of the distance d. For a fixed distance d and two dish antennae, it follows from (5) and (6) that the received signal power grows as the square of frequency. Consequently, for long distances we have to use high frequencies.

In the presence of static obstacles, which is usually the case for terrestrial microwave links, ray tracing is frequently used. The paths of wavefronts are depicted by rays, which allows the modelling of reflection

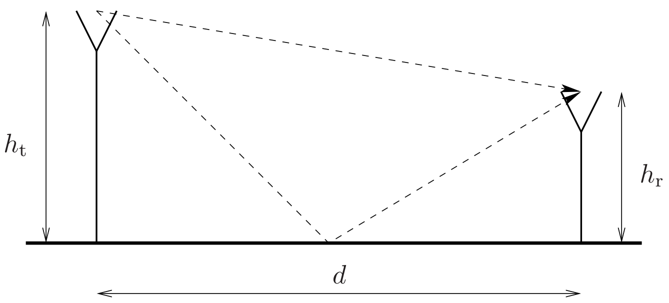


Figure 4: Two-path case with total reflection.

and scattering. A simple classical case of two paths with total reflection is depicted in Figure 4. The path loss for this case is given by

$$L_\mathrm{p} = \left(\frac{\lambda}{4\pi d}\right)^2 \left(\frac{4\pi h_t h_r}{\lambda d}\right)^2 = \frac{h_t^2 h_r^2}{d^4}, \tag{7}$$

where h_t and h_r are the heights of transmit and receive antenna, respectively. Note that in this case the received power decreases as the forth power of distance. A large portion of simulation packages for radio propagation is based on the ray tracing technique. In order to obtain representative results, detailed site information is required.

In reality, propagation scenarios can be much more complicated than the two-ray case considered above. Often these scenarios are modelled by a total path loss

$$L_\mathrm{p} = \underbrace{P_\mathrm{r}(d_0)/P_t}_{K} \left(\frac{d_0}{d}\right)^\alpha, \tag{8}$$

where the constant K denotes the ratio of receive and transmit power for a given reference distance d_0 and α is the path loss exponent, which depends on wavelength and environment and is typically in the range $2-8$ for 1 GHz.

Propagation mechanisms in terrestrial links (standard microwave link: 50 km, 4 GHz) and atmospheric wave propagation effects (medium wave band (300 kHz – 3 MHz) and short wave band (3 MHz – 30 MHz)) and their explanation based on physical grounds is a field of its own, which we will not discuss hereinafter. We only list a few most commonly encountered consequences:

- Reflection may lead to multi-path propagation and destructive interference—the free-space loss is replaced by a propagation loss which is independent of wavelengths but depends strongly on geometric parameters like distance, antennae heights, structure of the reflecting surface, etc.
- Refraction may result in a bent beam, which has to be taken into account when pointing at the sink antenna with a highly directional transmit antenna.
- Diffraction from near-LOS (line of sight) objects may contribute considerably to the total path loss. The local topology, like high risers, hills, etc. has to be taken into account during link planning.
- Atmospheric attenuation and absorption: water vapour and molecular oxygen cause absorption peaks at 24 GHz and 60 GHz, respectively.

3.2 Wireless channel with mobile terminal(s)

Most often only one terminal is moving, in the following referred to as the mobile terminal (MT), while the other one, referred to as fixed terminal (FT), does not move (usually a base station or an access point). In the following, we will observe transmission from the FT to the MT, the so called downlink direction. Due to reciprocity, it does not matter whether we transmit from the FT to the MT or vice versa, the propagation effects and the invoked arguments remain the same.

The two dominant effects of the wireless channel on the transmitted signal are dispersion in time and dispersion in frequency.

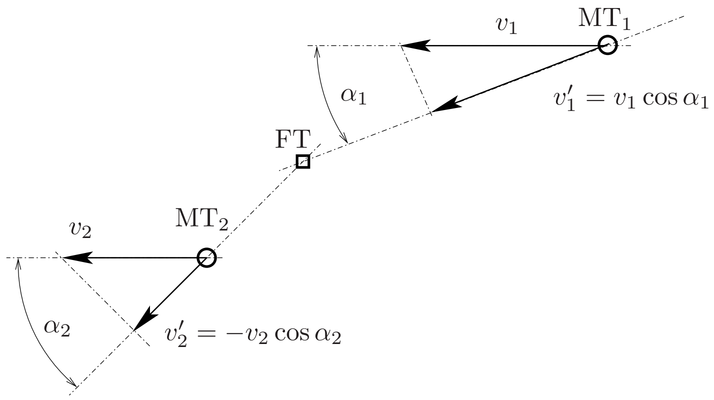


Figure 5: Two mobile terminals MT_1 and MT_2, their absolute velocities v_1 and v_2 and the velocities of approach v_1' and v_2' of the LOS components with respect to the FT.

- Dispersion in time

  In general, a beam transmitted by the FT is reflected and scattered due to obstacles along the way, and what arrives at the MT is rarely a single beam but many beams from different directions. This effect is referred to as *multi-path propagation.* If these multi-path components arrive with different delays, we receive a series of delayed (and scaled) versions of the transmit signal. This effect is known as *dispersion in time.* Often, there is neither a direct beam from the FT to the MT, nor a strong stationary reflection, both of which are referred to as *line of sight (LOS)* components and cause dominant stationary elements in the received signal. Dispersion in time leads to *frequency selectivity* of the channel, which is a direct consequence of the Fourier transform properties.

- Dispersion in frequency

  An effect that occurs whenever the transmitter and/or the receiver of a signal are moving, is the Doppler effect: When the FT transmits a signal with frequency $f_0 = c/\lambda$, the MT receives this signal at frequency $f_0 + \nu = f_0 + v'/\lambda$, where v' is the relative velocity of the MT with respect to FT. Note that v' is a signed quantity: $v' > 0$ indicates that MT is approaching FT, $v' < 0$ indicates that MT is moving away from FT. Figure 5 illustrates the relative velocity and its relation to the absolute velocity for two MTs. The frequency shift ν is referred to as Doppler shift.

  Assume the FT transmits a signal with frequency f_0. If the MT and/or the surrounding scatterers and reflectors are moving, each multipath component arrives at the MT with a different relative velocity. Each component has a different Doppler shift ν_i, consequently, the spectrum of the received signal contains many components with different frequencies $f_0 + \nu_i$. This effect is referred to as *dispersion in frequency.*

  Motion is not the only cause of dispersion in frequency. In general, dispersion in frequency is caused by time-varying propagation conditions. For example, assume that there is a single receive component whose envelope $a(t)$ changes over time due to variation of the channel. This can simply be viewed as amplitude modulation of the transmit signal by the time-varying channel attenuation $a(t)$. Consequently, the spectrum of the receive signal contains not only the transmit frequency f but is widened by the spectrum of the “modulating signal” $a(t)$.

To summarise, the wave propagation between the FT and the MT has time-varying multipath nature. The resulting fluctuations of the receive signal amplitude are referred to as *fading*, which can be modelled statistically.

Characterisation of amplitude fluctuations (fading models)

- Small-scale (microscopic) fading:

  The receive signal amplitude is viewed as a random variable and is described by a probability density function.

- *Rayleigh fading*

  The FT emits a wave of frequency f_0 corresponding to the signal

  $$\mathrm{Re}\left\{e^{j2\pi f_0 t}\right\} = \cos(2\pi f_0 t)$$

  Due to scattering and reflection, many waves reach the MT from different directions and there is *no LOS*. The dashed arrows in Figure 6 illustrate two of these waves and their angles of arrival ψ_1 and ψ_2, respectively. The relative velocity of the MT with respect to an arriving wave depends on the angle of arrival. The relative velocity of waves arriving from angles ψ close to 0 is essentially equal to the velocity v of the MT, while the relative velocity of waves arriving from ψ close to π is $-v$. The relative velocity of waves arriving from ψ close to $\pi/2$ or $3\pi/2$ is zero. Consequently, the Doppler shift is different for each wave, which leads to dispersion in frequency of the transmitted signal. At any given point in time, the received signal is the sum of K wave components:

  $$\begin{aligned} r(t) &= \sum_{k=1}^{K} A_k \cos(2\pi(f_0 + \frac{v\cos\psi_k}{\lambda})t + \theta_k) = \\ &= \underbrace{\sum_{k=1}^{K} A_k \cos(2\pi(\frac{v\cos\psi_k}{\lambda})t + \theta_k)}_{A_I(t)} \cos(2\pi f_0 t) - \underbrace{\sum_{k=1}^{K} A_k \sin(2\pi(\frac{v\cos\psi_k}{\lambda})t + \theta_k)}_{A_Q(t)} \sin(2\pi f_0 t), \end{aligned}$$

  where phases θ_k account for the different distances the waves travel and the amplitudes A_k model the different attenuation of the waves. We recognise in the above expression the inphase and the quadrature components of the received signal. Their amplitudes are $A_I(t)$ and $A_Q(t)$, respectively. According to the central limit theorem (CLT), no matter what the actual probability distributions of the random phases θ_k and amplitudes A_k are, the amplitudes $A_I(t)$ and $A_Q(t)$ follow Gaussian distribution (for K sufficiently large) with zero mean and variance $\sigma^2 = \frac{1}{2}\mathsf{E}\left\{|A_k|^2\right\}$. Moreover, due to the orthogonality of cos and sin, $A_I(t)$ and $A_Q(t)$ are uncorrelated. The envelope of the received signal is

  $$U(t) = \sqrt{A_I^2(t) + A_Q^2(t)} \tag{9}$$

  and the phase is

  $$\phi(t) = \arctan\frac{A_Q(t)}{A_I(t)}. \tag{10}$$

  Using probability theory, it can be easily shown that the envelope follows Rayleigh distribution, given by

  $$p_U(u) = \frac{u}{\sigma^2}\mathrm{e}^{-\frac{u^2}{2\sigma^2}}, \qquad u \geq 0. \tag{11}$$

  The mean value of U is $\mathsf{E}\{U\} = \sigma\sqrt{2}\Gamma(3/2) = \sigma\sqrt{\pi/2}$, the variance of U is $\mathrm{Var}\{U\} = (2-\pi/2)\sigma^2$. The phase has uniform distribution, i.e., $\phi \sim \mathcal{U}[0, 2\pi)$.

  The validity of the Rayleigh fading model has been verified by measurements. It turns out that only several incoming wave components suffice to observe a Rayleigh-distributed signal envelope (i.e., ”sufficiently large K” in the CLT is about 5-6). This model is obtained assuming no dominant (LOS) component in the receive signal. In case this component is present the distribution of the received signal amplitude changes. Figure 7 depicts the Rayleigh distribution for $\sigma = 1$.

- *Rician fading*

  In case there is a LOS component, the receive signal is given by

  $$r(t) = \underbrace{A_0\cos(2\pi f_0 t + \theta_0)}_{\text{LOS component}} + \sum_{k=1}^{K} A_k \cos(2\pi(f_0 + \frac{v\cos\psi_k}{\lambda})t + \theta_k), \tag{12}$$

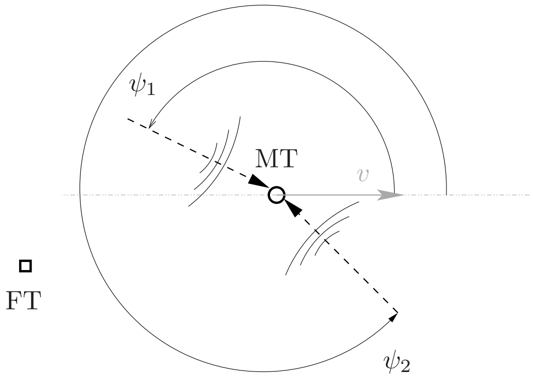


Figure 6: Grounds of Rayleigh fading: many waves arrive from arbitrary directions (two waves are shown in the figure).

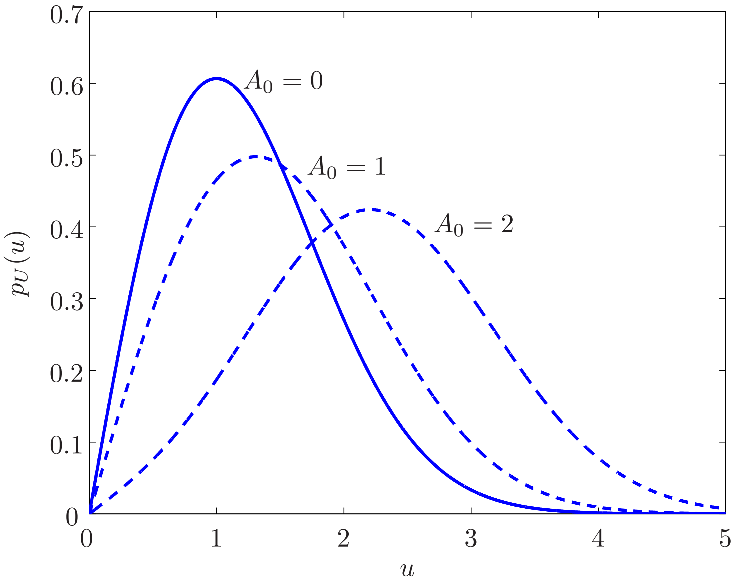


Figure 7: Rayleigh distribution (solid line) for $\sigma^2 = 1$ and Rice distributions (dashed lines) for $\sigma^2 = 1$, $A_0 = 0$, $A_0 = 1$, $A_0 = 2$. Rice distribution for $A_0 = 0$ reduces to Rayleigh distribution.

where A_0 and θ_0 are constant. Then the amplitude $A_I(t)$ of the inphase component and the amplitude $A_Q(t)$ of the quadrature component are Gaussian distributed with variance σ^2 and mean values $m_I = A_0 \cos(\theta_0)$ and $m_Q = A_0 \sin(\theta_0)$. Then the envelope $U(t)$ of the receive signal follows *Rice distribution* given by

$$p_U(u) = \frac{u}{\sigma^2} \mathrm{e}^{-\frac{u^2+s^2}{2\sigma^2}} I_0(\frac{us}{\sigma^2}), \qquad u \geq 0; \qquad s = \sqrt{m_I^2 + m_Q^2} = A_0; \tag{13}$$

where $I_n(x)$ is the nth order modified Bessel function of the first kind of x (in MATLAB: `besseli(n,x)`). Note that for no LOS, i.e., for $A_0 = 0$, the Rice distribution reduces to Rayleigh distribution ($I_0(0) = 1$). Figure 7 shows the Rice distribution for $\sigma = 1$ and different values of A_0.

– *Nakagami fading*

The *Nakagami distribution* can be viewed as a generalisation of the Rayleigh model. It has one additional parameter, which allows more freedom when fitting the fading model to measurement results. Figure 8 shows the Nakagami distribution for different values of the parameter m. For $m = 1$ the Nakagami distribution reduces to the Rayleigh distribution.

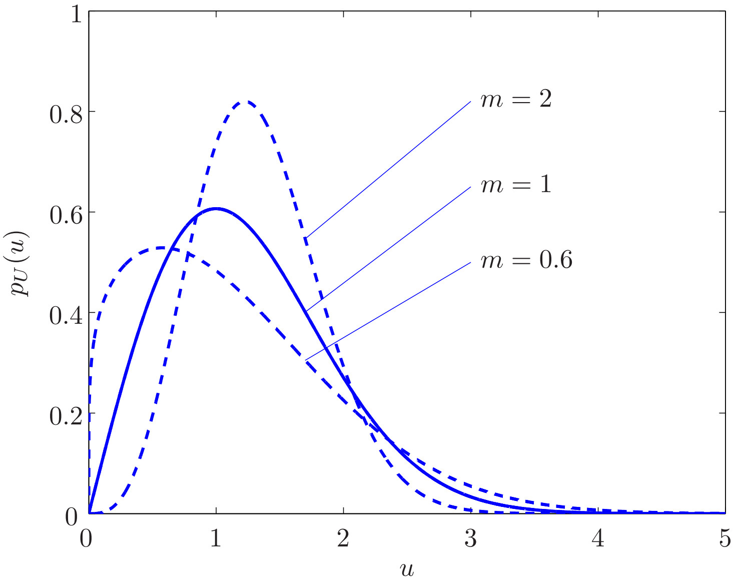


Figure 8: Rayleigh distribution (solid line) for $\sigma^2 = 1$ and Nakagami distributions (dashed lines) for three values of the parameter m. Nakagami distribution for $m = 1$ reduces to Rayleigh distribution. For $m > 1$ the tail decays faster than Rayleigh, for $m < 1$ the tail decays slower than Rayleigh.

- Large-scale (macroscopic) fading:

  Large-scale fading is the consequence of the effect that the environment is changing due to motion. A good way to model large-scale fading is to vary the mean value of the small-scale fading models discussed above with time. More precisely, the mean value of the Rayleigh or the Rician model is viewed as a random variable γ. The *log-normal distribution* of γ was found to yield a good match between the model and actual measurements. This means that the amplitude in decibels, $\gamma_{\text{dB}} = 20\log_{10}\gamma$, has Gaussian (normal) distribution

$$p_{\gamma_{\text{dB}}}(\gamma_{\text{dB}}) = \frac{1}{\sqrt{2\pi}\sigma_{\text{dB}}} \mathrm{e}^{-\frac{(\gamma_{\text{dB}} - m_{\text{dB}})^2}{2(\sigma_{\text{dB}})^2}}, \tag{14}$$

  where σ_{dB}, the standard deviation of γ_{dB}, is typically in the range of 6-12 dB. Then the distribution of $\gamma = 10^{\gamma_{\text{dB}}/20}$ is given by

$$p_\gamma(\gamma) = \frac{20}{\gamma\sqrt{2\pi}\sigma_{\text{dB}} \ln 10} \mathrm{e}^{-\frac{(20\log_{10}\gamma - m_{\text{dB}})^2}{2(\sigma_{\text{dB}})^2}}. \tag{15}$$

  Finally, small-scale, large-scale fading and path loss can be combined into a total probability density function. A realisation of a fading process including path loss, large-scale fading and small-scale fading is illustrated in Figure 9.

Characterisation of dispersion in time and frequency

If the mobile channel would be time-invariant, it would be fully described by its impulse response $h(\tau)$, which can be plotted versus the delay τ and does not change with time t (almost always, however, the impulse response of a continuous-time LTI system is denoted as $h(t)$, where t plays the role of τ). In general, the mobile channel is time-variant and space-variant and can be described by the impulse response $h(\tau, t)$, which is plotted versus delay τ for each time t. In a SISO channel, the transmit signal $s(t)$ is received as

$$r(t) = h(\tau, t) * s(t),$$

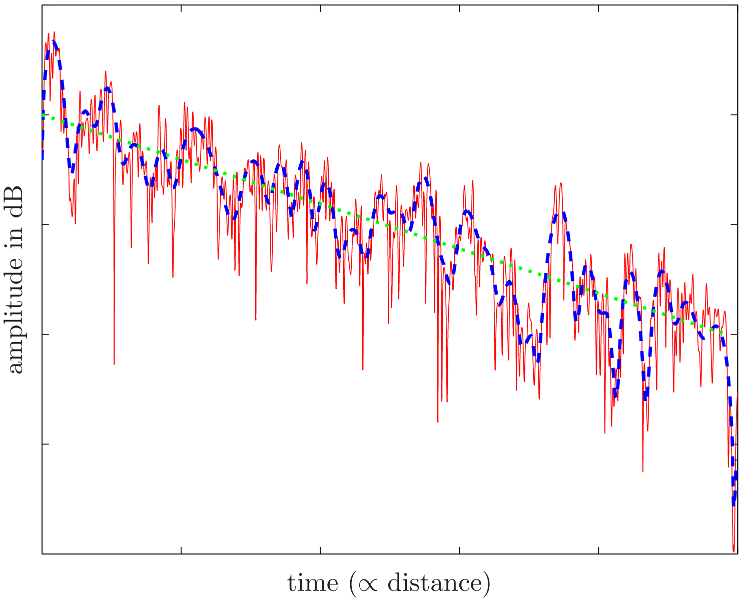


Figure 9: Illustration of amplitude fluctuations over time (fading): large-scale fading (dashed line), small-scale fading (solid line) and path loss (dotted line).

where $*$ denotes linear convolution. In a MIMO mobile channel, the signal is transmitted over N_s transmit antennae and received with N_r receive antennae,

$$\begin{bmatrix} r_1(t) \\ r_2(t) \\ \vdots \\ r_{N_r}(t) \end{bmatrix} = \begin{bmatrix} h_{11}(\tau,t) & h_{12}(\tau,t) & \cdots & h_{1N_s}(\tau,t) \\ h_{21}(\tau,t) & & & h_{2N_s}(\tau,t) \\ \vdots & & & \vdots \\ h_{N_r1}(\tau,t) & h_{N_r2}(\tau,t) & \cdots & h_{N_rN_s}(\tau,t) \end{bmatrix} * \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_{N_s}(t) \end{bmatrix}$$

where $h_{mn}(\tau,t)$ is the response observed by the mth receive antenna to the impulse launched at the nth transmit antenna.

The impulse response of a linear time-variant (LTV) system $h(\tau,t)$ is the response observed at time instant t to an impulse launched τ seconds ago (the discrete-time version of the LTV system is treated in Section 5.2). Given a realisation of the *time-variant impulse response* $h(\tau,t)$, obtained e.g. by measurements (referred to as channel sounding), a deterministic analysis can be carried out. The plot in the upper left corner of Figure 10 depicts an exemplary time-variant impulse response obtained by simulation. The time-dispersion of the impulse response is clearly visible as smearing along the τ-axis. The fluctuation of the impulse response over time t for each τ has Rayleigh characteristic[2]. The Fourier transform of $h(\tau,t)$ with respect to τ yields the *time-variant frequency response* $H(f,t)$, depicted in the upper right plot of Figure 10. The fluctuation of the transfer function's magnitude is different for each frequency f—the fading is clearly frequency-selective. The Fourier transform of $h(\tau,t)$ with respect to t yields the so called *delay Doppler spreading function* $s(\tau,\nu)$, shown in the lower left plot of Figure 10. For each delay τ, $s(\tau,\nu)$ depicts the dispersion in frequency along the ν-axis. In this example, the smearing over frequency is significant for $|\nu| \leq 50\,\text{Hz}$. The Fourier transform of $H(f,t)$ with respect to t yields the *output Doppler spreading function* $B(f,\nu)$, depicted in the lower right plot of Figure 10. For each frequency f, we observe the spectral dispersion along the ν-axis.

For a stochastic analysis, we need to agree on the meaning of an ensemble first. In order to evaluate the ensemble mean, we would need access to a very large number of wireless channels with identical properties. Since this is obviously infeasible, it is common practice in mobile radio channel characterisation to assume ergodicity (although the channel is time-variant *per se*) and average over time t. The autocorrelation function of the impulse response, $R_{hh}(\tau_1,\tau_2,t_1,t_2) = \mathsf{E}\left\{h^*(\tau_1,t_1)h(\tau_2,t_2)\right\}$, is in general

[2] In the simulation, it is assumed that fading of the multi-path coefficients is uncorrelated.

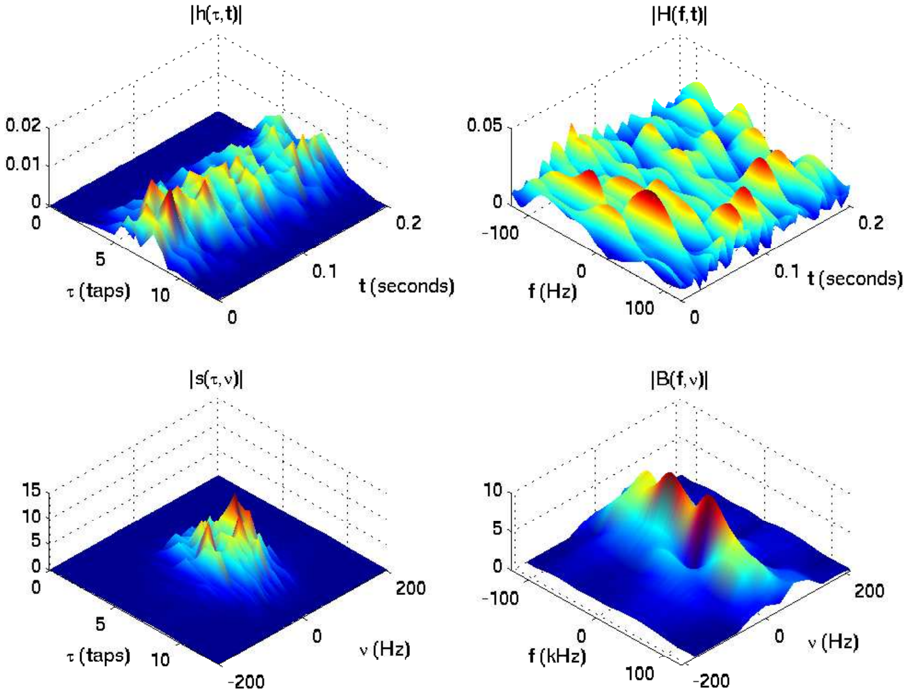


Figure 10: Exemplary simulation results of system functions of a wireless channel (14 taps uniformly spaced with 4 μs delay, each has Rayleigh fading characteristic, Doppler spectrum: 50 Hz lowpass, the fading of the 14 taps is uncorrelated.)

a function of two times instants t_1 and t_2 and two delays τ_1 and τ_2. Assuming *wide-sense stationarity (WSS)*, the autocorrelation function depends only on the difference $\Delta t = t_2 - t_1$, thus it is sufficient to observe $R_{hh}(\tau_1, \tau_2, \Delta t)$. Assuming *uncorrelated scattering (US)*, which simply states that the scatterers in different delay ellipsoids behave independently, it is sufficient to observe $R_{hh}(\tau, \Delta t)$ which is related to $R_{hh}(\tau_1, \tau_2, \Delta t)$ according to

$$R_{hh}(\tau_1, \tau_2, \Delta t) = \begin{cases} R_{hh}(\tau, \Delta t) & \tau_1 = \tau_2 = \tau \\ 0 & \text{otherwise} \end{cases},$$

frequently also written as $R_{hh}(\tau, \tau_1, \Delta t) = R_{hh}(\tau, \Delta t)\Delta(\tau - \tau_1)$. $R_{hh}(\tau, \Delta t)$ is referred to as *delay cross-power spectral density.* The Fourier transform of $R_{hh}(\tau, \Delta t)$ with respect to τ yields the *time-frequency correlation function* $R_{HH}(\Delta f, \Delta t) = \mathsf{E}\left\{H^*(f, t)H(f + \Delta f, t + \Delta t)\right\}$. The Fourier transform of $R_{hh}(\tau, \Delta t)$ with respect to Δt yields the *scattering function* $R_s(\tau, \nu)$. Finally, the Fourier transform of $R_{HH}(\Delta f, \Delta t)$ with respect to Δt yields the *Doppler cross-power spectral density* $R_B(\Delta f, \nu)$. Note that $R_s(\tau, \nu)$ and $R_B(\Delta f, \nu)$ in the stochastic analysis correspond to $s(\tau, \nu)$ and $B(f, \nu)$ in the deterministic analysis, respectively. Figure 12 summarises the system functions and the correlation functions and their relations.

Two functions that can be derived from the correlation functions and depend only on one variable are commonly used in practice.

1. The delay cross-power spectral density $R_{hh}(\tau, \Delta t)$ observed for $\Delta t = 0$ yields the *delay power density spectrum*

$$P(\tau) = R_{hh}(\tau, \Delta t)\big|_{\Delta t = 0}\,. \tag{16}$$

   The delay power density spectrum, also called the *power delay profile*, $P(\tau)$ specifies the magnitude of the impulse response and thus determines the time-dispersion characteristic, or equivalently, the frequency selectivity. For implementation of the channel model, usually a *tapped delay line* structure is used (cf. Figure 15, which shows a tapped delay line with taps equally spaced in time) and standards define the delay and the relative magnitude of the tap coefficients.

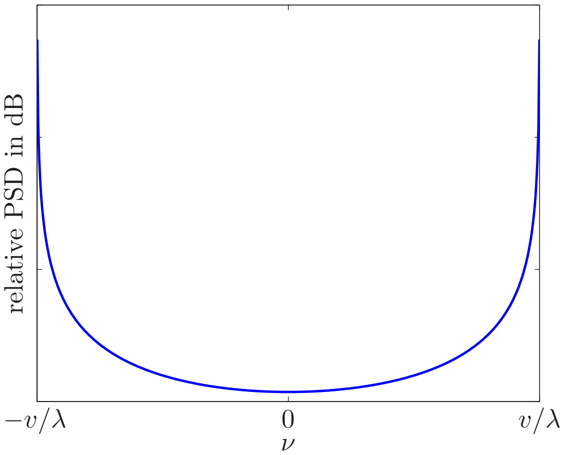


Figure 11: Jakes spectrum.

2. The Doppler cross-power spectral density $R_B(\Delta f, \nu)$ observed at $\Delta f = 0$ yields the *Doppler power density spectrum*
$$P(\nu) = R_B(\Delta f, \nu)|_{\Delta f = 0} . \tag{17}$$
The Doppler power density spectrum (often just referred to as Doppler spectrum) $P(\nu)$ specifies the frequency dispersion, or equivalently, the correlation over time of a given coefficient of the tapped delay line filter. An example of a commonly used Doppler spectrum, the so called Jakes spectrum, is depicted in Figure 11. The Jakes spectrum is the theoretical PSD received by a MT moving with velocity v, if the FT emits a tone of wavelength λ. $P(\nu)$ may be different for each tap and is thus specified together with the tap delay and the tap magnitude. The majority of models defined by various standardisation groups are specified in terms of $P(\tau)$ and $P(\nu)$.

The channel behaviour (i.e., its dispersiveness in time and frequency) is often described only by two scalar parameters. Naturally, this is a very condensed description, but it provides a rough idea about the channel properties.

1. The *multi-path spread* T_{multi}, which is the approximate support of the power delay profile $P(\tau)$, describes the approximate length of the channel impulse response. The larger T_{multi}, the more pronounced the time dispersion, and, equivalently, the more severe the frequency selectivity of the channel.

   The dual measure of T_{multi} in frequency domain is the *coherence bandwidth* $B_{\text{coh}} \approx 1/T_{\text{multi}}$, which is the approximate bandwidth over which the frequency correlation $R_H(\Delta f)$ is constant and non-zero, or equivalently, the bandwidth over which the channel transfer function is roughly frequency-flat(non-selective).

2. The *coherence time* T_{coh}, which is the approximate support of the time correlation function $R_H(\Delta t)$, defines the time during which the impulse response remains constant. Its dual frequency-domain measure is the *Doppler bandwidth* $B_{\text{doppler}} \approx 1/T_{\text{coh}}$, which is the bandwidth over which the Doppler spectrum is roughly constant and non-zero.

When designing a communication system, the mobile channel parameters have to be assessed with respect to the transmit symbol period[3] T_{sym}, or equivalently, the occupied bandwidth $B \approx 1/T_{\text{sym}}$. If, for a given channel and a given system, $T_{\text{sym}} < T_{\text{coh}}$, the fading is said to be *slow*, since the channel impulse response virtually does not change during a symbol interval. If $T_{\text{sym}} > T_{\text{coh}}$, the fading is said to be *fast*, since the channel impulse response changes during a symbol interval, which makes communication more difficult.

If $T_{\text{sym}} > T_{\text{multi}}$, or equivalently, $B < B_{\text{coh}}$, the channel is said to be *frequency-flat.* If $T_{\text{sym}} < T_{\text{multi}}$, or equivalently, $B > B_{\text{coh}}$, the channel is said to be *frequency-selective*, i.e., time-dispersion occurs. Figure 13 shows the classification of wireless channels according to different possible relations between T_{sym} and T_{coh}, T_{multi}.

[3] or chip duration in CDMA systems

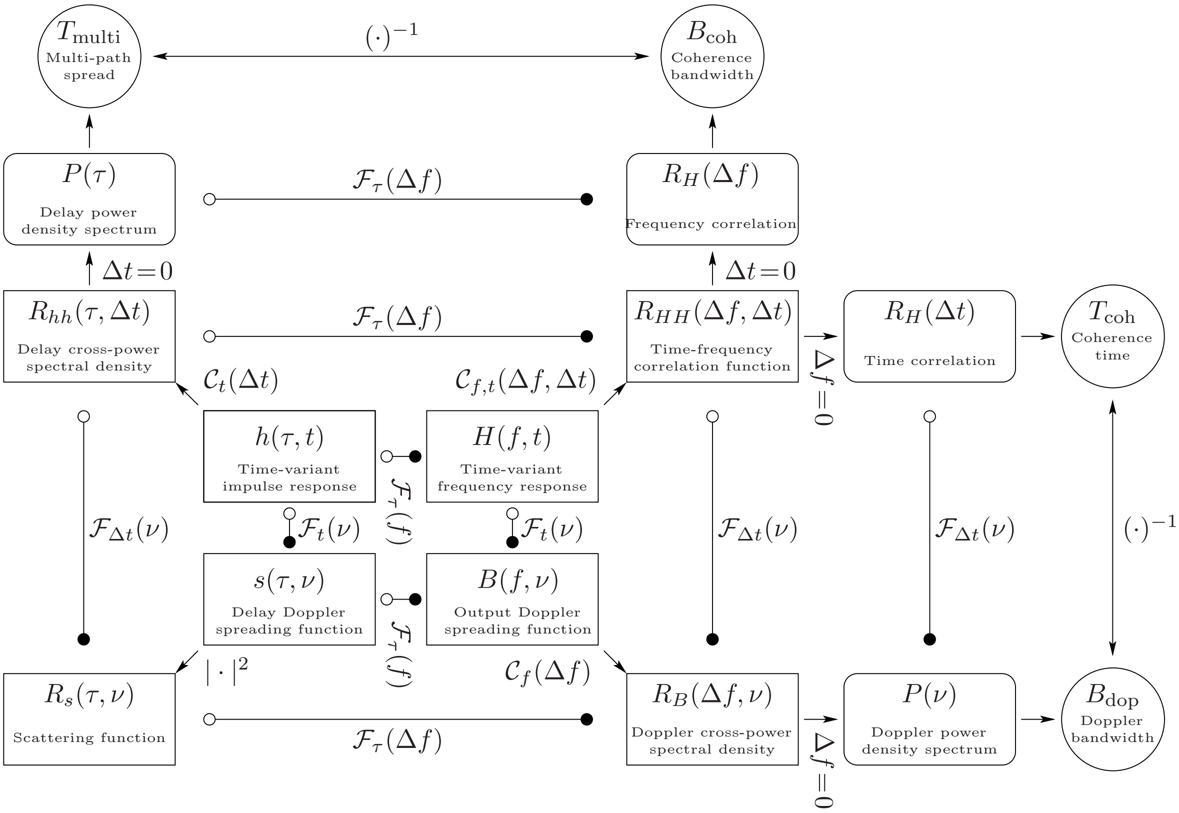

T_{multi}
Multi-path spread
$(\cdot)^{-1}$
B_{coh}
Coherence bandwidth
$P(\tau)$
Delay power density spectrum
$\mathcal{F}_\tau(\Delta f)$
$R_H(\Delta f)$
Frequency correlation
$\Delta t = 0$
$\Delta t = 0$
$R_{hh}(\tau, \Delta t)$
Delay cross-power spectral density
$\mathcal{F}_\tau(\Delta f)$
$R_{HH}(\Delta f, \Delta t)$
Time-frequency correlation function
$R_H(\Delta t)$
Time correlation
T_{coh}
Coherence time
$\Delta f = 0$
$\mathcal{C}_t(\Delta t)$
$\mathcal{C}_{f,t}(\Delta f, \Delta t)$
$h(\tau, t)$
Time-variant impulse response
$H(f, t)$
Time-variant frequency response
$\mathcal{F}_\tau(f)$
$\mathcal{F}_{\Delta t}(\nu)$
$\mathcal{F}_t(\nu)$
$\mathcal{F}_t(\nu)$
$\mathcal{F}_{\Delta t}(\nu)$
$\mathcal{F}_{\Delta t}(\nu)$
$(\cdot)^{-1}$
$s(\tau, \nu)$
Delay Doppler spreading function
$B(f, \nu)$
Output Doppler spreading function
$\mathcal{F}_\tau(f)$
$|\cdot|^2$
$\mathcal{C}_f(\Delta f)$
$R_s(\tau, \nu)$
Scattering function
$\mathcal{F}_\tau(\Delta f)$
$R_B(\Delta f, \nu)$
Doppler cross-power spectral density
$P(\nu)$
Doppler power density spectrum
B_{dop}
Doppler bandwidth
$\Delta f = 0$

Figure 12: Characterisation of wireless channels and relation between the measures ($\mathcal{F}_x(y)$ denotes the Fourier transform with respect to x as a function of y and $\mathcal{C}_x(y)$ denotes the correlation with respect x as function of y).

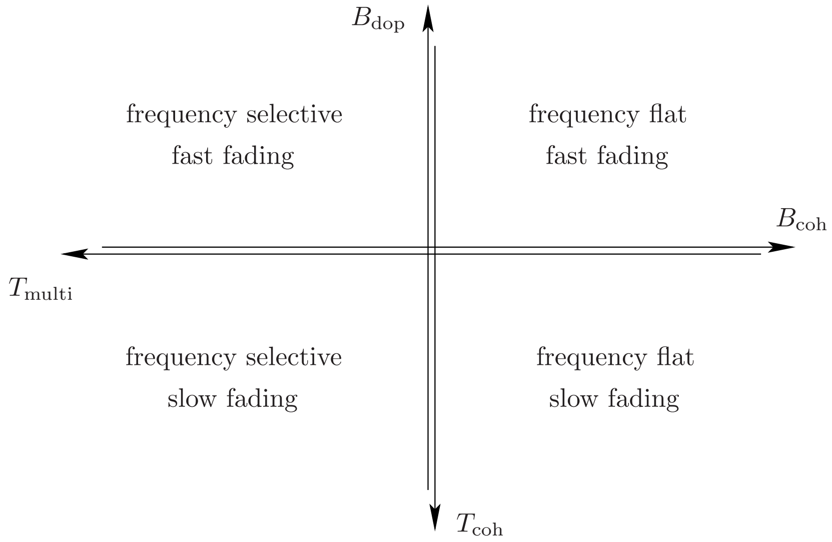


Figure 13: Assessment of a wireless channel: The symbol period of a given system is compared to the coherence time and to the delay spread.

4 Channel modelling

A useful description of a linear channel with additive noise is

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{n}$$

where $\boldsymbol{s} \in \mathbb{C}^S$ is the channel input, $\boldsymbol{r} \in \mathbb{C}^R$ is the channel output, $\boldsymbol{H} \in \mathbb{C}^{R \times S}$ is the channel matrix, and $\boldsymbol{n} \in \mathbb{C}^R$ is additive noise. This is a very general model and can describe dispersion in time, in frequency, and in space (MIMO, SIMO, MISO) simultaneously (cf. Example 22).

A classification of the channel variation over time that is often used in communications and information theory is the following:

- *Ergodic* channel: Consider the model

  $$\boldsymbol{r}_n = \boldsymbol{H}_n \boldsymbol{s}_n + \boldsymbol{n}_n.$$

  $\boldsymbol{H}_n$ are realizations of a random process. The assumption is, that a transmitted symbol (in uncoded transmission) or codeword (in coded transmission) denoted by $\boldsymbol{s}_n, n = 0, 1, \ldots, N, N \gg$ "sees" all channel states, which corresponds to averaging over $\boldsymbol{H}_n$. Fast fading channels are typically modelled as ergodic channels.

- *Nonergodic* channel: Consider the model

  $$\boldsymbol{r}_n = \boldsymbol{H}\boldsymbol{s}_n + \boldsymbol{n}_n,$$

  Here, $\boldsymbol{H}$ is constant over the codeword $\boldsymbol{s}_n, n = 0, 1, \ldots, N, N \gg$. The assumption is, that the channel variation is slow enough that a transmitted symbol/codeword "sees" only one state ($\boldsymbol{H}$). Slowly fading channels are typically modelled as nonergodic channels.

A frequently used notion is *block-fading*, which often occurs in the context of interleaved transmission. Codewords are spread out in time and/or frequency. A long interleaver can thus turn a nonergodic channel (where each code symbol of a codeword sees one channel state only) into an ergodic channel (where each code symbol of a codeword sees a different channel state). Block fading characterizes the situation in between those two extremes. If the interleaver is not long enough, blocks of code symbols see the same channel state.

A compact time-domain model of a frequency-selective MIMO channel with S inputs and R outputs is $\boldsymbol{H}[m] \in \mathbb{C}^{R \times S}, m = 0, \ldots, M$, where the element in row No. k and column No. ℓ is the impulse response $h_{k,\ell}[m]$ of the path from antenna (or port) ℓ to antenna (port) k.

5 Elementary Discrete-time linear channel models

5.1 Linear time-invariant (LTI) channel

The linear time-invariant (LTI) channel is completely characterised by its response $h(n), -\infty \leq n \leq \infty$ to an impulse $\delta(n), -\infty \leq n \leq \infty$ launched at time instant 0. Due to the time invariance, a time-shifted impulse $\delta(n-k), -\infty \leq n \leq \infty$ yields a time-shifted version $h(n-k), -\infty \leq n \leq \infty$ of the impulse response $h(n)$. Due to the homogeneity property of the LTI channel, a scaled impulse $s(k)\delta(n-k), -\infty \leq n \leq \infty$ yields a scaled response $s(k)h(n-k)$. Figure 14 shows exemplary responses for $k = 0, 1, 2$. Adding up all components corresponding to different values of k at the input yields the input signal

$$s(n) = \sum_k s(k)\delta(n-k), -\infty \leq n \leq \infty.$$

Due to the additivity property, the response to $s(n)$ is just the sum of the responses to the components $s(k)\delta(n-k), -\infty \leq n \leq \infty$ over all k, which yields

$$r(n) = \sum_k s(k)h(n-k), -\infty \leq n \leq \infty.$$

Homogeneity and additivity together form the linearity property. The filter structure corresponding to the convolution sum is depicted in Figure 15.

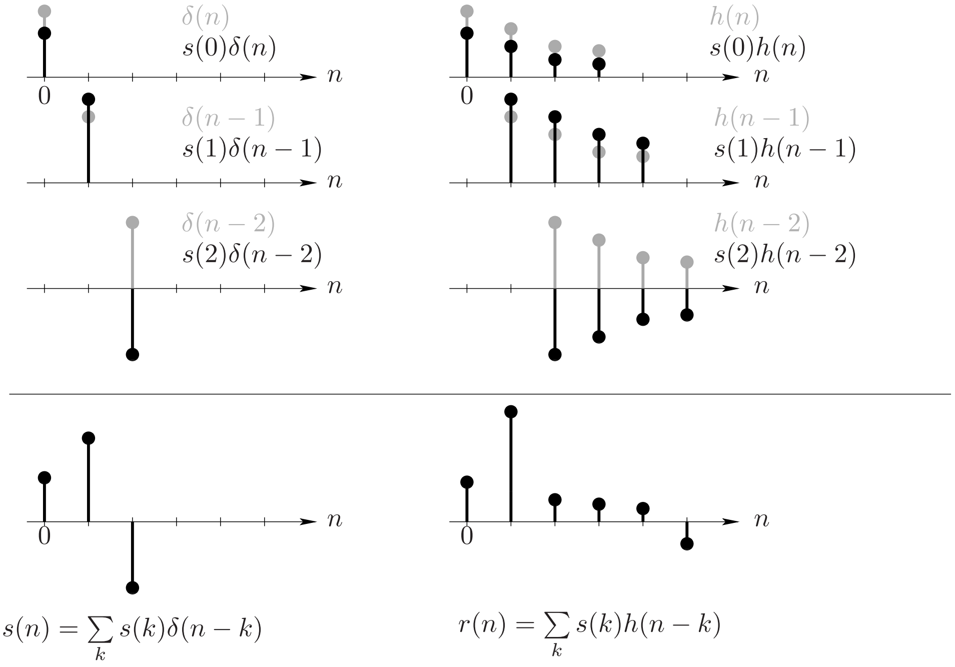


Figure 14: LTI channel.

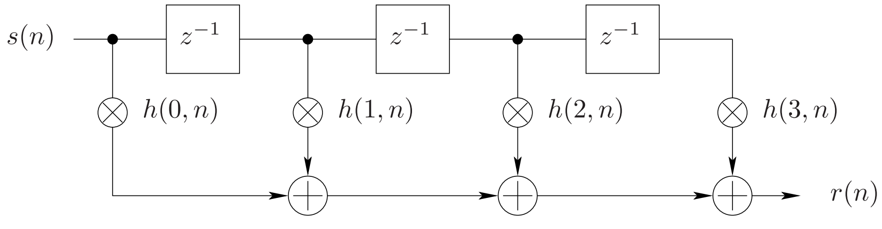


Figure 15: Channel model using responses $h(m, n)$.

5.2 Linear time-variant (LTV) channel

The response of the LTV channel to an impulse $\delta(n-k), -\infty \leq n \leq \infty$ launched at time k is $g(n,k), -\infty \leq n \leq \infty$, which can be viewed as a function of n with parameter k. The response $g(n,k)$ is sometimes referred to as the *response the transmitter sees.* Figure 16 shows exemplary responses for $k = 0, 1, 2$. Due to the homogeneity property, a scaled impulse $s(k)\delta(n-k), -\infty \leq n \leq \infty$ yields a scaled response $s(k)g(n-k,k), -\infty \leq n \leq \infty$. Again, the receive signal is obtained by adding up the scaled responses corresponding to different k values, which yields

$$r(n) = \sum_k s(k)g(n-k,k), -\infty \leq n \leq \infty.$$

This convolution sum, written in terms of $g(m,n)$, suggests the filter structure depicted in Figure 17. Now consider the following response

$$h(m,n) = g(m,n-m), -\infty \leq m \leq \infty,$$

which can be viewed as a function of m with parameter n and is sometimes referred to as the *response the receiver sees.* In fact, $h(m,n), -\infty \leq m \leq \infty$ is not the response to a single impulse but the component

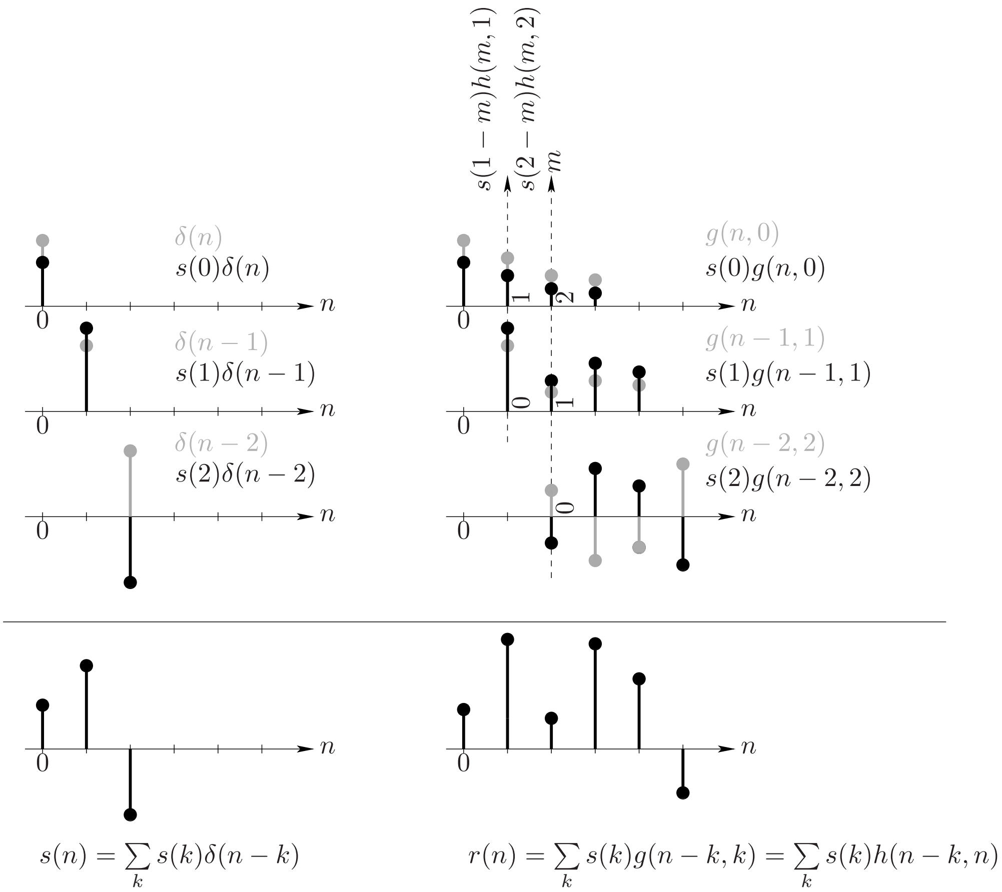


Figure 16: LTV channel.

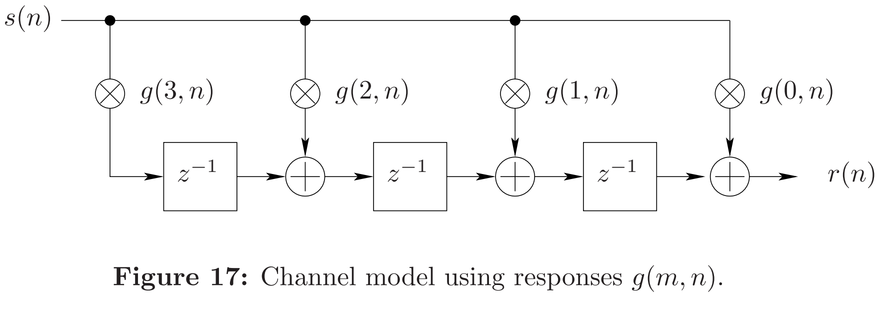


Figure 17: Channel model using responses $g(m,n)$.

observed at time n of the response to an impulse launched m samples ago, i.e., at time $n-m$. Thus, for each m, $h(m,n)$ is the component observed at time n of the response to the impulse launched m seconds ago. In terms of $h(m,n)$, the receive signal can be written as

$$\begin{aligned} r(n) = \sum_k s(k) g(\underbrace{n-k}_{\triangleq m}, k) = \sum_m s(n-m) g(m, n-m) = \sum_m s(\underbrace{n-m}_{\triangleq k}) h(m,n) = \\ \sum_k s(k) h(n-k, n), -\infty \leq n \leq \infty. \end{aligned} \tag{18}$$

The filter structure corresponding to the convolution sum written in terms of $h(m,n)$ is depicted in Figure 15.

References

[1] J.B. Anderson, *Digital Transmission Engineering*, IEEE Press, ISBN 0-7803-3457-4, 1999.

[2] A. F. Molisch, *Wideband Wireless Digital Communications*, Prentice Hall, ISBN 0-1302-2333-6, 2000.

[3] J. G. Proakis, *Digital Communications*, Mc Graw Hill, ISBN 0-07-232111-3, fourth edition, 2001.

[4] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, ISBN 0-7803-3483-3, 1999.

[5] T. Starr, J. M. Cioffi, and P. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice Hall, Englewood Cliffs, 1998.

Part III

Data Detection, Channel Equalisation, Channel Estimation

1 Data Detection

1.1 Optimal Receive Processing

The *optimum processing* — optimum in the sense of minimising the probability of block error — of the receive signal is *maximum likelihood (ML) sequence estimation/detection*. As the term suggests, such processing finds the *most probable* transmitted *sequence*. An intuitive explanation of why investigating a sequence instead of individual symbols (or samples) makes sense, is that the transmitted symbols (or samples) are dispersed in time by the channel (if the channel dispersion $M > 0$, which is often the case; for $M = 0$ and white noise, symbol-by-symbol detection performs as well as sequence detection). Consequently, the receiver has to consider a block of symbols (or samples) to capture all the information about a single transmit symbol (or sample). The complexity of ML sequence detection of a sequence of length n with symbols from an alphabet of size m is proportional to m^n, which is often prohibitive. There are several ways to decrease this complexity, which leads to approximations of ML sequence detection (e.g. iterative (often referred to as "turbo") techniques).

1.2 Coherent and Noncoherent Detection

We speak of *absolute modulation* when the information lies in the explicit choice of a symbol from given constellation. Absolute modulation at the transmitter implies that the receiver has to make decisions based on a reference that is coherent with the transmit constellation (cf. Figure 1). Consequently, *coherent detection* requires channel knowledge at the receiver, which in turn requires channel estimation. The performance of absolute modulation depends on the quality of the channel estimates and thus implicitly on the coherence properties of the channel as we will see shortly.

When the information lies in the difference between consecutive symbols, we speak of *differential modulation*. Assuming that the coherence of the channel is large enough to render the channel variation from symbol to symbol negligible, there is no need for channel equalisation (cf. Figure 1). Consequently, the receiver does not need to estimate the channel. As illustrated in Figure 2, differential modulation can be done

- in time (information lies in the difference to the symbol on the same subcarrier of the previous multicarrier symbol)

- in frequency (information lies in the difference to the symbol on a neighboring subcarrier of the same multicarrier symbol)

The performance of differential modulation clearly depends on the coherence properties of the channel. For example, consider differential phase modulation in time. Let $\underline{x}_{k,n} = e^{j\varphi_{k,n}}, \varphi_{k,n} \in \{\varphi^{(1)}, \ldots, \varphi^{(P)}\}$ denote the transmit symbol on subcarrier No. k of multicarrier symbol No. n, where P is the constellation

Chapter written by T. Magesacher.

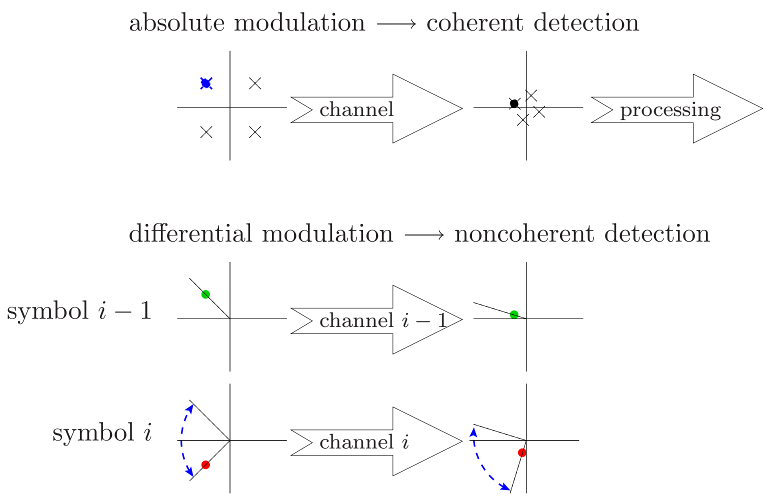

absolute modulation ⟶ coherent detection
channel
processing
differential modulation ⟶ noncoherent detection
symbol $i-1$
channel $i-1$
symbol i
channel i

Figure 1: Absolute modulation and coherent detection versus differential modulation and noncoherent detection.

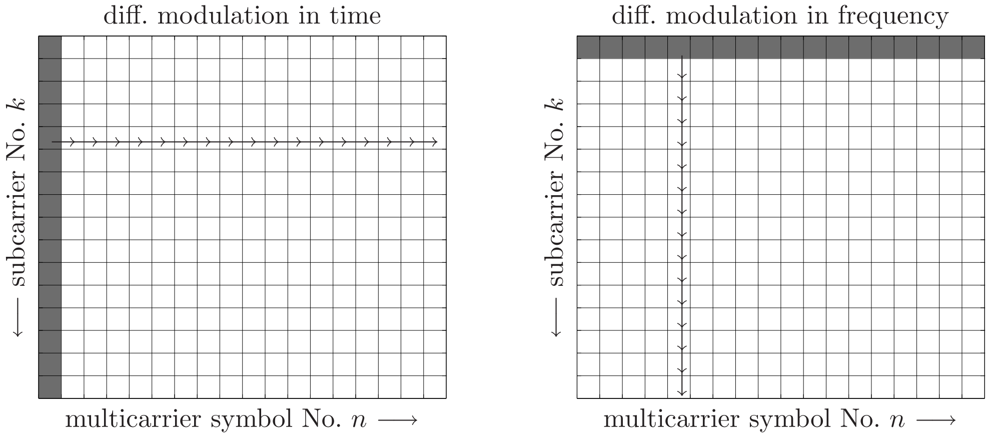

diff. modulation in time
⟵ subcarrier No. k
multicarrier symbol No. n ⟶
diff. modulation in frequency
⟵ subcarrier No. k
multicarrier symbol No. n ⟶

Figure 2: Differential modulation in time (left) and in frequency (right).

size. The channel multiplies $\underline{x}_{k,n}$ with $H[k,n] = c_{k,n}e^{j\phi_{k,n}}$ and adds noise denoted by $\underline{w}_{k,n}$. The receive symbol $\underline{y}_{k,n}$ is thus given by

$$\underline{y}_{k,n} = c_{k,n}e^{j(\varphi_{k,n}+\phi_{k,n})} + \underline{w}_{k,n}$$

A noncoherent differential detector in time correlates the current symbol $\underline{y}_{k,n}$ with the reference symbol $\underline{y}_{k,n-1}$, which yields

$$\hat{\underline{x}}_{k,n} = \underline{y}_{k,n}\underline{y}^*_{k,n-1} = c_{k,n}c_{k,n-1}e^{j(\varphi_{k,n}-\varphi_{k,n-1}+\phi_{k,n}-\phi_{k,n-1})} + \underline{w}_{k,n}\underline{w}^*_{k,n-1} + \underline{w}_{k,n}\underline{y}^*_{k,n-1} + \underline{w}^*_{k,n-1}\underline{y}_{k,n}$$

Apart from the noise, detection of the information $\varphi_{k,n} - \varphi_{k,n-1}$ is impaired by the term $\phi_{k,n} - \phi_{k,n-1}$, which is negligible if the channel coherence in time is sufficiently large.

2 Two Elements of Estimation Theory

Before continuing with channel estimation, we briefly review two elementary approaches from estimation theory that are frequently used in communication applications:

- Least squares (LS)
- Minimum mean square error (MMSE)

2.1 Least squares (LS) approach

Least squares (LS) is a very general concept with a vast number of applications in all fields of engineering. Consider the model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a strictly skinny ($m > n$) deterministic and known coefficient matrix, $\boldsymbol{y}$ is a deterministic and known vector and $\boldsymbol{x}$ is unknown. For most $\boldsymbol{y}$, there is no $\boldsymbol{x}$ that solves (1) (by solves we mean *exactly*). An approach to find a "good" approximation $\boldsymbol{x}$ that approximately solves (1) is the following:

- Define the error $\boldsymbol{e} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}$.
- Find $\boldsymbol{x}_{\mathrm{LS}}$ that minimises the Euclidean norm $\|\boldsymbol{e}\|_2 = \sqrt{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}$ of the error, *i.e.*,

$$\boldsymbol{x}_{\mathrm{LS}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{e}\|_2 \tag{2}$$

The approximate "solution" $\boldsymbol{x}_{\mathrm{LS}}$ is called the least squares solution. Note:

- Minimising $\|\boldsymbol{e}\|_2$ corresponds to minimising the energy $\|\boldsymbol{e}\|_2^2$ of the error.
- Both the parameter $\boldsymbol{x}$ and the error $\boldsymbol{e}$ are *deterministic*—least squares is a deterministic approach.

The LS approach has a neat geometric interpretation illustrated in Figure 3. The least squares solution $\boldsymbol{x}_{\mathrm{LS}}$ minimises the Euclidean distance between $\boldsymbol{A}\boldsymbol{x}_{\mathrm{LS}}$ and $\boldsymbol{y}$. $\mathcal{R}(\boldsymbol{A})$ is the n-dimensional subspace spanned by the columns of $\boldsymbol{A}$. $\boldsymbol{A}\boldsymbol{x}$ is a projection of the point $\boldsymbol{y}$ onto $\mathcal{R}(\boldsymbol{A})$ (and thus lies in $\mathcal{R}(\boldsymbol{A})$). Among all projections, the orthogonal projection $\boldsymbol{A}\boldsymbol{x}_{\mathrm{LS}}$ minimises $\|\boldsymbol{e}\|_2$

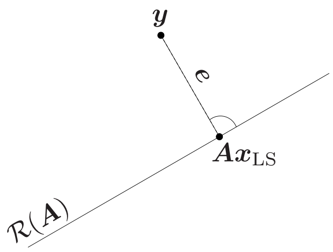

$\boldsymbol{y}$
$\boldsymbol{e}$
$\boldsymbol{A}\boldsymbol{x}_{\mathrm{LS}}$
$\mathcal{R}(\boldsymbol{A})$

Figure 3: Geometric interpretation of the least square approach.

Assume that $\boldsymbol{A}$ is full rank ($\operatorname{rank}\{\boldsymbol{A}\} = n$). $\boldsymbol{x}_{\mathrm{LS}}$ minimises both $\|\boldsymbol{e}\|_2$ and $\|\boldsymbol{e}\|_2^2$. Thus

$$\|\boldsymbol{e}\|_2^2 = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}$$

In order to minimise $\|\boldsymbol{e}\|_2^2$, we set the gradient $\nabla_{\boldsymbol{x}}\|\boldsymbol{e}\|_2^2$ to zero

$$\nabla_{\boldsymbol{x}}\|\boldsymbol{e}\|_2^2 = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \overset{!}{=} 0$$

This yields the so-called normal equations $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$. Since $\boldsymbol{A}$ is full rank, $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ is a nonsingular $n \times n$ matrix. The least squares solution is thus

$$\boldsymbol{x}_{\mathrm{LS}} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$$

Similarly, in the complex-valued case we obtain: $\boldsymbol{x}_{\mathrm{LS}} = \left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{y}$.

2.2 Minimum mean square error (MMSE) approach

In the least squares approach we did not make any assumptions about statistical properties of the transmit data $\boldsymbol{x}$ or the noise $\boldsymbol{w}$. In fact, least squares assumes that the data is deterministic and unknown and assumes no prior knowledge about signal or noise.

A different approach is to assume that data and noise are stochastic, which makes sense in particular if we have some knowledge about their statistical properties (*i.e.*, if we have prior information). Hereinafter, we assume that all stochastic variables have zero mean. Consequently, covariance corresponds to correlation. Consider a length-n parameter vector $\boldsymbol{x}$ to be estimated from a length-m observation vector $\boldsymbol{y}$. The parameters are assumed to be realisations of a random vector $\boldsymbol{x}$ with covariance matrix $\boldsymbol{C}_{\boldsymbol{x}\boldsymbol{x}} = \mathsf{E}\left\{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right\}$. An approach to find a "good" *linear* estimate $\hat{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{y}$ of $\boldsymbol{x}$ is the following:

- define the error $\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{G}\boldsymbol{y}$
- find $\boldsymbol{x}_{\mathrm{MMSE}}$ that minimises the mean of the squared error, *i.e.*,

$$\boldsymbol{x}_{\mathrm{MMSE}} = \arg\min_{\hat{\boldsymbol{x}}} \mathsf{E}\left\{\|\boldsymbol{e}\|_2^2\right\} \tag{3}$$

The solution $\boldsymbol{x}_{\mathrm{MMSE}}$ is called minimum mean square error (MMSE) estimate. Note:

- parameter $\boldsymbol{x}$ and error $\boldsymbol{e}$ are *stochastic*—MMSE is a Bayesian estimation approach

For the scalar case ($n = 1$) we have a nice geometric interpretation, illustrated in Figure 4, which moreover provides the key to finding a simple solution. The MMSE estimate x_{MMSE} minimises the Euclidean distance between x_{MMSE} and x. The components of $\boldsymbol{y}$ span an m-dimensional subspace $\mathcal{Y}$. $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{y}$ is a projection of x onto $\mathcal{Y}$ (and thus lies in $\mathcal{Y}$). Among all projections, the orthogonal projection $x_{\mathrm{MMSE}} = \boldsymbol{g}_{\mathrm{MMSE}}{}^{\mathrm{T}}\boldsymbol{y}$ of x onto $\mathcal{Y}$ minimises the mean of $\|\boldsymbol{e}\|_2$. The error $x - \boldsymbol{g}_{\mathrm{MMSE}}{}^{\mathrm{T}}\boldsymbol{y}$ is orthogonal to $\mathcal{Y}$, which implies $\mathsf{E}\left\{(x - \boldsymbol{g}_{\mathrm{MMSE}}{}^{\mathrm{T}}\boldsymbol{y})\boldsymbol{y}\right\} = \boldsymbol{0}$. The solution x_{MMSE} has to fulfil $\mathsf{E}\left\{(x - \boldsymbol{g}_{\mathrm{MMSE}}{}^{\mathrm{T}}\boldsymbol{y})\boldsymbol{y}\right\} = \boldsymbol{0}$. Consequently, $\mathsf{E}\left\{\boldsymbol{y}x\right\} - \mathsf{E}\left\{\boldsymbol{g}_{\mathrm{MMSE}}{}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}\right\} = \underbrace{\mathsf{E}\left\{\boldsymbol{y}x\right\}}_{\boldsymbol{c}_{\boldsymbol{y}x}} - \underbrace{\mathsf{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right\}}_{\boldsymbol{C}_{\boldsymbol{y}\boldsymbol{y}}}\boldsymbol{g}_{\mathrm{MMSE}} = \boldsymbol{0}$. This yields the so-called Wiener-Hopf equations $\boldsymbol{C}_{\boldsymbol{y}\boldsymbol{y}}\boldsymbol{g}_{\mathrm{MMSE}} = \boldsymbol{c}_{\boldsymbol{y}x}$, whose solution is $\boldsymbol{g}_{\mathrm{MMSE}} = \boldsymbol{C}_{\boldsymbol{y}\boldsymbol{y}}^{-1}\boldsymbol{c}_{\boldsymbol{y}x}$. The MMSE solution is thus

$$x_{\mathrm{MMSE}} = \boldsymbol{c}_{\boldsymbol{y}x}{}^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{y}\boldsymbol{y}}^{-1}\boldsymbol{y}$$

Similarly, in the complex-valued vector-case we obtain $\boldsymbol{x}_{\mathrm{MMSE}} = \boldsymbol{C}_{\boldsymbol{y}\boldsymbol{x}}^{\mathrm{H}}\boldsymbol{C}_{\boldsymbol{y}\boldsymbol{y}}^{-1}\boldsymbol{y}$.

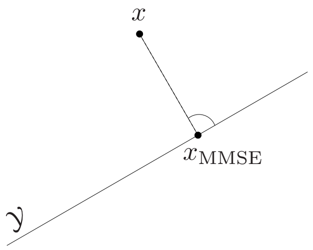


Figure 4: Geometric interpretation of the minimum mean square error approach.

3 Channel Equalisation

A *suboptimum* approach to data detection involves *processing* with the aim of mitigating ISI before symbol-by-symbol detection. This processing is referred to as *channel equalisation*.

In the following, we will consider both the scalar and the vector case. The scalar signal model is given by

$$y(n) = a(n) * x(n) + w(n) \tag{4}$$

where the transmit signal $x(n)$ is dispersed in time, modelled by the convolution of $x(n)$ with the channel impulse response $a(n)$, and disturbed by zero-mean additive noise $w(n)$ yielding the receive signal $y(n)$. The model (4) implies continuous transmission.

The signal model for the vector case is given by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \tag{5}$$

and implies block transmission.

Depending on the type of receive signal processing, we distinguish linear and nonlinear equalisation.

3.1 Linear Equalisation

Zero-forcing (ZF) equaliser

As the name says, the zero-forcing (ZF) equaliser tries to force the ISI to be zero, without considering the consequences this operation may have. In fact, the LS approach applied to channel equalization yields the ZF solution.

Scalar formulation We will briefly review the z-domain formulation of ZF, which is commonly used for singlecarrier (pulse-amplitude modulated) systems. Intuition suggests that the zero-forcing equaliser $G_{\mathrm{ZF}}(z)$ for a given channel $H(z)$, is the inverse of the channel (if this inverse exists):

$$G_{\mathrm{ZF}}(z) = \frac{1}{H(z)} = \frac{H^*(z^{-1})}{H(z)H^*(z^{-1})}. \tag{6}$$

In case this inverse does not exist, all we can do is find the best fit using the least-squares method (cf. time-domain equalisation). Problems that arise with the ZF equaliser are:

- If $H(z)$ is not minimum-phase (a minimum-phase filter has the property that all the zeros of its z-transform $H(z)$ lie inside the unit circle), its inverse $G_{\mathrm{ZF}}(z) = \frac{1}{H(z)}$ is not stable (the zeros of $H(z)$ are the poles of its inverse). From theoretical point of view there is no problem what so ever using an unstable inverse filter $G_{\mathrm{ZF}}(z)$: the output of the equaliser will be free of ISI. In practice, however, the impulse response of the unstable inverse filter contains elements of huge magnitude (and the magnitude grows with n), which causes problems in implementation.
- It may require infinitely many taps to realise $G_{\mathrm{ZF}}(z)$. In practice, only an approximation can be implemented.

The main problem, however, is that $G_{\mathrm{ZF}}(z)$, while removing or at least reducing the ISI, may amplify the noise $w(n)$. Note that the ZF equaliser only requires an estimate of the channel impulse response but not of the noise covariance.

Vector formulation In the following, we use the LS method and derive a matrix formulation of ZF for the multicarrier setup. The signal model is given by

$$\boldsymbol{y} = \boldsymbol{R}\boldsymbol{Z}_{\mathrm{rem}}(\boldsymbol{H}\boldsymbol{Z}_{\mathrm{add}}\boldsymbol{S}\boldsymbol{x} + \boldsymbol{w}') = \underbrace{\boldsymbol{R}\boldsymbol{Z}_{\mathrm{rem}}\boldsymbol{H}\boldsymbol{Z}_{\mathrm{add}}\boldsymbol{S}}_{\boldsymbol{A}}\boldsymbol{x} + \boldsymbol{w} \tag{7}$$

The least squares solution

$$\boldsymbol{x}_{\mathrm{LS}} = \underbrace{(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{H}}}_{\boldsymbol{G}_{\mathrm{ZF}}}\boldsymbol{y}$$

applied to this setup minimises $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2$, *i.e.*, the deviation between our actual receive vector $\boldsymbol{y}$ and what we would receive if we would transmit $\boldsymbol{x}_{\mathrm{ZF}}$ and there were no noise $\boldsymbol{w}$. Clearly, the least squares solution can be arbitrarily off if the noise is large. In a multicarrier system where $L \geq M$, $\boldsymbol{G}_{\mathrm{ZF}}$ is diagonal and multicarrier equalisation reduces to multiplication with a complex scalar per subchannel, sometimes referred to as frequency domain equaliser (FEQ). The ZF equaliser written in DFT-domain is

$$G_{\mathrm{ZF}}[k] = \frac{1}{H[k]} = \frac{H^*[k]}{|H[k]|^2}, \qquad k = 0, \ldots, N-1, \tag{8}$$

where k denotes the subcarrier (subchannel) index. Figure 5 depicts a block diagram of the channel and the receiver.

MMSE equaliser

Scalar formulation We begin with the z-domain formulation (singlecarrier PAM) before moving on to the multicarrier setup. Instead of fighting exclusively the ISI, the minimum mean square error (MMSE) equaliser, as the name suggests, aims at minimising

$$\mathsf{E}\left\{|e(n)|^2\right\}, \tag{9}$$

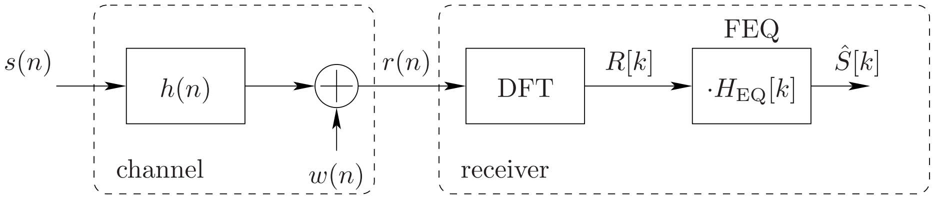


Figure 5: Multicarrier equalisation in frequency domain: as long as $L \geq M$, equalisation reduces to a complex scalar multiplication per subchannel.

which is the mean of the square of the error $e(n)$, given by

$$e(n) = s(n) - \hat{s}(n) = s(n) - (s(n) * h(n) + w(n)) * h_{\mathrm{EQ}}(n). \tag{10}$$

Solving this minimisation problem (→ problem solving session), which can be formulated as

$$g_{\mathrm{MMSE}}(n) = \arg \min_{h_{\mathrm{EQ}}(n)} \mathsf{E}\left\{|e(n)|^2\right\}, \tag{11}$$

yields the MMSE equaliser. Equation (11) describes the *MMSE criterion.* Note that the parameters $h_{\mathrm{EQ}}(n)$ are random variables. The z-transform of the solution of (11) is

$$G_{\mathrm{MMSE}}(z) = \frac{H^*(z^{-1})}{H(z)H^*(z^{-1}) + C_{ww}(z)}, \tag{12}$$

where $C_{ww}(z)$ is the z-transform of the autocorrelation function $r_{ww}(n)$ of the noise $w(n)$. The ZF equaliser (6) and the MMSE equaliser (12) are identical for $C_{ww}(z) = 0$, *i.e.*, in case there is no noise. The noise power spectral density $C_{ww}(z)$ limits the magnitude of the inverse and thus mitigates the noise enhancement problem. The MMSE equaliser requires an estimate of both the channel impulse response and the noise covariance.

Vector formulation Now we derive a matrix formulation for the MMSE equalizer in a multicarrier setup. Consider the model (7). Assuming that data and noise are stochastic and applying the MMSE solution, we obtain

$$\boldsymbol{x}_{\mathrm{MMSE}} = \underbrace{\boldsymbol{C_{xx}A}^{\mathrm{H}}\left(\boldsymbol{AC_{xx}A}^{\mathrm{H}} + \boldsymbol{C_{ww}}\right)^{-1}}_{\boldsymbol{G}_{\mathrm{MMSE}}}\boldsymbol{y} \tag{13}$$

In a multicarrier system where $L \geq M$, $\boldsymbol{G}_{\mathrm{MMSE}}$ is diagonal and multicarrier equalisation again reduces to multiplication with a complex scalar per subchannel. The MMSE equaliser written in DFT-domain is

$$G_{\mathrm{MMSE}}[k] = \frac{H^*[k]}{|H[k]|^2 + C_{ww}[k]}, \qquad k = 0, \ldots, N-1, \tag{14}$$

where $C_{ww}[k]$ describes the noise power (or equivalently, the noise variance for zero-mean noise) of the kth subchannel. Here, we tacitly assume that the noise power spectral density (PSD) is flat for each subchannel. The larger N, the better the chances that this approximation holds.

The linear MMSE equaliser derived above achieves the minimum estimation error when the parameter $\boldsymbol{x}$ has Gaussian distribution. In a communications system, we usually deal with finite alphabets (*e.g.*, QPSK or 16-QAM). Hence, instead of *estimating* we are interested in *detecting* the discrete-values of the transmitted symbol. Consequently, we have to normalise $\boldsymbol{G}_{\mathrm{MMSE}}$ to obtain

$$\boldsymbol{G}'_{\mathrm{MMSE}} = \boldsymbol{U}\boldsymbol{G}_{\mathrm{MMSE}}, \qquad \boldsymbol{U}[k,\ell] = \begin{cases} 1 + \left(\frac{E_{\mathrm{S}}}{N_0 N}|\boldsymbol{A}[k,k]|^2\right)^{-1}, & k = \ell = 1, \ldots, N \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

which is sometimes also referred to as unbiased MMSE detector. Note:

- Since the noise makes sure that $\left(\boldsymbol{A}\boldsymbol{C}_{\boldsymbol{xx}}\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{C}_{\boldsymbol{ww}}\right)$ is non-singular, the inverse always exists.
- For uncorrelated data ($\boldsymbol{C}_{\boldsymbol{xx}} = \boldsymbol{I}$) and low noise ($\sum_k \boldsymbol{C}_{\boldsymbol{ww}}[k,k] \ll \sum_k \boldsymbol{C}_{\boldsymbol{xx}}[k,k]$) the MMSE solution is close to the LS solution.

3.2 Decision Feedback Equalisation, GDFE, VBLAST

The basic idea of decision feedback equalisation (DFE) is to reconstruct the intersymbol-interference (ISI), via the so-called feedback filter, caused by the previous symbol(s) and subtract it from the receive signal. In the ideal case, there is only intrasymbol-interference left after the subtraction which is mitigated by the so-called feedforward filter. The reconstruction is based on hard decisions that are fed into the feedback filter, which tries to construct the same intersymbol-interference as the channel. Clearly, the assumption is that the hard decisions made in the receiver are correct, which holds for high-enough signal-to-noise ratios. An erroneous decision can cause a long lasting chain of erroneous decisions via the feedback path. A good introduction to DFE can be found in [1]. A detailed treatment of DFE design for block transmission (matrix formulation) given in [2].

Now we derive two signal models from (5) that will play an elementary role in developing a generalized version of DFE (GDFE). For simplicity, we assume white noise with variance $N_0/2$, *i.e.*, $\mathsf{E}\left\{\boldsymbol{w}\boldsymbol{w}^{\mathrm{H}}\right\} = \frac{N_0}{2}\boldsymbol{I}$. Matched filtering, matched to the channel, of the receive signal $\boldsymbol{y}$ given by (5) yields

$$\boldsymbol{z} = \boldsymbol{A}^{\mathrm{H}}\boldsymbol{y} = \underbrace{\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}}_{\boldsymbol{C}_{\mathrm{f}}}\boldsymbol{x} + \underbrace{\boldsymbol{A}^{\mathrm{H}}\boldsymbol{w}}_{\boldsymbol{w}'}, \tag{16}$$

which is referred to as the forward canonical model. Similarly, deriving the MMSE estimate of $\boldsymbol{x}$ given $\boldsymbol{z}$ yields the backward canonical model

$$\boldsymbol{x} = \underbrace{\left(\boldsymbol{C}_{\mathrm{f}} + (N_0/2)\boldsymbol{C}_{\boldsymbol{x}}^{-1}\right)^{-1}}_{\boldsymbol{C}_{\mathrm{b}}}\boldsymbol{z} + \boldsymbol{e}, \tag{17}$$

where $\boldsymbol{C}_{\boldsymbol{x}} = \mathsf{E}\left\{\boldsymbol{x}\boldsymbol{x}^{\mathrm{H}}\right\}$ and $\boldsymbol{e}$ denotes the MMSE error with covariance matrix $\mathsf{E}\left\{\boldsymbol{e}\boldsymbol{e}^{\mathrm{H}}\right\} = \boldsymbol{C}_{\mathrm{b}}$.

The ZF-DFE is based on the Cholesky factorization $\boldsymbol{C}_{\mathrm{f}} = \boldsymbol{G}^{\mathrm{H}}\boldsymbol{S}\boldsymbol{G}$ of $\boldsymbol{C}_{\mathrm{f}}$. The MMSE-DFE is based on the Cholesky factorization $\boldsymbol{C}_{\mathrm{b}}^{-1} = \boldsymbol{G}^{\mathrm{H}}\boldsymbol{S}\boldsymbol{G}$ of $\boldsymbol{C}_{\mathrm{b}}^{-1}$. In both cases, $\boldsymbol{G}$ is a upper triangular and monic (all ones the main diagonal) and $\boldsymbol{S}$ is diagonal. The DFE processing can be summarized as follows:

- $\tilde{\boldsymbol{z}} = \left(\boldsymbol{G}^{\mathrm{H}}\boldsymbol{S}\right)^{-1}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{y}$, $\tilde{\boldsymbol{G}} = \boldsymbol{I} - \boldsymbol{G}$
- for $k = N : -1 : 1$, $\boldsymbol{x}[k] = \text{hard-decision}\left(\tilde{\boldsymbol{z}}[k] + \tilde{\boldsymbol{G}}[k,:]\boldsymbol{x}\right)$, end

Note that $\left(\boldsymbol{G}^{\mathrm{H}}\boldsymbol{S}\right)^{-1}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} = \boldsymbol{G}$ is upper triangular and monic. The backward subsitution of the interference $\tilde{\boldsymbol{G}}[k,:]\boldsymbol{x}$ reconstructed based on decisions eliminates (in case of ZF-DFE) or mitigates (in case of MMSE-DFE) the ISI.

It can be shown that GDFE processing is equivalent to VBLAST (Vertical Bell-Labs Layered Space Time “coding”)—a standard technique to eliminate interference in multiple-input multiple-output (MIMO) setups such as multi-antenna scenarios. Denoting $\boldsymbol{a}_k$ the kth column of $\boldsymbol{A}$, VBLAST processing can be summarized as

- $\boldsymbol{v}_1 = \boldsymbol{y}$
- for $k = 1 : 1 : N$, $\boldsymbol{x}[N-k+1] = \text{hard-decision}\left(\boldsymbol{\omega}_{N-k+1}^{\mathrm{H}}\boldsymbol{v}_k\right)$, $\boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \boldsymbol{a}_{N-k+1}\boldsymbol{x}[N-k+1]$, end

where the vector $\boldsymbol{\omega}_{N-k+1}, k = 1,\ldots,N$ performs linear weighting of the elements in $\boldsymbol{v}_k, k = 1,\ldots,N$ such that the interference is eliminated (ZF-VBLAST) or mitigated (MMSE-VBLAST).

3.3 Multicarrier time-domain equalisation (TEQ)

As we have already seen, the whole multicarrier-concept falls apart if $L < M$, i.e., if the cyclic prefix is shorter than the dispersion of the channel. ISI and inter-carrier interference (ICI), which can also be interpreted as intra-symbol interference since it is not caused by preceding or succeeding symbols but by the symbol under consideration itself, occur for $L < M$ and the simple per-subchannel equalisation seizes to work. Per transmit symbol, L out of $L + N$ samples are cyclically repeated and thus redundant. In other words, $100\,L/(N + L)\%$ of the transmit signal neither carries information nor "redundancy added in a smart way" (like channel coding does with the aim of increasing the reliability of transmission), the cyclic extension is a plain repetition. Consequently, $100\,L/(N + L)\%$ of the achievable rate (or equivalently, of the transmit power) are inherently wasted. The smaller $L = M$, the better.

The goal of time-domain equalisation is to shorten the channel dispersion from M to a desired dispersion $M_1 < M$ (or equivalently, shorten the length of the channel impulse response from $M+1$ to a desired length $M_1 + 1$). Actually, it is rather a shortening of the channel impulse response than an equalisation operation. However, time-domain equalisation is an established term for this operation and we will stick to it. The simplest method is to apply a transversal filter $h_{\mathrm{TEQ}}(n)$ before the DFT operation at the receiver, as shown in Figure 6. In case the channel impulse response $h(n)$ is known and time-invariant (or slowly time-variant), the following offline approach, which is based on the method of least squares, can be applied. The desired result of the linear convolution of the transversal filter $h_{\mathrm{EQ}}(n), n = 0, \dots, K$ of length $K + 1$ and the channel can be described using convolution matrix notation:

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ r'(0) \\ r'(1) \\ \vdots \\ r'(M_1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \dots & 0 \\ h(1) & h(0) & & \vdots \\ & h(1) & \ddots & 0 \\ \vdots & & \ddots & h(0) \\ & & & h(1) \\ h(M) & & & \\ & h(M) & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & h(M) \end{bmatrix} \begin{bmatrix} h_{\mathrm{EQ}}(0) \\ h_{\mathrm{EQ}}(1) \\ \vdots \\ h_{\mathrm{EQ}}(K) \end{bmatrix} \tag{18}$$

Since we have no idea how the elements $r'(n)$ of the shortened channel impulse response may look like, we can simply omit the rows that yield $r'(n), 1 \leq n \leq M_1$ ("don't care"). In order to avoid the all zero solution, we must keep one row and fix that coefficient, i.e., we keep the row yielding $r'(0)$ and set $r'(0) = 1$. It is clear that this equation system may not have a unique solution, so we allow an error $\boldsymbol{e}$

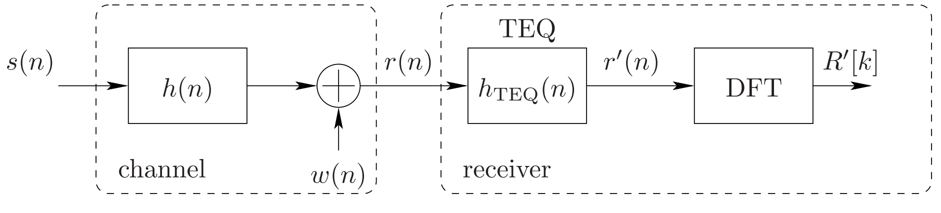


Figure 6: Time-domain equalisation (channel impulse response shortening).

that we then try to minimise. The problem can be formulated as follows: Given

$$\underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\boldsymbol{r}'} = \underbrace{\begin{bmatrix} h(0) & 0 & \dots & 0 \\ h(1) & h(0) & & \vdots \\ \vdots & & \ddots & h(0) \\ h(M') & & & h(1) \\ h(M) & & & \\ & h(M) & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & h(M) \end{bmatrix}}_{\boldsymbol{H}'} \underbrace{\begin{bmatrix} h_{\mathrm{EQ}}(0) \\ h_{\mathrm{EQ}}(1) \\ \vdots \\ h_{\mathrm{EQ}}(K) \end{bmatrix}}_{\boldsymbol{h}_{\mathrm{EQ}}} + \boldsymbol{e}, \tag{19}$$

find the $\boldsymbol{h}_{\mathrm{EQ}}$ that minimises the squared Euclidean norm $||\boldsymbol{e}||_2^2$ of the error $\boldsymbol{e}$. The impulse response that yields this minimum is denoted

$$\boldsymbol{h}_{\mathrm{TEQ}} = \arg\min_{\boldsymbol{h}_{\mathrm{EQ}}} ||\boldsymbol{e}||_2^2. \tag{20}$$

This minimisation has to be repeated for all possible positions of the fixed coefficient 1 in $\boldsymbol{r}'$. Equation (20) formulates the *least squares problem.* Note the difference compared to the MMSE criterion (11): the parameter $\boldsymbol{h}_{\mathrm{EQ}}$ is deterministic; we did not invoke any statistics, there is no expectation in (20). The solution to (20) is given by (→ problem solving session)

$$\boldsymbol{h}_{\mathrm{TEQ}} = \left(\boldsymbol{H}'^{\mathrm{H}}\boldsymbol{H}'\right)^{-1}\boldsymbol{H}'^{\mathrm{H}}\boldsymbol{r}'. \tag{21}$$

4 Channel Estimation

So far, we have tacitly assumed that the receiver knows the channel, which is never the case in practical applications. The receiver needs to estimate the channel. We distinguish pilot-based and decision-direct (blind) channel estimation methods.

4.1 Pilot-based channel estimation

A number of tones, the so-called pilots, in the time-frequency grid are modulation with pseudo-random sequences instead of data. Both the allocation of pilots and the pseudo-random sequences are known to the receiver. Based on this information, the receiver can estimate the channel on these time-frequency grid points and subsequently also estimate the channel in between these points. In order to make the interpolation work, the pilot allocation must fulfil the Nyquist theorem in time and frequency:

- pilot-spacing in time Δt must fulfil $\Delta t < T_{\mathrm{coh}} = 1/B_{\mathrm{doppler}}$
- pilot-spacing in frequency Δf must fulfil $\Delta f < B_{\mathrm{coh}} = 1/T_{\mathrm{multi}}$

The interpolation can be done in one dimension (time or frequency) or in both. Clearly, the quality of the channel estimates depends on the coherence properties of the channel with respect to the choice of the pilot allocation pattern. The pilot allocation pattern is usually agreed upon during the system standardisation. The actual pilot pattern depends on the type of system (continuous transmission or packet based). Figure 7 depicts exemplary patterns.

4.2 Decision-directed channel estimation

Decision direct channel estimation is based on (hopefully) correct data decisions which allows channel estimation using all causal subcarriers. Sometimes, a few pilots are transmitted at the beginning of transmission (typically, in a packet-based scenario) before switching to decision-directed estimation. Clearly, the performance depends again on coherence properties of the channel.

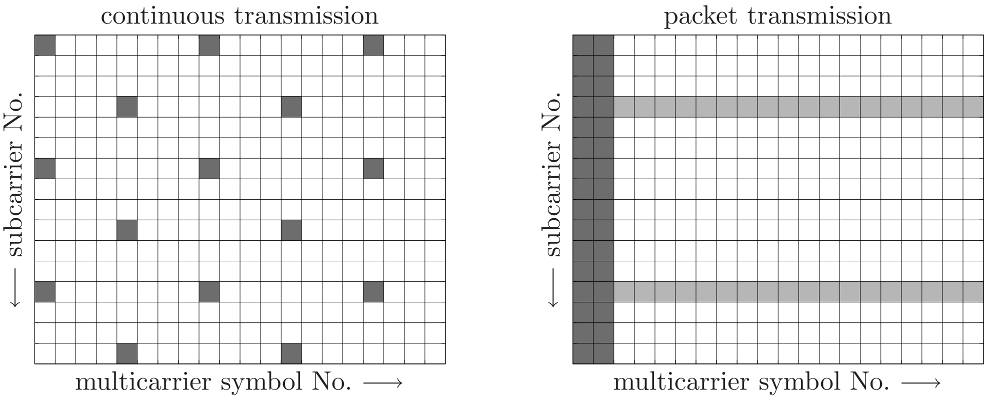


Figure 7: Example of channel-estimation pilot allocation for broadcast (left) and packet-based (right) applications. Tones marked light-grey are used for synchronization.

References

[1] J.B. Anderson, *Digital Transmission Engineering*, IEEE Press, ISBN 0-7803-3457-4, 1999.

[2] J. M. Cioffi, *Advanced Digital Communication*, class reader EE379C, Stanford University, 2005, Class web page `http://www.stanford.edu/class/ee379c/`.

Part IV

Bit- and Power-Loading

1 Introduction

We have already learnt that the multicarrier modulation technique partitions a given channel into N parallel independent subchannels. We have motivated the independence property by the orthogonality of our sinusoidal base functions in time domain. In frequency domain, we noticed that the transforms $R_i(f)$ and $R_j(f)$ of mutually orthogonal base functions $r_i(n)$ and $r_j(n)$, respectively, are "separable" in the sense that $R_i(f)$ is zero at frequency f_j while $R_j(f)$ reaches its maximum at f_j and vice versa. The subchannel No. i has frequency components that leak into subchannel No. j (and vice versa)—there is leakage for all frequencies $f \in [f_j - \Delta f/2, f_j + \Delta f/2)$ except for exactly f_j. The larger N, the smaller the subchannel width Δf and thus the smaller the mutual leakage among the subchannels.

Now let us have a look at the noise. Consider a channel with additive stationary Gaussian noise. The larger the N, the flatter (*i.e*, the more constant) the power spectral density (PSD) of the noise in each subchannel. The inverse Fourier transform of a PSD yields an autocorrelation function. The inverse Fourier transform of a flat PSD corresponds to a Dirac pulse (in continuous-time domain) or a unit-sample (in discrete-time domain), which is the autocorrelation function of an uncorrelated signal. To summarise, if N is sufficiently large, the noise is uncorrelated both in time and over subchannels.

Figure 1 depicts an equivalent model of a multicarrier system including multicarrier modulator, channel and multicarrier demodulator for a large number N of subchannels: N parallel independent complex-valued subchannels. This model reveals why uncoded and straightforward transmission over subchannels yields a bad performance. The worst subchannel, *i.e.*, the subchannel with the highest probability of error, determines the performance of the whole system.

In a single-carrier system, the information is 'spread' over the occupied bandwidth and the equaliser at the receiver 'fishes out' the portions with high signal-to-noise ratio (SNR). In a multicarrier system, we have the opportunity to spectrally allocate power and information—however, we must do it well in order to achieve good performance. Hence, in a multicarrier system, we have to focus on the performance of *all* subchannels by one or both of the following means:

- ensure reliability of transmission through *coding*
- adjust the amount of information transmitted over each subchannel according to quality of the subchannels through *bit-/power-loading*

In order to find a "good" way of distributing information and available power over subchannels, we will investigate the theoretical limit, given by the channel capacity. Besides telling us the limit, the channel capacity will also provide a hint on how to perform the optimal allocation.

2 Channel capacity and waterfilling

Although our multicarrier system consists of parallel complex-valued subchannels, we will start with a set of N real-valued subchannels, depicted in Figure 2, and later extend the results to the complex case.

Chapter written by T. Magesacher.

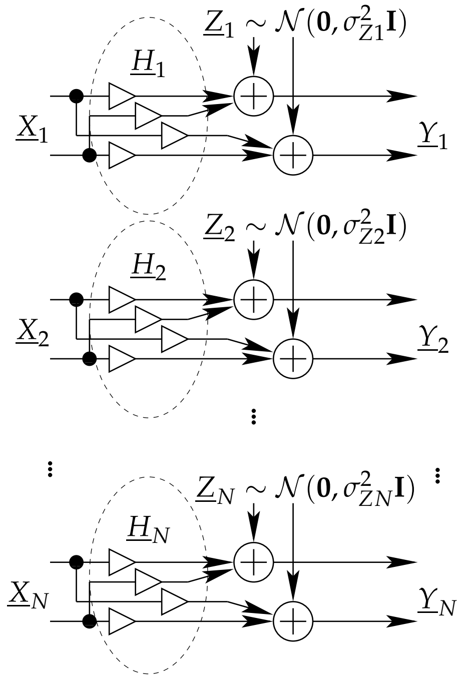

$\underline{Z}_1 \sim \mathcal{N}(\mathbf{0}, \sigma_{Z1}^2 \mathbf{I})$
$\underline{H}_1$
$\underline{X}_1$
$\underline{Y}_1$
$\underline{Z}_2 \sim \mathcal{N}(\mathbf{0}, \sigma_{Z2}^2 \mathbf{I})$
$\underline{H}_2$
$\underline{X}_2$
$\underline{Y}_2$
$\underline{Z}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{ZN}^2 \mathbf{I})$
$\underline{H}_N$
$\underline{X}_N$
$\underline{Y}_N$

Figure 1: Model of multicarrier modulator, channel and multicarrier demodulator for large N: parallel independent complex-valued subchannels.

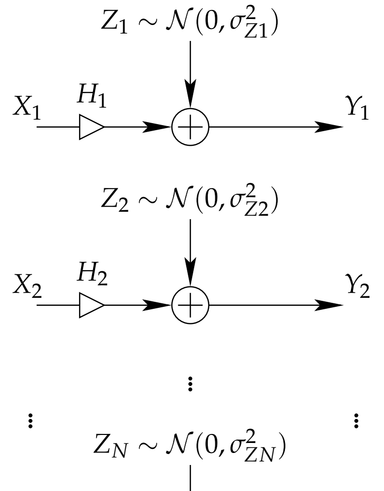

$Z_1 \sim \mathcal{N}(0, \sigma_{Z1}^2)$
X_1
H_1
Y_1
$Z_2 \sim \mathcal{N}(0, \sigma_{Z2}^2)$
X_2
H_2
Y_2
$Z_N \sim \mathcal{N}(0, \sigma_{ZN}^2)$
X_N
H_N
Y_N

Figure 2: N parallel independent real-valued subchannels.

First, we focus on a single subchannel. A real-valued subchannel, shown in Figure 3, is memoryless, discrete-time with a real-valued multiplicative scalar H and additive, real-valued, white, Gaussian noise (AWGN) of zero-mean and variance σ_Z^2:

$$Y = H \cdot X + Z, \qquad H \in \mathbb{R},\ Z \sim \mathcal{N}(0, \sigma_Z^2) \tag{1}$$

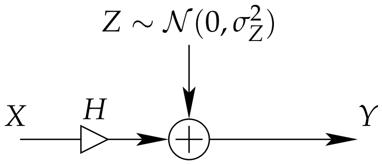


Figure 3: Single subchannel: memoryless, discrete-time, real-valued with real-valued multiplicative scalar H and additive, zero-mean, Gaussian noise with variance σ_Z^2.

The differential entropy of the noise is

$$h(Z) = h(Y|X) = \frac{1}{2}\log_2(2\pi e \sigma_Z^2)$$

The channel input is a continuous random variable X, with *finite power*:

$$\mathrm{E}(X^2) \leq P_X$$

The capacity of this channel is

$$\begin{aligned} C &= \max_{f_X(x):\mathrm{E}(X^2)\leq P_X} I(X;Y) = \max_{f_X(x):\mathrm{E}(X^2)\leq P_X} \{h(Y) - h(Y|X)\} = \max_{f_X(x):\mathrm{E}(X^2)\leq P_X} \{h(Y)\} - h(Z) \\ &= \frac{1}{2}\log_2(2\pi e(H^2\sigma_X^2 + \sigma_Z^2)) - \frac{1}{2}\log_2(2\pi e\sigma_Z^2) \\ &= \frac{1}{2}\log_2\left(1 + \frac{H^2\sigma_X^2}{\sigma_Z^2}\right) \quad \text{bit per channel use} \end{aligned} \tag{2}$$

The capacity-achieving input probability density function (PDF) is Gaussian $X \sim \mathcal{N}(0, \sigma_X^2)$ with $\sigma_X^2 = P_X$ which results in a Gaussian output PDF, $Y \sim \mathcal{N}(0, \sigma_Y^2)$, with $\sigma_Y^2 = H^2\sigma_X^2 + \sigma_Z^2$ and $\sigma_Z^2 = P_Z$. The ratio H^2/σ_Z^2 is a property of the subchannel and determines its quality. In the following, we will refer to H^2/σ_Z^2 as subchannel SNR, although it is not a true signal-to-noise power ratio since there is no signal power involved. However, multiplication of H^2/σ_Z^2 with the transmit signal power yields the true SNR, which somehow justifies our terminology.

Now we move on to N parallel real-valued subchannels shown in Figure 2. Assume we have a constraint on the total power, *i.e.*,

$$\sum_{i=1}^{N} P_{Xi} \leq P_X^{(\mathrm{tot})}$$

and there are no additional constraints for the power values P_{Xi} that are assigned to the individual subchannels. In other words, there is a total power constraint, but there is no power spectral density constraint. Our goal is to find the best distribution of $P_X^{(\mathrm{tot})}$ over the N subchannels (*i.e.*, to find the best values for P_{Xi}) in the sense that the right-hand side of the inequality

$$I(X_1X_2\ldots X_N; Y_1Y_2\ldots Y_N) \leq \frac{1}{2}\sum_{i=1}^{N}\log_2\left(1 + \frac{H_i^2 P_{Xi}}{P_{Zi}}\right) \tag{3}$$

is maximised, *i.e.*,

$$P_{Xi}^{(\mathrm{opt})} = \arg\max_{P_{Xi}} \frac{1}{2}\sum_{i=1}^{N}\log_2\left(1 + \frac{H_i^2 P_{Xi}}{P_{Zi}}\right), \ i = 1, 2, \ldots, N \qquad \text{subject to} \ \sum_{i=1}^{N} P_{Xi} \leq P_X^{(\mathrm{tot})}. \tag{4}$$

The constrained optimisation problem (4) can be solved by maximising the modified cost function

$$\frac{1}{2}\sum_{i=1}^{N}\log_2\left(1 + \frac{H_i^2 P_{Xi}}{P_{Zi}}\right) + \left(\sum_{i=1}^{N} P_{Xi} - P_X^{(\mathrm{tot})}\right)\lambda, \tag{5}$$

which includes the additional term $\left(\sum\limits_{i=1}^{N} P_{Xi} - P_X^{(\text{tot})}\right)\lambda$ ($\lambda \in \mathbb{R}$ is the so called Lagrange multiplier).

In order to determine the extrema of (5), we set the partial derivatives with respect to P_{Xi} equal to 0:

$$\frac{\partial\left(\frac{1}{2}\sum\limits_{i=1}^{N}\log_2\left(1+\frac{H_i^2 P_{Xi}}{P_{Zi}}\right)+\left(\sum\limits_{i=1}^{N}P_{Xi}-P_X^{(\text{tot})}\right)\lambda\right)}{\partial P_{Xi}}=0 \tag{6}$$

$$\frac{H_i^2}{P_{Zi}+H_i^2 P_{Xi}}\frac{1}{2\ln(2)}+\lambda=0 \tag{7}$$

Eventually, we obtain

$$P_{Xi}=\tilde{\lambda}-\frac{P_{Zi}}{H_i^2}, \qquad \text{with } \tilde{\lambda}=-\frac{1}{2\lambda\ln(2)} \tag{8}$$

Since $P_{Xi} \geq 0$ must hold, the optimum power allocation is given by

$$P_{Xi}=\max\{\tilde{\lambda}-\frac{P_{Zi}}{H_i^2},0\}, \tag{9}$$

where the constant $\tilde{\lambda}$ is chosen such that

$$\sum_{i=1}^{N}\max\{\tilde{\lambda}-\frac{P_{Zi}}{H_i^2},0\}=P_X^{(\text{tot})}. \tag{10}$$

Equations (9) and (10) describe the so called waterfilling solution. We have a bowl whose bottom has a profile shaped according to $\frac{P_{Zi}}{H_i^2}$. We fill this bowl with water, where the amount of water corresponds to $P_X^{(\text{tot})}$. The level of the water corresponds to $\tilde{\lambda}$ and the resulting filling profile corresponds to P_{Xi}.

Now we consider the set of N complex-valued subchannels. Each complex-valued subchannel can be modelled as

$$\underline{Y}=\underline{H}\cdot\underline{X}+\underline{Z}, \qquad \underline{H}\in\mathbb{C}, \begin{bmatrix}\text{Re}(Z)\\\text{Im}(Z)\end{bmatrix}\sim\mathcal{N}(\mathbf{0},\sigma_Z^2\boldsymbol{I}), \tag{11}$$

or, in matrix notation,

$$\underbrace{\begin{bmatrix}\text{Re}(Y)\\\text{Im}(Y)\end{bmatrix}}_{\boldsymbol{y}}=\underbrace{\begin{bmatrix}A\cos\phi & -A\sin\phi\\A\sin\phi & A\cos\phi\end{bmatrix}}_{\boldsymbol{H}}\underbrace{\begin{bmatrix}\text{Re}(X)\\\text{Im(X)}\end{bmatrix}}_{\boldsymbol{x}}+\underbrace{\begin{bmatrix}\text{Re}(Z)\\\text{Im(Z)}\end{bmatrix}}_{\boldsymbol{n}}, \tag{12}$$

where the complex-valued channel coefficient is $\underline{H}=Ae^{j\phi}, A\geq 0$. Singular value decomposition of $\boldsymbol{H}$ yields

$$\boldsymbol{H}=\boldsymbol{USV}, \tag{13}$$

where $\boldsymbol{S}$ is a diagonal matrix whose diagonal entries are the non-negative square roots of the eigenvalues of $\boldsymbol{HH}^{\text{H}}$, which can be easily computed:

$$\boldsymbol{HH}^{\text{H}}=\begin{bmatrix}A^2 & 0\\0 & A^2\end{bmatrix} \tag{14}$$

Thus, $\boldsymbol{S}$ is given by

$$\boldsymbol{S}=\begin{bmatrix}A & 0\\0 & A\end{bmatrix}. \tag{15}$$

Note that A is also the absolute value of the eigenvalues $\lambda_1=A\cos\phi+jA\sin\phi$ and $\lambda_2=A\cos\phi-jA\sin\phi$ of $\boldsymbol{H}$:

$$|\lambda_1|=|A\cos\phi+jA\sin\phi|=|\lambda_2|=|A\cos\phi-jA\sin\phi|=A. \tag{16}$$

Since both $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary ($\boldsymbol{UU}^{\text{H}}=\boldsymbol{I}$, $\boldsymbol{VV}^{\text{H}}=\boldsymbol{I}$), the modified channel model

$$\underbrace{\boldsymbol{U}^{\text{H}}\boldsymbol{y}}_{\tilde{\boldsymbol{y}}}=\boldsymbol{S}\underbrace{\boldsymbol{Vx}}_{\tilde{\boldsymbol{x}}}+\underbrace{\boldsymbol{U}^{\text{H}}\boldsymbol{n}}_{\tilde{\boldsymbol{n}}}, \tag{17}$$

obtained from (12) by multiplication with $\boldsymbol{U}^{\mathrm{H}}$ from the left, has the following properties:

- The noise vector $\tilde{\boldsymbol{n}}$ has the same mean vector and the same covariance matrix as $\boldsymbol{n}$
- The signal vector $\tilde{\boldsymbol{x}}$ has the same mean vector and the same covariance matrix as $\boldsymbol{x}$

Consequently, (12) and (17) are equivalent in the sense that they have the same channel capacity. The model (17) describes nothing else but a system of two real-valued independent channels, *i.e.*, precisely the system shown in Figure 2 for $N = 2$. In order to obtain the capacity of (17) we need to water-fill over the two real-valued subchannels. However, since $|\lambda_1| = |\lambda_2| = A = |\underline{H}|$, we know that equal distribution of the available power over both subchannels is optimal. The channel capacity of a complex-valued subchannel described by (11) is thus given by

$$C = \log_2 \left(1 + \frac{|\underline{H}|^2 P_X}{P_Z}\right) \text{ bit per channel use.} \tag{18}$$

We conclude that a complex-valued subchannel has, due to its particular structure, the property that splitting the available power into two equal halves for inphase-component $\mathrm{Re}(X)$ and quadrature component $\mathrm{Im}(X)$, respectively, is the optimal power allocation. This finding has two important implications:

- Rotationally-symmetric QAM-constellations inherently fulfill this requirement of equal power-distribution over the two dimensions.
- We can use the waterfilling result given by (9) and (10) to perform the allocation over complex-valued subchannels by replacing the coefficients H_i of the real-valued subchannels with the absolute values $|\underline{H}_i|$ of the complex-valued subchannels. Equations (9) and (10) tell us how to split the power across the complex-valued subchannels. Based on our investigation of a complex-valued subchannel, we then distribute this power equally over the two dimensions.

3 Gap analysis

A useful approach to incorporate quality-of-service constraints when carrying out bit/power-loading is the so called "gap analysis". The per-subchannel capacity

$$c_i = \log_2 \left(1 + \frac{|\underline{H}_i|^2 P_{X\,i}}{P_{Z\,i}}\right), \tag{19}$$

is replaced by

$$b_i = \log_2 \left(1 + \frac{|\underline{H}_i|^2 P_{X\,i}}{P_{Z\,i}\Gamma}\right), \tag{20}$$

where Γ denotes the "SNR gap". The SNR gap Γ is chosen such that uncoded transmission achieves a desired symbol error rate P_{s}. For the practically relevant case of QAM modulation, Γ is independent of the constellation size (for example, $\Gamma = 8.8\,\mathrm{dB}$ for $P_{\mathrm{s}} = 10^{-6}$).

In a practical transmission scheme, performance objective and constraints determine the way the loading is done. *Rate-adaptive* loading (also referred to as fixed-margin loading) aims at maximizing the total datarate with under a total power constraint:

$$\begin{array}{ll} \underset{P_{X\,i}}{\text{maximize}} & \sum b_i \\ \text{subject to} & \sum P_{X\,i} = \text{const} \end{array} \tag{21}$$

Often, the desired datarate is fixed (for example, through constraints imposed by the modulation/coding scheme). *(Performance) Margin adaptive* loading (also referred to as power-minimization loading or margin-maximization loading) aims at minimizing the total power under a total datarate constraint:

$$\begin{array}{ll} \underset{b_i}{\text{minimize}} & \sum P_{X\,i} \\ \text{subject to} & \sum b_i = \text{const} \end{array} \tag{22}$$

Performance margin refers to the amount by which the SNR can be decreased while maintaining P_{s} at a given Γ.

4 Waterfilling/loading algorithm

In the following, we discuss waterfilling algorithms that solve the rate-adaptive loading problem. The algorithms can be modified to solve the margin-adaptive loading problem. Solving the waterfilling problem defined by equations (9) and (10) is equivalent to solving the following set of $N+1$ linear equations

$$\begin{aligned} P_{X1} =& \tilde{\lambda} - \frac{P_{Z1}}{H_1^2}\Gamma \\ P_{X2} =& \tilde{\lambda} - \frac{P_{Z2}}{H_2^2}\Gamma \\ &\vdots \\ P_{XN} =& \tilde{\lambda} - \frac{P_{ZN}}{H_N^2}\Gamma \\ P_{X1} + P_{X2} + \ldots + P_{XN} =& P_X^{(\mathrm{tot})} \end{aligned} \tag{23}$$

under the constraint $P_{Xi} \geq 0, i = 1, \ldots, N$. Incorporating this constraint is simple:

Step 1: Set up a system of $N+1$ equations.

Step 2: Solve the system of equations.

Step 3: In case all power values are non-negative, the problem is solved. In case $P_{Xi} < 0$ for a set of indices $i \in \mathcal{I} = \{i_1, \ldots, i_K\}$, set up a new system of equations: start from the previous system, omit the K equations $P_{Xi} = \tilde{\lambda} - \frac{P_{Zi}}{H_i^2}\Gamma, i \in \mathcal{I}$ and omit the power values $P_{Xi}, i \in \mathcal{I}$ in the last equation, *i.e.*,

$$\sum_{i \notin \mathcal{I}} P_{Xi} = P_X^{(\mathrm{tot})},$$

which results in a new system of $N+1-K$ equations. Proceed with Step 2.

Although easy to grasp, the matrix-based version of the waterfilling algorithm requires in the worst case N matrix inversions. An alternative implementation is based on the observation that the waterfilling level for K subchannels of an arbitrary set $\mathcal{I} = \{i_1, \ldots, i_K\}$, in case *only* these subchannels are used, is given by

$$\tilde{\lambda} = \frac{P_X^{(\mathrm{tot})} + \Gamma(\frac{P_{Zi_1}}{H_{i_1}^2} + \ldots + \frac{P_{Zi_K}}{H_{i_K}^2})}{K}, \tag{24}$$

which can be readily derived by inserting the K equations

$$\begin{aligned} P_{Xi_1} =& \tilde{\lambda} - \Gamma\frac{P_{Zi_1}}{H_{i_1}^2} \\ P_{Xi_2} =& \tilde{\lambda} - \Gamma\frac{P_{Zi_2}}{H_{i_2}^2} \\ &\vdots \\ P_{Xi_K} =& \tilde{\lambda} - \Gamma\frac{P_{Zi_K}}{H_{i_K}^2} \end{aligned} \tag{25}$$

into the last equation

$$\sum_{k=1}^{K} P_{Xi_k} = P_X^{(\mathrm{tot})}$$

of the system (23). The second vital observation is the following: if subchannel No. i, whose subchannel SNR is $\Gamma\frac{H_i^2}{P_{Zi}}$, is not used because its subchannel SNR is too low, all subchannels with lower SNR are omitted as well. This suggests that we should consider the subchannels in sorted order. The corresponding algorithm can be formulated as follows:

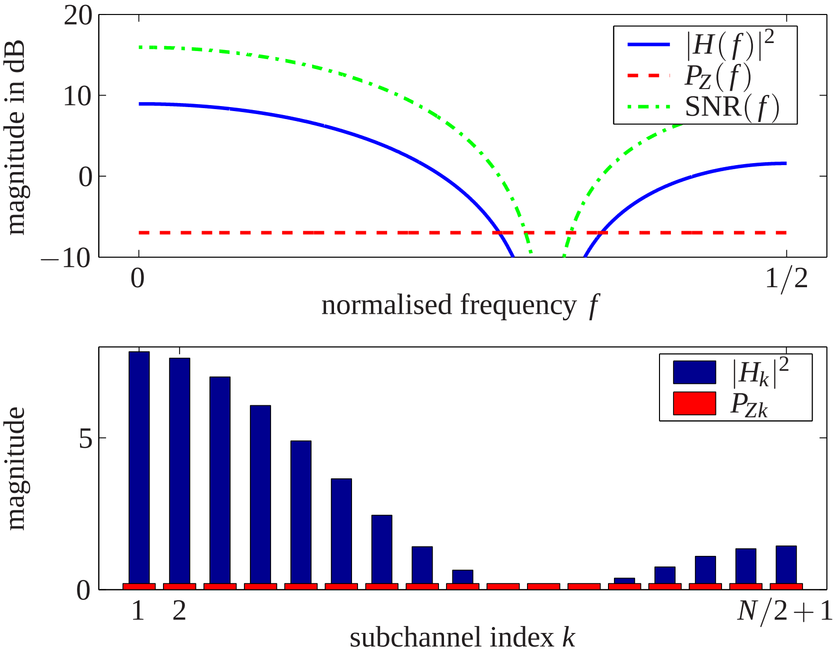


Figure 4: Waterfilling example: squared magnitude response $|H(f)|^2$ of the channel $H(z) = 1 + 0.8z^{-1} + z^{-2}$, PSD $P_Z(f)$ of the frequency-flat (white) noise with zero-mean and variance 0.2 per subchannel ($N = 32$) and the resulting channel signal-to-noise power ratio SNR$(f) = |H(f)|^2/P_Z(f)$.

Step 1: Sort the subchannels such that $\frac{H_{i_1}^2}{P_{Zi_1}} \geq \frac{H_{i_2}^2}{P_{Zi_2}} \geq \ldots \geq \frac{H_{i_N}^2}{P_{Zi_N}}$; set $k = 0$ and $\lambda_{\sum} = P_X^{(\mathrm{tot})}$

Step 2:: Set $k = k+1$ and compute the waterfilling level for the set of users $\{i_1, \ldots, i_k\}$: $\lambda_{\sum} = \lambda_{\sum} + \Gamma \frac{P_{Zi_k}}{H_{i_k}^2}$, $\lambda = \lambda_{\sum}/k$

Step 3:: If the power of the subchannel with the lowest subchannel SNR is still larger than zero, *i.e.*, if $\lambda - \Gamma \frac{P_{Zi_k}}{H_{i_k}^2} > 0$, go to Step 2. Otherwise proceed with Step 4.

Step 4:: Undo the increment corresponding to subchannel No. i_k: $\lambda_{\sum} = \lambda_{\sum} - \Gamma \frac{P_{Zi_k}}{H_{i_k}^2}$, $\lambda = \lambda_{\sum}/(k-1)$; Compute the subchannel power values:

$$P_{Xi_j} = \begin{cases} \lambda - \Gamma \frac{P_{Zi_j}}{H_{i_j}^2}, & j = 1, \ldots, k-1 \\ 0, & \text{otherwise} \end{cases}$$

Let us illustrate the waterfilling solution by an example. Consider the channel $H(z) = 1 + 0.8z^{-1} + z^{-2}$ with additive zero-mean uncorrelated Gaussian noise (the noise PSD is frequency-flat). The noise variance per subchannel is 0.2. Figure 4 depicts the squared magnitude response, the noise PSD and the resulting channel SNR. Our multicarrier system has $N = 32$ subchannels and the baseband multiplex should be real-valued. Thus, there are 17 subchannels over which we have full control, the data on the remaining 15 subchannels is determined by the Hermitian symmetry condition. We know the channel at the transmitter. Waterfilling with a total power $P_X^{(\mathrm{tot})} = 1$ yields the solution depicted in the upper plot of Figure 5. The power filling is proportional to the subchannel SNR and the worst subchannels are omitted. The lower plot of Figure 5 shows the waterfilling solution for $P_X^{(\mathrm{tot})} = 4$. With $P_X^{(\mathrm{tot})} = 4$, we are using some of the subchannels that we have omitted in the solution for $P_X^{(\mathrm{tot})} = 1$. The waterfilling solution depends both on the channel and on the total available power. Note that subchannels No. 1 and No. $N/2+1$ (assuming

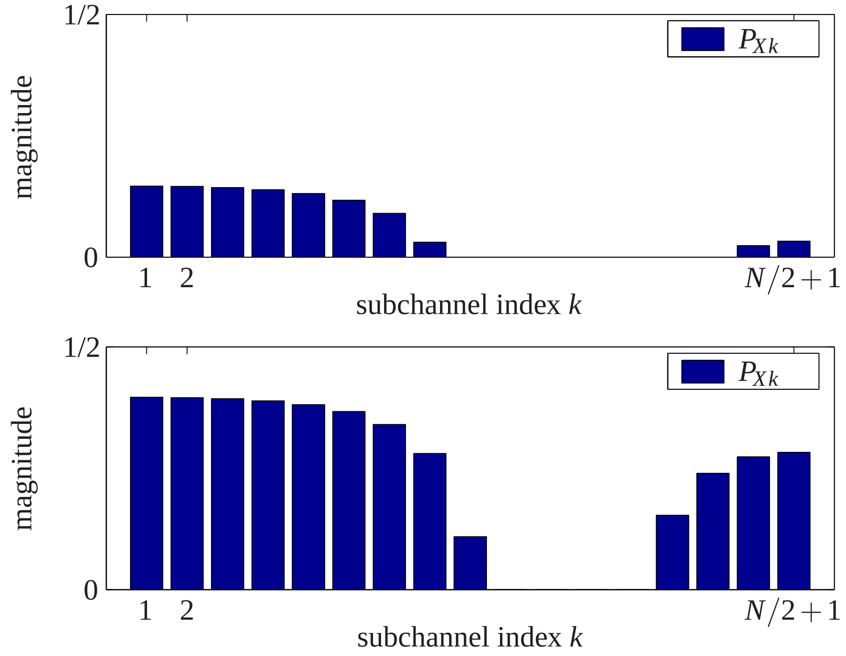


Figure 5: Waterfilling example: the upper plot shows the solution for $P_X^{(\mathrm{tot})} = 1$, the lower plot shows the solution for $P_X^{(\mathrm{tot})} = 4$.

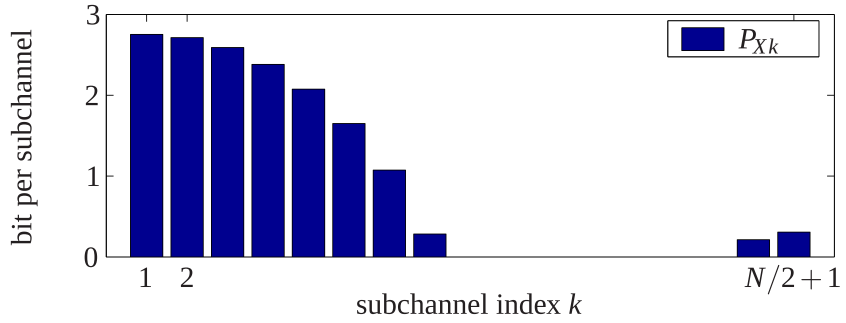


Figure 6: Waterfilling example: bit loading corresponding to the energy loading for $P_X^{(\mathrm{tot})} = 1$.

even N) must be used with real-valued data in order to assure a real-valued baseband multiplex, *i.e.*, pulse amplitude modulation (PAM) must be used on these subchannels rather than QAM. In practical DMT systems, these subchannels often exhibit low subchannel-SNR values due to filters, DC-blocking components, *etc.* and are thus not used in the first place.

5 Discrete power/bit-loading

Bit loading in practice often requires an integer number of bits per subchannel. The waterfilling solution will, in general, return a non-integer number of bits per subchannel. For our example, the resulting bit loading is shown in Figure 6. Straightforward rounding of the number of bits to the nearest integer would yield a worse bit error performance on subchannel where we round upwards and would result in a waste of energy on subchannels where we round downwards. The finite granularity issue gave rise to the development of several special algorithms. The Levin-Campello algorithm finds the optimal solution.

5.1 Levin-Campello algorithm

The most well-known ones are the so called Chow algorithm and the Levin-Campello algorithm. The details of these algorithms are out of scope of this text.

The "greedy"-principle postulates cost minimization to achieve a certain sub-goal. The idea applied to bit loading can be summarized as follows:

Step 1: find subcarrier k that requires least energy, denoted ϵ_k, to increase b_k by 1

Step 2: find subcarrier ℓ that saves most energy, denoted ϵ_ℓ, when decreasing b_ℓ by 1

Step 3: if more energy is saved than spent ($\epsilon_\ell > \epsilon_k$), move a bit from subcarrier ℓ to subcarrier k and goto Step 1

Every iteration improves the energy-efficiency while keeping the total number of bits constant and eventually yields a so-called *efficient distribution*.

Two additional concepts are *energy-tightness* and *bit-tightness* of a bit distribution $\boldsymbol{b}$. A distribution $\boldsymbol{b}$ is *energy-tight* if no additional bit can be added without violating the energy constraint. A distribution $\boldsymbol{b}$ is *bit-tight* if it achieves the target number of bits.

The Levin-Campello algorithm for rate-adaptive loading can be summarized as follows:

Step 1: choose any $\boldsymbol{b}$

Step 2: make the result of Step 1 efficient

Step 3: make the result of Step 2 energy-tight

The Levin-Campello algorithm for margin-adaptive loading can be summarized as follows:

Step 1: choose any $\boldsymbol{b}$

Step 2: make the result of Step 1 efficient

Step 3: make the result of Step 2 bit-tight

6 Practical aspects of bit and power loading

As mentioned already, there are two principally different types of constraints:

- constraint on total power (transmitted on all subchannels)
- constraint on power spectral density (PSD), which is equivalent to a per-subchannel power constraint if the number of subchannels is large

The waterfilling solution based on channel capacity discussed above yields the solution to maximum data rate if there is a constraint on the total power and no constraint on the PSD. In practice, however, there is often a PSD limit as well. In the absence of such a limit, it would theoretically be possible to cram all the available transmit power into a single subchannel (using a huge constellation)—in case the waterfilling solution would suggest it (*i.e.*, in case this single subchannel has a performance which is just so much better than all the others that it does not make sense to use the others). Consequently, the spectrum would exhibit a PSD peak at the subchannel's centre frequency (and the corresponding sidelobes). Such a peak is likely to disturb other systems (either directly through coupling or indirectly through its out-of-band emissions caused by its sidelobes). In case there is a PSD constraint, *i.e.*, the maximum power per subchannel is given, and there is a total power constraint large enough such that every subchannel can be used with its maximum power, there is no need for waterfilling. The power on each subchannel and thus the signal-to-noise ratio on each subchannel is given. The problem then changes into finding an adequate constellation size for each subchannel such that a prescribed probability of error can be guaranteed.

Often, provision of the maximum rate implies that the rate is variable. Depending on the channel state and the interference present, the rate may have to be updated from time to time and there is no guarantee that the 'maximum' rate can be delivered. In practice, often a minimum rate is required to deliver certain services. In such a case, an algorithm for bit and power loading could focus on minimising the required power per symbol to deliver the required rate.

Note that the capacity calculation assumes Gaussian signalling. With increasing QAM constellation size, this approximation becomes more and more accurate. An extension of the waterfilling that does not assume Gaussian signalling is the so-called *mercury/waterfilling*.

Finally, we should remind ourselves that any kind of waterfilling, bit-loading or power-loading requires knowledge of the channel at the transmitter.

Part V

Coding for Multicarrier Modulation

1 Introduction

Multicarrier modulation efficiently combats dispersion in time by dividing a wideband channel into many parallel narrowband subchannels and separating blocks by guard intervals (cyclic perfixing, zero padding). If the number of subchannels is sufficiently large, each subchannel can be regarded as frequency-flat. The attenuation the substreams experience can vary greatly from subchannel to subchannel. Moreover, in a fading channel, these attenuations vary over time. Transmitting over subchannels with equal power and equal constellations, the weakest subchannel would exhibit the largest amount of symbols or bits received in error and thus determine the overall performance. Coding helps to protect the weakest subcarrier.

The ultimate goal of a communication system is to transmit information *efficiently* and *reliably* from a source to a destination. On its way to the destination, often also referred to as sink, the information signal is transmitted over a physical channel, which, as we learnt from the previous lectures, introduces various disturbances. As a result, what reaches the receiver is a corrupted, distorted version of what the transmitter actually sent. When the receiver is not able to recover the transmitted message, we say that an *error event* occurred. In order to protect the information signal from errors events, i.e., in order to minimise the probability that an error occurs, we apply *channel coding*, also known as *error correction coding*. As we will see in the sequel, *a channel code* adds smart redundancy to the information data, such that the receiver can easily *detect* and, more importantly, *correct* possible errors that occurred. The process of introducing the smart redundancy (i.e. structure) in the information data is called *encoding*, and it is performed at the transmitter immediately after data compression, that is, in case of OFDM, *before* modulation, IDFT, and cyclic prefix extension. The dual process at the receiver is *decoding* – recovering the original data, and it is performed *after* cyclic prefix removal, DFT, channel equalisation and demodulation.

To investigate channel coding in more detail, we consider the model of a communication system depicted in Figure 1. The block labelled ”equivalent source” is the equivalent binary source obtained by applying (optimum) data compression (source encoding) to the physical source of data; the ”channel” block is an equivalent digital channel that *includes* the whole chain: modulation, CP extension, IDFT, physical channel, CP removal, DFT, equalisation, demodulation.

Chapter written by M. Lončar and T. Magesacher.

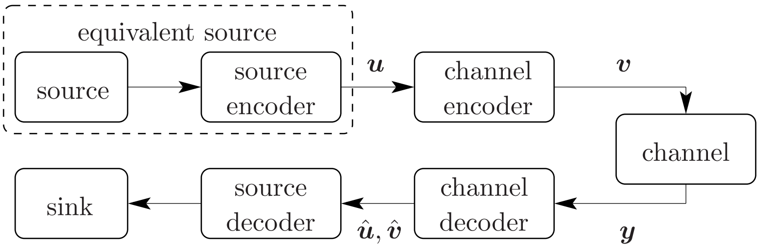


Figure 1: Model of communication system.

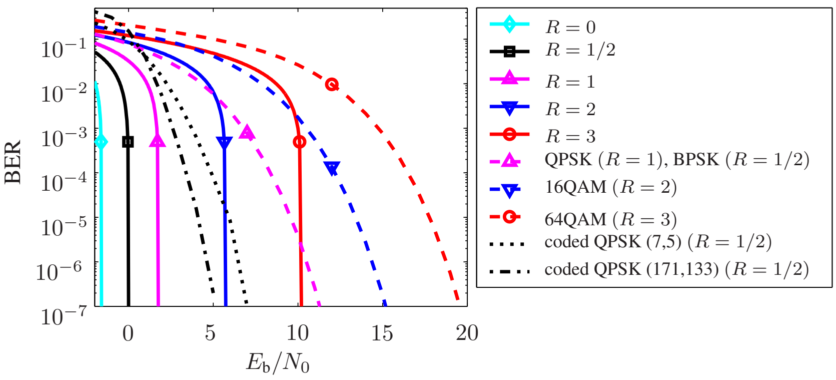


Figure 2: BER versus E_b/N_0 for AWGN channel. Solid lines: Shannon bound for various rates. Dashed lines: uncoded modulation. Dotted line: rate-1/2 (7,5)-code with QPSK modulation. Dashed-dotted line: rate-1/2 (171,133)-code with QPSK modulation.

The performance of a coding scheme is assessed in terms of the achieved bit error rate (BER) for a given SNR E_b/N_0. E_b/N_0 is a measure for the received energy per information bit (normalized by the noise power spectral density N_0; note that E_b/N_0 is indeed a signal-to-noise power ratio, where the signal power $P_s = E_b/T$, the noise power $P_n = N_0B$, and symbol rate T and bandwidth B are related as $B = 1/T$). A code improves the reliability of transmission by adding redundancy. R denotes the number of information bits per symbol. Figure 2 depicts the BER versus E_b/N_0 of various schemes for the AWGN channel. The lower the BER for a given SNR E_b/N_0 (or equivalently, the lower the required SNR E_b/N_0 to achieve a certain BER), the better the scheme. The Shannon limit (solid lines) provides an ultimate performance bound for coding schemes. There exists no coding scheme with code rate R that can operate at a (E_b/N_0, BER)-point left of the corresponding Shannon-limit curve. The smaller R (i.e., the more redundancy is added), the smaller is the E_b/N_0 required to achieve a given BER. On the other hand, the higher R, the higher is the number of transmitted information bits per symbol (or, equivalently, per $1/T$ seconds). When designing a transmission system, R is determined by the signal-to-noise power ratio, i.e., by the channel's noise and attenuation and by the admissible transmit power (i.e, by the SNR). The name of the game is to design coding/modulation schemes that allow to operate as closely as possible to the Shannon limit for a given R.

2 Block Codes

2.1 Definitions

In block coding, the output of the source is parsed into blocks (sequences) of K information symbols u_k, $1 \le k \le K$. We denote them as *row* vectors $\boldsymbol{u} = [u_1\, u_2 \dots u_K]$. In general, the source uses a q-ary alphabet, i.e., $u_i \in \{0, 1, \dots, q-1\}$, $1 \le i \le K$. Thus, the number of possible information sequences (or "messages") is $M = q^K$. If $q = 2$, i.e., if the information symbols u_k are binary, we call them *bits.*

A channel *code* $\mathcal{C}$ is a *set* of M distinct codewords $\mathcal{C} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_M\}$. Codewords are sequences of code symbols of length N, and we denote them as row vectors $\boldsymbol{v} = [v_1\, v_2 \dots v_N]$. Code symbols v_i, $1 \le i \le N$, in general, belong to a q-ary alphabet, $v_i \in \{0, 1, \dots, q-1\}$, where $q \ge 2$. When $q = 2$, we have a *binary* block code. However, even if the v_i are binary, we call them *code symbols* (or, shortly, symbols), and not bits, in order to distinguish them from the information bits.

The *code rate* R of a block code of length N with M different codewords is defined as a ratio

$$R = \frac{\log_q M}{N} = \frac{K}{N}, \tag{1}$$

and it describes how much redundancy is added by coding: lower rate means we have introduced more redundancy, i.e., one information symbol is represented with more code symbols. The rate defined in this way[1] satisfies $0 \leq R \leq 1$.

An *encoder* is a device that performs *mapping* from information sequences $\boldsymbol{u}$ to codewords $\boldsymbol{v} \in \mathcal{C}$. We consider only *one-to-one* mappings: every information sequence is mapped onto a different codeword. Note that there are in total $M(M-1) \cdot ... \cdot 1 = M!$ possible ways to map M information sequences onto M codewords of a given code. In other words, every code has many different encoders!

The *decoder* is a device that estimates which codeword was transmitted, i.e., it performs mapping from the received sequence $\boldsymbol{y}$ to the codeword estimate $\hat{\boldsymbol{v}}$, or equivalently, to the corresponding information sequence $\hat{\boldsymbol{u}}$. This mapping is denoted with a function $g(\boldsymbol{y})$ whose values are from the set of message indexes $\{1, 2, ..., M\}$. Thus, if, for a given received sequence $\boldsymbol{y}$, we have that $g(\boldsymbol{y}) = i$, $1 \leq i \leq M$, this means that whenever this sequence $\boldsymbol{y}$ is received, the decoder decides that $\hat{\boldsymbol{v}} = \boldsymbol{v}_i$ was transmitted.

If the channel output alphabet is of size J, there are J^N possible received sequences $\boldsymbol{y}$. For each of them, there are M possible choices for the decision function $g(\boldsymbol{y})$. Thus, there are in total M^{J^N} possible decoders for the given channel and a given code! Naturally, among all those decoders, there are many "stupid" ones that will not help us correct any errors. Therefore, we will now formulate the optimal rule how to design a good decoder.

2.2 Optimum decoder

Before proceeding any further, we will briefly agree on the notation that will be used in the following section. In probability theory, random variables are always denoted with capital letters, and their realisations with small letters. Since the source produces random information sequences (messages), we will denote its output with $\boldsymbol{U} = [U_1\, U_2 \ldots U_K]$ which denotes a K-dimensional random variable that can take any of the $M = q^K$ possible realisations $\boldsymbol{u}$. Let W denote the index of the transmitted information sequence, that is, W is a discrete random variable whose realisations are $i \in \{1, 2, ..., M\}$. Similarly, an encoder output is a random codeword $\boldsymbol{V}$, whose possible realisations are $\boldsymbol{v} \in \mathcal{C}$ and the channel output is an N-dimensional random variable $\boldsymbol{Y}$, whose realisations are $\boldsymbol{y}$. Finally, the decoder decision rule is determined by $\widehat{W} = g(\boldsymbol{Y})$, which is a random variable with possible realisations $i \in \{1, 2, ..., M\}$. Furthermore, we have the following notation for the probabilities

- $p(\boldsymbol{v}) = \Pr(\boldsymbol{V} = \boldsymbol{v})$ – *a priori* probability that the encoder transmits a codeword $\boldsymbol{v}$
- $p(\boldsymbol{y}|\boldsymbol{v}) = \Pr(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{V} = \boldsymbol{v})$ – *likelihood* of a codeword $\boldsymbol{v}$ is the conditional probability that $\boldsymbol{y}$ is observed at the receiver, given that $\boldsymbol{v}$ was transmitted (this probability is specified by the channel).
- $p(\boldsymbol{v}|\boldsymbol{y}) = \Pr(\boldsymbol{V} = \boldsymbol{v}|\boldsymbol{Y} = \boldsymbol{y})$ – *a posteriori* probability of a codeword $\boldsymbol{v}$ is the conditional probability that $\boldsymbol{v}$ was transmitted, given that $\boldsymbol{y}$ is received (i.e. it is the probability of $\boldsymbol{v}$ *after* observing $\boldsymbol{y}$).

The *Bayes' formula* provides a connection of these three probabilities

$$p(\boldsymbol{v}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{v})p(\boldsymbol{v})}{p(\boldsymbol{y})}. \tag{2}$$

Due to the one-to-one mapping $\boldsymbol{u} \to \boldsymbol{v}$ specified by the encoder, in all the above expressions, we can replace $\boldsymbol{v}$ with the corresponding $\boldsymbol{u}$, without changing the values of the probabilities, and thus we would respectively define *a priori*, *likelihood* and *a posteriori* probabilities of the information sequence $\boldsymbol{u}$ that is mapped onto the codeword $\boldsymbol{v}$.

If the estimated information sequence $\widehat{\boldsymbol{U}}$ is not equal to the sequence $\boldsymbol{U}$ that was actually transmitted (or, equivalently, if $\widehat{\boldsymbol{V}} \neq \boldsymbol{V}$, or, $\widehat{W} \neq W$), we say that a *block* (or a *word*) error occurred. The probability

[1] Note that the *information rate* in bits per symbol, i.e., in bits per channel use is defined as $R_b = \log_2 M/N = K \log_2 q/N$, and it tells us how much information is transferred through the channel.

of a block error is denoted by P_B and it equals

$$\begin{aligned} P_B &= \Pr(\boldsymbol{U} \neq \widehat{\boldsymbol{U}}) = \Pr(\boldsymbol{V} \neq \widehat{\boldsymbol{V}}) = \Pr(W \neq \widehat{W}) \\ &= 1 - \Pr(W = \widehat{W}) = 1 - \Pr(W = g(\boldsymbol{Y})) \\ &= 1 - \sum_{\boldsymbol{y}} \Pr(W = g(\boldsymbol{Y}) | \boldsymbol{Y} = \boldsymbol{y}) \Pr(\boldsymbol{Y} = \boldsymbol{y}) = \\ &= 1 - \sum_{\boldsymbol{y}} p(g(\boldsymbol{y})|\boldsymbol{y}) p(\boldsymbol{y}) = 1 - \sum_{\boldsymbol{y}} p(\boldsymbol{y}|g(\boldsymbol{y})) p(g(\boldsymbol{y})) = \\ &= 1 - \sum_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{v}_{g(\boldsymbol{y})}) p(\boldsymbol{v}_{g(\boldsymbol{y})}). \end{aligned} \tag{3}$$

The *optimal* decoder is the decoder that *minimises* the block error probability. From (3) it follows that minimising P_B is equivalent to *maximising* $p(\boldsymbol{y}|\boldsymbol{v}_{g(\boldsymbol{y})})p(\boldsymbol{v}_{g(\boldsymbol{y})})$ *for every* $\boldsymbol{y}$, and from Bayes' formula we see that this is equivalent to maximising the *a posteriori* probability of $\boldsymbol{v}$ (note that $p(\boldsymbol{y})$ is a scaling factor that does not depend on $\boldsymbol{v}$ and thus has no influence on the maximisation). Thus, we conclude that the optimum decoder is a *maximum a posteriori* (MAP) decoder: for every received sequence, it chooses the $\hat{\boldsymbol{v}}$ that has highest *a posteriori* probability:

$$\hat{\boldsymbol{v}}_{\mathrm{MAP}} = \arg\min_{\boldsymbol{v}\in\mathcal{C}} P_B = \arg\max_{\boldsymbol{v}\in\mathcal{C}} (p(\boldsymbol{y}|\boldsymbol{v})p(\boldsymbol{v})) = \arg\max_{\boldsymbol{v}\in\mathcal{C}} p(\boldsymbol{v}|\boldsymbol{y}) \tag{4}$$

In order to perform MAP decoding, we need to know the *a priori* probabilities $p(\boldsymbol{v})$, or, equivalently, $p(\boldsymbol{u})$, which are determined by the statistics of the information source. For every different source, a MAP decoder is different, "tuned" to that source.

If all the information sequences are equiprobable, i.e., if $p(\boldsymbol{v}) = p(\boldsymbol{u}) = 1/M$, then maximising *a posteriori* probabilities is equivalent to maximising likelihoods $p(\boldsymbol{y}|\boldsymbol{v})$.

The decoder that minimises the block error probability when the codewords are equally likely is the *maximum likelihood* (ML) decoder:

$$\hat{\boldsymbol{v}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{v}\in\mathcal{C}} p(\boldsymbol{y}|\boldsymbol{v}) \tag{5}$$

The ML detector can be applied for any source. If the source outputs are equiprobable, it coincides with the MAP decoder. In case the output is non-uniformly distributed, the ML decoder does not yield the minimum P_B. However, in practice, we always assume that the source sequences are equiprobable, and then, the ML detector is optimum (this assumption is justified by the fact that a perfect data compression algorithm applied on an arbitrary source yields equiprobable sequences).

In the sequel, we will turn our attention to a special class of block codes - namely, *linear* block codes. They are of particular practical and theoretical interest since they posses many useful structural properties which enable easier analysis. More importantly, the *encoders* and *decoders* for linear codes are very simple to realise. This is why all existing communication systems utilise only linear codes.

3 Linear Block Codes

3.1 Definition

Definition: *A linear (N, K) q-ary block code of length N and with q^K codewords is a K-dimensional subspace of the vector space $GF(q)^N$ of all N-tuples over the field $GF(q)$.*

This formal definition might seem somewhat complicated, so we will explain what hides behind it. First, GF denotes *Galois Field*, which is simply a field with a finite number of elements. By *field* we mean an algebraic structure - for example, $\mathbb{R}$ is a field of real numbers (and it is not a Galois field). A binary field $\{0, 1\}$ is the smallest finite field $GF(2)$. Roughly speaking, a field is a set of elements in which we can perform operations of addition, subtraction, multiplication and division without leaving the set; and addition and multiplication satisfy commutative, associative and distributive laws. Without proving it, we mention here that a set with q elements can constitute a field only if q is a *prime* number, or a power of a prime number p, i.e., $q = p^m$, $m \geq 1$. If q is prime, all operations in $GF(q)$ are performed modulo-q and if $q = p^m$, all operations in $GF(p^m)$ are performed modulo-p.

When codesymbols v_i are q-ary symbols from $GF(q)$, then the N-symbol long codewords are vectors from the field $GF(q)^N$. Moreover, we say that this field constitutes a vector *space* over $GF(q)$, which means that any linear combination of vectors $GF(q)^N$, scaled by numbers from $GF(q)$ yields again a vector from $GF(q)^N$. Thus, we can perform multiplication by scalars and vector addition without leaving the vector space.

If we pick $M = q^K$ vectors (codewords) from $GF(q)^N$ such that this set preserves the property that we can perform multiplication by scalars from $GF(q)$ and vector addition without leaving this set, we say that this set is a *subspace* of the original space of all the vectors. Such a set is a linear code.

Thus, *a linear (N, K) q-ary code is a set of codewords $\boldsymbol{v}$ such that any linear combination $\sum_i a_i \boldsymbol{v}_i$ with coefficients $a_i \in GF(q)$ is again a codeword.*

This statement can be used as equivalent definition of a linear code.

For $q = 2$ we have *binary* linear codes, for which the above statement simplifies to

A binary linear code is a set of codewords such that a sum of any two codewords is a codeword (where summation is performed modulo-2).

Before proceeding further, note that, due to the definition of the linear code, *every linear code must contain the all-zero codeword $\boldsymbol{v} = \boldsymbol{0}$* (this codeword is a neutral element for addition, which must exist in every linear space).

3.2 Generator matrix

From the definition of a linear code it follows that it is possible to find exactly K linearly independent codewords $\boldsymbol{g}_1, ..., \boldsymbol{g}_K \in \mathcal{C} \subset GF(q)^N$ such that every codeword $\boldsymbol{v}$ can be written as a linear combination of these codewords:

$$\boldsymbol{v} = u_1 \boldsymbol{g}_1 + u_2 \boldsymbol{g}_2 + ... + u_K \boldsymbol{g}_K,$$

where $u_k \in GF(q)$. We can rewrite this equation as

$$\boldsymbol{v} = [u_1 \ u_2 \ ... \ u_K] \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_K \end{bmatrix},$$

or, equivalently,

$$\boldsymbol{v} = \boldsymbol{u}\boldsymbol{G}, \tag{6}$$

where the K-tuple $\boldsymbol{u}$ is the information sequence that is to be encoded and the $K \times N$ matrix $\boldsymbol{G}$ is the so called *generator matrix* of a code. Its name comes from the fact that linear combinations of its rows $\boldsymbol{g}_k$ generate all codewords of a given code. A linear (N, K) block code is usually specified by its generator matrix $\boldsymbol{G}$. Then, all the N-tuples $\boldsymbol{v}$ that can be written as (6) are the codewords of that code.

Equation (6) specifies the *encoding* rule, i.e. the mapping from the information sequences $\boldsymbol{u}$ to codewords $\boldsymbol{v}$. Since *any* set of K linearly independent codewords can be chosen as rows of $\boldsymbol{G}$, one code has many different generator matrices! In other words, all these matrices yield the same set of codewords, but they specify different *encoders*, i.e., different mappings between the codewords and the information sequences.[2] One thing that is for sure common to all of the encoders is that the all-zero information sequence $\boldsymbol{u} = \boldsymbol{0}$ is always mapped to the all-zero code sequence $\boldsymbol{v} = \boldsymbol{0}$.

We can obtain different generator matrices (i.e. different encoders) for the given code by performing elementary *row* operations on $\boldsymbol{G}$. In other words, by exchanging any two rows, by adding one row to another multiplied by a constant from $GF(q)$ we change the mapping, but we do not change the code - the set of codewords is the same.

If we apply *column* operations on $\boldsymbol{G}$ (i.e., we permute the columns) we change the positions of the codesymbols in a codeword, and thus, we change the code. This leads us to the definition of an *equivalent code.*

Definition: *A code $\mathcal{C}'$ is said to be equivalent to the code $\mathcal{C}$ if there exists a permutation π such that every codeword $\boldsymbol{v}' \in \mathcal{C}'$ is obtained as a permutation π of a codeword $\boldsymbol{v} \in \mathcal{C}$, i.e., $\boldsymbol{v}' = \pi(\boldsymbol{v})$.*

Equivalent codes have the same distance properties and the same error-rate performance.

[2] We can easily compute the exact number of generator matrices for a q-ary linear (N, K) code: as a first row of $\boldsymbol{G}$ we can use any non-zero codeword – we have $M - 1 = q^K - 1$ choices. As a second row, we can use any non-zero codeword, different from the previously chosen one or any scaled version of it – we have $q^K - q$ choices, etc. In total, there are exactly $\prod_{i=0}^{K-1}(q^K - q^i)$ distinct generator matrices of the given code.

One particularly interesting encoder type is the so called *systematic* encoder of a given code. This is the encoder that specifies the mapping $\boldsymbol{u} \to \boldsymbol{v}$ such that the information K-tuple $\boldsymbol{u}$ appears unchanged as the K symbols of the corresponding codeword $\boldsymbol{v}$. Usually we assume that these are the first K positions, then the systematic codeword is of the form

$$\boldsymbol{v}_{\text{sys}} = [u_1\ u_2\ ...\ u_K\ v_{K+1}...v_N] = [\boldsymbol{u}\ v_{K+1}...v_N].$$

The first K codeword symbols are called systematic symbols, and, the following $N-K$ symbols are called *parity check* symbols. The reason for this name will become clear very soon. The corresponding *systematic generator matrix*, that specifies the systematic encoding rule is of the form

$$\boldsymbol{G}_{\text{sys}} = [\boldsymbol{I}_K \,|\, \boldsymbol{P}], \tag{7}$$

where $\boldsymbol{I}_K$ is the $K \times K$ identity matrix and $\boldsymbol{P}$ is some $K \times N - K$ matrix. We denote the elements of the matrix $\boldsymbol{P}$ with p_{kn}, $1 \leq k \leq K$, $K+1 \leq n \leq N$. If the code is binary, $p_{kn} \in \{0,1\}$, then the parity check symbols specify the parity equations of the information symbols: $v_n = u_1 p_{1n} + ... + u_K p_{Kn}$.

Every matrix $\boldsymbol{G}$ can be reduced to the systematic form by elementary row operations, but the K systematic positions do not necessarily have to appear as the first K positions. In that case, however, we can always permute the columns so that we obtain an equivalent code with the first K systematic symbols. Finally, we recall once more that being systematic is a property of an encoder, not of a code, i.e., every code has systematic and non-systematic encoders.

3.3 Parity-check matrix and syndrome

Besides the generator matrix, there is one more matrix that can be associated with every linear block code – the so called *parity check matrix* denoted as $\boldsymbol{H}$. This is an $(N-K) \times N$ matrix whose rows are linearly independent and they are *orthogonal* to the rows of the generator matrix $\boldsymbol{G}$, i.e.,

$$\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}. \tag{8}$$

Since any codeword $\boldsymbol{v} \in \mathcal{C}$ can be written as $\boldsymbol{v} = \boldsymbol{u}\boldsymbol{G}$, we immediately obtain that

$$\boldsymbol{v}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0} \tag{9}$$

has to hold for any $\boldsymbol{v} \in \mathcal{C}$. In fact, (9) can be used as an alternative way to generate a code (instead of the generator matrix)[3] – we say that a sequence $\boldsymbol{v}$ is a codeword of a code $\mathcal{C}$ if and only if it satisfies (9) for the given parity check matrix of the code $\mathcal{C}$. Note that a parity check matrix is not unique for a given code – every code has many parity check matrices (as it has many generator matrices).

When the generator matrix is in the systematic form (7), the corresponding parity check matrix is

$$\boldsymbol{H}_{\text{sys}} = [-\boldsymbol{P}^{\mathrm{T}} \,|\, \boldsymbol{I}_{(N-K)}], \tag{10}$$

where minus "$-$" should be interpreted modulo q. For binary codes $-\boldsymbol{P}^{\mathrm{T}} = \boldsymbol{P}^{\mathrm{T}}$.

The $(N-K)$ rows of the parity check matrix generate a $(N, N-K)$ code $\mathcal{C}^{\perp}$ which is called the *dual code* of $\mathcal{C}$. All the codewords $\boldsymbol{v}_{\perp}$ of the dual code are orthogonal to the codewords $\boldsymbol{v}$ of $\mathcal{C}$, i.e., $\boldsymbol{v}_{\perp} \cdot \boldsymbol{v}^{\mathrm{T}} = 0$.

Now we will briefly illustrate how the decoder uses the parity check matrix to detect errors: Assume that a linear binary block code $\mathcal{C}$ was used for transmission over an equivalent binary symmetric channel. The received sequence is given by

$$\boldsymbol{y} = \boldsymbol{v} + \boldsymbol{e}, \tag{11}$$

where $\boldsymbol{e}$ is the *error sequence* with a 1 on every position where the error occurred (the code symbol got flipped by the channel), and zeros elsewhere. The decoder computes the vector

$$\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^{\mathrm{T}} = \underbrace{\boldsymbol{v}\boldsymbol{H}^{\mathrm{T}}}_{=\mathbf{0}} + \boldsymbol{e}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{e}\boldsymbol{H}^{\mathrm{T}},$$

[3] A well-known example of the codes that are defined via their parity check matrices (rather than via the generator matrices) are the LDPC codes – Low Density Parity Check codes. As the name suggests, these are (very long) binary block codes whose parity check matrix is *sparse*, i.e., it contains very few ones in all rows or columns, and all the other elements are zero. When properly designed, these codes can achieve excellent performance, better than most of other known codes.

which is called a *syndrome* and it depends on the error pattern: clearly, if $\boldsymbol{s} \neq \mathbf{0}$ this means that at least one error occurred, i.e., $\boldsymbol{e} \neq \mathbf{0}$. Thus, by evaluating a syndrome of the received sequence, the decoder can immediately detect whether an error occurred during transmission (moreover, some errors can be corrected using the value of the syndrome, but we will not go deeper into this consideration). When the syndrome is zero, the decoder declares that the transmission was error-free. Note, however, that $\boldsymbol{s} = \mathbf{0}$ if $\boldsymbol{e} = \mathbf{0}$ (the transmission was indeed error-free), *or* if some errors occurred, but the error pattern is such that $\boldsymbol{e}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$, which means that the error pattern is equal to some codeword. Such error patterns are *not detectable*. There are in total $2^K - 1$ non-detectable error patterns (we exclude the all-zero codeword). The remaining $2^N - 2^K$ patterns are detectable. For a q-ary block code, the same reasoning applies (we consider a q-ary symmetric channel where received sequences and error-patterns are q-ary).

Mentioning error-detecting and error-correcting capabilities of a code leads us to the definition of a very important parameter of a linear code – *the minimum distance* of a code, which determines these capabilities.

3.4 Minimum distance

First, we need the following

Definition: *The Hamming weight $w_{\mathrm{H}}(\boldsymbol{v})$ of a codeword $\boldsymbol{v}$ is the number of non-zero symbols in $\boldsymbol{v}$.*

From the definition it follows that $0 \leq w_{\mathrm{H}}(\boldsymbol{v}) \leq N$, where $w_{\mathrm{H}}(\boldsymbol{v}) = 0$ corresponds to the all-zero codeword. For binary codes, the non-zero symbols are equal to 1, and thus the Hamming weight of a codeword is obtained simply by summing all its symbols.

Definition: *The Hamming distance $d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}')$ between the two codewords, $\boldsymbol{v}$ and $\boldsymbol{v}'$ is the number of positions where these codewords differ.*

Again, it follows that $0 \leq d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') \leq N$, where $d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') = 0$ means that the two codewords coincide, $\boldsymbol{v} = \boldsymbol{v}'$. The Hamming distance is a proper *metric* (or a proper *distance*) in the mathematical sense, which means that it fulfils the three metric axioms:

- non-negativity: $d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') \geq 0$, with $d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') = 0$ iff $\boldsymbol{v} = \boldsymbol{v}'$
- symmetry: $d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') = d_{\mathrm{H}}(\boldsymbol{v}', \boldsymbol{v})$
- triangle inequality: $d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') + d_{\mathrm{H}}(\boldsymbol{v}', \boldsymbol{v}'') \geq d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}'')$

It follows from the definition that the Hamming distance between any two codewords $\boldsymbol{v}$ and $\boldsymbol{v}'$ is *equal to the Hamming weight of the sum sequence $\boldsymbol{v} + \boldsymbol{v}'$*, where summation is performed modulo q:

$$d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}') = w_{\mathrm{H}}(\boldsymbol{v} + \boldsymbol{v}') \tag{12}$$

Now we are ready for the following

Definition: *The minimum distance $d_{\min}$ of a block code $\mathcal{C}$ is the minimum Hamming distance between any two distinct codewords of that code:*

$$d_{\min} = \min_{\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{C},\ \boldsymbol{v} \neq \boldsymbol{v}'} \{d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{v}')\} \tag{13}$$

From (12) we see that $d_{\min}$ is the minimum Hamming weight of the sum of any two distinct codewords. We know from before that *for a linear block code, the sum of any two codewords is a codeword*, i.e., $(\boldsymbol{v} + \boldsymbol{v}') \in \mathcal{C}$, $\forall \boldsymbol{v}, \boldsymbol{v}' \in \mathcal{C}$.

Thus, we arrive to the very important result: *The minimum distance of a **linear** block code $\mathcal{C}$ is equal to the minimum Hamming weight of its non-zero codewords:*

$$d_{\min} = \min_{\boldsymbol{v} \in \mathcal{C},\ \boldsymbol{v} \neq \mathbf{0}} \{w_{\mathrm{H}}(\boldsymbol{v})\}, \quad \text{for linear } \mathcal{C} \tag{14}$$

Thus, the Hamming weight of any codeword $\boldsymbol{v}$ of a linear block code $\mathcal{C}$ satisfies $w_{\mathrm{H}}(\boldsymbol{v}) \geq d_{\min}$. The minimum distance of any $(N,\, K)$ linear block code $\mathcal{C}$ satisfies the so called *Singleton bound*

$$d_{\min} \leq N - K + 1 \tag{15}$$

The codes that satisfy this inequality with equality are called *maximum distances separable* (MDS) codes – as the name suggests, they have the largest possible minimum distance for the given values of N, K.

As we will see later, *Reed-Solomon codes are MDS codes*! Moreover, except for the "dummy" repetition codes, they are the only binary codes that are MDS.

In order to find the minimum distance of a given code, one would hypothetically need to check weights of all non-zero codewords and find the smallest one. For long codes, checking $2^K - 1$ candidates can be a tedious task. What is often used as a quicker way to obtain $d_{\min}$ is the following result:

The minimum distance $d_{\min}$ of a linear code is equal to the smallest number d such that d columns of the code's parity check matrix $\boldsymbol{H}$ are linearly dependent. In other words, any set of $d_{\min} - 1$ or less columns of $\boldsymbol{H}$ are guaranteed to be linearly independent. This result follows directly from (9), which can be written as

$$v_1 \boldsymbol{h}_1 + v_2 \boldsymbol{h}_2 + ... + v_N \boldsymbol{h}_N = 0,$$

where $\boldsymbol{h}_n$ denote rows of $\boldsymbol{H}^T$, which are columns of $\boldsymbol{H}$. For a codeword of weight equal to $d_{\min}$ the left-hand sum in the above equation has $d_{\min}$ non-zero elements (corresponding to $d_{\min}$ non-zero symbols v_n) and since this sum equals 0, that means that the $d_{\min}$ columns of $\boldsymbol{H}$ are linearly dependent.

3.5 Error-detection and error-correction capabilities of a code

Let us consider again the linear block code used for transmission over a (in general q-ary) symmetric channel, where the received sequence is given by (11). From the previous section we now know that any error pattern $\boldsymbol{e}$ of Hamming weight $w_{\mathrm{H}}(\boldsymbol{e}) \leq d_{\min} - 1$ cannot be a codeword, and thus, its syndrome will be non-zero, $\boldsymbol{s} \neq \boldsymbol{0}$. This means that *a linear code with minimum distance $d_{\min}$ can detect every error pattern with $d_{\min} - 1$ or fewer errors.*

Actually, as mentioned before, a q-ary linear code can detect many more error patterns, in total $q^N - q^K$ of them, where some of them have weight $d_{\min}$ or larger. Now we just want to stress that if a weight of an error pattern is $\leq d_{\min} - 1$, it is *guaranteed* that this pattern will be detected, whereas for $w_{\mathrm{H}}(\boldsymbol{e}) \geq d_{\min}$ this guarantee can not be given (because some error patterns might pass undetected).

More often, we want to employ a code to *correct* errors, rather than only to detect them. To this end, for a given linear code with minimum distance d_{min} we introduce a positive integer t such that

$$2t + 1 \leq d_{\min} \leq 2t + 2,$$

which is equivalent to

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor, \tag{16}$$

where the floor function $\lfloor x \rfloor$ returns the largest integer smaller than or equal to x (note that when $d_{\min}$ is an odd number, $d_{\min} - 1$ is even, and flooring is not necessary). Now we can state a fundamental result:

A block code of minimum distance $d_{\min}$ is capable of correcting all error patterns with t or fewer errors, where t is given by (16). We call such a code a t-error correcting code. An intuitive verification of this result is illustrated in Figure 3.

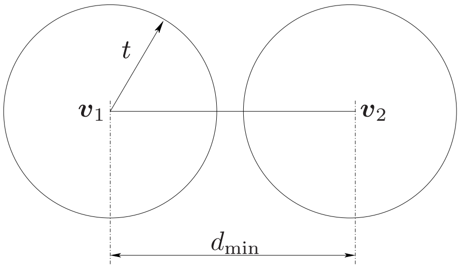


Figure 3: Illustration of error correction capability of a code: we can represent codewords $\boldsymbol{v}$ and received sequences $\boldsymbol{y}$ as points in the $N-$dimensional space. Then, all points $\boldsymbol{y}$ at distance $\leq t$ from $\boldsymbol{v}$ lie inside a sphere with centre in $\boldsymbol{v}$ and radius t and they will be decoded as $\boldsymbol{v}$. Decoding is correct when the spheres do not overlap (and do not touch each other). In order to avoid overlapping spheres, it must hold that $d_{\min} \geq 2t + 1$.

3.6 ML decoding for binary block codes

First, let us recall the definition of the optimum decoder. We assume that the codewords are equiprobable, and thus the maximum-likelihood (ML) decoder defined by (5) is the optimum decoder. In this section, we will formulate the ML decoding rule in more detail. For simplicity, we consider a binary linear block code, and two important channel models: the binary symmetric channel (BSC), which corresponds to the case when a hard decision device follows the demodulator in the receiver chain; and an additive white Gaussian noise (AWGN) channel, which corresponds to the case when a soft decision device (with a continuum of decision levels) follows the demodulator.

BSC

The binary received sequence $\boldsymbol{y}$ is given by (11). Let d denote the number of errors that occurred during the transmission of one codeword, i.e., $w_{\mathrm{H}}(\boldsymbol{e}) = d$. This is nothing else but the distance between the received and the transmitted sequence

$$d_{\mathrm{H}}(\boldsymbol{y}, \boldsymbol{v}) = w_{\mathrm{H}}(\boldsymbol{e}) = d.$$

The *crossover* probability of a BSC, i.e., the probability that the codesymbol v_n, $1 \leq n \leq N$ will be inverted by the channel is denoted as p

$$p = \Pr(y_n = 0 | v_n = 1) = \Pr(y_n = 1 | v_n = 0), \quad p \leq \frac{1}{2}.$$

The probability of correct reception of a codesymbol v_n is then simply

$$1 - p = \Pr(y_n = 0 | v_n = 0) = \Pr(y_n = 1 | v_n = 1).$$

The BSC is memoryless and without feedback—that means that the output of the channel y_n at time instant n depends only on the transmitted codesymbol x_n (and not on previous codesymbols, or previous outputs). The likelihood of the codeword $\boldsymbol{v}$ is then

$$p(\boldsymbol{y}|\boldsymbol{v}) = \prod_{n=1}^{N} p(y_n|v_n) = p^d (1-p)^{(N-d)} = \left(\frac{p}{1-p} \right)^d (1-p)^N, \tag{17}$$

where the second equality is obtained from the fact that d errors occurred, each with probability p, and the remaining $N - d$ symbols were received correctly, each with probability $1 - p$.

Now note that in the above likelihood expression, $(1-p)^N$ is a constant, determined by the code length N and channel's crossover probability p. Thus, if we want to maximise the likelihood over all codewords, this constant term plays no role and can be omitted. Furthermore, we note that, for a given $p \leq 1/2$, $p/(1-p) \leq 1$, and thus *maximising* $(p/(1-p))^d$ is equivalent to *minimising* d. Thus we have arrived to the following important result:

$$\hat{\boldsymbol{v}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{v} \in \mathcal{C}} p(\boldsymbol{y}|\boldsymbol{v}) = \arg\max_{\boldsymbol{v} \in \mathcal{C}} \left(\frac{p}{1-p} \right)^d = \arg\min_{\boldsymbol{v} \in \mathcal{C}} d. \tag{18}$$

The ML decoder is the *minimum-distance decoder*, i.e., it chooses the codeword which is *closest* to the received sequence in terms of the Hamming distance.

The larger the minimum distance of the code, the more errors d need to occur such that the received sequence is closer to an erroneous codeword than to the transmitted codeword (i.e., such that an ML decoder commits a decoding error).

AWGN channel

In case of the AWGN channel model, the received sequence is given by

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w},$$

where $\boldsymbol{w}$ is the Gaussian noise sequence with zero mean and per-sample variance σ^2 and $\boldsymbol{x}$ is the codeword $\boldsymbol{v}$ mapped onto a modulation symbol (for BPSK or QAM, $v_n = 0$ is mapped onto $x_n = -1$, and $v_n = 1$ onto $x_n = 1$, using the rule $x_n = 2v_n - 1$).

Now the likelihood of the codeword $\boldsymbol{v}$, i.e., of the corresponding bipolar sequence $\boldsymbol{x} = 2\boldsymbol{v} - \mathbf{1}$ is the Gaussian pdf:

$$p(\boldsymbol{y}|\boldsymbol{v}) = p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}||\boldsymbol{y}-\boldsymbol{x}||^2\right) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}(||\boldsymbol{y}||^2 - 2\boldsymbol{y}\boldsymbol{x}^{\mathrm{T}} + ||\boldsymbol{x}||^2)\right)$$

We observe the following: maximisation of the likelihood $p(\boldsymbol{y}|\boldsymbol{v})$ is equivalent to minimisation of the term in the exponent, $||\boldsymbol{y}-\boldsymbol{x}||^2 = ||\boldsymbol{y}||^2 - 2\boldsymbol{y}\boldsymbol{x}^{\mathrm{T}} + ||\boldsymbol{x}||^2$, which is the squared *Euclidean distance* between $\boldsymbol{y}$ and $\boldsymbol{x}$. Furthermore, for a given received sequence, $||\boldsymbol{y}||^2$ is a constant and thus has no influence on the minimisation, and, for any bipolar sequence $\boldsymbol{x}$ we have $||\boldsymbol{x}||^2 = N$. Thus, minimisation of the $(||\boldsymbol{y}||^2 - 2\boldsymbol{y}\boldsymbol{x}^{\mathrm{T}} + ||\boldsymbol{x}||^2)$ is equivalent to maximisation of $\boldsymbol{y}\boldsymbol{x}^{\mathrm{T}}$, which is nothing else but the *correlation* between the received and the transmitted sequence. To summarise we have obtained

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{v}\in\mathcal{C}} p(\boldsymbol{y}|\boldsymbol{x}) = \arg\min_{\boldsymbol{v}\in\mathcal{C}} ||\boldsymbol{y}-\boldsymbol{x}||^2 = \arg\max_{\boldsymbol{v}\in\mathcal{C}} \left(\boldsymbol{y}\boldsymbol{x}^{\mathrm{T}}\right) = \arg\max_{\boldsymbol{v}\in\mathcal{C}} \left(\sum_{i=1}^{N} x_n y_n\right). \tag{19}$$

The ML decoder is the *minimum-Euclidean distance decoder*, i.e., it chooses the codeword which is *closest* to the received sequence in terms of Euclidean distance. Equivalently, we can say that the ML decoder is the *maximum-correlation decoder*, i.e., it chooses the sequence which has highest correlation (i.e., highest similarity) with the received sequence.

The exact value of the block error rate P_B for the ML decoder of a given code is very hard to compute. However, we can upper-bound it. The most frequently used bound is the so called *union bound* which says that the probability of block error is not larger than the sum of probabilities of all pair-wise error events (event that a transmitted codeword $\boldsymbol{v}$ is misinterpreted as another codeword $\boldsymbol{v}'$ at distance d from the transmitted codeword):

$$P_B \leq \sum_{d=d_{\min}}^{N} A_d Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right), \tag{20}$$

where A_d is the number of codewords of a given linear code of weight d, and $R = K/N$ is the code rate. This bound is tight for very high signal-to-noise ratios (SNRs) E_b/N_0. As the SNR increases, the Q functions in the above sum decrease rapidly. The dominant term is the first term in the sum, determined by the minimum distance

$$A_{d_{\min}} Q\left(\sqrt{2d_{\min}R\frac{E_b}{N_0}}\right),$$

and this term governs the behaviour of P_B and high SNR. Thus we conclude that the *minimum distance of the code determines its error rate performance at high SNR.*

For very low SNR, all the terms in the above sum contribute to the bound. In this region, minimum distance of the code is not the most important parameter that dictates the performance of the code - larger weights d and their multiplicities A_d are even more important. As a well-known example of codes designed to operate at very low SNR we mention *turbo codes* - these codes have quite poor minimum distance but have other good parameters, which results in their remarkable performance in very low SNRs, and relatively poor performance in high SNR. However, these codes are out of the scope of our tutorial.

To go back to higher SNR region, we conclude that, in order to achieve low P_B we should design codes such that they have *large* $d_{\min}$ and *small* number of codewords $A_{d_{\min}}$ of weight equal to $d_{\min}$ (this number is also called number of nearest neighbours). This design guideline is perfectly valid for higher SNR and it is the reason why the coding theory has been focused on designing large-$d_{\min}$ codes. As a remarkable example of such codes we mention Reed-Solomon codes, the most important and the most widely used block codes today.

In the sequel, we will study Reed-Solomon codes in more detail, since they are used in almost all OFDM systems today (and, by the way, in all the CD players!). They belong to a special class of linear block codes, called *cyclic codes*, which we will introduce in the next section.

4 Cyclic Codes

4.1 Elements of finite-fields algebra

Cyclic codes posses plenty of interesting structural properties - to present and analyse them, we need to acquire some basic mathematical tools – namely, the basic concepts of the finite-field algebra. The purpose and the usefulness of the concepts presented here will become clear in the next chapter, when we apply them to analyse cyclic codes. We warn the reader that condensing a whole science on one page is a rather difficult task. Most of the important results will be stated shortly, without proofs.

As a start, we recall that a *finite field* is a field with a finite number of elements. We call such a field a Galois field, and denote it $GF(q)$. The number of elements q is called the *order of the field*, and it is either a prime number, or a power of a prime number p, i.e., $q = p^m$, for some integer $m \geq 1$. All the operations in such a field are performed modulo-p. The field $GF(q^n)$, where $n \geq 1$, is called the *extension field* of $GF(q)$. In fact, for any positive integer n, we can construct an extension field $GF(q^n)$ of the field $GF(q)$. Furthermore, $GF(q)$ is a *subfield* of the extension field $GF(q^n)$.

For a field $GF(q)$ we define the *characteristic* of a field as a smallest integer λ, such that $\sum_{i=1}^{\lambda} 1 = 0$ (don't forget that summation is modulo-p). This number is always prime (for any q). In particular, for $GF(p)$, $\lambda = p$.

Consider any non-zero field element $a \in GF(q)$. Since the field is closed under multiplication, the elements $\{a, a^2, ..., a^n, a^{n+1}, ...\}$ all belong to the field. Since the field is finite, these powers cannot all be distinct. Thus, at some point in this chain we must start having a repetition of the previous powers. In other words, *for any non-zero field element $a \in GF(q)$, there exists an integer n such that $a^n = 1$, where* $n \leq q-1$. The *smallest* such integer is called the *order* of the element a. The elements $\{a, a^2, ..., a^n = 1\}$ form a multiplicative group. A group is called *cyclic* if there exists a group element whose powers constitute the whole group.

For any non-zero $a \in GF(q)$ it can be shown that $a^{q-1} = 1$ holds. On the other hand, we know that the order of a is the smallest $n \leq q-1$ such that $a^n = 1$. It can be proved that *the order of any element* $a \in GF(q)$ *divides* $q-1$, i.e., $n|(q-1)$. If the order of an element $a \in GF(q)$ is exactly $n = q-1$, such an element is called a *primitive element* of the field $GF(q)$, and we denote it with α. Clearly, all powers of the primitive element generate all non-zero elements of the field, $\{\alpha, \alpha^2, ..., \alpha^{q-2}, \alpha^{q-1} = 1\}$. Every finite field $GF(q)$ has its primitive element, and it is not necessarily unique (one field can have several distinct primitive elements). For example, if we consider $GF(7)$, its elements are {0,1,...,6} and the order of each element is either 1, 6, 3 or 2, since these are all possible numbers n that divide $q-1 = 6$. The primitive element must have order $n = 6$, and it is $\alpha = 3$, since its powers are all non-zero elements of $GF(7)$, $3^1 = 3,\ 3^2 \mod (7) = 2,\ 3^3 = (3^2) \cdot 3 = 2 \cdot 3 = 6,\ 3^4 = (3^3) \cdot 3 = 4,\ 3^5 = (3^4) \cdot 3 = 5,\ 3^6 = (3^5) \cdot 3 = 1$.

At this point we can clarify how are the operations of addition, subtraction, multiplication and division performed in $GF(q)$. As we mentioned before, addition and multiplication are simple modulo-p operations, for $q = p^m$, and p prime. Subtraction is defined as modulo-p addition of the "negative" element, i.e., $a_1 - a_2 = a_1 + (-a_2)$, where $-a_2 = 0 - a_2 = cq - a_2$, where c is some positive integer, such that $0 \leq cq - a_2 \leq q-1$. For example, -1 in the field $GF(7)$ means $7 - 1 = 6$; $a - 9$, for $a \in GF(7)$ is equivalent to $a + (2 \cdot 7 - 9) = a + 5 \mod (7)$. Note that for $GF(2^n)$, -1 is the same as 1, since $2 - 1 = 1$. Similarly, division is defined as the modulo-p multiplication by the "inverse" element, i.e., $a_1/a_2 = a_1 \cdot (a_2^{-1})$, where $(a_2^{-1}) = a_2^{0-1} = (a_2^0)^{-1} = (a_2^n)^{-1} = a_2^{n-1}$, where n is the order of a_2.

Now, we consider *polynomials* $a(x) = a_0 + a_1 x + ... + a_n x^n$, whose coefficients a_i, $0 \leq i \leq n$, are elements of the field $GF(q)$. We say for such $a(x)$ that it is a *polynomial over* $GF(q)$. The largest exponent n of the x^n with the non-zero coefficient a_n is called the *degree* of the polynomial, and denoted as $\deg(a(x)) = n$. If $a_n = 1$, we say that $a(x)$ is a *monic* polynomial. A polynomial of degree 0 is a scalar $a(x) = a_0 \in GF(q)$. For an arbitrary degree-n polynomial, the n coefficients from a_0 to a_{n-1} can have any of the q values from $GF(q)$, while a_n can have $q-1$ possible values ($a_n = 0$ is not allowed since then the degree cannot be n). Thus, there are in total $q^n(q-1)$ polynomials of degree n over $GF(q)$.

A *root* (or a zero) of a polynomial $a(x)$ is a number β such that $a(x = \beta) = 0$. A polynomial of degree n has exactly n roots β_i, and can be written as a product of n degree-1 polynomials $(x - \beta_i)$, i.e., as $a(x) = (x - \beta_1)(x - \beta_2)...(x - \beta_n)$. That means, if a polynomial $a(x)$ has a root $x = \beta$, then $a(x)$ is divisible by $(x - \beta)$.

In the field of real numbers $\mathbb{R}$, we know that some polynomials do not have real roots, but they have complex roots, from the field of complex numbers $\mathbb{C}$. This property translates to finite fields: *In general, roots of the degree-n polynomial over $GF(q)$ are from the extension field $GF(q^n)$.* Recall that $GF(q)$ is a subfield of $GF(q^n)$ (in the same sense as $\mathbb{R}$ is a subfield of $\mathbb{C}$). Thus, roots of some polynomials over $GF(q)$ belong to $GF(q)$, while others do not, but they always belong to $GF(q^m)$.

Now we can state the following important result: *The $q^n - 1$ roots of the polynomial $(x^{q^n-1} - 1)$ are all $q^n - 1$ non-zero elements of $GF(q^n)$* . Furthermore, since a root of x is 0, we can conclude that *the q^n roots of $x(x^{q^n-1} - 1) = (x^{q^n} - x)$ are all the elements of $GF(q^n)$.* Now recall that a primitive element of $GF(q^n)$ is an element $\alpha \in GF(q^n)$ of order $q^n - 1$, whose powers $\{\alpha, \alpha^2, ..., \alpha^{q^n-2}, \alpha^{q^n-1} = 1\}$ constitute all the non-zero elements of $GF(q^n)$. We conclude that *the primitive element of $GF(q^n)$ and all its powers are the roots of $x^{q^n-1} - 1$.*

Since roots of any degree-n polynomial over $GF(q)$ are some elements of $GF(q^n)$, which are some of the roots of $x^{q^n} - x$, we conclude that *any polynomial of degree n over $GF(q)$, divides the polynomial $x^{q^n} - x$.*

Now we proceed to the final step – we define some important special types of polynomials. A polynomial $a(x)$ of degree n over $GF(q)$ is said to be *irreducible* if it is not divisible by any polynomial over $GF(q)$ of degree $d < n$ (and $d > 0$). For example, consider a binary field $GF(2)$. There are 2 polynomials of degree 1 over $GF(2)$, namely, x and $x+1$. There are 4 polynomials of degree 2: x^2, x^2+x, x^2+1 and x^2+x+1. The only irreducible polynomial of degree 2 is x^2+x+1, since it is not divisible by neither of the two degree-1 polynomials. The other 3 polynomials are not irreducible since $x^2 = x \cdot x$ is divisible by x, $x^2 + x = x(x+1)$ is divisible by both x and $x+1$, and $x^2+1 = (x+1)(x+1)$ is divisible by $x+1$. This example illustrates a general property that *all polynomials over the binary field $GF(2)$ with even number of elements are divisible by $(x+1)$.* Thus, the candidates for the irreducible polynomials are only those polynomials that have odd number of terms.

Since an irreducible polynomial is not divisible by x, this means that it does not have a root in 0. That implies, in accordance with the previous section that *any irreducible polynomial of degree n over $GF(q)$, divides the polynomial $x^{q^n-1} - 1$.*

In general, every irreducible polynomial of degree n over $GF(q)$ divides a polynomial $x^l - 1$, for some $l \leq q^n - 1$. If, for some irreducible polynomial, the smallest such l is exactly $l = q^n - 1$, then this irreducible polynomial is called the *primitive* polynomial, and it is usually denoted as $p(x)$. Note the analogy of these statements with the definitions of the order of the element and of the primitive element! Since $p(x)|(x^{q^n-1} - 1)$, we arrive to the final crucial result, which follows from the previous section: *a root of a primitive polynomial $p(x)$ of degree n over $GF(q)$ is a primitive element α of the extension field $GF(q^n)$.* In other words, we say that a primitive polynomial $p(x)$ (or, more precisely, its roots) *generates the field $GF(q^n)$.*

Example: To illustrate the above statements, we consider the case when $q = 2$, $n = 3$, i.e., the field is $GF(2^3)$, which is the extension field of $GF(2)$. The primitive polynomials of degree $n = 3$ over $GF(2)$ are: $p'(x) = 1 + x^2 + x^3$ and $p''(x) = 1 + x + x^3$ (cf. Appendix). Any of these two polynomials can be used to generate the field $GF(2^3)$. Let us choose $p'(x)$. As we said before, its root is a primitive element $\alpha \in GF(2^3)$. Thus we have

$$p'(\alpha) = 0 \quad \Longrightarrow \quad 1 + \alpha^2 + \alpha^3 = 0 \quad \Longrightarrow \quad \alpha^3 = \alpha^2 + 1 \tag{21}$$

Every element of $GF(2^3)$ can be represented as a power of α. Additionally, using relation (21), we can represent every element of $GF(2^3)$ as a *polynomial in α over $GF(2)$ of degree* $\leq 3 - 1 = 2$, i.e., as

$$a(\alpha) = a_2\alpha^2 + a_1\alpha + a_0,$$

where $a_2, a_1, a_0 \in \{0, 1\}$ and $\deg(a(\alpha)) \leq 2$. The strategy is the following: as a start, (21) tells us that α^3 can be written as $\alpha^2 + 1$. Then α^4 can be written as $\alpha^4 = \alpha^3 \cdot \alpha = (\alpha^2+1)\alpha = \alpha^3 + \alpha = \alpha^2 + \alpha + 1$, and so on. Using this procedure we obtain the different representations of the elements of $GF(2^3)$ as shown in Table 1.

Generalisation: At the end, we generalise the previous example: the elements of the field $GF(q^n)$ can be generated by the primitive polynomial $p(x)$ over $GF(q)$ of degree n, whose root is a primitive element $\alpha \in GF(q^n)$. Every field element is a power of α. The primitive polynomial $p(x)$ is not necessarily unique. The property that $p(\alpha) = 0$ specifies the relation of the form $\alpha^n = p_{n-1}\alpha^{n-1} + ... + p_1\alpha + p_0$, which enables us to represent every field element not only as α^i, $0 \leq i < q^n - 1$, but also as a polynomial

$a(\alpha) = a_{n-1}\alpha^{n-1} + ... + a_1\alpha + a_0$, of degree $\leq n-1$ and coefficients $a_i \in GF(q)$. The corresponding q-ary sequence $[a_{n-1} \; ... \; a_1 \; a_0]$ is also one possible representation of the field elements.

Table 1: Table of non-zero field elements of $GF(2^3)$ generated by the primitive polynomial $p(x) = x^3 + x^2 + 1$.

α^i	polynomial	binary $[a_2 \, a_1 \, a_0]$	decimal
α^0	1	0 0 1	1
α^1	α	0 1 0	2
α^2	α^2	1 0 0	4
α^3	α^2+1	1 0 1	5
α^4	$\alpha^2+\alpha+1$	1 1 1	7
α^5	$\alpha+1$	0 1 1	3
α^6	$\alpha^2+\alpha$	1 1 0	6

Appendix

In Table 2 we list all irreducible polynomials over the binary field $GF(2)$, of the degree $n = 1, 2, 3, 4, 5$. In general, there is no quick way to check whether an arbitrary polynomial is irreducible or not – one needs to try dividing it with all polynomials of lower degree, and if it is not divisible by any, one can declare it irreducible. The polynomials listed in the table are obtained in this way, which is not a big effort for relatively low degree n.

Table 2: Primitive and irreducible polynomials over $GF(2)$ of degree $n = 1, 2, 3, 4, 5$.

degree n	primitive polynomials	irreducible, but not primitive
1	$x+1$	x (by convention)
2	x^2+x+1	-
3	$x^3+x+1, \;\; x^3+x^2+1$	-
4	$x^4+x+1, \;\; x^4+x^3+1$	$x^4+x^3+x^2+x+1$
5	$x^5+x^2+1, \;\; x^5+x^3+1,$ $x^5+x^4+x^3+x^2+1, \;\; x^5+x^4+x^2+x+1$ $x^5+x^4+x^3+x+1, \;\; x^5+x^3+x^2+x+1$	-

Note that, for example, $p(x) = x^4 + x^3 + x^2 + x + 1$ is an irreducible polynomial, but it is not primitive because $l = 2^4 - 1 = 15$ is not the smallest integer for which $x^l + 1$ is divisible by $p(x)$. The smallest one is $l = 5$, since $x^5 + 1 = (x + 1)(x^4 + x^3 + x^2 + x + 1)$.

After this, not so short introduction to finite fields, we are ready to proceed with the definition and the properties of cyclic codes.

4.2 Definition of cyclic codes

Definition: *A linear block code $\mathcal{C}$ is called a cyclic code if every cyclic shift of any codeword $\boldsymbol{v} \in \mathcal{C}$ is also a codeword of $\mathcal{C}$.*

Before proceeding any further, we will adopt a slight change of notation: the index k of the information symbols u_k will from now on be in the range $0 \leq k \leq K-1$ (instead of $1 \leq k \leq K$ used before). Thus, the K-symbol long q-ary information sequences are of the form $\boldsymbol{u} = [u_0 \, u_1 \, ...u_{K-1}]$. Similarly, the codesymbol index n is in the range $0 \leq n \leq N-1$, i.e., the codewords of an (N, K) cyclic code are $\boldsymbol{v} = [v_0 \; v_1 \; ...v_{N-1}]$. The cyclic shift of the codeword for i positions is of the form $[v_{N-i} \; v_{N-i+1} \; ... \; v_{N-1} \; v_0 \; v_1 \; ... \; v_{N-i-1}]$. The code $\mathcal{C}$ is cyclic if every such shift, $0 \leq i \leq N-1$, is another codeword.

The generator matrix of an (N, K) cyclic code is of the form

$$\boldsymbol{G} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{N-K} & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{N-K-1} & g_{N-K} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & g_0 & g_1 & \cdots & g_{N-K} \end{bmatrix}, \tag{22}$$

where $g_0 \neq 0$, $g_{N-K} \neq 0$. This shape of $\boldsymbol{G}$ is a consequence of the cyclic property: if the first row $\boldsymbol{g}_1 = [g_0\ g_1\ ...g_{N-K-1}\ g_{N-K}\ 0\ ...\ 0]$ is a generator codeword with weight $\leq N-K$, and non-zero at the first and at the $(N-K)$th position, then any cyclic shift of it is also a codeword, linearly independent of it, and is thus also a generator codeword. Thus, the generators $\boldsymbol{g}_k$ are simply K cyclic shifts of the first row of $\boldsymbol{G}$.

The corresponding parity-check matrix, that satisfies $\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}$ is of the form

$$\boldsymbol{H} = \begin{bmatrix} h_K & h_{K-1} & ... & h_0 & 0 & ... & 0 \\ 0 & h_K & ... & h_1 & h_0 & ... & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & ... & h_K & h_{K-1} & ... & h_0 \end{bmatrix}, \tag{23}$$

where $h_0 \neq 0$, $h_K \neq 0$. Clearly, representation of a cyclic code via $\boldsymbol{G}$ or $\boldsymbol{H}$ is redundant—it suffices to know only the coefficients of the first row to be able to write the whole matrix. Therefore, this representation is seldom used for cyclic codes. Instead we use *polynomial representation* of q-ary cyclic codes, where polynomials are over the field $GF(q)$.

Instead of the information sequences $\boldsymbol{u}$, code sequences $\boldsymbol{v}$, generator matrix $\boldsymbol{G}$ and parity check matrix $\boldsymbol{H}$, we introduce

- Information polynomial: $u(D) = u_0 + u_1 D + ... + u_{K-1} D^{K-1}$
- Code polynomial: $v(D) = v_0 + v_1 D + ... + v_{N-1} D^{N-1}$
- Generator polynomial: $g(D) = g_0 + g_1 D + ... + g_{N-K} D^{N-K}$
- Parity polynomial: $h(D) = h_0 + h_1 D + ... + h_K D^K$

Note that we now write polynomials in D (instead of x) which is motivated by the fact that D denotes a delay element in the corresponding encoder realisation.

Now we can state that *the generator polynomial of a q-ary cyclic* $(N,\ K)$ *code is a polynomial* $g(D)$ *over* $GF(q)$ *of degree* $N-K$ and *it is the code polynomial of the smallest degree.* Moreover, this polynomial is *unique.* For most of cyclic codes used in practice, $q = 2^m$, for some m.

The generator polynomial $g(D)$ of a (N, K) cyclic code has the property that it divides the polynomial $D^N - 1$. In fact, it can be proved that any polynomial of degree $N-K$ that divides $D^N - 1$ is a generator polynomial of some (N, K) cyclic code.

The linear encoding rule $\boldsymbol{v} = \boldsymbol{u}\boldsymbol{G}$ in vector-matrix notation, corresponds to the following encoding rule in polynomial representation:

$$v(D) = u(D)g(D), \tag{24}$$

According to (24) we can state that $v(D)$ *is a code polynomial of a cyclic code if and only if it is divisible by the generator polynomial of that code.* For a $(N,\ K)$ cyclic code, the degree of a code polynomial is $\deg(v(D)) \leq N-1$, and the degree of an information polynomial is $\deg(u(D)) \leq K-1$.

As all other linear codes, cyclic codes also have a systematic encoder. The systematic generator matrix can be obtained from the generator matrix by row operations. The systematic code sequence $\boldsymbol{v}_{\mathrm{sys}} = [u_0\ u_1\ ...\ u_{K-1}\ v_K\ ...\ v_{N-1}]$, with first K positions equal to the information sequence $\boldsymbol{u}$, followed by $N-K$ parity check symbols, can be represented as the *systematic code polynomial* $v_{\mathrm{sys}}(D)$

$$v_{\mathrm{sys}}(D) = u(D) + p(D), \tag{25}$$

where the polynomial $p(D) = v_K D^K + v_{K+1} D^{K+1} + ... + v_{N-1} D^{N-1}$ contains the parity check symbols as its coefficients and it has degree $K \leq \deg(p(D)) \leq N$, or, equivalently, $\deg(D^{-K} p(D)) \leq N-K$. To obtain $p(D)$ we recall that every code polynomial must be divisible by the generator polynomial $g(D)$. Thus it holds

$$v_{\mathrm{sys}}(D) = u(D) + p(D) = a(D)g(D), \tag{26}$$

for some polynomial $a(D)$, or, equivalently, $u(D) = a(D)g(D) - p(D)$. From here we conclude that $p(D)$ is obtained as the *remainder of the long division*[4] *of* $u(D)$ *by* $g(D)$

$$p(D) = -\mathrm{Rem}\left(\frac{u(D)}{g(D)}\right),$$

[4] Long division, as opposed to the ”normal” polynomial division means that we divide polynomials starting from their lowest degree element. For example, long division $(1 + D + D^2)/(1 + D)$ yields $1 + D^2 + D^3 + ...$

where the division is stopped as soon as we obtain a remainder of degree $\geq K$. That remainder is $p(D)$.

Any cyclic code can also be defined via its *parity polynomial* $h(D)$, which is related to the generator polynomial via

$$g(D)h(D) = D^N - 1 \quad (\bmod\ (D^N - 1) = 0). \tag{27}$$

Note that this equation corresponds to the matrix equation $\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$. The parity polynomial is a polynomial of degree $\deg(h(D)) = K$, obtained from the generator polynomial via division

$$h(D) = \frac{D^N - 1}{g(D)}$$

Since $v(D) = u(D)g(D)$, from (27) we immediately obtain that every code polynomial must satisfy

$$v(D)h(D) = u(D)(D^N - 1) \quad (\bmod\ (D^N - 1) = 0) \tag{28}$$

which corresponds to the matrix equation $\boldsymbol{v}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$. We can rewrite (28) in an equivalent form

$$v(D) = \frac{u(D)}{h(D)}(D^N - 1) \tag{29}$$

The equations (24), (25) and (29) specify three different encoding rules for the same code. These encoders can be realised as linear sequential circuits using delay elements and modulo-p adders, as specified in the next section.

4.3 Encoding of cyclic codes

The following three encoders realise mappings defined by the equations (24), (25) and (28), respectively. They are often referred to as *time-domain* encoders. Note that for all encoders, $g_0 = 1$, $h_0 = 1$.

1. **Encoder 1:** *Non-systematic* encoder that generates codewords according to $v(D) = u(D)g(D)$, i.e., by multiplication of the information polynomial by the generator polynomial, as shown in Figure 4. This encoder is nothing else but an FIR filter (or a shift register) whose tap coefficients are the coefficients g_k of the generator polynomial. The codesymbols appear at the filter output during N clock cycles. Before the encoding begins, all delay elements are assumed to be in the zero state.

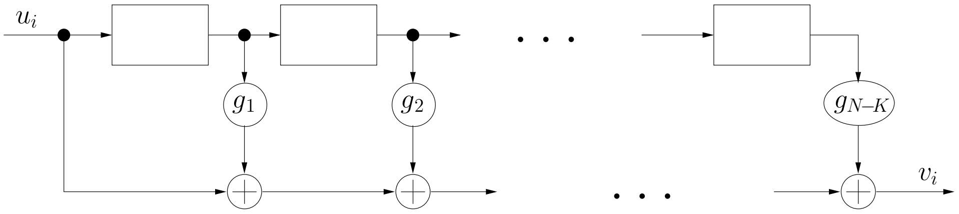


Figure 4: Encoder 1.

2. **Encoder 2:** *Systematic* encoder that outputs information symbols as the first K code symbols and subsequently generates parity checks via long division by the generator polynomial, $v(D) = u(D) + p(D), \quad p(D) = -\mathrm{Rem}\left(\frac{u(D)}{g(D)}\right)$, as shown in Figure 5. Before the encoding starts, all delay elements are in zero state. During the first K clock cycles, the switches are in position (1), and during the next $(N - K)$ clocks they are in position (2).

3. **Encoder 3:** *Systematic* encoder that generates the code symbols according to the division $v(D) = \frac{u(D)}{h(D)}(D^N - 1)$ realised as a linear feedback shift register (LFSR) whose coefficients are coefficients h_k of the parity polynomial, as shown in Figure 6. We say that $h(D)$ is the *connection polynomial* for this LFSR. Before the encoding starts, the K information symbols are loaded into the LFSR, such that they appear at the output during the first K clock cycles.

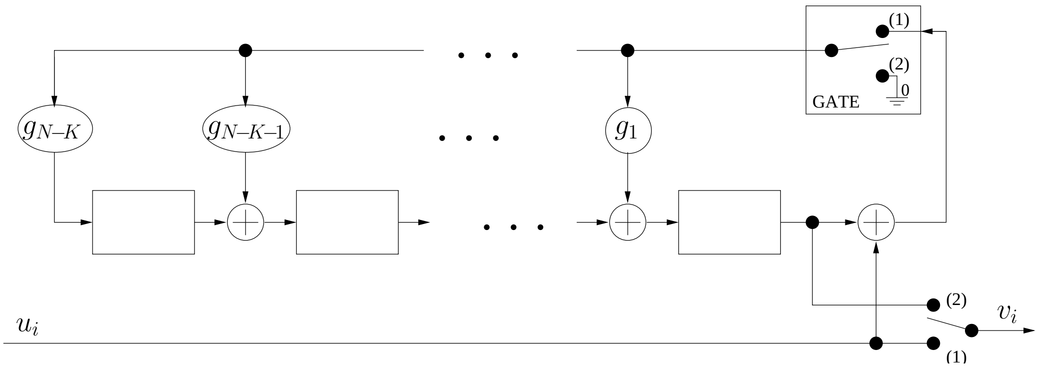


Figure 5: Encoder 2.

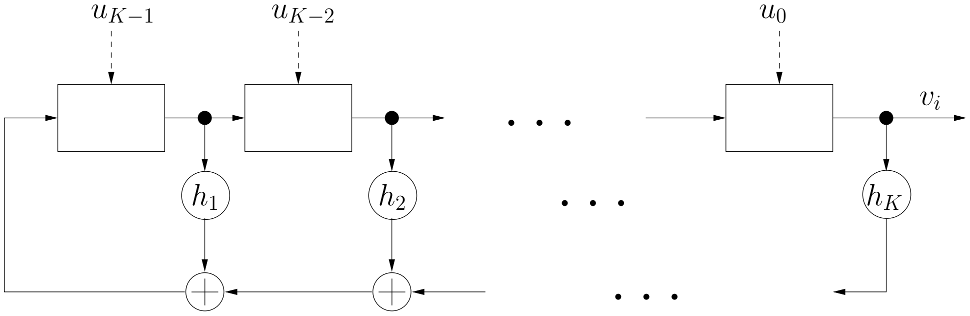


Figure 6: Encoder 3.

Note that the above encoders have complexity (in terms of the number of delay elements required for realisation) $N-K$, $N-K$ and K, respectively. If we want to use the non-systematic encoder, the complexity is $N-K$ delay elements. For systematic encoder, however, we have a choice between $N-K$ and K. For a given code, we should first evaluate these values and pick the realisation with the lower complexity.

4.4 Decoding of cyclic codes

For hard decision decoding, the received sequence $\boldsymbol{y}$ can be represented as a received polynomial $y(D) = y_0 + y_1 D + ... + y_{N-1} D^{N-1}$ of degree $\leq N-1$, and it equals

$$y(D) = v(D) + e(D) = a(D)g(D) + e(D),$$

where $e(D)$ is an error-pattern polynomial of degree $\leq N-1$. By dividing $y(D)$ with the generator polynomial $g(D)$ we obtain

$$\frac{y(D)}{g(D)} = a(D) + s(D),$$

where *the remainder of the division* is the *syndrome polynomial* $s(D)$ of degree $\deg(s(D)) \leq N-K-1$. The $N-K$ coefficients of the syndrome polynomial form the syndrome vector $\boldsymbol{s} = [s_{N-K-1} \; ... \; s_1 \; s_0]$, which is used for error detection and error correction, as for all other linear block codes. The computation of the syndrome and the decoder can also be realised as a linear sequential circuit with low complexity. We will not, however, go into more details on this topic.

Instead, we turn our attention now to one particular type of cyclic codes - namely, Reed-Solomon codes. We will study the basic properties of RS codes, as well as the encoding and the decoding methods.

5 Reed-Solomon (RS) codes

5.1 Definition and parameters

Definition: *A t-error correcting $(N,\,K)$ RS code is a cyclic q-ary code, whose generator polynomial $g(D)$ is a polynomial over $GF(q)$ that has $2t$ consecutive zeros in ω^0, ω^1, ω^2, ... ω^{2t-1}, i.e.,*

$$\begin{aligned} g(D) &= (1-D)(1-\omega^{-1}D)(1-\omega^{-2}D)...(1-\omega^{-(2t-1)}D) \\ &= 1+g_1D+g_2D^2+...+g_{2t}D^{2t} \end{aligned} \tag{30}$$

where ω is the element of $GF(q)$ of order N (i.e., $\omega^N=1$ and $\omega^k\neq 1,\ \forall k<N$)

Parameters:

- The field order q is, in practice, usually $q=2^m$, for some $m\geq 2$. (If $q=2^m$, all operations are modulo-2, and thus, every "minus" in the above definition can be replaced by a "plus").
- The length N of an (N,K) RS code is such that $N|(q-1)$. The usual choice is $N=q-1$, in which case the root ω is the primitive element α of $GF(q)$.
- Since the generator polynomial of any (N,K) cyclic code is of the degree $N-K$, it follows from the above definition that, for RS codes,

  $$N-K=2t \quad\Longrightarrow\quad K=N-2t$$

- The minimum distance of a t-error correcting $(N,\,K)$ RS code is

  $$d_{\min}=N-K+1=2t+1$$

  Thus, RS codes are maximum distance separable (MDS) codes.
- The parity polynomial is a degree-K polynomial obtained by $h(D)=(D^N-1)/g(D)$.

5.2 Frequency-domain analysis

Reed-Solomon codes belong to the larger group of so called BCH codes (named after their inventors, Bose, Chaudhuri and Hocquenghem). All the codes from this large group can be analysed in *frequency domain* as follows.

Discrete Fourier transform

A DFT of the codeword $\boldsymbol{v}$ is called the *frequency spectrum* of $\boldsymbol{v}$. It is an N-long q-ary sequence

$$\boldsymbol{V}=[V_0\ V_1\ ...\ V_{N-1}]=\mathrm{DFT}(\boldsymbol{v}),\quad V_j\in GF(q),\quad 0\leq j\leq N-1$$

These DFT coefficients specify the so called *spectrum polynomial*

$$V(D)=V_0+V_1D+...+V_{N-1}D^{N-1}$$

where the j-th spectral component V_j equals the code polynomial $v(D)$ evaluated at $D=\omega^j$:

$$V_j\triangleq v(D=\omega^j)=\sum_{i=0}^{N-1}v_i\omega^{ij},\ 0\leq j\leq N-1. \tag{31}$$

From (31) it follows directly that $D=\omega^j$ *is a root of the generator polynomial $g(D)$ (i.e., of every code polynomial $v(D)$) if and only if the j-th spectral component is* $V_j=0$.

Since the generator polynomial of an (N,K) RS code has $2t=N-K$ zeros in ω^j, $0\leq j<N-K$, we conclude that *the frequency spectrum of all codewords of an (N,K) RS code has zeros at the first $N-K$ positions*, $V_j=0$, $0\leq j<N-K$. This fact is often used as an alternative definition of the RS codes.

Alternative definition: *An (N,K) RS code is a set of N-tuples $\boldsymbol{v}$ whose frequency spectrum vanishes over a frequency range* $0\leq j<N-K$.

Thus, the frequency spectrum of RS codewords is of the form $\boldsymbol{V} = [\ \underbrace{0\,0 \ldots 0}_{N-K\text{times}}\ \ V_{N-K}\, V_{N-K+1} \ldots V_{N-1}]$.

The equivalent matrix notation of the DFT is $\boldsymbol{V} = \boldsymbol{v} \cdot \boldsymbol{T}$, where

$$\boldsymbol{T} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega & \omega^2 & \ldots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \ldots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \ldots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

Inverse Discrete Fourier transform

By the inverse DFT we can obtain a RS codeword from its frequency spectrum: $\boldsymbol{v} = \text{IDFT}(\boldsymbol{V})$. The jth codesymbol is obtained as

$$v_j = \frac{1}{N^*} V(D = \omega^{-j}) = \frac{1}{N^*} \sum_{i=0}^{N-1} V_i \omega^{-ij},\ j = 0, 1, \ldots, N-1, \tag{32}$$

where $N^* = N \mod (\lambda)$, where λ is the characteristic of $GF(q)$. Note that for the usual case when $q = 2^m$, the characteristic of $GF(2^m)$ is $\lambda = 2$. Then, if $N = q - 1 = 2^m - 1$, we have that $N^* = (2^m - 1) \mod (2) = 1$, and we do not need to worry about any scaling.

The equivalent matrix notation of the IDFT is $\boldsymbol{v} = \boldsymbol{V} \cdot \boldsymbol{T}^{-1}$

$$\boldsymbol{T}^{-1} = \frac{1}{N^*} \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \ldots & \omega^{-(N-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \ldots & \omega^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \ldots & \omega^{-(N-1)(N-1)} \end{bmatrix}$$

Note that $\omega^N = 1$ and thus, $\omega^{-k} \triangleq \omega^{N-k}$. Also, we have $\omega^{(N-1)(N-1)} = \omega^{N \cdot N} \cdot \omega^{-2N} \omega^1 = \omega$, etc.

5.3 Encoding and decoding of RS codes

As any other cyclic codes, RS codes can be encoded in time domain using any of the three encoders presented before: one non-systematic encoder with $N - K = 2t$ delay elements, and two systematic encoders with $N - K = 2t$ and K elements, respectively. For RS codes used in practice, $N - K < K$, and thus, realisations with $N - K$ elements are preferred.

Additionally, RS codes can be encoded in the *frequency domain* using the following *non-systematic* encoding procedure:

Step 1: *K non-zero spectral components are set equal to information symbols:*

$$\boldsymbol{V} = [\ \underbrace{0\,0 \ldots 0}_{N-K\text{times}}\ \ u_0\, u_1 \ldots u_{K-1}]$$

Step 2: Codeword $\boldsymbol{v}$ that corresponds to the encoded information sequence $\boldsymbol{u}$ is obtained via IDFT:

$$\boldsymbol{v} = [v_0\ v_1\ \ldots\ v_{N-1}] = \text{IDFT}(\boldsymbol{V}) = \boldsymbol{V} \cdot \boldsymbol{T}^{-1}.$$

Frequency domain is conveniently used for decoding of RS codes. If the received sequence is given by $\boldsymbol{y} = \boldsymbol{v} + \boldsymbol{e}$, where $\boldsymbol{e}$ is a q-ary error pattern, then the DFT of the received sequence yields

$$\boldsymbol{Y} = \boldsymbol{V} + \boldsymbol{E},$$

where $\boldsymbol{E} = \text{DFT}(\boldsymbol{e})$. The main idea of the decoding algorithm is to find the components of $\boldsymbol{E}$ in frequency domain, and then perform an IDFT to obtain the error-pattern $\boldsymbol{e}$. If $\boldsymbol{e}$ is properly estimated, we obtain the transmitted codeword by subtracting $\boldsymbol{e}$ from the received sequence. This decoding method is based on the *Berlekamp-Massey algorithm*.

5.4 Shortened RS codes (in general, cyclic codes)

Many classes of cyclic codes cannot be constructed for any arbitrary length N and number of information bits K. For example, the parameters of Reed-Solomon codes used in practice fulfill $N = 2^m - 1$, $m \geq 2$, and $K = N - 2t$, $t \geq 1$. Hence, the length of an RS codeword can be $N \in \{3, 7, 15, 31, 63, 127, 255, ...\}$. In standardized communication systems, the length of a data block is predefined, and, it often happens that the code of a suitable length cannot be found. In such cases, *shortening* of a cyclic code can be used as a technique to tailor the length of a code to the given system parameters.

Let $\mathcal{C}$ be an $(N,\, K,\, d_{\min})$ q-ary cyclic code. Code $\mathcal{C}$ has q^K codewords. With systematic encoding, information symbols appear as the first K symbols in a codeword. Now consider a set $\mathcal{C}_l$ of codewords that have the first l symbols equal to zero, $0 \leq l < K$ (in the limit case, for $l = 0$, $\mathcal{C}_0 = \mathcal{C}$). There are in total q^{K-l} such codewords and they are obtained by systematic encoding of information sequences whose first l positions are zero. The set $\mathcal{C}_l$ is a *subcode* of the code $\mathcal{C}$. If the l leading zeros are deleted from the codewords of $\mathcal{C}_l$ we obtain a set $\mathcal{C}_l^*$ of q^{K-l} words of length $N - l$. This set $\mathcal{C}_l^*$ is $(N - l, K - l)$ *shortened cyclic code*, and it is not cyclic! The minimum distance of $\mathcal{C}_l^*$ is at least $d_{\min}$; hence, the shortened cyclic code has at least the same error-correcting capabilities as the original (N, K) cyclic code $\mathcal{C}$.

An important feature of shortened cyclic codes is that the encoding and decoding can be easily accomplished by the same circuits used for the original code. When encoding, the $K - l$ information symbols are first padded with l leading zeros and fed into a systematic encoder for the code $\mathcal{C}$. The l leading zeros are deleted from the encoded codeword to obtain the $(N - l)$-symbol codeword. When decoding, the $(N - l)$-symbol sequence received from the channel is first padded with l leading zeros and subsequently fed into the decoder of the original cyclic code.

To conclude, by proper choice of l, a shortened cyclic code of any length $N - l$ and dimension $K - l$ can be obtained, without losing the capabilities of the original cyclic code, and without needing to devise new encoder and decoder.

5.5 Conclusions

To conclude our discussion about RS code, we briefly review the main features of this class of codes. We first recall that RS codes are MDS codes, hence, for given parameters N and K, they have largest possible minimum distance $d_{\min} = N - K + 1$, which guarantees that any error pattern with $t = (N - K)/2$ or fewer errors can be corrected. RS codes are q-ary codes, that is, codesymbols are elements of the field $GF(q)$. In practice, $q = 2^m$, for $m = 2, 3, 4, ...$, thus, codesymbols can be represented as binary m-tuples. If the error-correcting capability of an RS code is t (q-ary symbols), this in effect means that all blocks of tm bit-errors can be corrected. For example, if $m = 8$, each codesymbol is 1 byte (8 bits). If $t = 5$, blocks of as much as 40 erroneous bits can be corrected.

Note that if a code can correct t symbol errors that this does not mean that any error pattern of tm bit-errors can be corrected. It is necessary that these bit-errors do not affect more than t symbols. A codesymbol (binary m-tuple) is erroneous if at least one bit is in error. If a symbol is to be corrected, the code ”does not care” whether this error occurred due to 1 or m erroneous bits. To conclude, from an RS error-correction point of view, it is preferable that bit-errors appear *in blocks*, rather than *sparse*. As we will see later, quite the opposite holds for convolutional codes.

6 Convolutional codes

6.1 Introduction

In this section, we consider a ”non-block” coding technique known as *trellis* coding. We will restrict our attention to binary linear trellis codes, which are called binary *convolutional codes*. (Unlike block codes, convolutional codes are virtually always binary, and thus, we will hereinafter omit this attribute). In some sense, convolutional codes are a generalization of block codes, but in another, they can be viewed as a special type of block codes. (These statements will be clarified throughout this chapter.) However, the design criteria, and the decoding methods of convolutional codes differ greatly from traditional block codes. In many theoretical and practical aspects, convolutional codes are found to be superior to traditional block codes. This is the reason why convolutional codes are the most widely used codes in existing communication systems.

6.2 Convolutional codes, encoders and encoding matrices

Convolutional codes can be defined in a very formal mathematical way, which is somewhat hard to grasp when we encounter this type of codes for the first time. Therefore, the approach followed is this tutorial will be rather to explain everything in a descriptive way and through illustrative examples.

First, we recall that for the purpose of block coding the information sequence was parsed into blocks of length K, and each block was *independently* encoded as a block of N code symbols. For convolutional codes, we follow a different approach:

The sequence of information symbols is fed into a *linear sequential circuit with memory* in a *continuous way*, so that the symbols of the output sequence depend not only on the current input, but also on the previous inputs. This linear sequential circuit is called a *convolutional encoder*, and the *set* of output code sequences is called a *convolutional code.* The attribute "convolutional" stems from the fact that the output code sequence is obtained as a convolution of the input information sequence and the impulse response of the encoder. An encoder is built using modulo-2 adders and memory (delay) elements.

Note that hypothetically, if we do not stop our encoding process, the input and the output sequences can be of *infinite* length – as long as our encoding circuit is in operation, and there are information bits at its input to be encoded, it will produce code symbols at the output. Therefore, it is inconvenient to define the code rate as $R = K/N$, as for block codes, because now K and N may be infinitely large. Instead, we can observe our encoder and see that, in general, at any time instant, a small group of b bits enters the encoder, and c code symbols appear at the output. Therefore, the *rate of a convolutional code* is defined as

$$R = \frac{b}{c},$$

and, as before, $0 \leq R \leq 1$.

To illustrate what we have said so far, let us consider an example of the convolutional encoder shown in Figure 7. The encoder is realized with 2 delay elements, thus its memory is $m = 2$. At any time instant t, one information bit u_t enters the encoder, and two code symbols, $v_t^{(1)} v_t^{(2)}$ appear at the output. Thus, this is an encoder of a rate $R = 1/2$ convolutional code.

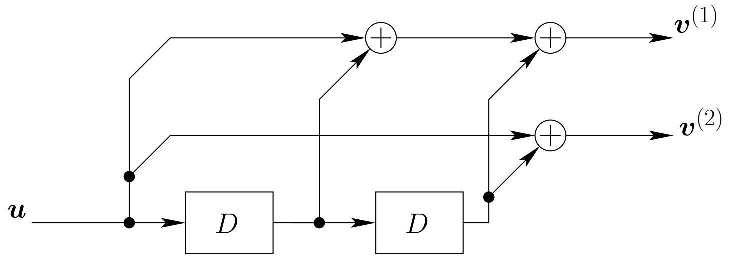


Figure 7: An encoder of a rate $R = 1/2$ convolutional code, realized in controller canonical form.

We can write the input-output relations for this encoder:

$$\begin{aligned} v_t^{(1)} &= u_t \oplus u_{t-1} \oplus u_{t-2}, \\ v_t^{(2)} &= u_t \oplus u_{t-2} \end{aligned} \tag{33}$$

Now, we can group the b information bits at any time instant t into a vector $\boldsymbol{u}_t = [u_t^{(1)} \; u_t^{(2)} \; ... \; u_t^{(b)}]$ of length b, where in our case $b = 1$ and thus $\boldsymbol{u}_t = [u_t]$ (superscript $^{(1)}$ omitted). Similarly, we group the c output code symbols into a vector $\boldsymbol{v}_t = [v_t^{(1)} \; v_t^{(2)} \; ... \; v_t^{(c)}]$ of length c, which for our example yields $\boldsymbol{v}_t = [v_t^{(1)} \; v_t^{(2)}]$. Then the relations (33) can be rewritten as

$$\boldsymbol{v}_t = \boldsymbol{u}_t \boldsymbol{G}_0 + \boldsymbol{u}_{t-1} \boldsymbol{G}_1 + \boldsymbol{u}_{t-2} \boldsymbol{G}_2 \tag{34}$$

where the $b \times c = 1 \times 2$ matrices $\boldsymbol{G}_i$ are

$$\begin{aligned} \boldsymbol{G}_0 &= \begin{bmatrix} 1 & 1 \end{bmatrix} \\ \boldsymbol{G}_1 &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \boldsymbol{G}_2 &= \begin{bmatrix} 1 & 1 \end{bmatrix} \end{aligned}$$

In general, for an encoder with memory m, the relation (34) is

$$\boldsymbol{v}_t = \boldsymbol{u}_t\boldsymbol{G}_0 + \boldsymbol{u}_{t-1}\boldsymbol{G}_1 + ... + \boldsymbol{u}_{t-m}\boldsymbol{G}_m, \tag{35}$$

where the $\boldsymbol{G}_i$ describe the input-output relation for delays $0 \leq i \leq m$. These matrices are *time-invariant*, i.e., the above input-output relation is valid *for any time t*. Without loss of generality, assume that the encoding process started at time instant $t = 0$. Then, the complete information sequence $\boldsymbol{u}$ is a sequence of input b-tuples $\boldsymbol{u}_t$ that have appeared at time instants $t = 0, 1, 2, ...$

$$\boldsymbol{u} = [\boldsymbol{u}_0\ \boldsymbol{u}_1\ \boldsymbol{u}_2\ ...].$$

Analogously, the complete code sequence $\boldsymbol{v}$ is a sequence of output c-tuples $\boldsymbol{v}_t$ that have appeared at time instants $t = 0, 1, 2, ...$

$$\boldsymbol{v} = [\boldsymbol{v}_0\ \boldsymbol{v}_1\ \boldsymbol{v}_2\ ...].$$

Then we can write (35) simply as

$$\boldsymbol{v} = \boldsymbol{u}\boldsymbol{G}, \tag{36}$$

where

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_0 & \boldsymbol{G}_1 & \dots & \boldsymbol{G}_m & \boldsymbol{0} & \dots \\ \boldsymbol{0} & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \dots & \boldsymbol{G}_m & \\ \vdots & & \ddots & \ddots & & \ddots \\ & & & & & \end{bmatrix}$$

is the *generator matrix* of the convolutional code. Thus, we have arrived to the same *linear encoding rule* as for linear block coding. We assume that before the encoding started, the encoder was in the *all-zero* state. Hypothetically, $\boldsymbol{u}$ and $\boldsymbol{v}$ can be *semi-infinite* if the encoding goes on forever. In practice, we have to *terminate* these sequences at some point, say after transmitting n b-tuples. For the encoder from our example, we will terminate the transmission by sending additional $mb = 2$ "dummy" zeros, which will force the encoder back to the all-zero state, where it started (this termination ensures that the last information bits are equally well protected as all the others). Thus, "zero-tail" termination yields finite information sequences $\boldsymbol{u}$ of length $K = nb$, which are encoded as code sequences $\boldsymbol{v}$ of length $N = nc+mc$, where the encoding rule is given by the finite generator matrix $\boldsymbol{G}$ of size $K \times N$.

In our example, zero-tail termination yields the generator matrix

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_0 & \boldsymbol{G}_1 & \boldsymbol{G}_2 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \boldsymbol{G}_2 & & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{G}_0 & \boldsymbol{G}_1 & \boldsymbol{G}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Thus, *by terminating a convolutional code we obtain a linear block code* of the rate $R_B = K/N = (nb)/(nc + mc) = R \cdot n/(n + m)$. Note that termination results in a slight rate reduction ($R_B < R$), but for large n, this is negligible ($R_B \approx R$).

Now, we introduce the so called *D-domain* code representation, which is more convenient than the time-domain representation considered so far. The information sequences $\boldsymbol{u}$, code sequences $\boldsymbol{v}$ and generator matrix $\boldsymbol{G}$ can be written as *polynomials over binary field in D*:

$$\begin{aligned} \boldsymbol{u}(D) &= \boldsymbol{u}_0 + \boldsymbol{u}_1 D + \boldsymbol{u}_2 D^2 + ... \\ \boldsymbol{v}(D) &= \boldsymbol{v}_0 + \boldsymbol{v}_1 D + \boldsymbol{v}_2 D^2 + ... \\ \boldsymbol{G}(D) &= \boldsymbol{G}_0 + \boldsymbol{G}_1 D + ... + \boldsymbol{G}_m D^m, \end{aligned} \tag{37}$$

which yields the linear encoding rule in D-domain

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)\boldsymbol{G}(D).$$

For our example, the generator matrix in D-domain is

$$\boldsymbol{G}(D) = \boldsymbol{G}_0 + \boldsymbol{G}_1 D + \boldsymbol{G}_2 D^2 = \begin{bmatrix} 1 + D + D^2 & 1 + D^2 \end{bmatrix} \tag{38}$$

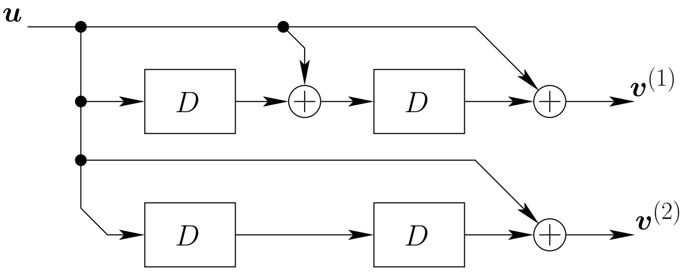


Figure 8: A rate $R = 1/2$ encoder for the $(7,5)_8$ generator matrix, realized in observer canonical form.

Note that the coefficients of the D-polynomials in the matrix can be written as binary $(m+1)$-tuples (where m is the memory). Thus we have coefficients $(111,\ 101)_2$, or, in octal representation, $(7,5)_8$. This kind of notation is most commonly used to describe a generator matrix of a convolutional code.

The highest degree of D in the ith row of the generator matrix $\boldsymbol{D}$ is called i-th *constraint length* ν_i, $1 \leq i \leq b$. The sum of all the constraint lengths is the *overall constraint length* $\nu = \sum_{i=1}^{b} \nu_i$, and the maximum value is *memory* $m = \max_{1 \leq i \leq b} \nu_i$. In general, $\nu \leq mb$ holds. Clearly, for $b = 1$, i.e., for rate $R = 1/c$ codes we have $\nu_1 = \nu = m$.

The encoder realization of the type shown in Figure 7 is in the so called *controller canonical form* (CCF). In general, for a rate $R = b/c$ code, the *CCF encoder has b shift registers, one for each input, where the shift register for the i-th input has* ν_i *memory elements.* Thus the total number of memory elements is ν. For our example, the CCF of the $(7,5)_8$ encoder has $m = \nu = 2$ memory elements.

As we have already suspected, the CCF is not the only encoder realization of the given generator matrix $\boldsymbol{G}(D)$. In general, *every convolutional code has many generator matrices, and every generator matrix has many encoder realizations*!

One more commonly used encoder realization is the so called *observer canonical form* (OCF)[5]. In general, for a rate $R = b/c$ code, the *OCF encoder has c shift registers, one for each output.* To be more precise, these shift registers are not really shift registers as the ones in the CCF, since "the chain" of memory elements can be "broken" with modulo-2 adders between any two memory elements, such that intermediate elements in the chain can be accessed from outside by different inputs.

For illustration, Figure 8 depicts the OCF encoder realization of the $(7,5)_8$ generator matrix (38). Note that this realization requires twice as many memory elements as the CCF realization of the same $\boldsymbol{G}(D)$.

In general, we are always interested in finding the *minimal* realization of the encoder, i.e. the realization that requires the smallest number of memory elements. A generator matrix that has an encoder realization with smallest number of memory elements (among all realizations) is called the *minimal* generator matrix, and the corresponding realization is the *minimal encoder.* In general, the minimal encoder need not be in CCF or OCF, and in some cases, the problem of finding the minimal encoder is rather complicated. Therefore, we are particularly interested in a special class of polynomial generator matrices that have the nice property that *the CCF encoder realization is the minimal encoder.* We call such matrices *minimal-basic* matrices. Our matrix (38) is an example of a minimal-basic matrix.

As we mentioned before, one convolutional code can be generated by many different generator matrices. If two $b \times c$ generator matrices $\boldsymbol{G}(D)$ and $\boldsymbol{G}'(D)$ generate the same code, we say that these matrices are *equivalent.* Then it holds that there exists a non-singular $b \times b$ matrix $\boldsymbol{T}(D)$ such that

$$\boldsymbol{G}(D) = \boldsymbol{T}(D)\boldsymbol{G}'(D).$$

Using this definition of equivalence, it can be easily shown that *every generator matrix* $\boldsymbol{G}(D)$ *has an equivalent (rational) systematic generator matrix* $\boldsymbol{G}_{\text{sys}}(D)$. A systematic generator matrix is of the form

$$\boldsymbol{G}_{\text{sys}}(D) = [\boldsymbol{I}_b \mid \boldsymbol{R}(D)] \tag{39}$$

[5] The attribute "controller" in CCF implies that we can control the states of the memory elements by the inputs, while the "observer" in OCF means that we can observe the states by observing the outputs.

where $\boldsymbol{I}_b$ is the $b \times b$ identity matrix, and $\boldsymbol{R}(D)$ is a possibly *rational* $b \times (c-b)$ matrix. A systematic matrix defines the systematic encoding rule, $\boldsymbol{v}_{\text{sys}}(D) = \boldsymbol{u}(D)\boldsymbol{G}_{\text{sys}}(D)$, according to which, at any time instant, the b input information bits appear as the first b (out of c) encoder outputs.

Here we briefly point out what we have already intuitively understood: a generator matrix $\boldsymbol{G}(D)$ is said to be *polynomial* if all its elements are polynomials in D. If, however, there is at least one element in $\boldsymbol{G}(D)$ that is a *ratio* of two polynomials, then we say that such a matrix is *rational.* As we will soon see, the encoder realization of a rational matrix always contains a *feedback* loop (from the output to the input). Due to that, encoders of rational matrices are sometimes called *feedback* encoders, while the encoders of polynomial matrices are referred to as *feedforward* encoders.

To illustrate the structure of a rational systematic matrix, we return to our $(7,5)_8$ example, and we note that the generator matrix $\boldsymbol{G}(D) = [1+D+D^2 \;\; 1+D^2]$ can be written as

$$\boldsymbol{G}(D) = \underbrace{(1+D+D^2)\cdot}_{\boldsymbol{T}(D)} \underbrace{\begin{bmatrix} 1 & \dfrac{1+D^2}{1+D+D^2} \end{bmatrix}}_{\boldsymbol{G}_{\text{sys}}(D)}.$$

Thus, our polynomial non-systematic generator matrix $\boldsymbol{G}(D)$ is equivalent to a rational systematic generator matrix $\boldsymbol{G}_{\text{sys}}(D)$ (i.e., they both generate the same code, only the input-output mapping is different). The encoder realization for $\boldsymbol{G}_{\text{sys}}(D)$ is shown in Figure 9. We see that the input information bit appears as the first code output, $v_t^{(1)} = u_t$.

Systematic generator matrices have several important properties. First, they are always *minimal*, i.e., they have a realization with minimum number of memory elements (however, this realization need not necessarily be CCF or OCF). Furthermore, systematic matrices are always *non-catastrophic.*

We have not introduced this concept so far, but the name suggests that being catastrophic is something very bad, and indeed it is. We say that a convolutional generator matrix is *catastrophic* if it defines such a mapping that there exists an information sequence of infinite weight (with infinitely many ones) that is encoded into a finite-weight code sequence (with only a few ones). Why is this a "catastrophe"? – Because, if an optimum decoder decodes only a few code symbols erroneously this results in infinitely many bit errors!

Now it is clear why a systematic matrix can never be catastrophic - the systematic mapping ensures that infinite-weight input sequences result in infinite-weight output code sequences.

Note that being catastrophic is not a code property, but a generator matrix property, i.e., every code has catastrophic and non-catastrophic generator matrices.

Finally, we point out that we always consider generator matrices such that $\boldsymbol{G}(D=0) = \boldsymbol{G}_0$ is of full rank b (this ensures that the first zero-output c-tuple corresponds only to the zero-input b-tuple). Such generator matrices are called *encoding* matrices. All the generator matrices discussed here are encoding matrices.

The properties of the generator matrices and their relations are summarized in Figure 10.

At the end, we give an example of a rate $R = b/c$ code, where $b > 1$. Consider the rate $R = 2/3$

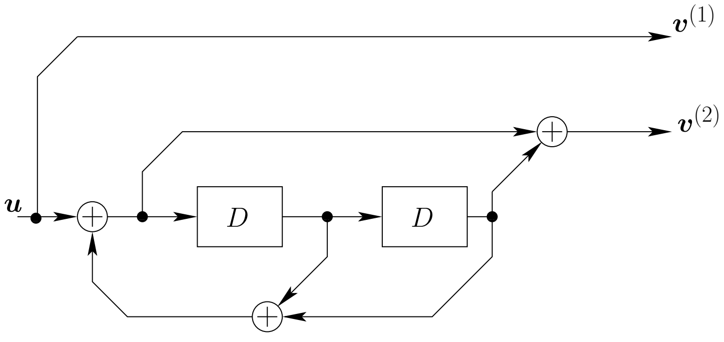


Figure 9: Rate $R = 1/2$ systematic encoder with feedback.

convolutional code whose generator matrix is

$$\boldsymbol{G}(D) = \boldsymbol{G}_0 + \boldsymbol{G}_1 D + \boldsymbol{G}_2 D^2 = \begin{bmatrix} 1+D & D & 1 \\ D^2 & 1 & 1+D+D^2 \end{bmatrix}$$

where

$$\boldsymbol{G}_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \; \boldsymbol{G}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; \boldsymbol{G}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

The generator matrix in time domain is

$$\boldsymbol{G} = \begin{bmatrix} 1\,0\,1 & 1\,1\,0 & 0\,0\,0 & & \\ 0\,1\,1 & 0\,0\,1 & 1\,0\,1 & & \\ & 1\,0\,1 & 1\,1\,0 & 0\,0\,0 & \\ & 0\,1\,1 & 0\,0\,1 & 1\,0\,1 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}$$

The constraint lengths of $\boldsymbol{G}(D)$ are $\nu_1 = 1$ and $\nu_2 = 2$. Thus, the memory is $m = 2$, and the overall constraint length $\nu = 3$. Thus, the CCF realization requires in total 3 memory elements, and it is shown in Figure 11. Finding the equivalent systematic generator matrix is left as an exercise.

6.3 State-transition diagrams and trellises

Since a convolutional encoder is a linear sequential circuit, i.e., a *finite-state machine*, its operation can be described by a *state-transition diagram*. A *state* $\boldsymbol{\sigma}_t$ of the encoder at time t is defined as the contents of its memory elements at time t. For a *controller canonical form* realization of the rate $R = b/c$ encoder, the state is defined by the previous encoder inputs. Since in CCF there are $\nu \leq mb$ memory elements, the encoder has 2^ν different states, and they can be written as binary ν-tuples

$$\boldsymbol{\sigma}_t = [u_{t-1}^{(1)}\, u_{t-2}^{(1)} \dots u_{t-\nu_1}^{(1)}\; u_{t-1}^{(2)} \dots u_{t-\nu_2}^{(2)} \dots u_{t-1}^{(b)} \dots u_{t-\nu_b}^{(b)}]$$

For a rate $R = 1/c$ memory m encoder in CCF, a state is simply given by

$$\boldsymbol{\sigma}_t = [u_{t-1}\, u_{t-2} \dots u_{t-m}].$$

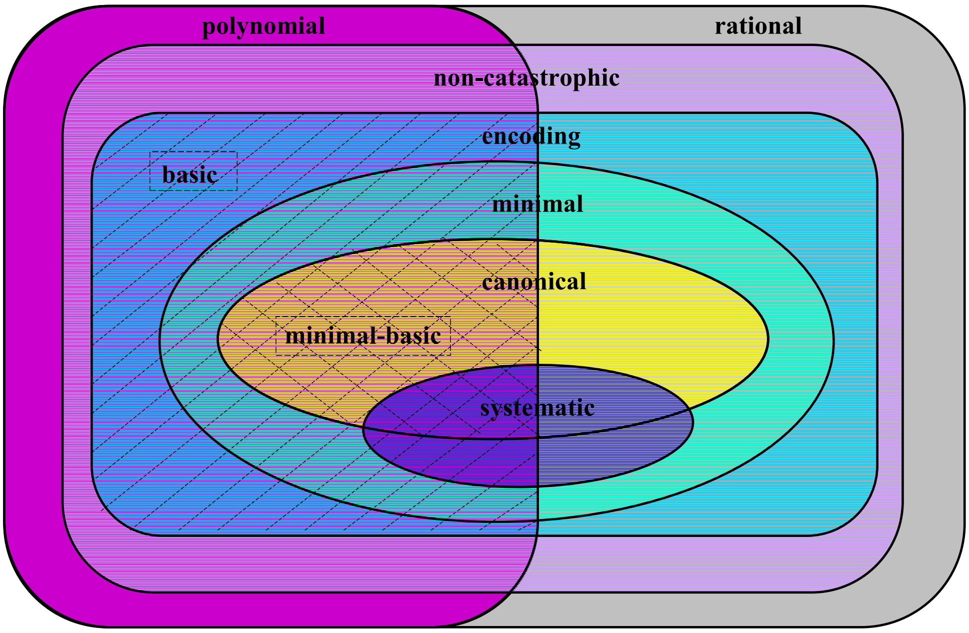


Figure 10: Different properties of a convolutional generator matrix and their relation.

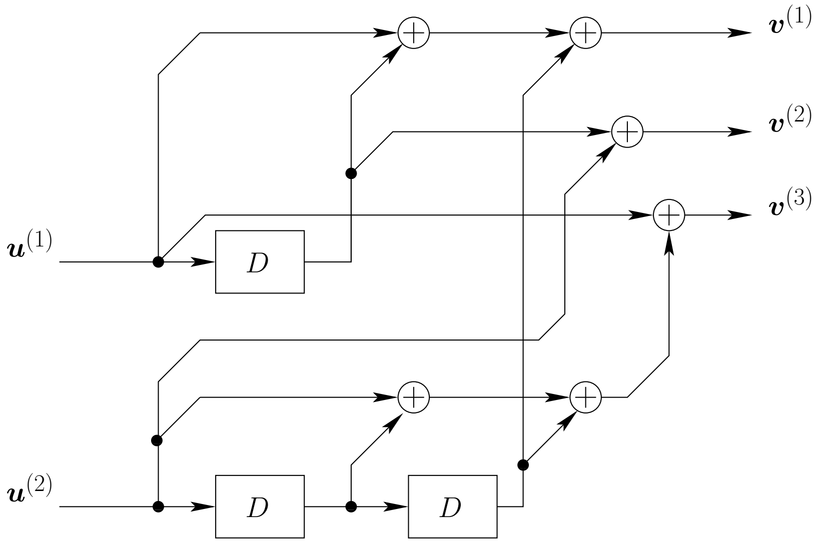


Figure 11: CCF realization of the convolutional encoder for the rate 2/3 code.

If we represent all possible states as nodes, and connect every state (by an oriented branch) to the states that are reachable from it, we obtain a graph that is called a state transition diagram. We label each branch with an input b-tuple that causes this transition and with the corresponding output c-tuple. For example, for a rate $R = 1/2$ encoder, labels are of the form $u_t/v_t^{(1)}v_t^{(2)}$.

The state-transition diagram for the encoder from Figure 7 is shown in Figure 12. From there we, for example, read: if the encoder is in state 00 (e.g. at the beginning of the encoding process), and a 0 appears at its input, clearly, the encoder stays in the zero-state, which is marked by a loop around this sate, and the output will be a pair of zeros. If, however, a 1 appears at its input, the decoder will output $v_t^{(1)}v_t^{(2)} = 11$ and move to a new state 10, and so on.

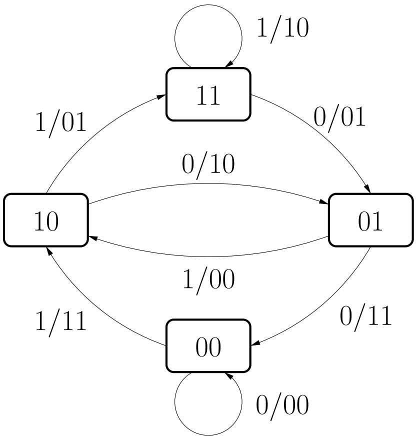


Figure 12: State transition diagram for $(7,5)_8$ rate 1/2 convolutional encoder realized in the CCF.

Another way to graphically represent the dynamics of the encoder, i.e., its operation over time is the *trellis diagram* (shortly, trellis). A trellis for the $(7,5)_8$ CCF encoder is shown in Figure 13. For every time-instant, the trellis has 2^ν nodes, representing possible encoder states. Trellis branches depict transitions from state to state, and they are labelled with the corresponding output c-tuple. Thus, *every path in the trellis, from its root till its end, corresponds to one code sequence.*

For a $R = b/c$ code, there are exactly 2^b branches leaving each node, and arriving to each node. They correspond to 2^b possible input b-tuples at every time instant. For time-invariant codes, which we consider here, we see that a trellis is regular – its structure is the same for every time instant.

For a rate $R = 1/c$ code, there are 2^m states at every trellis depth, and there are 2 branches leaving every state – the upper one corresponds to input 0, the lower one corresponds to input 1. We assume that the encoder is initially in the all-zero state. Thus, when the encoding begins, we have a "start-up" phase during which not all states are reachable (because it takes m steps to fill up the encoder's memory elements). After m sections, the trellis is fully developed and from that time point on, its structure is invariant. Zero-tail termination of the encoding process is performed by sending mb "dummy" input bits such that they force the encoder back into the all-zero state. Note that these dummy bits are zeros for the controller canonical form realization, but for other encoder realizations (e.g. for encoders with feedback!) they are non-zero bits. The last m trellis sections denote the termination phase, when only transitions that lead to the zero state occur. The trellis representation of convolutional codes is very convenient for both analyzing their properties and for performing decoding. Before explaining how we decode convolutional codes, we will now define a very important code parameter – the *free distance.*

6.4 Free distance and error correcting capability of a convolutional code

The free distance of a convolutional code is a parameter equivalent to the minimum distance of a block code.

Definition: *Free distance of a convolutional code $\mathcal{C}$ is the minimum Hamming distance between any two differing codewords,*

$$d_{\text{free}} = \min_{\boldsymbol{v},\boldsymbol{v}' \in \mathcal{C},\ \boldsymbol{v} \neq \boldsymbol{v}'} \{d_{\text{H}}(\boldsymbol{v}, \boldsymbol{v}')\}. \tag{40}$$

Due to the linearity of a convolutional code we immediately conclude that the free distance is the minimum Hamming weight of a non-zero code sequence

$$d_{\text{free}} = \min_{\boldsymbol{v} \in \mathcal{C},\ \boldsymbol{v} \neq \mathbf{0}} \{w_{\text{H}}(\boldsymbol{v})\}. \tag{41}$$

Now we recall that every code sequence corresponds to a path in the trellis. Thus, the free distance of the code is the smallest weight of a path that *diverges* (makes a detour) from the all-zero path and *merges* (returns back) to it. Note that this is not necessarily the weight of the shortest such path! Due to time-invariance, we can always assume that the diverging path starts at the root of the trellis.

For example, from the trellis from Figure 13, we conclude that the free distance of our $(7,5)_8$ code is $d_{\text{free}} = 5$. It is the weight of the path that goes through the following sequence of states : $0 \to 2 \to 1 \to 0 \to 0...$ (we use decimal notation for states) and outputs the sequence 11 01 11 followed by infinitely many zeros, which we do not write. In this example, the $d_{\text{free}}-$path is the shortest diverging path. Note that the smallest possible length of the $d_{\text{free}}-$path is equal to $(m+1)$ (branches), because this is how much time we need to force the encoder back to all-zero state (for encoder in CCF). Thus, we intuitively conclude that by increasing the memory we can obtain codes with larger free distance. We will return to this point shortly.

As for the linear block codes, the can prove that *a convolutional code with free distance d_{free} can*

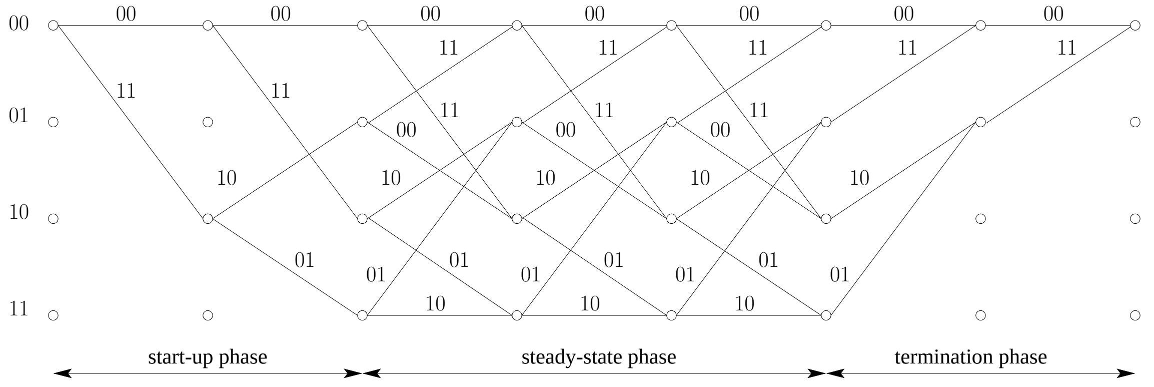


Figure 13: Trellis diagram of the $(7,5)$ rate $1/2$ convolutional encoder in CCF.

correct all error patterns of weight not larger than

$$t = \left\lfloor \frac{d_{\text{free}} - 1}{2} \right\rfloor .$$

The free distance determines the performance of the convolutional code at higher SNR. At low SNR, other distance measures are of larger importance, but we will not analyze them here.

Due to the linearity of the code, when we investigate the decoding of convolutional codes, we can always assume that the all-zero sequence is transmitted. Then, the decoder makes a decoding error when it diverges from the all-zero path. This happens if a channel introduces a *burst* of more than t *consecutive* errors. The larger the memory of the code (the larger the minimum distance), the more consecutive errors are necessary to force the decoder to leave the correct all-zero path, and thus the more unlikely it is that the decoder makes a decoding mistake. But, once it diverges from the correct path, it takes a lot of time to come back to it, if the memory is large! Thus, a decoder of a strong convolutional code will very rarely make a mistake, but when it does, it will cause a burst of many bit errors.

On the other hand, if the channel introduces many *sparse* errors - this is not a problem for a strong code (because errors need to appear in bursts to force the decoder to err). In fact, over a long sequence, if channel errors are sparse (where their sparseness is observed relative to the code's memory) the number of channel errors can be much greater than t - the code will correct them all! Thus we have arrived to the fundamental result *if the channel errors are sparse, a convolutional code can correct many more errors than what is guaranteed by its free distance*! We can thus say that convolutional codes "prefer sparse errors to error bursts". Note that this behaviour is exactly dual to what we have said about Reed-Solomon codes (to remind you, RS codes "prefer error bursts")! So, why not combining these two codes somehow? This is precisely the motivation for *concatenated coding schemes*, which are explained in the last section of this tutorial.

Before explaining concatenated schemes, we need to explain how we perform the optimum (ML) decoding of convolutional codes.

6.5 Viterbi algorithm

The *Viterbi algorithm* (named after its inventor Andrew Viterbi) is the maximum-likelihood decoding algorithm that finds the ML codeword by searching the code trellis. Recall from before that ML decoding is the *minimum distance decoding* – for hard-decision decoding we minimize the Hamming distance between the received and the estimated codeword (for soft-decision decoding we minimize the Euclidean distance).

The Viterbi algorithm proceeds through the code trellis and finds the path with the smallest distance from the received sequence – this is the ML path! For rate $R = 1/c$ codes, there are 2 branches merging at every node. At each step, the path metrics are compared, and the worse one is discarded. The metric accumulates along the trellis. At the end, the surviving path is the ML path.

To illustrate the Viterbi algorithm, consider again our $(7, 5)_8$ code and its trellis. Assume we have used this code to encode information sequences of length $n = 3$. To force the encoder back into the all-zero state, we need $m = 2$ dummy zeros. Thus, the encoded sequence has length 5 2-tuples. Assume we communicate over a binary symmetric channel and we have received a sequence $\boldsymbol{y} = [11\,00\,11\,00\,10]$. The Viterbi algorithm for finding the ML codeword estimate is shown in Figure 14 below. Bold letters denote the accumulated Hamming weight at each node. At depth > 2, at every node, two arriving paths are compared and the one with worse metric is discarded (discarded paths are shown in dashed lines). At the end, we have one surviving path (bold line) which is the ML solution: $\hat{\boldsymbol{v}} = [11\,10\,11\,00\,00]$, which corresponds to the information sequence $\hat{\boldsymbol{u}} = [1\,0\,0]$. The distance of $\hat{\boldsymbol{v}}$ from the received sequence is $d = 2$. If the estimated codeword was indeed the transmitted one, we have managed to correct two errors.

6.6 Punctured convolutional codes

A trellis of a rate $R = b/c$ convolutional code has 2^b branches leaving from and arriving to each node. With increased rate (i.e., b), branch complexity exponentially increases, which makes the trellis-based decoding algorithms, such as the Viterbi algorithm, computationally very costly. One way of obtaining a high rate convolutional code, while keeping the branch complexity equal to 2 per each node (as is the

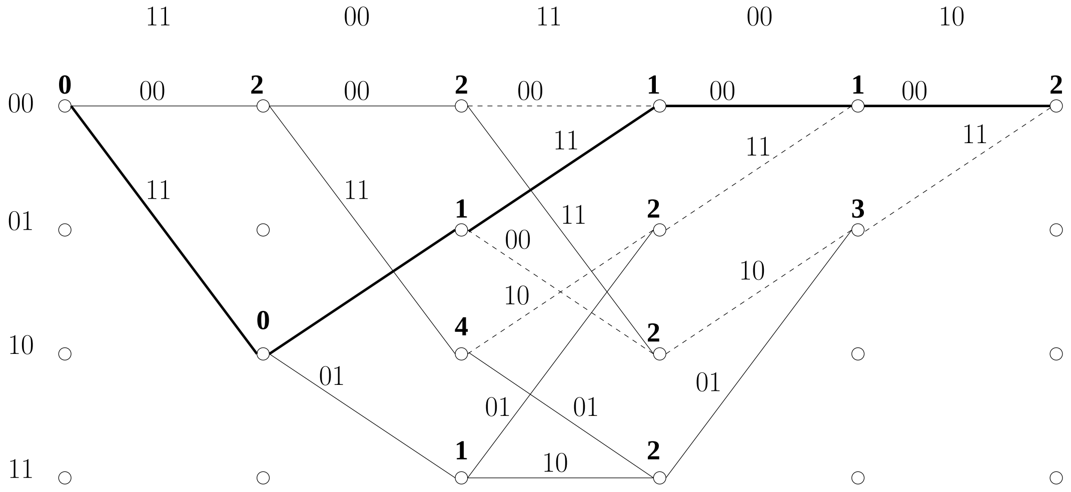


Figure 14: Decoding example.

case with $R = 1/c$ codes) is to *puncture* the original low-rate convolutional code, which is also called *the mother code*. Puncturing a code simply means periodically deleting certain code symbols from the code sequence of a mother code. Since the length of the encoded information sequence is kept the same, while some codesymbols from the corresponding code sequence are deleted, it is clear that puncturing always *increases the rate* as compared to the mother code. Using a mother code of rate $R = 1/2$, convolutional codes of rates $R = (n-1)/n$, where $n \geq 3$, can be obtained by proper puncturing.

Puncturing rule is specified by the *puncturing matrix* $\boldsymbol{P}$. This is a binary matrix of size $c \times T$, where c is the number of encoder outputs of the rate $R = 1/c$ mother code, and T is the length of the period of the puncturing sequence. Each row is the matrix $\boldsymbol{P}$ specifies the puncturing pattern for the corresponding encoder's output: a '0' denotes that the symbol is deleted (punctured), and a '1' denotes that the symbol is left unchanged. The total number of ones in the matrix $\boldsymbol{P}$, denoted as $s = \sum_{i,j}(\boldsymbol{P})_{ij}$ is the number of codesymbols that the encoder of the punctured code outputs for every T inputs. Hence, the rate of the punctured code is $R_p = s/T$.

Example: Let the puncturing matrix for a rate $R = 1/2$ mother code be

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The first row indicates that the puncturing sequence for the first encoder's output is 0101010101..., that is, every other symbol should be deleted, while the second row specifies that every fourth symbol should be deleted from the second encoder's output. Hence, out of 8 symbols that the encoder produces in $T = 4$ time intervals, only $s = 3$ are left unpunctured, which yields the rate of the punctured code $R_p = 3/4$.

In general, puncturing patterns need to be carefully designed, so that the newly obtained punctured codes have good properties.

Decoding of punctured codes is easily performed using the trellis of the mother code. When computing the path metric along the trellis, the punctured positions do not contribute to the metric.

In many practical applications, a communication system needs to support several different code rates (corresponding to different levels of quality of service). Then it is convenient to use *rate-compatible punctured convolutional codes* (RCPCCs) – a set of codes of different rates obtained from the same mother code in such a way that the codewords of a higher-rate code are obtained from the codewords of a lower-rate code by deleting additional symbols. Hence, the puncturing matrices of the RCPCCs all have the same period T and have the property that starting from the lowest-rate code up to the highest, simply more and more 1s are replaced by 0s. All the codes are decoded on the same trellis of the mother code. RCPCCs are particularly convenient for automatic-repeat-request (ARQ) systems, where the transmission starts with a high rate code and, upon a retransmission request, only punctured symbols are subsequently transmitted, resulting in a lower-rate code and thus better error-protection.

6.7 Code concatenation

The motivation of code concatenation is to exploit the virtues of both RS codes and convolutional codes. The scheme is depicted in Figure 15. A convolutional code is used as inner code (closer to the channel). It will correct most of the channel errors but it may from time to time produce bursts of bit errors. A RS code used as outer coder (closer to the source encoder/decoder) is very well suited to cope with these bursts, as long as not too many symbols of a RS codeword are destroyed.

A way to combat very long error bursts at the cost of delay (often referred to as latency) is interleaving/deinterleaving: the deinterleaver breaks up error bursts such that at most t RS code symbols per RS codeword are affected.

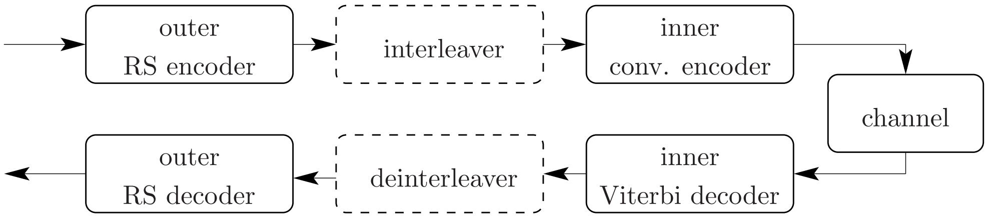


Figure 15: Concatenation of RS code (outer code) and convolutional code (inner code). The purpose of the interleaver/deinterleaver is to spread long error bursts that may be caused by the Viterbi decoder over several codewords such that the RS decoder can correct them.

References

[1] S. Lin and D. J. Costello, *Error Control Coding*, Prentice Hall, ISBN 0-13-042672-5, second edition, 2004.

[2] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, ISBN 0-7803-3483-3, 1999.

[3] J. L. Massey, *Applied Digital Information Theory—Lecture Notes, ETH Zurich*, www.isi.ee.ethz.ch/education/public/pdfs/aditI.pdf.

Part VI

Synchronisation for OFDM(A)

Abstract

Synchronization is a vital issue in OFDM(A). The entire orthogonality concept, including both the orthogonality among subcarriers of an individual user in an OFDM system as well as the orthogonality among users in an OFDMA system, falls apart if synchronization is poor. The chapter begins with a brief review of different types of offsets, their causes and their impact (Section 1). Basics of estimating and correcting timing (symbol boundary) offsets and carrier-frequency offsets in a single-user setup (OFDM, OFDMA downlink) are introduced in Section 2. Section 3 is devoted to the OFDMA uplink, where offsets of several users have to be handled to avoid mutual interference. Section 4 outlines the synchronization procedure of an operational OFDMA system following the WiMAX OFDMA standard (IEEE 802.16e). Finally, Section 5 presents a literature survey and Section 6 concludes the chapter. The purpose of this chapter is twofold. First, it points out the importance of synchronization in OFDM in general and in OFDMA uplink in particular. Second, it highlights elementary concepts and algorithms to tackle the synchronization challenge.

Notation

ΔF_c:	Carrier frequency offset in Hz.
ΔF_s:	Sampling frequency offset in Hz.
F_{c}:	Carrier frequency in Hz.
$F_{\mathrm{s}} = 1/T_{\mathrm{s}}$:	Sampling frequency in Hz.
K:	Number of users.
N:	OFDM symbol length in samples (excluding the cyclic prefix).
$T^{(\mathrm{CP})}$:	Cyclic prefix length in seconds.
$T^{(\mathrm{MC})} = NT_{\mathrm{s}}$:	OFDM symbol length (excluding cyclic prefix) in seconds.
$T_{\mathrm{s}} = 1/F_{\mathrm{s}}$:	Sampling period in seconds.
$T^{(\mathrm{W})}$:	Cyclic OFDM symbol extension in seconds for time-domain windowing.
$\tau^{(\mathrm{CIR})}$:	Channel impulse response length in seconds.
$\tau_i^{(\mathrm{DL})}$:	Down-link path delay of user i in seconds.
$\tau_i^{(\mathrm{UL})}$:	Up-link path delay of user i in seconds.

Chapter written by Thomas Magesacher, Jungwon Lee, Per Ödling, Per Ola Börjesson.

Abbreviations

BS:	Base station.
CIR:	Channel impulse response.
CP:	Cyclic prefix
DL:	Downlink
DFT:	Discrete Fourier transform.
FFT:	Fast Fourier transform.
ICI:	Inter-carrier interference.
ISI:	Inter-symbol interference.
MAI:	Multiple access interference.
UL:	Uplink

1 Introduction

The whole synchronisation issue arises from the fact that transmitter and receiver are spatially separated and not connected in any sense except for the fact that the receiver processes a signal, which is a modified version of what the transmitter once sent. Synchronization-related tasks required in every multicarrier system are:

- *Timing* acquisition

  After down-conversion and analog-to-digital conversion, the receiver sees a sequence of baseband samples. Before the block processing (cyclic-prefix removal, FFT, etc.) can begin, the symbol boundaries have to be determined. Timing offsets are caused by the inherent lack of a timing reference at the receiver.

  Even with access to the transmitter's timing reference (which is not available in a practical system though), there are distance-dependent path delays of significant duration. For example, the time a block needs to travel from a BS to a user that is five miles (ca. eighth kilometers) away, is roughly[1] $27\,\mu$s, which corresponds to about a quarter of the symbol duration (roughly $100\,\mu$s) in a WiMAX system (IEEE 802.16 Wireless MAN standard).

- *Frequency* offset

  After down-conversion in the receiver, the resulting baseband signal may exhibit an unwanted frequency offset of ΔF_c Hz for two reasons. First, down-conversion is based on a reference carrier generated by a local oscillator, which may have a slightly different frequency compared to the carrier generated in the transmitter. Second, the signal itself may experience a motion-incurred Doppler shift in frequency on its way through the channel.

  Frequency offsets caused by carrier-oscillator mismatch can be significant. For example, in a system operating at $F_\mathrm{c} = 5.2\,\mathrm{GHz}$ with carrier oscillators of 1 ppm[2] precision, the frequency offset[3] can be as large as 10.4 kHz, which is in the order of the subcarrier spacing (roughly 11.16 kHz) for a WiMAX system.

  Compared to oscillator-incurred offsets, motion-incurred frequency offsets are rather harmless. For example, a user moving at 40 mph (ca. 64 km/h) experiences a motion-incurred Doppler shift[4] of roughly 308 Hz, which corresponds to roughly 2.7 % of the subcarrier spacing. A simple motion-incurred shift occurs only in a single-path propagation scenario. In a multipath propagation scenario, the signal components arrive from several different angles and thus with several different

[1] Assuming a propagation velocity of $c = 3 \cdot 10^8$ m/s, the path delay is given by $8047/c \approx 26.83\,\mu$s.

[2] ppm stands for parts per million, a common measure to specify the precision of an oscillator.

[3] Assuming that the offsets of transmitter and receiver have opposite signs, we have $\Delta F_\mathrm{c} = 2 \cdot 10^{-6} \cdot F_\mathrm{c} = 10.4\,\mathrm{kHz}$.

[4] Assuming a propagation velocity of $c = 3 \cdot 10^8$ m/s, the Doppler shift is given by $\Delta F_\mathrm{c} = 64/3.6/(F_\mathrm{c}/c) \approx 308.15\,\mathrm{Hz}$.

relative velocities which yields several different Doppler shifts. Consequently, the receive signal experiences a dispersion in frequency rather than a shift. The practical approach to handle motion-incurred dispersion in frequency is to ensure a large enough subcarrier spacing.

- *Clock* offset

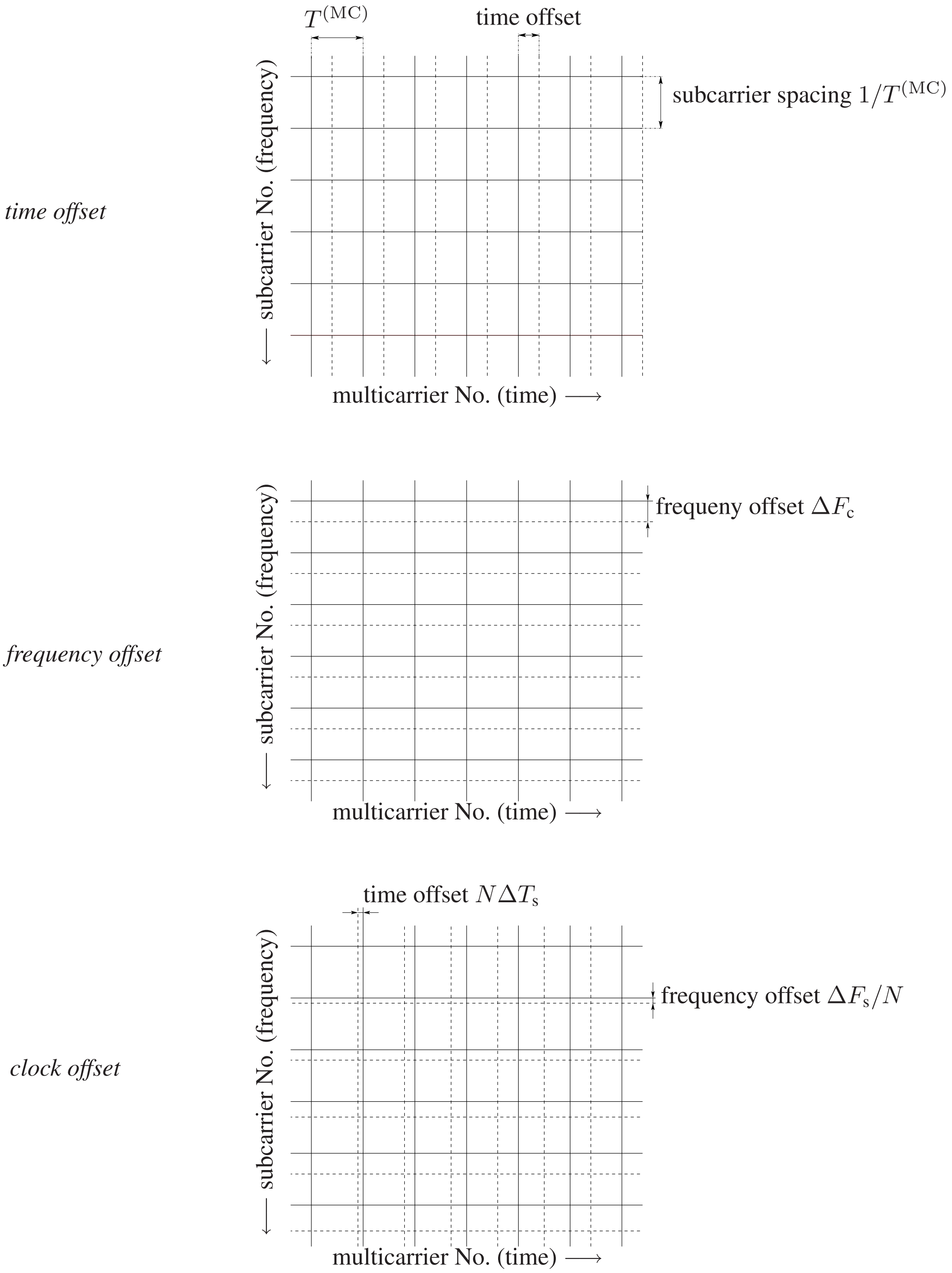

$T^{(\mathrm{MC})}$
time offset
subcarrier spacing $1/T^{(\mathrm{MC})}$
time offset
← subcarrier No. (frequency)
multicarrier No. (time) ⟶
frequeny offset ΔF_c
frequency offset
← subcarrier No. (frequency)
multicarrier No. (time) ⟶
time offset $N\Delta T_\mathrm{s}$
frequency offset $\Delta F_\mathrm{s}/N$
clock offset
← subcarrier No. (frequency)
multicarrier No. (time) ⟶

Figure 1: Illustration of timing offset, frequency offset, and clock offset in the time-frequency plane: Vertical and horizontal lines indicate symbol boundaries and subcarriers' center frequencies, respectively (for simplicity, $T^{(\mathrm{CP})} = 0$). The receiver assumes the time/frequency grid illustrated by dashed lines while the true time/frequency grid is marked by solid lines.

> At some point in the receive chain, the continuous-time receive signal is sampled in time-domain (and, in practice, quantised in amplitude at the same time by an analogue-to-digital converter). If the difference $\Delta F_s = F_s^{(\mathrm{rx})} - F_s^{(\mathrm{tx})}$ between the sampling clocks of the receiver and the transmitter is smaller (larger) than zero, the elements of the receive signal's DFT have a frequency spacing that is smaller (larger) than the spacing of the transmit signal's DFT. Furthermore, the receiver's sampling period is larger (smaller) than the transmitter's sampling period.
>
> While a carrier-oscillator mismatch causes a frequency offset, a clock offset results in a subcarrier-spacing mismatch. Consequently, subcarriers at the band edges experience stronger ICI while subcarrier close to the band center experience almost no ICI. The resulting frequency mismatch is small compared to a carrier-oscillator incurred offset. For example, in a WiMAX system with 2048 subcarriers (sampling frequency $B = 22.857\,\mathrm{MHz}$), clock oscillators with 1 ppm cause an offset[5] of up to ca. 23 Hz, which corresponds to roughly 0.2 % of the subcarrier spacing. The time error accumulates and corresponds to roughly 1.3 OFDM symbols per minute.

Figure 1 illustrates the offsets discussed above in the time-frequency plane. Time offsets and frequency offsets deserve special attention in OFDM(A) systems and are thus central to this chapter. Clock offsets can be dealt with using standard techniques applied in digital transceivers. Furthermore, the sampling clock can be derived from the carrier clock both at the BS and at the user side. Thus, once the carrier frequency is corrected, the sampling clock offset is negligible.

In general, synchronisation consists of two tasks: *estimation* of appropriate parameters (frequency offset, time offset) and *correction* of offsets based on these estimates. Estimation techniques are generally classified as *pilot-based methods*, which are supported by deliberately inserted synchronisation-assisting signals (pilot signals, synchronisation symbols) or *blind methods*, which operate without dedicated pilots. Synchronization is usually carried out in two steps: an initial synchronization procedure (often referred to as *acquisition*) and periodic update steps to cope with varying conditions (often referred to as *tracking*).

2 Synchronization in OFDM and OFDMA DL

The techniques discussed in the following are elementary in the sense that they can be applied directly to single-user (point-to-point) systems (traditional OFDM) and constitute the basic building blocks for multiuser (point-to-multipoint) setups (OFDMA downlink).

2.1 Symbol-timing acquisition

Tradeoff: symbol-synchronization precision versus robustness to time dispersion

Before dealing with dedicated synchronization techniques, it is helpful to realize that a cyclic prefix longer than the channel delay spread can ease the symbol-timing problem. Figure 2 schematically depicts a sequence of time-domain blocks both before and after passing through the channel. An OFDM symbol of length $T^{(\mathrm{MC})} = NT_{\mathrm{s}}$ extended by a cyclic prefix of length $T^{(\mathrm{CP})}$ is sent through a time-dispersive channel of length $\tau^{(\mathrm{CIR})}$. Often, multicarrier symbols are extended by an additional cyclic prefix and an additional cyclic suffix, each of length $T^{(\mathrm{W})}$, followed by time-domain windowing of these additional extensions in order to reduce out-of-band emissions.

Taking an FFT at the true symbol boundary captures exclusively the channel's steady-state response to the current symbol. When the symbol-boundary estimate is ahead of the true boundary (too early in time), the FFT captures a portion of the transient stemming from the cyclic prefix of the current symbol (which causes ICI) as well as a piece of the falling transient of the previous symbol (which causes ISI). Similarly, when the symbol-boundary estimate lags behind the true boundary (too late in time), the FFT captures a portion of the current symbol's falling transient (causing ICI) and a part of the succeeding symbol's rising transient (causing ISI). The performance decay caused by ISI and ICI induced by the deviation from the optimal symbol boundary is gradual, where late timing typically results in a larger performance decay than early timing. The reason is that most practical channels exhibit a decaying channel delay profile. Consequently, late timing results in capturing "early-response"-components from the succeeding symbol which are stronger than "tail"-components from the preceding symbol.

[5] The maximum frequency error is half the maximum sampling-clock difference, given by $\Delta F_s = 2 \cdot 10^{-6} B = 45.71\,\mathrm{Hz}$.

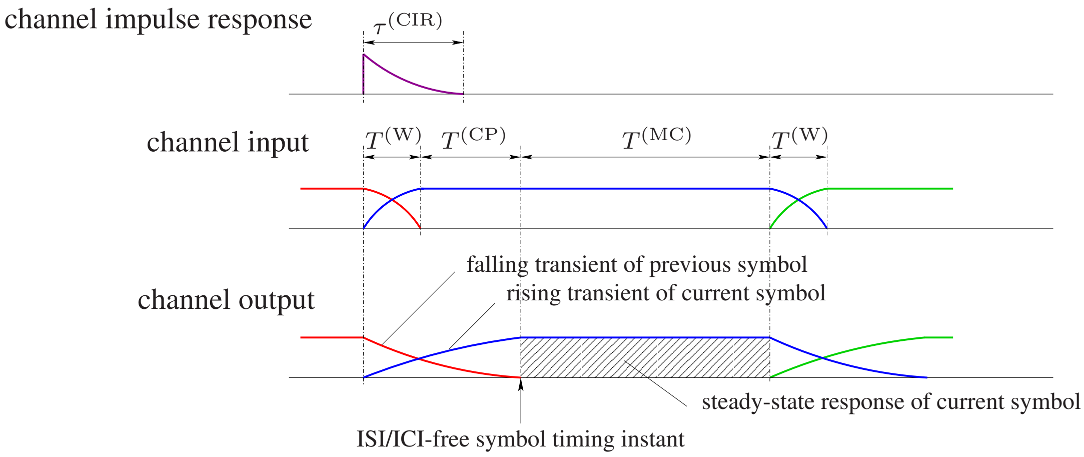


Figure 2: Symbol timing when cyclic prefix length is equal to channel dispersion ($T^{(\mathrm{CP})} = \tau^{(\mathrm{CIR})}$): the ISI/ICI-free timing instant is unique.

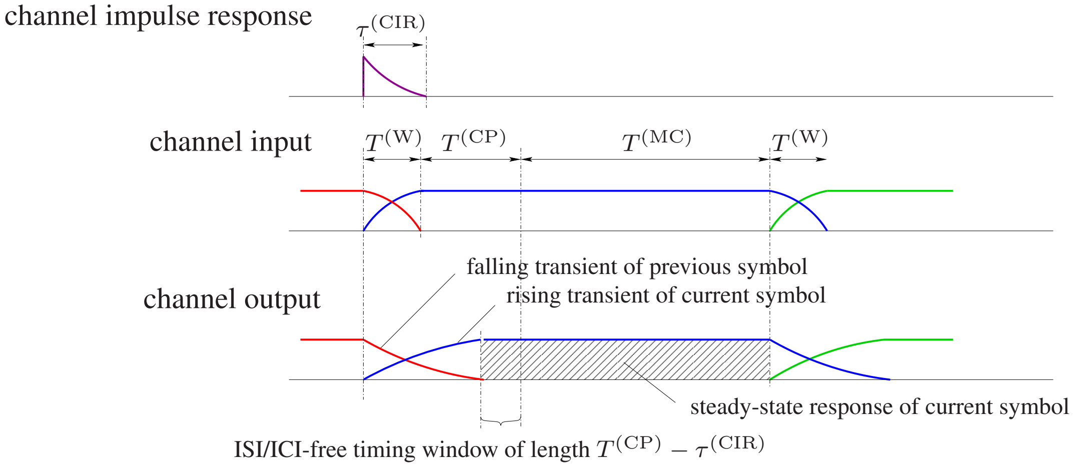

channel impulse response
$\tau^{(\mathrm{CIR})}$
channel input
$T^{(\mathrm{W})}$
$T^{(\mathrm{CP})}$
$T^{(\mathrm{MC})}$
$T^{(\mathrm{W})}$
channel output
falling transient of previous symbol
rising transient of current symbol
steady-state response of current symbol
ISI/ICI-free timing window of length $T^{(\mathrm{CP})} - \tau^{(\mathrm{CIR})}$

Figure 3: Symbol timing when cyclic prefix is longer than channel dispersion ($T^{(\mathrm{CP})} > \tau^{(\mathrm{CIR})}$) yields an ISI/ICI-free timing window of length $T^{(\mathrm{CP})} - \tau^{(\mathrm{CIR})}$.

Clearly, there is a tradeoff between the delay spread that can be handled and the admissible timing error. A channel impulse response (CIR) of length $\tau^{(\mathrm{CIR})} < T^{(\mathrm{CP})}$ increases the length of the received block's steady-state part by $T^{(\mathrm{CP})} - \tau^{(\mathrm{CIR})}$, as shown in Figure 3. Consequently, the symbol boundary can be up to $T^{(\mathrm{CP})} - \tau^{(\mathrm{CIR})}$ seconds ahead of the true timing instant without capturing the falling transient of the preceding block (that is, there is no ISI) nor capturing a transient of the current symbol (that is, there is no ICI). However, compared to the receive block captured at the true timing instant, the receive block captured n samples earlier corresponds to a version that is cyclically right-shifted by n samples. A cyclic time-domain right-shift of $r(n), n = 0, \ldots, N-1$ by s samples, resulting in the block $r'(n) = r((n-s) \bmod N)$, corresponds to a phase rotation of all subcarriers in the DFT domain. The kth subcarrier is rotated by $2\pi sk/N$:

$$r'(n) = r((n-s) \bmod N) \longleftrightarrow R'(k) = R(k)\mathrm{e}^{-j2\pi sk/N}, k = 0, \ldots, N-1 \quad (1)$$

Differential modulation in time (in combination with noncoherent detection) is inherently immune to time-invariant phase rotations. For absolute modulation (in combination with coherent detection), these phase rotations can be viewed as part of the channel and are thus implicitly taken care of by frequency-domain equalization.

Elements of pilot-based timing synchronization

The basic idea of pilot-based synchronisation is to transmit a piece of signal (a so called synchronization symbol or pilot symbol) that is known to the receiver and can thus be detected to mark a reference timing instant. Most detectors are based on evaluating the similarity of a piece of receive signal and a piece of reference signal through computing their correlation. Unless the pilot symbol is simply a sequence of zeros, it is modified by the time-dispersive channel and thus correlation at the receiver with the sent pilot symbol will perform poorly[6]. Most practically relevant schemes thus revert to transmitting a periodic piece of signal, which yields (apart from the channel noise) a periodic response (assuming that a long-enough CP is applied). Rather than correlating with a known reference, the receive stream is correlated with a shifted version of itself (shifted exactly by one period)—an idea that has been used in single-carrier systems long before OFDM became popular. Under the (reasonable) assumption that the data does not exhibit periodic behavior, strong correlation will pinpoint the pilot symbol.

More concretely, the classical pilot-based synchronization method [1] tackling both time and frequency offsets, uses a time-domain pilot symbol with two identical halves, which can be generated easily by setting odd subcarriers to zero. The normalized correlation measure

$$M_{\text{time}}(d) = \frac{\left| \sum\limits_{m=d}^{d+N/2-1} r^*(m) r(m+N/2) \right|}{\sum\limits_{m=d}^{d+N-1} |r(m)|^2} \tag{2}$$

computed at the receiver based on the receive stream $r(n)$ measures the similarity between a length-$N/2$ block starting at time instant d and the consecutive length-$N/2$ block. The timing instant

$$\widehat{d} = \arg\max_{d} M_{\text{time}}(d) \tag{3}$$

yields the maximum value of $M_{\text{time}}(d)$ and pinpoints the first sample of the pilot symbol's steady-state part. For $\tau^{(\text{CIR})} = T^{(\text{CP})}$, the correlation metric $M_{\text{time}}(d)$ yields a unique peak and the desired symbol-boundary instant (first sample following the pilot symbol) is given by $\widehat{d} + N$. When the CP is longer than the channel delay spread ($\tau^{(\text{CIR})} < T^{(\text{CP})}$), $M_{\text{time}}(d)$ exhibits a non-unique plateau-like maximum. Clearly, choosing the last time-instant of the plateau yields largest immunity to time dispersion—however, reliably pinpointing this instant in the presence of channel noise using a detection algorithm may be non-trivial.

CP-based timing acquisition

In setups that do not use dedicated pilots for acquisition, like broadcast systems, the receiver can exploit the repetitive pattern of the receive stream for identifying the symbol boundaries. Similarly to the pilot-based approach, CP-based timing acquisition uses a correlation measure (cf. [2])

$$\gamma(d) \mathrel{\widehat{=}} \sum_{m=d}^{d+L-1} r^*(m) r(m+N) \tag{4}$$

and a power measure

$$\Phi(d) \mathrel{\widehat{=}} \frac{1}{2} \sum_{m=d}^{d+L-1} |r(m)|^2 + |r(m+N)|^2$$

Joint maximum-likelihood estimation of both timing offset and frequency offset yields the estimate pinpointing the first sample of the pilot symbol's steady-state part

$$\widehat{d} = \arg\max\{|\gamma(d)| - \rho\Phi(d)\} + L, \tag{5}$$

where ρ is the magnitude of correlation coefficient between two N-spaced samples.

In an AWGN channel, $r(m)$ and $r(m+N)$ are identical up to the noise introduced by the channel. A time-dispersive channel, however, additionally introduces ISI and ICI, which reduces the estimator's

[6] Note that one could, in principle, use the channel's response to the pilot symbol for correlation. However, computation of the latter is difficult since, during the synchronization phase, there is usually no channel knowledge available yet.

performance. However, power-delay profiles found in real systems often decay rapidly enough (for example, exponentially) in order to yield sufficient correlation between $r(m)$ and $r(m+N)$ to achieve an acceptable performance.

2.2 Carrier-frequency acquisition

Elements of pilot-based frequency acquisition

Like timing acquisition schemes, also carrier-frequency synchronization methods used in practice almost exclusively rely on pilot symbols. Preferably, the timing pilots are exploited also for frequency synchronization reducing overhead and taking advantage of the channel-independence property of periodic pilot signals.

A carrier-frequency offset between transmitter and receiver causes an unwanted, steadily growing phase rotation: Denoting $r'(n)$ the baseband receive signal without carrier-frequency offset ($\Delta F_c = 0$), the receive stream with carrier-frequency offset can be written as $r(n) = r'(n)\,\mathrm{e}^{j2\pi(n_0+n)\Delta F_c/F_\mathrm{s}}$. Frequency-offset measures rely on the following observation: if $r'(n)$ is periodic with period P, any two samples $r(n)$ and $r(n+P)$ are identical up to a constant phase difference

$$\arg(r(n+P)) - \arg(r(n)) = \arg(r(n+P)r^*(n)) = 2\pi P\Delta F_c/F_\mathrm{s} + 2\pi i \tag{6}$$

and noise. The integer multiple $i \in \mathbb{Z}$ of 2π accounts for the fact that $\arg(\cdot) \in [-\pi, \pi)$. Averaging over P samples starting at $\widehat{d}$, which captures the (periodic) pilot symbol and mitigates the noise, yields the fractional part[7] of the frequency-offset estimate

$$\widehat{\Delta F_c} = \frac{F_\mathrm{s}}{2\pi P}\arg(\sum_n r(\widehat{d}+n+P)r^*(\widehat{d}+n))$$

If the frequency offset is known to be small enough so that $|\Delta F_c| \leq \frac{F_\mathrm{s}}{2P}$ can be guaranteed, then $i=0$ and the fractional part yields the correct estimate. Thus, a smaller value P yields an larger acquisition range. However, there is a tradeoff since smaller P results in less averaging and thus more susceptibility to noise. The scheme proposed in [1], for example, uses $P = N/2$, which corresponds to an offset range of $\pm\frac{F_\mathrm{s}}{N}$ (± one subcarrier spacing).

The fractional frequency offset is easily corrected in time-domain before FFT processing through counter-rotation yielding $r'(n) = r(n)\,\mathrm{e}^{-j2\pi n\widehat{\Delta F_c}/F_\mathrm{s}}$. Estimating the remaining integer part of the frequency offset can be done with the help of a second training symbol that carries differentially modulated data with respect to the pilot symbol used up to now. The pilot symbols in FFT domain denoted by $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$, exhibit no (or only residual) ICI. However, all subcarriers are shifted by iF_s/P (for the scheme [1], this corresponds to $i2F_\mathrm{s}/N$—an integer multiple of twice the subcarrier spacing), which is the basis for estimating the integer part of the frequency offset. Differential modulation be formulated as $\boldsymbol{R}_2 = \boldsymbol{R}_1\boldsymbol{P}$, where $\boldsymbol{P}$ is a pseudo-random sequence and assuming MSK with points on the unit circle. In order to find the integer part of the offset, the similarity between $\boldsymbol{R}_2(k)/\boldsymbol{R}_1(k)$ and a shifted version $\boldsymbol{P}(k-iN/P)$ of $\boldsymbol{P}$ is measured by[8]

$$M_\mathrm{freq}(i) = \sum_k |\frac{\boldsymbol{R}_2(k)}{\boldsymbol{R}_1(k)}\boldsymbol{P}(k-iN/P)| = \sum_k \frac{|\boldsymbol{R}_2(k)\boldsymbol{R}_1^*(k)\boldsymbol{P}(k-iN/P)|}{|\boldsymbol{R}_1(k)|^2}$$

and its maximum indicates the integer part

$$\widehat{i} = \arg\max_i M_\mathrm{freq}(i)$$

of the frequency offset, which can then be included in the time-domain rotation.

Finally, a carrier-phase offset $\mathrm{e}^{j2\pi n_0\Delta F_c/F_\mathrm{s}}$ remains, which can either be ignored (differential modulation in time) or is taken care of through estimation and equalization of the channel.

[7] Fraction of $\frac{F_\mathrm{s}}{P}$

[8] Note that the sum index k includes only subcarriers for which $\boldsymbol{R}_1(k) \neq 0$ and $k - iN/P \in [1, N]$.

CP-based frequency acquisition

In the absence of dedicated acquisition pilots, the receiver can exploit the CP also for estimating the fractional frequency offset using the same principle as before: repetitive signal parts are identical up to a phase difference. Joint maximum-likelihood estimation of time and frequency offsets (cf. [2]) yields the frequency offset

$$\widehat{\Delta F_c} = \frac{F_\mathrm{s}}{2\pi N}\,\arg(\gamma(\widehat{d})) \tag{7}$$

where the correlation measure $\gamma(d)$ is given by (4). Note that (7) is identical to (6) with $P = N$. Like for CP-based timing acquisition, the performance of CP-based frequency acquisition degrades with stronger time dispersion and with increasing channel noise.

2.3 Tracking

After acquisition, time offsets and frequency offsets are updated on a regular basis exploiting dedicated tracking pilots, which are arranged in time and frequency following a predefined pattern known to the receiver. A small timing offset manifests itself in frequency-domain as a phase offset that increases linearly with the subcarrier index, which is a consequence of the circular shift theorem (1). Knowing both the pilot symbols and the channel, the time shift corresponding to this phase offset can be estimated. The frequency offset can be tracked using again the same principle as for acquisition, exploiting the linearly increasing phase offset between consecutive pilots in time (cf. (6)).

3 Synchronization in OFDMA UL

Synchronization in UL direction in an OFDMA system with K users bears the challenge that the receive stream is a superposition of K components, which pass through potentially different channels and thus experience different path delays (reflecting different distances of users from the BS) and different frequency offsets (reflecting different velocities of users). Furthermore, in an asynchronous setup, the transmission of each component may begin at different points in time.

Two aspects render synchronization in UL direction more difficult than synchronization in DL direction: First, the parameter space increases by a factor of K and, unless structured subcarrier assignment is applied, these parameters have to be estimated jointly. Second, straightforward correction of an individual user's offset applied to the UL stream affects the offsets of all other users. Even if all parameters are known, correction of offsets in UL direction is thus not at all straightforward. Hence, as appealing the principle idea of using OFDM as access scheme may seem, the requirement of synchronized users in OFDMA UL introduces a "chicken-egg problem": separating users' streams through FFT processing is only possible after correcting offsets of individual users—which, in turn, requires access to individual users' streams.

Clearly, synchronization in UL direction is closely connected to multiuser detection. A rigorous approach embracing the synchronization problem in OFDMA-UL is joint detection of all users' data sequences. Such a multiuser detector that is optimal in the maximum *a posteriori* probability (MAP) sense is formulated in [3]. Instead of separately estimating and correcting offsets, the multiuser detector jointly decides the coded transmitted sequences of all users in the presence of timing offsets and frequency offsets for arbitrary subcarrier allocation. However, the underlying trellis, which includes the three dimensions time, frequency, and users, quickly grows in complexity to a level that prohibits implementation for real-world parameters.

Practical schemes thus separate the tasks of parameter estimation and offset correction, treated in Subsections 3.2 and 3.3, respectively. Structured subcarriers assignment allows dedicated low-complexity solutions. Moreover, as discussed next in Subsection 3.1, dedicated timing schemes exploiting the DL timing reference can greatly simplify the synchronization problem.

3.1 Synchronous and asynchronous multiuser systems

In OFDMA UL, we distinguish the following elementary timing schemes:

- In an *asynchronous system*, the users transmit their UL blocks without considering a timing reference. Additionally, the UL path delay may vary greatly from user to user (as a consequence of

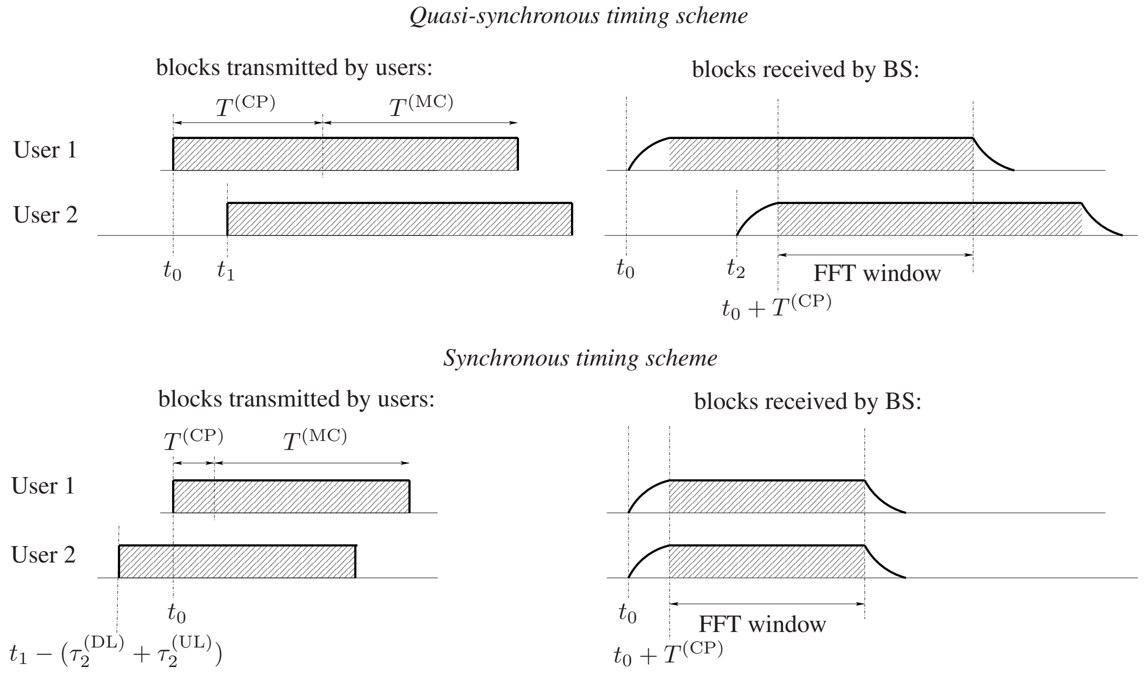

Quasi-synchronous timing scheme
blocks transmitted by users:
blocks received by BS:
$T^{(\mathrm{CP})}$
$T^{(\mathrm{MC})}$
User 1
User 2
t_0
t_1
t_0
t_2
FFT window
$t_0 + T^{(\mathrm{CP})}$
Synchronous timing scheme
blocks transmitted by users:
blocks received by BS:
$T^{(\mathrm{CP})}$
$T^{(\mathrm{MC})}$
User 1
User 2
t_0
$t_1 - (\tau_2^{(\mathrm{DL})} + \tau_2^{(\mathrm{UL})})$
t_0
FFT window
$t_0 + T^{(\mathrm{CP})}$

Figure 4: Quasi-synchronous (top) and synchronous (bottom) OFDMA UL timing scheme: UL block of User 1 (with path delay ≈ 0) and User 2 (with longest path delay among all users) at the user side (left) and at the BS (right).

different distances from the BS). Consequently, different users' UL components arrive with different time offsets and different frequency offsets (as a consequence of different velocities) at the BS, which renders the asynchronous case very challenging.

- In a *quasi-synchronous system*, each terminal exploits the timing reference established through DL synchronization. Together with using a long enough CP, timing acquisition can be entirely avoided. More concretely, the BS transmits a DL block at time instant t_0. User No. k receives the block $\tau_k^{(\mathrm{DL})}$ seconds later, implicitly identifies this time instant $t_0 + \tau_k^{(\mathrm{DL})}$ in the course of DL synchronization, and starts transmitting its UL block at $t_0 + \tau_k^{(\mathrm{DL})}$. Figure 4 depicts the UL blocks of two users: user 1 is very close to the BS ($\tau_1^{(\mathrm{DL})} \approx 0$) and thus transmits its UL block virtually at t_0; user 2 is farthest away for the BS among all users ($\tau_2^{(\mathrm{DL})} = \max_k \tau_k^{(\mathrm{DL})}$) and starts its UL transmission at $t_1 = t_0 + \tau_2^{(\mathrm{DL})}$. Under the assumption of quasi-reciprocal path delays ($\tau_\cdot^{(\mathrm{UL})} \approx \tau_\cdot^{(\mathrm{DL})}$) and negligible processing delays, the UL components from user 1 and user 2 reach the BS at t_0 and $t_2 = t_1 + \tau_2^{(\mathrm{UL})}$, respectively. In a quasi-synchronous system, the CP is chosen long enough to accommodate the maximum roundtrip delay as well as the maximum channel delay spread:

$$T^{(\mathrm{CP})} \geq \max_k \tau_k^{(\mathrm{UL})} + \max_k \tau_k^{(\mathrm{DL})} + \max_k \tau_k^{(\mathrm{CIR})} \qquad (8)$$

The BS takes an FFT of the aggregate UL stream beginning at $t_0 + T^{(\mathrm{CP})}$. As illustrated in Figure 4, the choice of the CP length ensures that the FFT window extends exclusively over steady-state parts of both the component with the largest and with the shortest delay: $t_0 + T^{(\mathrm{CP})}$ is the earliest possible timing to avoid the rising transient of the component with the largest delay (user 2); at the same time, $t_0 + T^{(\mathrm{CP})}$ is the latest possible timing to avoid the falling transient of the component with the shortest delay (user 1). In this manner, MAI is entirely avoided without explicit estimation of path delays. Note, however, that the FFT window in general captures cyclically shifted versions of individual users' components. A cyclic shift in time domain corresponds to phase rotations in frequency domain (cf. (1)), which can be easily corrected by a per-tone phase-shift carried out by the equalizer without explicitly estimating the phase factors or, equivalently, the number of shifted samples.

An alternative way of viewing a quasi-synchronous system is the following: Assume that all UL blocks are sent at t_0 and treat the roundtrip delays $\tau_k^{(\mathrm{DL})} + \tau_k^{(\mathrm{UL})}$ as part of the channel impulse responses. Consequently, there are no timing offsets to be estimated or corrected explicitly.

To summarize, in a quasi-synchronous OFDMA UL system, there is no need for estimating and/or correcting timing offsets. The price to pay is reduced efficiency caused by increasing the CP length by the largest roundtrip delay. Frequency offsets, however, remain and have to be estimated and corrected.

- In a *synchronous system*, each user estimates its roundtrip delay $\tau_k^{(\mathrm{DL})} + \tau_k^{(\mathrm{UL})}$ and transmits its UL block $\tau_k^{(\mathrm{DL})} + \tau_k^{(\mathrm{UL})}$ seconds before the arrival of the DL block. Consequently, the UL blocks of all users arrive aligned at t_0 at the BS. Figure 4 again depicts the two users closest to and farthest from the BS. The CP is chosen long enough to accommodate the maximum channel delay spread:

$$T^{(\mathrm{CP})} \geq \max_i \tau_i^{(\mathrm{CIR})} \tag{9}$$

The BS takes an FFT of the aggregate UL stream beginning at $t_0 + T^{(\mathrm{CP})}$, which ensures that the FFT window extends exclusively over steady-state parts of both the component with the largest and with the shortest delay.

To summarize, in a synchronous system, offsets are corrected by the users such that blocks arrive in an aligned manner at the BS. The CP only needs to enclose the maximum channel delay spread. The offsets are either estimated by the users themselves or by the BS and subsequently fed back to the users.

3.2 Offset estimation in OFDMA UL

Considering the complexity of this task, it comes to no surprise that open literature does not seem to include a viable solution for jointly estimating time and frequency offsets in an asynchronous setup. There has been, however, considerable progress under certain simplifying assumptions:

- *Consecutive estimation of parameters* can be applied assuming that users begin to access the UL channel one at a time, which eliminates the need for joint estimation [4]. In practice, however, it may be difficult to exclude the case when two users enter the network simultaneously.

- *Structured subcarrier allocation* can greatly simplify the estimation task. Assigning groups of adjacent subcarriers forming subbands to individual users, for example, essentially allows separation of users in frequency domain applying filterbank-like techniques. Another popular allocation scheme is regular interleaving, which assigns subcarrier k and each consecutive K-spaced subcarrier to user No. $k \in \{1, \dots, K\}$. Provided that $N/K \in \mathbb{N}$, using every Kth subcarrier yields a periodic time-domain signal with period N/K—a distinct signal structure to be exploited by estimation techniques such as MUSIC [5]. However, subcarrier allocation through resource allocation schemes aiming at maximizing some performance metric, which in general yields an arbitrary, unstructured allocation, is one of the most appealing advantages of OFDMA. Structured subcarrier allocation may thus be a too stringent assumption.

- *Quasi-synchronous timing* leaves only frequency offsets to be estimated. Dedicated pilot-based estimation schemes have been proposed in [6],[7] using the EM-based SAGE algorithm [8],[9] described in the following. Under the quasi-synchronous timing assumption, a receive block arriving at the base station after purging the cyclic extension can be written as

$$\boldsymbol{r} = \sum_{k=1}^{K} \mathrm{diag}\{\begin{bmatrix} 1 & \mathrm{e}^{j2\pi\epsilon_k/N} & \cdots & \mathrm{e}^{j2\pi\epsilon_k(N-1)/N} \end{bmatrix}\} \boldsymbol{S}_k \boldsymbol{h}_k + \boldsymbol{n} \tag{10}$$

where ϵ_k is the kth user's frequency offset, $\boldsymbol{S}_k$ is a Toeplitz matrix containing the kth user's pilot data known to the receiver, $\boldsymbol{h}_k$ is the channel impulse response, and $\boldsymbol{n}$ is the noise. A possibly existing phase offset is absorbed by $\boldsymbol{h}_k$ in (10) and can be taken care of by the equalizer.

The SAGE algorithm applied to the signal model (10) iteratively improves estimates of ϵ_k and $\boldsymbol{h}_k$ in a two-step procedure and approaches the maximum likelihood solution. Given initial estimates $\widehat{\epsilon}_i^{(0)}$ and $\widehat{\boldsymbol{h}}_i^{(0)}$, initial estimates of the users' receive signal can be calculated as

$$\widehat{\boldsymbol{r}}_i^{(0)} = \mathrm{diag}\{\begin{bmatrix}1 & \mathrm{e}^{j2\pi\widehat{\epsilon}_i^{(0)}/N} & \dots & \mathrm{e}^{j2\pi\widehat{\epsilon}_i^{(0)}(N-1)/N}\end{bmatrix}\}\boldsymbol{S}_i\widehat{\boldsymbol{h}}_i^{(0)}$$

In each iteration $j = 1, 2, \dots$, for each user $k = 1, 2, \dots, K$, the algorithm performs the following steps:

- E-step (compute expectation): compute an estimate of the kth user's receive signal based on the (best available) estimates of all other users' signals according to

$$\bar{\boldsymbol{r}}_k^{(j)} = \boldsymbol{r} - \sum_{i=1}^{k-1}\widehat{\boldsymbol{r}}_i^{(j)} - \sum_{i=k+1}^{K}\widehat{\boldsymbol{r}}_i^{(j-1)}$$

- M-step (maximize expectation): compute improved estimates of ϵ_k and $\boldsymbol{h}_k$ by minimizing

$$||\bar{\boldsymbol{r}}_k^{(j)} - \mathrm{diag}\{\begin{bmatrix}1 & \mathrm{e}^{j2\pi\bar{\epsilon}_k/N} & \dots & \mathrm{e}^{j2\pi\bar{\epsilon}_k(N-1)/N}\end{bmatrix}\}\boldsymbol{S}_k\bar{\boldsymbol{h}}_k||^2$$

  with respect to $\bar{\epsilon}_k$ and $\bar{\boldsymbol{h}}_k$. Since the problem allows to decouple $\bar{\epsilon}_k$ and $\bar{\boldsymbol{h}}_k$, we can first estimate the frequency offset according to

$$\widehat{\epsilon}_k^{(j)} = \arg\max_{\bar{\epsilon}_k}\{||\boldsymbol{S}_k(\boldsymbol{S}_k^{\mathrm{H}}\boldsymbol{S}_k)^{-1}\boldsymbol{S}_k^{\mathrm{H}}\mathrm{diag}\{\begin{bmatrix}1 & \mathrm{e}^{j2\pi\bar{\epsilon}_k/N} & \dots & \mathrm{e}^{j2\pi\bar{\epsilon}_k(N-1)/N}\end{bmatrix}\}\bar{\boldsymbol{r}}_k^{(j)}||^2\}$$

  and subsequently use $\widehat{\epsilon}_k^{(j)}$ to calculate

$$\widehat{\boldsymbol{h}}_k^{(j)} = (\boldsymbol{S}_k^{\mathrm{H}}\boldsymbol{S}_k)^{-1}\boldsymbol{S}_k^{\mathrm{H}}\mathrm{diag}\{\begin{bmatrix}1 & \mathrm{e}^{j2\pi\widehat{\epsilon}_k^{(j)}/N} & \dots & \mathrm{e}^{j2\pi\widehat{\epsilon}_k^{(j)}(N-1)/N}\end{bmatrix}\}\bar{\boldsymbol{r}}_k^{(j)}$$

  Finally, using these estimates of frequency offset and channel, an update of the kth user's receive signal can be computed for the next E-step(s):

$$\widehat{\boldsymbol{r}}_k^{(j)} = \mathrm{diag}\{\begin{bmatrix}1 & \mathrm{e}^{j2\pi\widehat{\epsilon}_k^{(j)}/N} & \dots & \mathrm{e}^{j2\pi\widehat{\epsilon}_k^{(j)}(N-1)/N}\end{bmatrix}\}\boldsymbol{S}_k\widehat{\boldsymbol{h}}_k^{(j)}$$

Note that this scheme can be applied when unstructured subcarrier allocation is used. Though appealing because simple, quasi-synchronous timing can pose serious limits on the allowed roundtrip time and thus on the system's cell size, or equivalently, on the system's bandwidth efficiency.

3.3 Offset correction in OFDMA UL

In OFDMA UL, there are in principle two ways to correct offsets:

- Correction by users: The synchronous timing scheme discussed in Subsection 3.1 can be extended to frequency offsets. Both offsets in time and in frequency are corrected by users to achieve alignment at the BS. Offset estimates determined by the BS must thus be fed back to the users, which causes a bandwidth loss in DL direction.
- Correction by the BS is the most sophisticated and most challenging approach avoiding the feedback of parameter estimates.

In the following, we focus on dedicated low-complexity techniques to correct offsets. Most published work assumes a synchronous or quasi-synchronous timing scheme and focuses on the frequency offset. Again, structured subcarrier allocation simplifies the problem.

Subband allocation inherently separates users in frequency domain to a certain degree. Figure 5 depicts a straightforward solution to frequency-offset compensation, hereinafter referred to as *direct frequency-offset correction* method. Let $\boldsymbol{r}$ denote a receive block after purging the cyclic extension. For each user $k \in \{1, \dots, K\}$, there is a dedicated processing path including the correction of the frequency offset in time domain via multiplication by $e^{-j2\pi n\widehat{\epsilon}_k/N}$, where $\widehat{\epsilon}_k$ is the frequency-offset estimate, followed

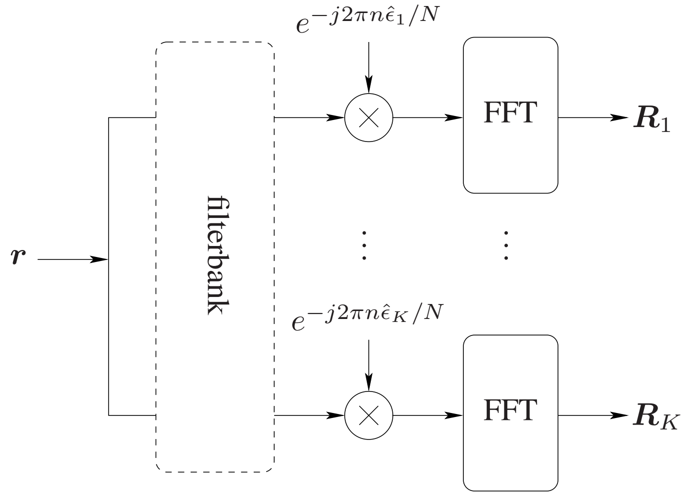


Figure 5: Direct frequency-offset correction.

by an N-point FFT yielding $\boldsymbol{R}_k$. In order to improve the separation of users in frequency domain, which is impaired by spectral leakage, a time-domain filterbank may be employed.

Direct frequency-offset correction requires one full-size FFT per user, which results in a computational complexity of $\mathcal{O}(KN\log_2 N)$ (excluding the filterbank). An approach to reduce complexity, hereinafter referred to as *CLJL scheme* [10], is depicted in Figure 6. The idea of CLJL is to perform offset correction after FFT processing of $\boldsymbol{r}$ and thus use only a single FFT yielding $\boldsymbol{R}$. At first glance, this does not seem to reduce complexity since the multiplication $\boldsymbol{r}[n+1]e^{-j2\pi n\widehat{\epsilon}_k/N}, n = 0, \ldots, N-1$ in discrete-time domain (complexity $\mathcal{O}(N)$), corresponds to circular convolution of $\boldsymbol{R}$ and $\boldsymbol{C}_k$ in frequency domain (complexity $\mathcal{O}(N\log_2 N)$), where $\boldsymbol{C}_k[\ell] = \sum_{n=0}^{N-1} e^{-j2\pi n\widehat{\epsilon}_k/N} e^{-j2\pi(\ell-1)n/N}, \ell = 1, \ldots, N$ is the N-point DFT of $e^{-j2\pi n\widehat{\epsilon}_k/N}, n = 0, \ldots, N-1$. However, using only the set $\mathcal{S}_k$ of subcarriers that are actually assigned to user k when computing the circular convolution yields a computational complexity of $\mathcal{O}(N\log_2(N/K))$. In other words, instead of $\boldsymbol{R}$, the vectors

$$\boldsymbol{R}_{\mathcal{S}_k}[\ell] = \begin{cases} \boldsymbol{R}[\ell], & \ell \in \mathcal{S}_k \\ 0, & \text{otherwise} \end{cases}$$

are used to compute the circular convolutions. The rationale behind this approximation is that most of the energy corresponding to the kth user's signal should be concentrated in the kth user's subband $\mathcal{S}_k$, as long as the carrier-frequency offset is small compared to the subcarriers spacing. The CLJL scheme in fact outperforms the direct method without filterbank in terms of bit error rate since processing only the subcarriers assigned to the corresponding user has a filter-effect and thus improves separation of users.

Both the direct frequency-offset correction method and the CLJL scheme rely on subband allocation. In order to keep the ICI among users sufficiently low, frequency-domain guard-bands may be necessary, which reduces spectral efficiency. An approach to correct frequency errors for interleaved and arbitrary assignment is *parallel interference cancellation* [11], a nonlinear multiuser-detection scheme that builds on the direct method or on the CLJL scheme. The basic idea is to iteratively improve estimates of users' receive signals. The direct offset correction method or the CLJL scheme can be used to obtain a set of K initial estimates for the separated receive signals. In each iteration, for each user $k \in \{1, \ldots, K\}$, a new receive signal estimate is computed as follows: first, the receive signal components of all other users $\ell \in \{1, \ldots, K\}\backslash k$ impaired by their frequency-offsets are reconstructed by imposing the corresponding frequency offset on each user's current (best available) frequency-corrected receive signal estimate; second, these receive components are subtracted from the aggregate receive signal, which yields a (hopefully) improved estimate of the kth user's receive signal (still containing the frequency offset); third, the frequency offset of the kth user is corrected (using one of the methods discussed above) to obtain a new estimate of the current user's frequency-corrected receive signal. Eventually, after a number of iterations, decisions on the data based on these estimates are taken.

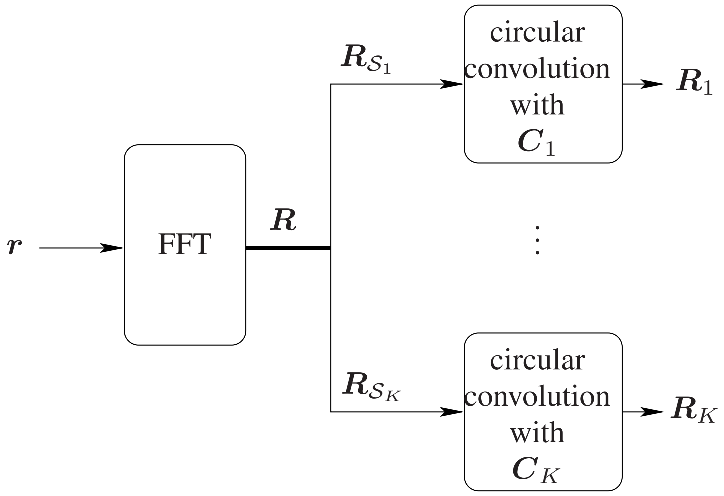

$\boldsymbol{r}$
FFT
$\boldsymbol{R}$
$\boldsymbol{R}_{\mathcal{S}_1}$
circular convolution with $\boldsymbol{C}_1$
$\boldsymbol{R}_1$
$\vdots$
$\boldsymbol{R}_{\mathcal{S}_K}$
circular convolution with $\boldsymbol{C}_K$
$\boldsymbol{R}_K$

Figure 6: Carrier-frequency offset correction in FFT-domain (CLJL scheme [10]).

Another approach to frequency-offset correction is *linear multiuser detection* [12]. The idea here is to start from the linear signal model

$$\boldsymbol{R} = \boldsymbol{\epsilon}(\epsilon_1, \ldots, \epsilon_K) \underbrace{\begin{bmatrix} \boldsymbol{R}'_1 \\ \vdots \\ \boldsymbol{R}'_K \end{bmatrix}}_{\boldsymbol{R}'} + \boldsymbol{n},$$

where $\boldsymbol{R}'_k \in \mathbb{C}^{|\mathcal{S}_k| \times 1}$ denotes the data of the kth user multiplied by the corresponding frequency-domain channel coefficients (but excluding the impact of spectral leakage) and $\boldsymbol{\epsilon}(\epsilon_1, \ldots, \epsilon_K) \in \mathbb{C}^{N \times \sum_i |\mathcal{S}_i|}$ models the spectral leakage caused by the frequency offsets $\epsilon_k, k \in \{1, \ldots, K\}$. Then, parameter estimation is used to obtain frequency-offset-free receive signal estimates $\boldsymbol{R}'_k$ that are optimal in the sense of minimizing the least squares (LS) error or the minimum mean square error (MMSE). The LS estimate, given by

$$\boldsymbol{R}'_{\mathrm{LS}} = (\boldsymbol{\epsilon}^{\mathrm{H}} \boldsymbol{\epsilon})^{-1} \boldsymbol{\epsilon}^{\mathrm{H}} \boldsymbol{R},$$

is sometimes also referred to as linear decorrelating detector when used together with a decision device (and thus detecting data). The MMSE estimate, given by

$$\boldsymbol{R}'_{\mathrm{MMSE}} = \boldsymbol{C}_{\mathbf{R}'} \boldsymbol{\epsilon}^{\mathrm{H}} (\boldsymbol{\epsilon} \boldsymbol{C}_{\mathbf{R}'} \boldsymbol{\epsilon}^{\mathrm{H}} + \boldsymbol{C}_{\mathbf{n}})^{-1} \boldsymbol{R},$$

requires *a priori* knowledge about the noise and the parameter to be estimated in form of their covariance matrices $\boldsymbol{C}_{\mathbf{n}}$ and $\boldsymbol{C}_{\mathbf{R}'}$, respectively. In practice, the received data $\boldsymbol{R}'$ is often assumed to be uncorrelated, yielding $\boldsymbol{C}_{\mathbf{R}'} = \boldsymbol{I}$. Also the noise $\boldsymbol{n}$ can often be assumed uncorrelated, yielding $\boldsymbol{C}_{\mathbf{n}} = \sigma_n^2 \boldsymbol{I}$, where σ_n^2 is the per-sample noise variance. The advantage of the MMSE estimator over the LS estimator is that it mitigates the noise enhancement problem.

4 Example: synchronization for WiMAX

As an example of an operational scheme, this section outlines the synchronization procedure in mobile WiMAX (IEEE 802.16e), a wireless metropolitan area network standard [13]. WiMAX embraces several of the principles discussed so far and is thus a well suited example to round off this chapter. Synchronization for mobile WiMAX is described for DL first, followed by a discussion of the UL.

For WiMAX DL, each user needs to perform both timing synchronization and carrier frequency synchronization. Apart from finding the symbol boundary, a user also needs to identify the frame boundary (a frame consists of a number of symbols). For carrier frequency synchronization, the user synchronizes its carrier oscillator with the received carrier, whose frequency corresponds to the BS's carrier frequency plus the offset due to the Doppler spread.

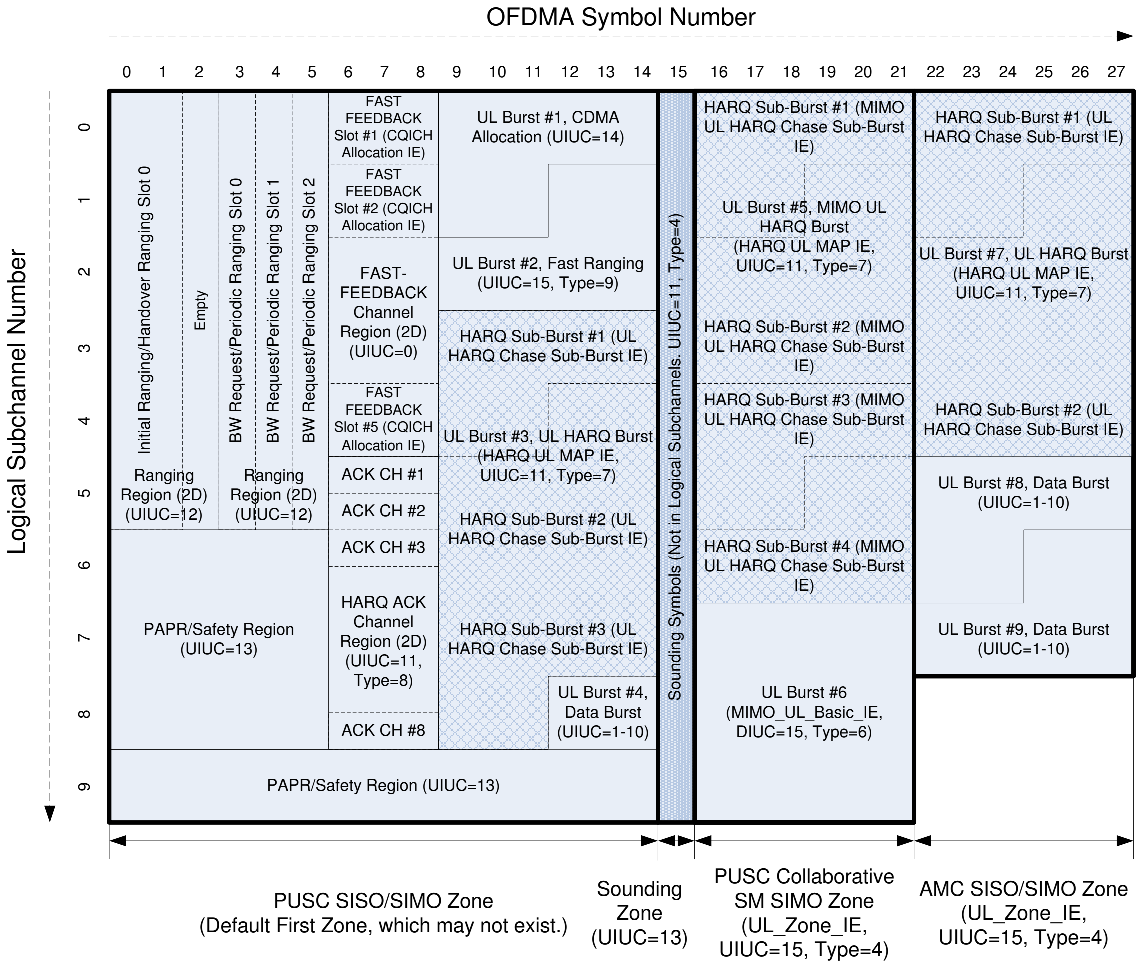
In WiMAX, the DL acquisition can be done based on a preamble inserted at the beginning of each frame. The preamble OFDMA symbol has the special property that only every third subcarrier is used, which results in a time-domain signal that has three identical parts within the useful part of an OFDMA symbol. These three identical parts can be used for both symbol timing synchronization and carrier frequency synchronization, based on the principle explained in Section 2. Since only the preamble OFDMA-symbol exhibits these identical parts, symbol-timing acquisition identifies the frame boundary as well. With the initial acquisition based on the preamble OFDMA symbol, the user can do FFT processing with reasonably small ISI and ICI. Subsequently, users perform tracking using dedicated tracking-pilot subcarriers. Alternatively, CP-based tracking can be performed. The requirement for the symbol-timing error is 25% of the CP length according to the IEEE 802.16e standards and 6.25% of the CP length according to the WiMAX system-profile document that lists the requirement for the real WiMAX system deployed. The carrier-frequency offset must be below 2% of subcarrier spacing. This requirement may not be satisfied right after the initial acquisition, but tracking can help to satisfy this requirement within acceptable time.

After the DL synchronization is completed, a user is allowed to start transmitting signals to a BS in order to begin the process of UL synchronization. The main goal of UL synchronization is estimation and correction of the path delay. Given the strict requirement for the symbol timing and carrier frequency for DL, both symbol timing and carrier frequency for UL are expected to be quite accurate. However, the user cannot estimate the path delay based on the DL signal only. Thus, the path delay is estimated by the BS. In addition, some residual symbol-timing error and carrier-frequency offset can be estimated by the BS. In essence, WiMAX OFDMA employs a synchronous timing scheme. The synchronization parameters are estimated by the BS, the correction itself of both time and frequency offset is performed by the users. Parameter estimation is simplified through dedicated pilot symbols with periodic structure. Users employ individual pilot symbols with low mutual cross-correlation. Typically, one or a few users will try to establish UL synchronization simultaneously, which corresponds to a scenario in between consecutive and joint estimation of UL parameters.

Before attempting to initiate UL communication, a user must establish DL synchronization to be able to listen to control information broadcasted by the BS. In a time-division duplexed system, the frequency offset acquired in the process of DL synchronization can decrease the frequency offset in UL[9]. The first step is to acquire important UL transmission parameters, the so called "uplink channel descriptor (UCD)" and "uplink map (UL-MAP)". With the UL transmission parameters, a user can figure out the uplink subframe structure. One example uplink subframe structure is shown in Fig. 7. Then the so called "initial ranging" procedure, which establishes time offset and transmit power, begins: the user randomly picks one out of a set of predefined PN sequences (so called "ranging codes"), which is used to modulate the ranging subchannels (as defined in the UL-MAP). The so-designed symbol is transmitted (with proper prefix and postfix extension) in accordance with ranging-opportunity information (time slots available for ranging in the next UL frame as specified in the UCD) and repeated immediately yielding a synchronization sequence that is two symbols long. In Fig. 7, the initial ranging symbols are located in the first two OFDMA symbols. At the BS, it is thus guaranteed that out of three consecutive symbols, at least one (in the best, although very unlikely case, two) symbols contain exclusively ranging information on the ranging subcarriers. Although in Fig. 7, no data subcarriers are assigned at the same time as ranging subcarriers, ranging subcarriers can be multiplexed with data subcarriers before and after ranging-opportunity slots. PN sequences are used to cope with users simultaneously trying to establish UL communication (which is not too unlikely since ranging opportunities are limited). After successful identification of the PN sequence (or sequences, in case more than one user performs initial ranging) and estimation of the associated time offset(s), the BS broadcasts this information. Finally, aspiring users apply UL timing correction and continue with negotiating further physical layer parameters before eventually commencing their UL communication.

Users carrying out initial ranging are not time-aligned yet and thus cause MAI disturbing synchronized users. In order to keep the MAI low, initial ranging begins with the lowest possible transmit power level. If synchronization fails (that is, the user's PN-sequence identifier is not included in the broadcasted DL information), the initial ranging procedure is repeated with a larger transmit power level.

[9] Note that using the carrier frequency acquired during DL synchronization for UL transmission results in an UL frequency offset if the Doppler shift ΔF_c changes from DL reception to UL transmission.


OFDMA Symbol Number
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
Logical Subchannel Number
0 1 2 3 4 5 6 7 8 9
Initial Ranging/Handover Ranging Slot 0
Empty
BW Request/Periodic Ranging Slot 0
BW Request/Periodic Ranging Slot 1
BW Request/Periodic Ranging Slot 2
Ranging Region (2D) (UIUC=12)
Ranging Region (2D) (UIUC=12)
PAPR/Safety Region (UIUC=13)
FAST FEEDBACK Slot #1 (CQICH Allocation IE)
FAST FEEDBACK Slot #2 (CQICH Allocation IE)
FAST-FEEDBACK Channel Region (2D) (UIUC=0)
FAST FEEDBACK Slot #5 (CQICH Allocation IE)
ACK CH #1
ACK CH #2
ACK CH #3
HARQ ACK Channel Region (2D) (UIUC=11, Type=8)
ACK CH #8
UL Burst #1, CDMA Allocation (UIUC=14)
UL Burst #2, Fast Ranging (UIUC=15, Type=9)
HARQ Sub-Burst #1 (UL HARQ Chase Sub-Burst IE)
UL Burst #3, UL HARQ Burst (HARQ UL MAP IE, UIUC=11, Type=7)
HARQ Sub-Burst #2 (UL HARQ Chase Sub-Burst IE)
HARQ Sub-Burst #3 (UL HARQ Chase Sub-Burst IE)
UL Burst #4, Data Burst (UIUC=1-10)
PAPR/Safety Region (UIUC=13)
Sounding Symbols (Not in Logical Subchannels. UIUC=11, Type=4)
HARQ Sub-Burst #1 (MIMO UL HARQ Chase Sub-Burst IE)
UL Burst #5, MIMO UL HARQ Burst (HARQ UL MAP IE, UIUC=11, Type=7)
HARQ Sub-Burst #2 (MIMO UL HARQ Chase Sub-Burst IE)
HARQ Sub-Burst #3 (MIMO UL HARQ Chase Sub-Burst IE)
HARQ Sub-Burst #4 (MIMO UL HARQ Chase Sub-Burst IE)
UL Burst #6 (MIMO_UL_Basic_IE, DIUC=15, Type=6)
HARQ Sub-Burst #1 (UL HARQ Chase Sub-Burst IE)
UL Burst #7, UL HARQ Burst (HARQ UL MAP IE, UIUC=11, Type=7)
HARQ Sub-Burst #2 (UL HARQ Chase Sub-Burst IE)
UL Burst #8, Data Burst (UIUC=1-10)
UL Burst #9, Data Burst (UIUC=1-10)
PUSC SISO/SIMO Zone (Default First Zone, which may not exist.)
Sounding Zone (UIUC=13)
PUSC Collaborative SM SIMO Zone (UL_Zone_IE, UIUC=15, Type=4)
AMC SISO/SIMO Zone (UL_Zone_IE, UIUC=15, Type=4)

Note: The number of OFDMA symbols in a UL subframe cannot exceed 21. Here, 28 OFDMA symbols are chosen for illustrative purpose.

Figure 7: Example of UL subframe in WiMAX showing ranging regions at the beginning of the subframe.

5 Literature

As a complement to the foray into selected synchronization techniques presented so far, this section aims at briefly summarizing the most important contributions available in open literature.

Pilot-based single-user synchronization techniques include [1][14][15][16][17][18][19]. [14] is one of the first correlation-based timing-synchronization schemes for OFDM, however, it assumes that the impact of time dispersion is negligible. [15] uses two successive identical OFDM blocks to estimate the frequency offset ($P = N$). In [16], a null-symbol followed by a pilot symbol is used to detect the beginning of a frame. [17] uses CAZAC (Constant-Amplitude-Zero-Autocorrelation) sequences. [18] and [19] investigate modifications of [1] avoiding the second pilot symbol: [18] introduces a pilot symbol with more than two identical parts and [19][20] uses a pilot symbol based on a repeated maximum length sequence.

Another class of methods, here referred to as *semi-blind single-user synchronization techniques*, relies on exploiting inherent deterministic signal structure originally not introduced to aid synchronization. The signal repetition in time domain introduced by the CP provides a great amount of structure to be exploited for timing synchronization [2][21][22][23][24]. [2] presents the maximum-likelihood solution for jointly estimating time and frequency offsets based on the CP. A very similar approach employing weighted moving average correlators is described in [22]. [23] improves the range of timing offsets using a modified likelihood function. A principle weakness of CP-based techniques is that they profit from an excess CP. Consequently, there is a tradeoff between CP-based synchronization performance and immunity to delay spread. [24] presents a CP-based time offset estimator that is immune to small frequency offsets and exhibits a certain degree of robustness to time dispersion. Another type of redundancy that can be exploited for synchronization are nulled subcarriers [25][26][27][28]. In a more rigorous approach, [29] exploits quasi-cyclostationarity (second order), introduced not only by a CP but also by frequency-domain guard bands or pulse shaping.

Truly *blind single-user synchronization methods* rely on higher-order statistical properties of the OFDM receive signal. Examples for kurtosis-based methods are [30], which measures distance from Gaussianity and [31], which exploits the spectral line corresponding to carrier frequency offset.

An excellent tutorial overview on *multiuser synchronization* in OFDMA UL in general and BS-based estimation and correction techniques in particular is [32]. Parameter estimation in a consecutive fashion (one user at a time) is discussed in [4].

Multiuser parameter estimation techniques for systems with subband subcarrier allocation in combination with filterbank-techniques can be borrowed from single-user setups [2][33]. A dedicated frequency estimation scheme for interleaved subcarrier allocation based on multiple signal classification (MUSIC) [34] is presented in [5]. Synchronization schemes for unstructured subcarrier allocation in quasi-synchronous schemes using the EM-based [9] SAGE algorithm [8] are introduced in [6][7]. [35] considers estimation of both frequency and time offsets for synchronous OFDMA UL. Detection algorithms for initial ranging at the BS in WiMAX OFDMA are discussed in [36].

Multiuser offset correction by users for OFDMA systems was first mentioned in [33]. For subband subcarrier allocation, again filterbank techniques and individual correction can be applied [32]. A low-complexity correction scheme based on circular convolution is proposed in [10]. Offset correction for unstructured subcarrier allocation includes interference cancellation [11] and linear multiuser detection [12].

6 Summary and conclusion

Synchronization in OFDMA is a challenge. In DL direction, many concepts can be borrowed from single-user OFDM systems. In the UL direction, however, time and frequency misalignment of different users' receive components destroys the orthogonality among users. Both estimation and correction of offsets is more complicated compared to the DL case, which renders the intuitively appealing multiple-access concept of user separation in frequency domain less attractive. While research on the ultimate scheme of joint parameter estimation and offset correction at the BS is ongoing, currently operational systems revert to estimating parameters of one or a few users at a time and offset correction at the user side.

References

[1] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997.

[2] J.-J. van de Beek, M. Sandell, and P. O. Börjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1800–1805, July 1997.

[3] A. M. Tonello, "Multiuser detection and turbo multiuser decoding for asynchronous multitone multiple access systems," in *Proc. 56th IEEE Vehicular Technology Conference (VTC '02-Fall)*, 2002, vol. 2, pp. 970–974.

[4] M. Morelli, "Timing and frequency synchronization for the uplink of an OFDMA system," *IEEE Transactions on Communications*, vol. 52, no. 2, pp. 296–306, Feb. 2004.

[5] Z. Cao, U. Tureli, and B Y. D. Yao, "Deterministic multiuser carrier-frequency offset estimation for interleaved OFDMA uplink," *IEEE Transactions on Communications*, vol. 52, no. 9, pp. 1585–1594, Sept. 2004.

[6] M. O. Pun, M. Morelli, and B C.-C. J. Kuo, "Maximum-likelihood synchronization and channel estimation for OFDMA uplink transmissions," *IEEE Transactions on Communications*, vol. 54, no. 4, pp. 726–736, Apr. 2006.

[7] M. O. Pun, M. Morelli, and C.-C. J. Kuo, "Iterative detection and frequency synchronization for OFDMA uplink transmissions," *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 629-639, Feb. 2007.

[8] J. A. Fessler and A. O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2664–2677, Oct. 1994.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1997.

[10] J. Choi, C. Lee, H. W. Jung, and B Y. H. Lee, "Carrier frequency offset compensation for uplink of OFDM-FDMA systems," *IEEE Transactions on Communications*, vol. 4, no. 12, pp. 414–416, Dec. 2000.

[11] Defeng Huang and K.B. Letaief, "An interference-cancellation scheme for carrier frequency offsets correction in OFDMA systems," *IEEE Transactions on Communications*, vol. 53, no. 7, pp. 1155–1165, July 2005.

[12] Zhongren Cao, U. Tureli, Yu-Dong Yao, and P. Honan, "Frequency synchronization for generalized OFDMA uplink," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '04)*, Dec 2004, vol. 2, pp. 1071–1075.

[13] "IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: Physical and medium access control layersfor combined fixed and mobile operation in licensed bands and corrigendum 1," *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004) Std.*, 2006.

[14] W. D. Warner and C. Leung, "OFDM/FM frame synchronization for mobile radio data communication," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 3, pp. 302–313, Aug. 1993.

[15] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Transactions on Communications*, vol. 42, no. 10, pp. 2908–2914, Oct. 1994.

[16] H. Nogami and T. Nagashima, "A frequency and timing period acquisition technique for OFDM systems," in *Proc 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '95)*, Sep 1995, vol. 3.

[17] U. Lambrette, M. Speth, and H. Meyr, "OFDM burst frequency synchronization by single carrier training data," *Communications Letters, IEEE*, vol. 1, no. 2, pp. 46–48, Mar 1997.

[18] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," *IEEE Communications Letters*, vol. 3, no. 3, pp. 75–77, Mar 1999.

[19] A. J. Coulson, "Maximum likelihood synchronization for OFDM using a pilot symbol: algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 12, pp. 2486–2494, Dec 2001.

[20] A. J. Coulson, "Maximum likelihood synchronization for OFDM using a pilot symbol: analysis," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 12, pp. 2495–2503, Dec 2001.

[21] M. Schmidl and D. C. Cox, "Blind synchronisation for OFDM," *Electronics Letters*, vol. 33, no. 2, pp. 113–114, Jan 1997.

[22] M.-H. Hsieh and C.-H. Wei, "A low-complexity frame synchronization and frequency offset compensation scheme for OFDM systems over fading channels," *IEEE Transactions on Vehicular Technology*, vol. 48, pp. 1596–1609, Sept. 1999.

[23] N. Lashkarian and S. Kiaei, "Class of cyclic-based estimators for frequency-offset estimation of OFDM systems," *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2139–2149, Dec 2000.

[24] D. Landström, S.K. Wilson, J.-J. van de Beek, P. Ödling, and P.O. Börjesson, "Symbol time offset estimation in coherent OFDM systems," *IEEE Transactions on Communications*, vol. 50, no. 4, pp. 545–549, Apr 2002.

[25] H. Liu and U. Tureli, "A high-efficiency carrier estimator for OFDM communications," *IEEE Communication Letters*, vol. 2, pp. 104–106, Apr. 1998.

[26] X. Ma, C. Tepedelenlioglu, G. B. Giannakis, and S. Barbarossa, "Non-data-aided carrier offset estimators for OFDM with null subcarriers: Identifiability, algorithms, and performance," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 2504–2515, Dec. 2001.

[27] S. Barbarossa, M. Pompili, and G. B. Giannakis, "Channel-independent synchronization of orthogonal frequency-division multiple-access systems," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 474–486, Feb. 2002.

[28] M. Ghogho and A. Swami, "Blind frequency-offset estimator for OFDM systems transmitting constant-modulus symbols," *IEEE Communication Letters*, vol. 6, pp. 343–345, Aug. 2002.

[29] H. Bölcskei, "Blind estimation of symbol timing and carrier frequency offset in wireless OFDM systems," *IEEE Transactions on Communications*, vol. 49, pp. 988–999, June 2001.

[30] Y. Yao and G. B. Giannakis, "Blind carrier frequency offset estimation in SISO, MIMO, and multiuser OFDM systems," *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 173–183, Jan. 2005.

[31] M. Luise, M. Marselli, and R. Reggiannini, "Low-complexity blind carrier frequency recovery for OFDM signals over frequency-selective radio channels," *IEEE Transactions on Communications*, vol. 50, no. 7, pp. 1182–1188, July 2002.

[32] M. Morelli, C.-C.J. Kuo, and M.-O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394–1427, July 2007.

[33] J.-J. van de Beek, P.O. Börjesson, M.-L. Boucheret, D. Landström, J.M. Arenas, P. Ödling, C. Östberg, M. Wahlqvist, and S.K. Wilson, "A time and frequency synchronization scheme for multiuser OFDM," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 11, pp. 1900–1914, Nov. 1999.

[34] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar 1986.

[35] Y. Zhang, R. Hoshyar, and R. Tafazolli, "Timing and frequency offset estimation scheme for the uplink of OFDMA systems," *IET Communications*, vol. 2, no. 1, pp. 121–130, January 2008.

[36] H. A. Mahmoud, H. Arslan, and M. K. Ozdemir, "Initial ranging for WiMAX (802.16e) OFDMA," Oct 2006, pp. 1–7.

Part VII

Reduction of PAR

1 Peak-to-average power ratio and its reduction

1.1 Introduction

One of the biggest challenges when implementing multicarrier modulation schemes is the high *peak-to-average power ratio (PAR)* of the transmit signal multiplex $s(n)$. For a given block $s(n), n = 0, \dots, N-1$, this ratio is quantified by

$$\mathrm{PAR} = \frac{\max\limits_{n} |s(n)|^2}{\frac{1}{N} \sum\limits_{n=0}^{N-1} |s(n)|^2}. \tag{1}$$

An example of a signal with a low PAR-value is a constant signal, which has PAR = 1 (0 dB). Figure 1 shows three signals with the same average power but different PAR-values.

The problem with transmit signals that have a high PAR-value is that the whole transmitter chain from the IDFT to the antenna (wireless applications) or to the output connectors (wireline applications) needs to have linear characteristic in the wide amplitude range from the negative to the positive peak amplitude value. Although the average power to be delivered to the antenna or to the line is only a fraction 1/PAR of the peak power, the analogue components—in particular the power amplifier (wireless application) or the line driver (wireline application)—must be designed to deliver the peak power while sustaining linearity. High PAR-values thus cause a *large power consumption*, which affects in particular power-critical systems (such as hand-held devices) and systems that are compactly packed in tight non air-conditioned locations and thus suffer from the heat they generate (such as cabinets with highly-integrated line-card boards).

An example of a non-linearity that can occur in case the transmit chain is excited with a too high peak value is *clipping*. The signal amplitude $s(n)$ is simply limited to the certain value $s_{\max}$, or equivalently, a distortion $d(n)$ is added to the output signal

$$s_{\text{out}}(n) = s(n) + d(n),$$

where

$$d(n) = \begin{cases} \mathrm{sign}\{s(n)\}(s_{\max} - |s(n)|), & |s(n)| > s_{\max} \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

The distortion $d(n)$ is often referred to as *clip noise*. If clipping happens only rarely, which should be the case in properly designed systems, it is likely that there will be only a single clip per multicarrier symbol $s(n), n = 0, \dots, N-1$. Then the distortion $d(n)$ consists of a single sample at the position of the peak and the DTFT of $d(n)$ is a constant. Consequently, rare clipping events introduce white noise, *i.e.*, they introduce distortion of all subcarriers.

Depending on the channel, the receive signal can also exhibit a high PAR-value. For channels that do not introduce intersymbol-interference (ISI), the PAR value of the receive signal is comparable to the

Chapter written by T. Magesacher.

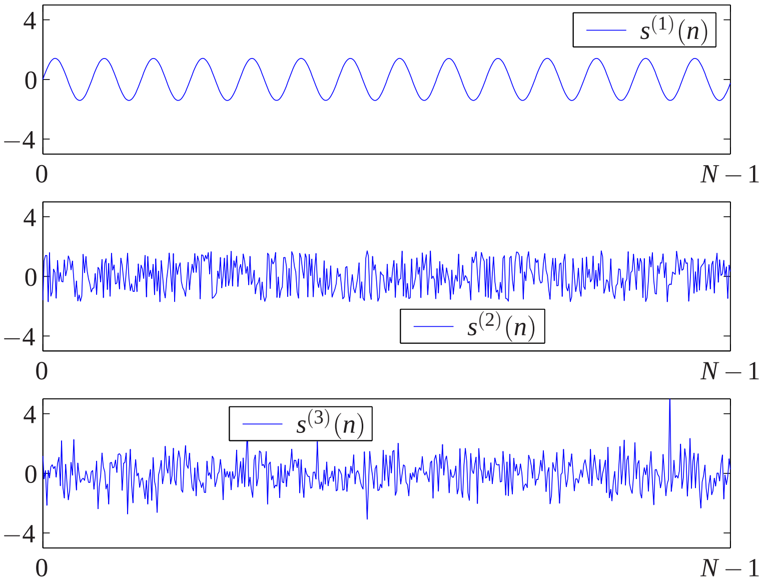


Figure 1: Peak-to-average (PAR) power ratio of three different signals with unit average power ($\frac{1}{N}\sum|s^{(1)}(n)|^2 = \frac{1}{N}\sum|s^{(2)}(n)|^2 = \frac{1}{N}\sum|s^{(3)}(n)|^2 = 1$); $s^{(1)}(n)$: sinusoidal signal, PAR = 3 dB, $\max|s^{(1)}(n)| = 1.41$; $s^{(2)}(n)$: samples are realisations of uniform distribution, PAR = 4.7 dB, $\max|s^{(2)}(n)| = 1.71$; $s^{(3)}(n)$: samples are realisation of Gaussian distribution, PAR = 15.3 dB, $\max|s^{(3)}(n)| = 5.78$.

PAR-value of the transmit signal. The problems discussed above apply also to the receiver. Channels that introduce ISI perform averaging of the transmit signal, which reduces the PAR-value of a signal with high PAR.

The transmit multiplex

$$s(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{j2\pi\frac{kn}{N}}$$

of a multicarrier signal can have high PAR-values. For a large number of subcarriers N and large constellation sizes (many different amplitude and phase values per subcarrier), the transmit signal $s(n)$ is a sum of many independent random variables (if the data is random, which is usually the case). According to the central limit theorem, the distribution of the samples of the transmit signal is close to Gaussian. The plot on the bottom of Figure 1 shows a signal whose samples are drawn from a Gaussian distribution. The PAR value of such a signal is unfortunately rather high compared to transmit signals generated by other modulation methods.

Assuming that $s(n)$ is a real-valued signal with independent and identically Gaussian distributed samples (DMT), *i.e.*, $s(n) \sim \mathcal{N}(0, \sigma^2)$, the probability that $|s(n)|$ exceeds a certain value p is given by

$$\Pr(|s(n)| > p) = 2\frac{1}{\sqrt{2\pi}\sigma} \int_{x=p}^{\infty} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} \mathrm{d}x = 2\mathrm{Q}\left(\frac{p}{\sigma}\right), \tag{3}$$

where the Q-function Q(x) is the complementary Gaussian $\mathcal{N}(0,1)$ cumulative distribution function. Assuming that $s(n)$ is a complex-valued signal with independent and identically, proper, complex Gaussian distributed samples (OFDM), *i.e.*, $\mathrm{Re}\{s(n)\} \sim \mathcal{N}(0, \sigma^2/2)$, $\mathrm{Im}\{s(n)\} \sim \mathcal{N}(0, \sigma^2/2)$, $\mathsf{E}\{\mathrm{Re}\{s(n)\}\,\mathrm{Im}\{s(n)\}\} = 0$, the probability that $|s(n)|$ exceeds a certain value p is given by (cf. Example 51)

$$\Pr(|s(n)| > p) = \mathrm{e}^{-(p/\sigma)^2}, \tag{4}$$

Figure 2 depicts these CCDFs versus PAR level. While (1) measures the PAR-value of a deterministic

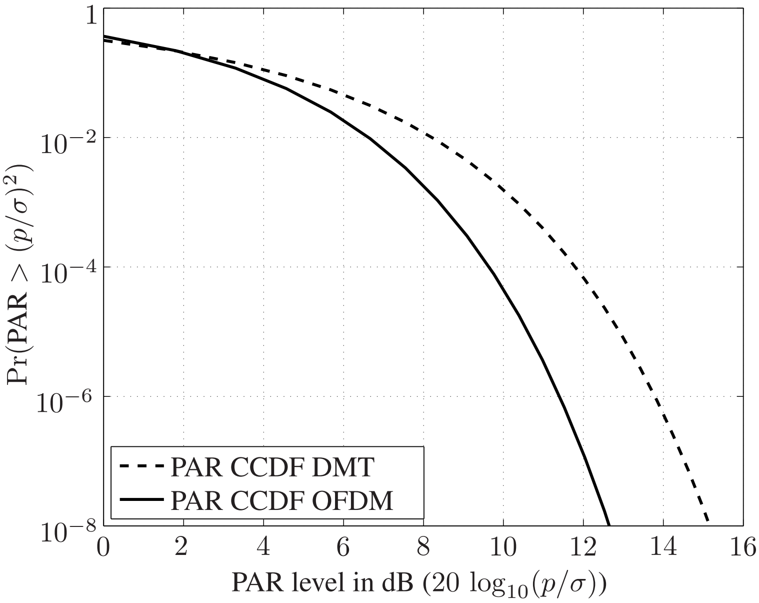


Figure 2: Stochastic characterisation of PAR: probability that the PAR exceeds the PAR level for DMT and OFDM.

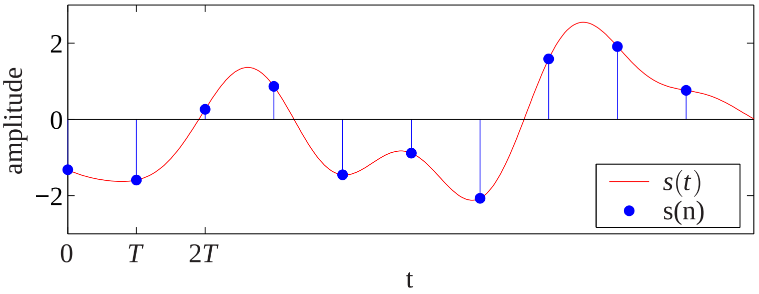


Figure 3: PAR re-growth problem: the continuous-time signal $s(t)$ exhibits larger amplitude values than the sequence $s(n)$. Reducing the PAR of $s(n)$ to a certain level in general does not guarantee that the PAR of $s(t)$ will be reduced to the same value—the PAR value "grows" in the discrete-time to continuous-time conversion process.

signal (or a given realisation of a random signal), (3) and (4) are probabilistic measures for random signals.

Another more block-oriented measure is the probability that the peak-PAR of a length-N multicarrier symbol is below a certain PAR level (CDF) or exceeds a certain PAR level (CCDF). Clearly, this block-PAR (or peak-PAR) CCDF grows with N (cf. Example 52).

An important aspect in the context of PAR is the so called *re-growth problem*: the continuous-time signal $s(t)$, obtained by discrete-time to continuous-time conversion (using interpolation filters and a digital-to-analogue converter), may exhibit larger values than the sequence $s(n)$. Figure 3 illustrates this fact. It has been noticed that reducing the PAR of $s(n)$ to a certain level does not guarantee that the PAR of $s(t)$ will be the same—the PAR value "grows" again. The level of re-growth depends strongly on the discrete-time to continuous-time conversion chain and is still a topic of ongoing research. In practice, the re-growth is often tolerated and methods to combat PAR focus on the baseband signal multiplex $s(n)$. In the following, the most important PAR reduction methods are summarised.

1.2 PAR-reduction methods

There is a number of methods to reduce the PAR of a multicarrier signal. Reducing the PAR is always a tradeoff and involves a reduction of the achievable rate. Almost all PAR reduction methods either introduce noise or consume bandwidth.

- *Clipping and clip modifications*
- *Codes*
- *Selected mapping / interleaving*
- *Partial transmit sequences*
- *Active Constellation Extension*
- *Tone-reservation method*

  In the sequel, we will use the expression 'peaks of a signal' for the samples with the largest absolute values. The idea of tone-reservation is to add a peak-annihilating signal $c(n)$ to the transmit multiplex $s(n)$. The peak-annihilating signal $c(n)$ is chosen such that it has counter-peaks (peaks of opposite sign at the same positions as the peaks of $s(n)$) so that the peaks of the sum $s(n)+c(n)$ are lower than the peaks of $s(n)$. The peak-annihilating signal $c(n)$ is generated by sacrificing a few subcarriers specified by the set $\mathcal{T}_{\mathrm{tr}} = \{t_1, \ldots, t_T\}$. The subcarriers (tones) in the set $\mathcal{T}_{\mathrm{tr}}$ are reserved for constructing the peak-annihilating signal, thus the name *tone reservation* method. Consequently, the values $x_k, k \in \mathcal{T}_{\mathrm{tr}}$ do not carry information but they are chosen such that

  $$c(n) = \frac{1}{\sqrt{N}} \sum_{k \in \mathcal{T}_{\mathrm{tr}}} x_k e^{j2\pi \frac{kn}{N}}$$

  is a “good” peak-annihilator. The desired “good” peak-annihilating property can be formulated as minimisation problem

  $$\begin{array}{ll} \underset{x_k, k \in \mathcal{T}_{\mathrm{tr}}}{\text{minimize}} & \max\limits_{n} |s(n) + c(n)| \\ \text{subject to} & \frac{1}{N} \sum\limits_{n=0}^{N-1} |s(n) + c(n)|^2 \leq \sigma^2 \end{array} \tag{5}$$

  yielding the optimum values $\left[x_{t_1}^{\mathrm{opt}} \quad \ldots \quad x_{t_T}^{\mathrm{opt}}\right]^{\mathrm{T}}$, where σ^2 is the average power per signal sample. For OFDM, (5) can be formulated via a linear objective with quadratic constraints (which is a special case of a quadtratically constrained quadratic program). For DMT, (5) reduces to a linear program. Note that (5) must be solved for every multicarrier symbol, *i.e.*, roughly 4300 times per second for an ADSL system and roughly 300000 times per second for a HiperLAN/2 system.

  A less complex approach that, although suboptimal, performs very well in practice is based on the gradient of the clip noise power with respect to the symbols $x_k, k \in \mathcal{T}_{\mathrm{tr}}$. The gradient specifies the direction and the magnitude for every dimension of the space spanned by the symbols $x_k, k \in \mathcal{T}_{\mathrm{tr}}$ such that the minimum of the error surface (clip noise power surface) is approached via the path of steepest descent.

  The tone reservation method does not require any coordination between the transmitter and the receiver. Often, the receiver uses an algorithm to determine the quality (in terms of signal-to-noise ratio) of the subcarriers by listening to a known training sequence sent by the transmitter. The transmitter can simply pretend that the tones in $\mathcal{T}_{\mathrm{tr}}$ are useless for information transmission by sending pseudo noise sequences instead of the training sequences on these tones. The receiver will then identify these tones as useless for information transmission, which allows the transmitter to exploit them for tone reservation.

- *Tone injection*

  Tone injection is based on the same idea as tone reservation, *i.e.*, on adding a peak-annihilating signal to the transmit signal multiplex. In contrast to tone reservation, however, tone injection does not reserve tones, *i.e.*, it does not keep them from being used for information transmission, but rather attempts to add the peak-annihilating signals on data-carrying tones. The possibilities

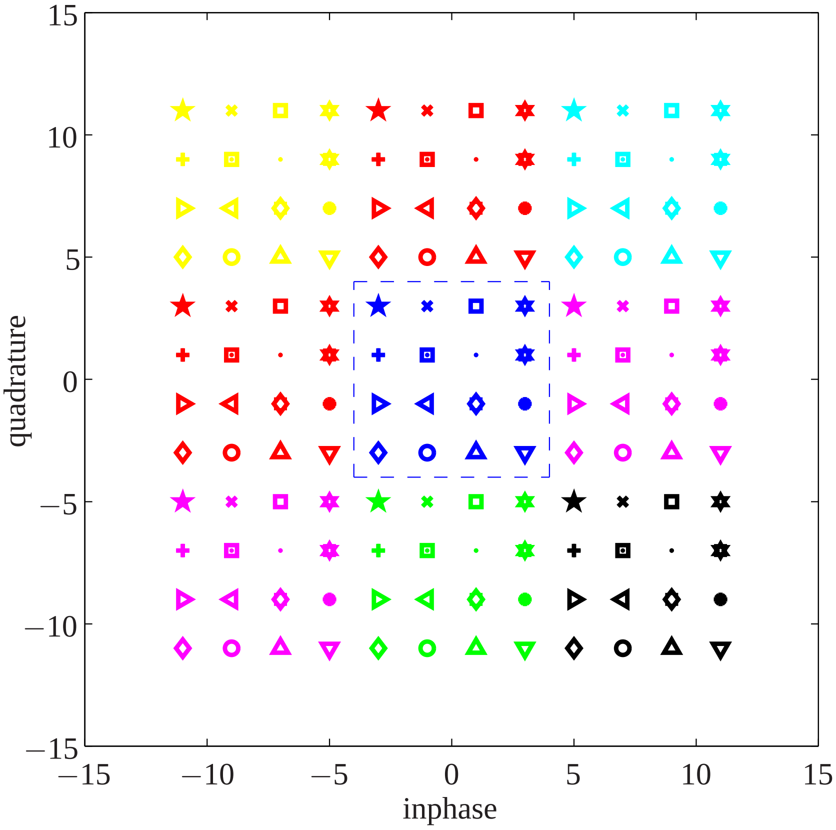


Figure 4: Extension of the 16-QAM constellation for tone injection.

to generate a peak-annihilating signal without introducing additional distortion (*i.e.*, allowing the receiver to decode the transmitted information without performance degradation) are limited. Tone injection is based on the modulo-principle: instead of transmitting a symbol $\underline{x}_k$ from, e.g., a 16-QAM constellation, a corresponding symbol from the extended constellation can be chosen. Figure 4 illustrates the extension of the 16-QAM constellation. Clearly, the receiver has to know that the transmitter may chose to send points from the extended constellation. Modifying the decision device such that it copes with the extended constellation is simple. Note that selecting symbols from the extended constellation increases the average transmit power. If peaks to be combated occur rarely, the average excess power remains low. The tone injection method can be interpreted as an attempt to reduce time-domain peaks, which have a negative impact on the whole system, at the expense of frequency-domain peaks, which only introduce a small excess power.

In addition to the choice of equivalent symbols in the extended constellation (8 possibilities for each point of the 16-QAM constellation in Figure 4), the choice of tones to be used is now very large—any tone could be selected. An exhaustive search over all tones would be too complex. Practical implementations search over a limited set of tones and consider only those tones with symbols from the outer parts of the constellation. The advantage of tone injection compared to tone reservation is that no tone that can be used for information transmission is “wasted”. However, the performance of tone injection is worse compared to tone reservation. Furthermore, the receiver needs to know about it and adapt its decision rule.

References

Part VIII

Spectrum efficiency in OFDM(A)

Abstract

Spectral efficiency or bandwidth efficiency, *i.e.*, exploiting the resource “bandwidth” in the best possible way, is a key prerequisite for communication systems using licensed bands. The spectral leakage of state-of-the-art FFT-based multicarrier modulation is significant. As a consequence, spectral efficiency of real-world OFDMA systems is limited by out-of-band emission and intercarrier interference causing multiuser interference when synchronization is not perfect. While guard bands are an acceptable solution for unlicensed and thus free bands, remedies to reduce spectral leakage and thus increase the number of subcarriers that can be loaded with reasonably high power values are paramount for licensed bands. This chapter deals with physical-layer techniques to improve spectral efficiency in FFT-based multicarrier systems by reducing unwanted spectral leakage.

The chapter begins with an assessment of the resulting emission and intercarrier interference levels in Section 1. Section 2 introduces a suitable system model and Section 3 reviews notion and calculation of spectra, which serve as a measure for emission and interference levels. Section 4 presents a unified framework for techniques that reduce and control both in-band and out-of-band spectral leakage. The focus is on schemes that enhance the spectral efficiency of OFDMA while preserving the orthogonality of subcarriers in time-dispersive channels. Section 6 points out open issues and summarizes the chapter.

To conclude, there is a number of physical-layer remedies that significantly reduce out-of-band leakage and thus improve spectral efficiency in OFDMA setups. The performance of different techniques depends strongly on parameters such as admissible out-of-band levels or length of the cyclic prefix, which prevents a global ranking of different techniques. However, the presented framework, applying techniques in an optimal way in the sense of maximizing the throughput while obeying a PSD limit, allows a fair comparison of techniques for the parameters at hand.

Chapter written by T. Magesacher.

$f \in [0,1)$:	Discrete-time frequency.
$F \in \mathbb{R}_{+}$:	Frequency in Hertz.
$L \in \mathbb{Z}_{+}$:	Total leading cyclic extension in samples.
$L' \in \mathbb{Z}_{+}$:	Total trailing cyclic extension in samples.
$L'' \in \mathbb{Z}_{+}$:	Trailing cyclic extension purged by receiver.
$L_{\mathrm{cp}} \in \mathbb{Z}_{+}$:	Cyclic prefix length in samples.
$L_{\mathrm{cs}} \in \mathbb{Z}_{+}$:	Cyclic suffix length in samples.
$L_{\mathrm{f}} \in \mathbb{Z}_{+}$:	Emission reducing transmit filter's length in samples.
$L_{\mathrm{nyq}} \in \mathbb{Z}_{+}$:	Nyquist-window extension in samples.
$L_{\mathrm{w}} \in \mathbb{Z}_{+}$:	Overlap of consecutive symbols in samples.
$M \in \mathbb{Z}_{+}$:	Channel dispersion (length of channel impulse response minus one) in samples.
$\boldsymbol{m} \in \mathbb{R}_{+}^{Q\times 1}$:	Transmit PSD mask at Q points in the frequency interval $[0,1)$.
$N \in \mathbb{Z}_{+}$:	Symbol length in samples (or equivalently, FFT/IFFT size).
$N' = N + L + L'$:	Symbol length in samples including cyclic extensions.
$N'' = N' - L_{\mathrm{w}}$:	Effective symbol length (N plus extensions minus overlap L_{w}).
$\boldsymbol{p} \in \mathbb{R}_{+}^{N\times 1}$:	Vector containing power values.
$\boldsymbol{q} \in \mathbb{R}_{+}^{Q\times 1}$:	Transmit PSD at Q points in the frequency interval $[0,1)$.
$\mathcal{S}_{\mathrm{i}} \in \{1,\ldots,N\}$:	Set of information-carrying subcarriers.
$\mathcal{S}_{\mathrm{c}} \in \{1,\ldots,N\}$:	Set of compensation subcarriers.
$\boldsymbol{W} \in \mathbb{C}^{N\times N}$:	DFT matrix scaled by $\frac{1}{\sqrt{N}}$ ($\boldsymbol{W}^{\mathrm{H}}$ is the IDFT matrix and $\boldsymbol{W}\boldsymbol{W}^{\mathrm{H}} = \boldsymbol{I}$).
$\boldsymbol{Z}_{\mathrm{add}} \in \{0,1\}^{N'\times N}$:	Matrix adding leading and trailing cyclic extensions.
$\boldsymbol{Z}_{\mathrm{rem}} \in \{0,1\}^{N\times N''}$:	Matrix purging leading and trailing cyclic extensions.

Frequently used abbreviations

CP:	Cyclic Prefix
CS:	Cyclic Suffix
DFT:	Discrete Fourier Transform
DL:	Downlink
FFT:	Fast Fourier Transform
IDFT:	Inverse DFT
IFFT:	Inverse FFT
PSD:	Power Spectral Density
SNR:	Signal-to-Noise Power Ratio
UL:	Uplink

1 Introduction

Spectral efficiency or bandwidth efficiency is an important property of modern communication systems. In general, bandwidth can be a very costly resource. For example, the 3G mobile communication licenses in Germany and in the United Kingdom were auctioned off for an average price of roughly €850 per Hertz bandwidth. Furthermore, spectral efficiency can be the key prerequisite to applying OFDMA in certain setups. OFDM-based overlay systems for aviation communication, for example, share a frequency band with already existing systems. High spectral efficiency is required to allow communication without degrading the performance of in-place systems by temporarily exploiting those parts of the common frequency band that are currently not used [1].

Spectral leakage, *i.e.*, leakage of spectral energy into undesired parts of the spectrum, can significantly lower spectral efficiency in practical systems. Figure 1 illustrates the importance of spectral leakage in OFDMA systems. Like all communication systems, also OFDMA-based systems have to maintain spectral compatibility with systems occupying neighbouring bands—both in downlink (DL) and in uplink (UL) direction (cf. Figure 1a). Spectral energy outside the band covered by the subcarriers is eliminated by continuous-time post-filtering, which constructs the analog transmit signal by suppressing the mirror images of the digital-to-analog converter's output. A primary design goal in practical systems is to keep the requirements for the continuous-time post-filter fairly relaxed for the sake of hardware complexity and power consumption. Therefore, often up to a fifth of the available subcarriers are nulled to form guard bands with reduced spectral energy close to the two band edges.

Besides out-of-band emission, spectral leakage causes interference among users. In an ideal OFDMA system, subcarriers and thus users are orthogonal at the receiver(s). In practice, however, timing offsets, frequency offsets, phase noise, etc. disturb the perfect orthogonality, which causes spectral leakage and thus reduces spectral efficiency. Even when a *synchronous UL timing* scheme is employed, users are initially not synchronized which, in general, results in severe intercarrier interference among adjacent subcarriers belonging to different users (or equivalently, to "out-of-band emission" caused by one user in another user's band) and has a detrimental effect on the performance of synchronization techniques. Typically, this problem is approached via a subband allocation scheme with frequency-domain guard bands (cf. Figure 1b), which allows the base station to separate adjacent users via filterbanks. However, guard bands reduce the exploitable bandwidth and thus lower the system's spectral efficiency.

When general subcarrier allocation and *asynchronous UL timing* are applied, the orthogonality among users is lost, which causes significant interference and thus reduces the achievable throughput in UL direction. Although an asynchronous timing scheme in UL direction annihilates the inherent orthogonality among users, asynchronous systems may be of interest in certain applications. For example, asynchronous timing allows very simple user terminals since all the complexity is shifted to the base station (which performs parameter detection and user alignment in time and frequency). Furthermore, there is no need to feed timing information back to the users. Another motivation is the desire to exploit the resource-scheduling capabilities of OFDMA to the full, which may lead to growing interest in general subcarrier assignment together with asynchronous transmission in future and may thus also emphasize the challenge of keeping out-of-band emissions low.

The out-of-band emission of a rectangularly-windowed FFT-based OFDM subcarrier or a band of subcarriers is significant. Figure 2 depicts the power spectral densities (PSDs) caused by a spectral neighbour using OFDM versus distance in subcarriers. The smaller the distance to the neighbour and the wider the neighbour's band, the more significant spectral leakage becomes. Assuming that the neighbour occupies more than just a few subcarriers, roughly 15 subcarriers closest to the neighbouring band experience interfering PSD levels that are only 10 to 20 dB below the nominal PSD level.

An elegant aspect of OFDM as well as the basis for OFDMA is the fact that, in a perfectly synchronized scenario, the spectra of adjacent subcarriers overlap while orthogonality in time-dispersive channels is preserved. However, in the presence of timing errors and frequency errors, orthogonality is compromised and intercarrier interference occurs. Clearly, the better the alignment in time and frequency (*i.e.*, the better the synchronization of the interfering and the impaired system), the lower the interference level. Figure 3 depicts the worst-case interference power level (relative to the power of the disturbing signal) caused by a *timing offset* versus distance to the disturbing neighbour's band edge for different band widths of the neighbouring OFDM system [2]. A derivation similar to the one presented in [2], yields the worst-case interference levels caused by a *frequency offset* shown in Figure 4.

To conclude, the worst-case interference levels can lie above the interference level suggested by the PSD. However, it should be noted that the actual values of timing offset and frequency offset, which lead

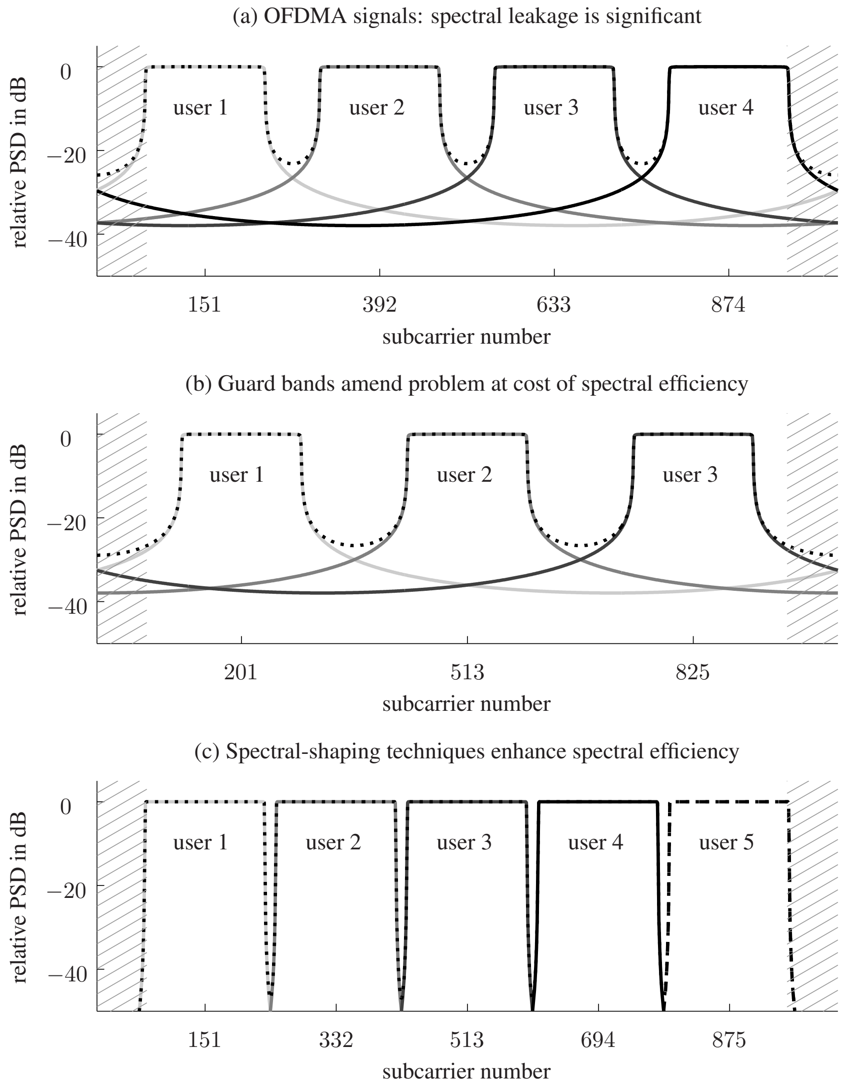


Figure 1: Impact of spectral leakage on spectral efficiency in OFDMA: (a) Spectral leakage of standard FFT-based OFDMA signals is significant and may cause two problems: First, spectral compatibility with coexisting systems needs to be assured which requires power-backoff on subcarriers close to band edges (critical frequency regions are marked by pattern) both in OFDMA-UL and OFDMA-DL; Second, spectral leakage causes interference among users both in DL direction as well as in UL direction of practical schemes, where imperfect synchronization destroys orthogonality. (b) The problem can be amended to a certain extend via frequency-domain guard bands at the cost of spectral efficiency (*i.e.*, accommodating only three users instead of four in the example at hand). (c) Spectral-shaping techniques enhance spectral efficiency of FFT-based OFDMA (accommodating five users instead of four in the example at hand). ($N = 1024$ subcarriers in total. Each subband occupies 165 subcarriers. Dotted line: aggregate power spectral density (PSD) of all users.)

to the worst-case interference levels, vary with distance from the neighbour. Thus, it is very unlikely to experience the worst-case interference levels over a wide range of subcarriers at the same time. The PSD

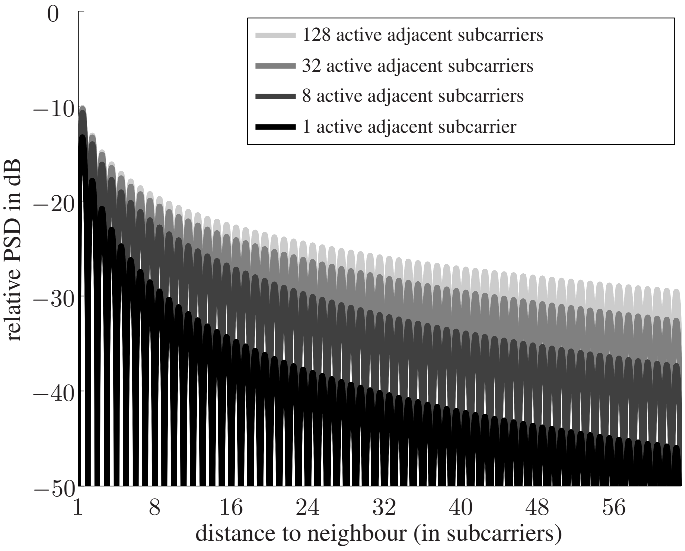


Figure 2: Spectral leakage PSD caused by an asynchronous neighbour versus distance ($N = 1024$).

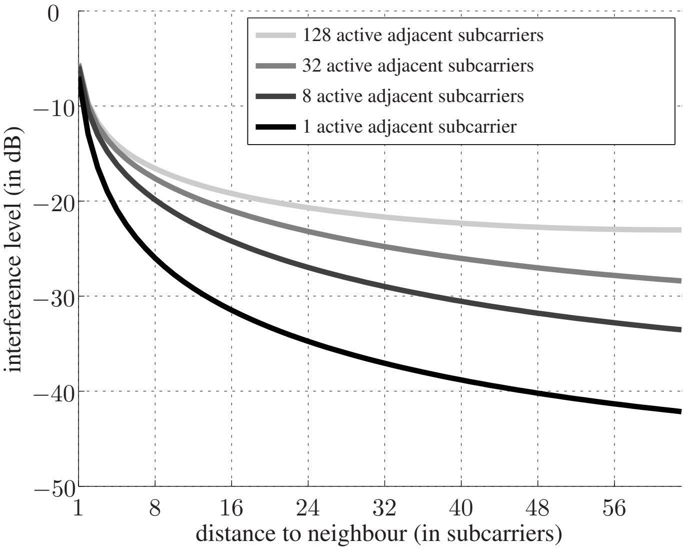


Figure 3: Worst-case intercarrier interference caused by a neighbour (occupying a band of 1, 8, 32, or 128 adjacent subcarriers) with *timing offset* versus distance according to [2] ($N = 1024$).

is thus a reasonable measure for both out-of-band emission and intercarrier interference levels experienced by the impaired system.

Clearly, there is a tradeoff between throughput and out-of-band emission: even the most stringent spectral constraints can be met by reducing the transmit power on subcarriers, which in turn reduces the achievable throughput. A viable approach, and the central topic of this chapter, are physical-layer remedies to reduce spectral leakage and thus increase the exploitable bandwidth (cf. Figure 1c). An optimization-based framework for emission-reduction techniques is introduced. The goal is to exploit the available bandwidth in the best possible way (in the sense of maximizing the throughput) while obeying a PSD limit.

The chapter's focus is on methods that preserve orthogonality among subcarriers in time-dispersive (frequency selective) channels. An alternative approach that has received a lot of attention is the design of pulse shapes (or equivalently, of basis functions) aiming at better spectral confinement in both time-

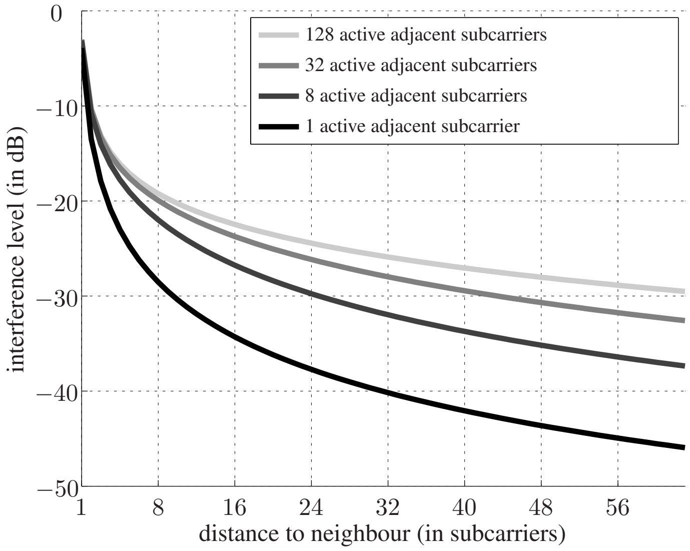


Figure 4: Worst-case intercarrier interference caused by a neighbour (occupying a band of 1, 8, 32, or 128 adjacent subcarriers) with *frequency offset* versus distance ($N = 1024$).

domain and frequency-domain [3]-[8]. Most of these designs aim at basis functions that are orthogonal at the transmitter. However, once they pass the time-dispersive channel, orthogonality is lost, which results in both intercarrier interference and intersymbol interference.

2 System model

In order to formalize the ideas, a matrix-based model of a standard FFT-based multicarrier system with N subchannels is introduced in the following. The ith time-domain OFDM transmit symbol $\boldsymbol{t}_i \in \mathbb{C}^{N' \times 1}$ can be written as vector

$$\boldsymbol{t}_i = \underbrace{\boldsymbol{Z}_{\text{add}}\, \boldsymbol{W}^{\text{H}} \text{diag}\{\sqrt{\boldsymbol{p}}\}}_{\boldsymbol{A}}\, \boldsymbol{x}_i, \tag{1}$$

where the vector $\boldsymbol{x}_i \in \mathbb{C}^{N \times 1}$ contains the transmit data, diag$\{\cdot\}$ denotes a diagonal matrix with the argument vector on the main diagonal, and $\boldsymbol{p} \in \mathbb{R}_{+}^{N \times 1}$ contains the power values assigned to individual subcarriers. The matrix $\boldsymbol{W} \in \mathbb{C}^{N \times N}$ defined as

$$\boldsymbol{W}[n,k] = \frac{1}{\sqrt{N}} \mathrm{e}^{-j2\pi(n-1)(k-1)/N}, \;\; n,k = 1,\ldots,N$$

is the normalized discrete Fourier transform (DFT) matrix. The matrix $\boldsymbol{Z}_{\text{add}} \in \mathbb{R}^{N' \times N}$ defined by

$$\boldsymbol{Z}_{\text{add}} = \begin{bmatrix} \boldsymbol{0}_{L \times (N-L)} \;\; \boldsymbol{I}_L \\ \boldsymbol{I}_N \\ \boldsymbol{I}_{L'} \;\; \boldsymbol{0}_{L' \times (N-L')} \end{bmatrix} \tag{2}$$

introduces cyclic extensions of length L and L' at the beginning and at the end of each symbol, respectively. For the standard transmitter, $L = L_{\text{cp}}$ and $L' = L_{\text{cs}}$ holds, *i.e.*, the lengths of the extensions are equal to the lengths L_{cp} and L_{cs} of the cyclic prefix (CP) and the cyclic suffix (CS), respectively. The total length of each such transmit symbol is $N' = N + L + L'$. The matrix $\boldsymbol{A}$ describes the transmit information processing.

In general, consecutive OFDM symbols may be transmitted with an overlap of L_{w} samples (cf. intersymbol windowing), which yields the transmit sequence

$$t(n) = \begin{cases} \boldsymbol{t}_{\lfloor n/N'' \rfloor}[(n \bmod N'') + 1] + \boldsymbol{t}_{\lfloor n/N'' \rfloor - 1}[(n \bmod N'') + 1 + N''], & (n \bmod N'') = 0, \ldots, L_{\text{w}} - 1 \\ \boldsymbol{t}_{\lfloor n/N'' \rfloor}[(n \bmod N'') + 1], & \text{otherwise} \end{cases},$$

where $N'' = N' - L_\mathrm{w}$ is the effective symbol length accounting for all extensions and overlaps. For the standard transmitter, which transmits symbols without overlap, $N'' = N'$ holds.

Hereinafter, two assumptions are made that greatly simplify further analysis while introducing only a mild loss of generality. First, the data is assumed to be uncorrelated both over time and over subcarriers, *i.e.*,

$$\mathsf{E}\{\boldsymbol{x}_k[m]\boldsymbol{x}_\ell[n]\} = 0, \quad k \neq \ell \text{ or } m \neq n, \tag{3}$$

which is a reasonable assumption for coded (and interleaved) symbols. Second, the data is assumed to be proper complex Gaussian distributed with zero mean and unit variance:

$$\boldsymbol{x}_k[m] \sim \mathcal{CN}(0, 1), \tag{4}$$

The Gaussian assumption is only an approximation when finite alphabets are employed. However, it greatly simplifies the computation of the achievable throughput yielding upper bounds and is thus commonly used.

The time-dispersive Gaussian channel performs linear convolution of the transmit signal with the impulse response $h_n, n = 0, \ldots, M$ of length $M+1$ ($h_0 \neq 0$, $h_M \neq 0$, $h_n = 0$ for $n < 0$ and for $n > M$). Hereinafter, $L_\mathrm{cp} \geq M$ is assumed, which eliminates intersymbol interference and intercarrier interference after removal of the CP so that the received time-domain symbol can be written as $\boldsymbol{r}_i = \boldsymbol{H}\boldsymbol{t}_i + \boldsymbol{n}_i$ (assuming perfect synchronization). The noise vector $\boldsymbol{n}_i \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{C_n})$ is a random vector of length N' with zero-mean proper complex Gaussian entries and covariance matrix $\boldsymbol{C_n} = \mathsf{E}\{\boldsymbol{n}_i\boldsymbol{n}_i^\mathrm{H}\} \in \mathbb{C}^{N' \times N'}$. The channel convolution matrix $\boldsymbol{H} \in \mathbb{C}^{N' \times N'}$ is defined as

$$\boldsymbol{H}[n, k] = h_{n-k}, \qquad n, k = 1, \ldots, N'.$$

The frequency-domain receive signal at the output of the equalizer is given by

$$\boldsymbol{y}_i = \underbrace{\boldsymbol{G}\boldsymbol{W}\boldsymbol{Z}_\mathrm{rem}}_{\boldsymbol{B}}\,\boldsymbol{r}_i.$$

The matrix $\boldsymbol{Z}_\mathrm{rem} \in \mathbb{R}^{N \times (L+N+L'')}$, defined as

$$\boldsymbol{Z}_\mathrm{rem} = \begin{bmatrix} \boldsymbol{0}_{N \times L} & \boldsymbol{I}_N & \boldsymbol{0}_{N \times L''} \end{bmatrix},$$

purges leading and trailing cyclic extensions. The diagonal matrix $\boldsymbol{G} \in \mathbb{C}^{N \times N}$ performs linear per-subchannel equalization and the matrix $\boldsymbol{B}$ describes the receive information processing.

A performance measure that allows a fair comparison of different system variants is the normalized information rate

$$R = \frac{N}{N''} \sum_{k \in \mathcal{S}_\mathrm{i}} \log_2 \left(1 + \frac{\left(\boldsymbol{BHA}(\boldsymbol{BHA})^\mathrm{H}\right)[k, k]}{\left(\boldsymbol{B}\boldsymbol{C_n}\boldsymbol{B}^\mathrm{H}\right)[k, k]} \right) \tag{5}$$

in bit per multicarrier symbol, where $\mathcal{S}_\mathrm{i}$ denotes the set of all information-carrying subcarriers. Note that R/N is a measure for the number of bits per subchannel (of width $1/T_\mathrm{sym}$) per multicarrier symbol (of length T_sym), which corresponds to the number of bits per second per Hertz.

3 Transmit spectra

Open literature on spectrum-shaping techniques includes several measures for spectral energy, which are often just loosely referred to as “spectra”. This section presents a more careful view on these measures and their suitability. Furthermore, power spectral density (PSD) is postulated as the actual measure of interest.

In the following, different spectral measures of a single-frequency finite-length complex exponential, which is the basis function (subcarrier) of standard rectangularly-windowed OFDM, are discussed. In discrete time, this signal can be written as vector $\boldsymbol{w}_k \in \mathbb{C}^{N \times 1}$ given by

$$\boldsymbol{w}_k[n] = \frac{1}{\sqrt{N}} \mathrm{e}^{j2\pi(n-1)(k-1)/N}, \qquad n = 1, \ldots, N,$$

where $k \in \{1, \ldots, N\}$ denotes the subcarrier number and $(k-1)/N$ is the subcarrier's frequency. Note that the IDFT matrix $\boldsymbol{W}^{\mathrm{H}}$ contains the basis functions: $\boldsymbol{W}^{\mathrm{H}} = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \cdots & \boldsymbol{w}_N \end{bmatrix}$. The corresponding continuous-time version is given by

$$w_k(t) = \frac{1}{T} \mathrm{e}^{j2\pi F_k t}, \qquad 0 \le t < T,$$

where $F_k = (k-1)/T$ is the frequency in Hertz and T is the symbol length in seconds.

The Fourier transform $W_k(f)$ of $\boldsymbol{w}_k$ is given by

$$W_k(f) = \sum_n \boldsymbol{w}_k[n]\, \mathrm{e}^{-j2\pi f n} = \frac{1}{\sqrt{N}} \frac{\sin(\pi N(f - k/N))}{\sin(\pi(f - k/N))} \mathrm{e}^{-j\pi(f-k/N)(N-1)} \tag{6}$$

and features in all spectral measures in one way or another. A frequently used measure is $\frac{1}{\sqrt{N}}|W_k(f)|$ depicted in Figure 5. Scaling by $\frac{1}{\sqrt{N}}$ yields $\frac{1}{\sqrt{N}}|W_k(k/N)| = 1$ (note that $\lim_{f \longrightarrow k/N} \frac{\sin(\pi N(f-k/N))}{\sin(\pi(f-k/N))} = N$).

Another measure is the Fourier transform $W_k(F)$ of $w_k(t)$, given by

$$W_k(F) = \int_t w_k(t) \mathrm{e}^{-j2\pi F t} \mathrm{d}t = \frac{\sin(\pi(F - F_k)T)}{\pi(F - F_k)T} \mathrm{e}^{-j\pi(F-F_k)T},$$

whose absolute yields the classical "sinc-F" function[1]

$$|W_k(F)| = \mathrm{sinc}((F - F_k)T).$$

Although both $|W_k(f)|$ and $|W_k(F)|$ are proportional to the energy distribution over frequency, the measure actually describing it is power spectral density (PSD), which is discussed next.

The block-wise processing of OFDM symbols results in a transmit sequence $t(n)$ which can be modeled as cyclostationary random process [9]. The autocorrelation sequence

$$r(n, m) = \mathsf{E}\{t(n) t^*(n - m)\}, \quad n, m = -\infty, \ldots, \infty \tag{7}$$

of $t(n)$ thus depends both on the time instant n and on the lag m and is periodic in n with period N'': $r(n, m) = r(n + N'', m)$. The averaged autocorrelation sequence [10] is given by

$$r(m) = \frac{1}{N''} \sum_{n=n_0}^{n_0+N''-1} r(n, m) \tag{8}$$

where the average is taken over a period of N'' samples starting at an arbitrary time instant n_0. The transmit PSD is the Fourier transform of $r(m)$:

$$\mathit{PSD}(f) = \sum_m r(m)\, \mathrm{e}^{-j2\pi f m} \tag{9}$$

Let us now look at the PSD of an OFDM signal when only a single subcarrier, say subcarrier No. k, is in use. The transmit symbols are then given by $\boldsymbol{t}_i = \boldsymbol{w}_k \sqrt{\mathbf{p}[k]} \boldsymbol{x}_i[k]$. Assuming $L_{\mathrm{cp}} = L_{\mathrm{cs}} = 0$ and no overlap of consecutive blocks, $N'' = N$ holds and the averaged autocorrelation function is given by

$$r_k(m) \overset{(7),(8)}{=} \frac{1}{N} \sum_n \boldsymbol{w}_k[n] \sqrt{\mathbf{p}[k]}\, \underbrace{\mathsf{E}\{\boldsymbol{x}_i[k] \boldsymbol{x}_i^*[k]\}}_{\overset{(4)}{=} 1} \sqrt{\mathbf{p}[k]}\, \boldsymbol{w}_k^*[n - m]$$

which yields the transmit PSD

$$\mathit{PSD}_k(f) \overset{(9)}{=} \sum_m r_k(m)\, \mathrm{e}^{-j2\pi f m} = \frac{\mathbf{p}[k]}{N} \underbrace{\sum_n \boldsymbol{w}_k[n] \mathrm{e}^{-j2\pi f n}}_{\overset{(6)}{=} W(f)} \underbrace{\sum_\ell \boldsymbol{w}_k^*[\ell] \mathrm{e}^{j2\pi f \ell}}_{\overset{(6)}{=} W^*(f)} = \frac{\mathbf{p}[k]}{N} |W_k(f)|^2$$

[1] $\mathrm{sinc}(x) \hat{=} \sin(\pi x)/(\pi x)$

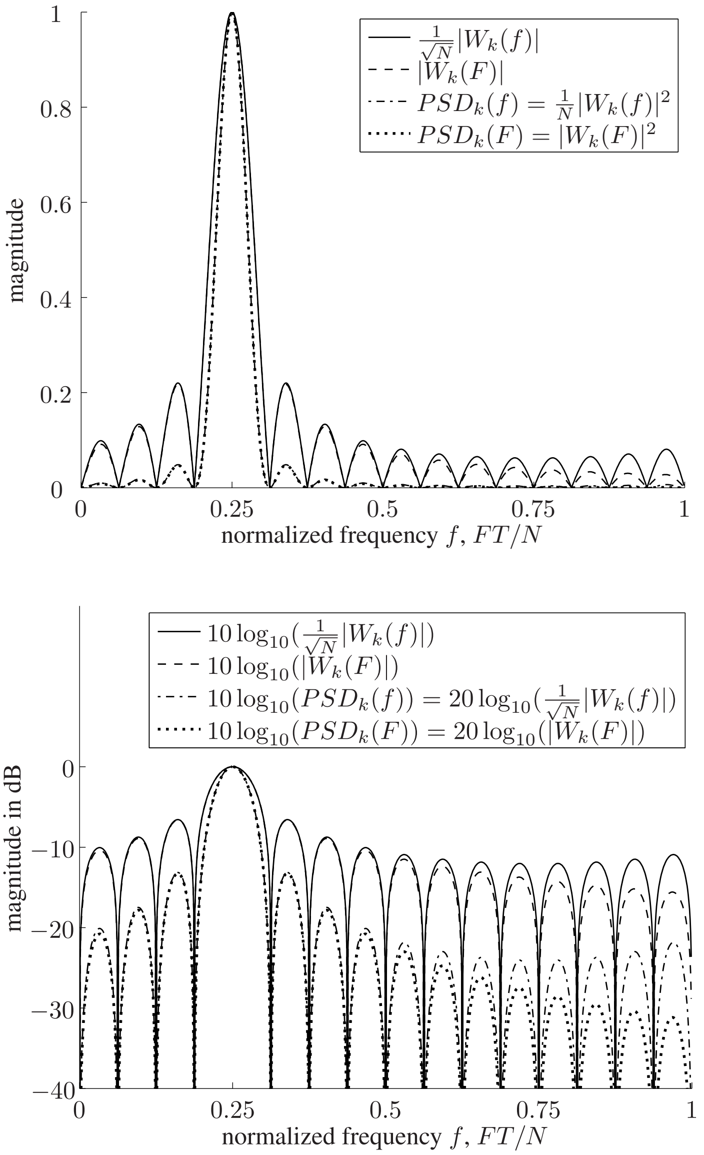


Figure 5: "Spectra" of finite-length complex exponential (OFDM basis function): Fourier transforms and PSDs. $N = 16$, $L_{\text{cp}} = 0$, $k = 5$, $\mathbf{p}[5] = 1$.

Analogously, the PSD of $w_k(t)$ is

$$PSD_k(F) = |\, W_k(F)|^2.$$

Figure 5 depicts Fourier transforms and PSDs both in linear and logarithmic scale. The Fourier transforms are only proportional to the square root of energy distributed over frequency and thus not fully adequate as approximations for actual PSD levels. Still, they may be preferred in some settings since they are linear functions of the data modulating the basis functions. The sidelobes of $W_k(F)$ and $PSD_k(F)$ decay steadily with distance from the mainlobe's center frequency k/T. The periodicity of $W_k(f)$ and $PSD_k(f)$ causes the sidelobes to rise again for frequencies more than 0.5 away from k/N.

The larger the number of subcarriers in a system, the faster the decay of the sidelobes' magnitudes, as the basis functions depicted in the band center in Figure 6 show. At first glance, spectral sidelobes do not seem to be a problem for large systems (for example, $N = 2048$). However, the larger the system, the more subcarriers contribute to the aggregate out-of-band PSD. In Figure 6, subcarriers within the normalized frequency range $k/N \in [1/8, 7/8]$ are in-band, while the rest of the carriers are out-of-band.

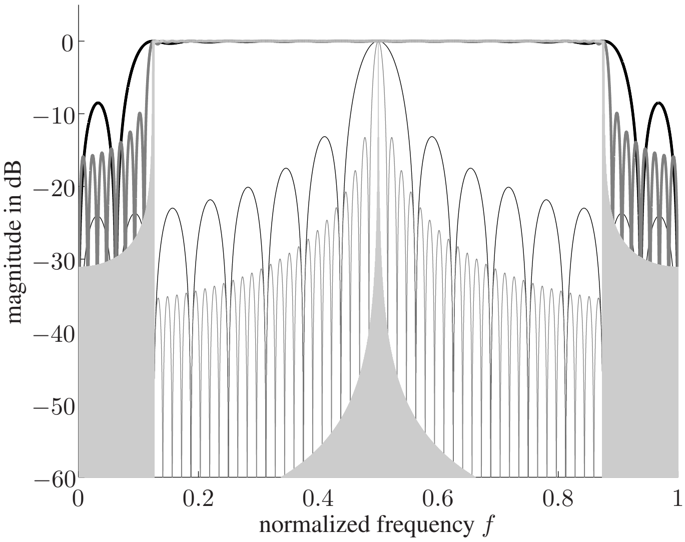


Figure 6: PSDs $PSD(f)$ of basis function (thin lines) and aggregate PSD (thick lines) for different number of subcarriers: $N = 16$ (black), $N = 64$ (gray), $N = 2048$ (light gray). Basis functions are depicted for frequency $k/N = 1/2$. Aggregate PSDs include all subcarriers with frequencies in the in-band range $k/N \in [1/8, 7/8]$.

Hereinafter, the out-of-band region is referred to as the union of all subbands within the normalized frequency range $[0, 1)$ that are not used by active subcarriers. Typically, subcarriers in the vicinity of the band edges (that is, subcarriers with normalized frequencies close to 0 and close to 1) cause spectral energy that is critical in terms of spectral compatibility. Even in very large systems, the aggregate out-of-band PSD decays only slowly with distance from the in-band region.

Note that in the course of discrete-time to continuous-time conversion, the spectra and PSDs of discrete-time signals depicted in the previous figures are repeated in frequency and weighted by a function depending on the actual interpolation (for example, a simple hold-element producing a step-like waveform corresponds to weighting with a sinc-function). "True" out-of-band emission, *i.e.*, spectral energy outside the band corresponding to frequencies $[0, 1)$ caused by these replica, have to be taken care of by continuous-time filtering (or, equivalently, higher-quality interpolation in the course of discrete-time to continuous-time conversion). The critical frequency part are the band edges where the continuous-time filter is faced with the challenge of providing a brickwall-like magnitude response. Hereinafter, the focus is on techniques to shape the spectrum in the above-defined out-of-band part of the frequency range $[0, 1)$, which can be influenced by discrete-time processing and greatly relaxes the requirements on the continuous-time filter.

In case the data streams transmitted on individual subcarriers are mutually uncorrelated, the aggregate PSD, as for example depicted in Figure 6, is the sum of the PSDs of all in-band subcarriers. In case the data is correlated over subcarriers, the PSD is computed according to (9). In order to cast optimization problems, a matrix-based formulation for the transmit PSD $\boldsymbol{q} \in \mathbb{R}_+^{Q\times 1}$, given at Q points in the normalized frequency range $[0, 1)$, as a function of the transmit signal's autocorrelation matrix

$$\boldsymbol{C}_t \mathrel{\hat{=}} \mathsf{E}\{\boldsymbol{t}_i \boldsymbol{t}_i^{\mathrm{H}}\} \overset{(1),(3),(4)}{=} \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$$

proves useful and can be written as

$$\boldsymbol{q} = \boldsymbol{Q}\,\mathrm{vec}\,(\boldsymbol{C}_t)\,. \tag{10}$$

The transform matrix $\boldsymbol{Q}$ computes the averaged autocorrelation function and subsequently takes a length-Q DFT according to (8) and (9), respectively, and $\mathrm{vec}\,(\cdot)$ stacks the columns of its matrix argument into one column vector.

4 Egress reduction techniques

4.1 Filtering

Filtering may be the most straightforward approach to reducing out-of-band emission. Filtering of the transmit sequence $t(n)$, or an oversampled version of it, results in an effective impulse response of transmit filter (length $L_{\mathrm{f}}+1$) and channel (length $M+1$) of length $M+L_{\mathrm{f}}+1$. In order to preserve orthogonality (avoid ISI and ICI) in time-dispersive channels, the cyclic prefix has to be extended to length $L_{\mathrm{cp}} = M+L_{\mathrm{f}}$. Hence, one goal of the filter design for this purpose is clearly to keep L_{f} low. Apart from the desired suppression in the stopband, the filter may introduce an undesired ripple in the passband, which has a negative impact on the bit error rate performance. A predistortion in frequency domain (before the IDFT) can be employed to mitigate this ripple.

4.2 Power loading

Instead of best-effort emission reduction by simply nulling ("turning off") subcarriers close to band edges, a more advanced approach is to maximize the throughput under emission limits. Finding the power values $\mathbf{p}^{(\mathrm{opt})}$ that maximize R given by (5) under a PSD constraint can be formulated as

$$\mathbf{p}^{(\mathrm{opt})} = \begin{array}{l} \arg\max\limits_{\mathbf{p}} R \\ \text{subject to} \quad \boldsymbol{q} \leq \boldsymbol{m} \end{array}, \tag{11}$$

where $\boldsymbol{m} \in \mathbb{R}_{+}^{Q\times 1}$ denotes the PSD mask and $\boldsymbol{q}$ is given by (10). Problem (11) can be cast in convex form [11] and is, in essence, a constrained waterfilling problem. For the power-loading transmitter, all subcarriers potentially carry information ($\mathcal{S}_{\mathrm{i}} = \{1,\ldots,N\}$). Each symbol is extended by CP ($L = L_{\mathrm{cp}}$) and CS ($L' = L_{\mathrm{cs}}$) before transmission without overlap ($L_{\mathrm{w}} = 0$). The receiver purges the leading $L = L_{\mathrm{cp}}$ and the trailing $L'' = L_{\mathrm{cs}}$ samples. Note that through the choice $\mathcal{S}_{\mathrm{i}} = \{1,\ldots,N\}$ no subcarriers are *a priori* nulled—power loading will pick the best tradeoff between a subcarrier's contribution to R given by (5) and the spectral emission it causes.

4.3 Transmit windowing

In general, the term *windowing* is used to denote multiplication of a signal by a finite-length weighting function—the so called window (or window function). Based on the observation that the sharp transitions in time domain at the symbol borders due to changing data cause high-frequency components in the frequency domain, it is intuitively clear that a good window function should mitigate these transitions by scaling down the power at symbol borders. In fact, it can be shown that the quality of a continuous-time window function with respect to its out-of-band emission is proportional to its number of continuous derivatives. It is not surprising that the rectangular window (that is, no windowing) used in standard OFDM, which is not even continuous itself, has rather high out-of-band components.

It is worth noting that transmit windowing is equivalent to pulse-shape design. Hereinafter, the focus is on window designs that preserve orthogonality among subcarriers in time-dispersive channels, which may require window-related receive processing. Dedicated design of pulse shapes (or equivalently, basis functions) usually aims at improving the confinement of the pulse in time-domain and frequency domain in order to enhance its immunity to time-offsets and frequency-offsets. Hereinafter, the goal is to maximize the achievable information rate under a PSD constraint.

Intersymbol windowing

Intersymbol transmit windowing [11][12] introduces an additional cyclic extension of L_{w} at both the beginning and the end of a multicarrier symbol before shaping these extended parts with a window function, as depicted in Figure 7. The overall cyclic extensions of beginning and end of each symbol are thus $L = L_{\mathrm{cp}} + L_{\mathrm{w}}$ and $L' = L_{\mathrm{cs}} + L_{\mathrm{w}}$, respectively. This kind of windowing appears in several communication standards, *e.g.*, in wireline communications [13]. A windowed transmit symbol of length N' can be written as

$$\boldsymbol{t} = \underbrace{\mathrm{diag}\{\boldsymbol{u}\}\boldsymbol{Z}_{\mathrm{add}}\,\boldsymbol{W}^{\mathrm{H}}\mathrm{diag}\{\sqrt{\boldsymbol{p}}\}}_{\boldsymbol{A}}\,\boldsymbol{x},$$

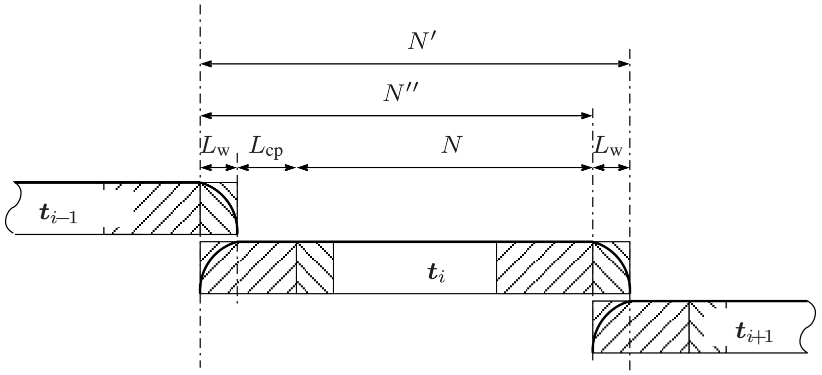


Figure 7: Intersymbol windowing: consecutive blocks are cyclically extended and only these extended parts are windowed. Consecutive blocks overlap and the sum of overlapping blocks is transmitted. Orthogonality among subcarriers in time-dispersive channels is preserved. (For simplicity, $L_\mathrm{cs} = 0$.)

where the vector $\boldsymbol{u} \in \mathbb{R}^{N' \times 1}$ is the time-domain transmit window which obeys

$$\boldsymbol{u}[k] = \begin{cases} \boldsymbol{u}[N' - k + 1], & k = 1, \ldots, L_\mathrm{w} \\ 1, & k = L_\mathrm{w} + 1, \ldots, N' - L_\mathrm{w} \end{cases} . \tag{12}$$

The excess intervals of consecutive OFDM symbols may overlap (cf. Figure 7), which justifies the name intersymbol windowing. The effective receive-block length is thus $N'' = L_\mathrm{w} + L_\mathrm{cp} + N + L_\mathrm{cs}$ samples and the length of the suffix removed by the receiver is $L'' = L_\mathrm{cs}$. Compared to standard OFDM, the symbols are thus L_w samples longer.

As an example, consider two practically relevant window shapes: the linear-slope window

$$\boldsymbol{u}[k] = k/(L_\mathrm{w} + 1), \quad k = 1, \ldots, L_\mathrm{w} \tag{13}$$

and the root raised-cosine window

$$\boldsymbol{u}[k] = \left(\frac{1}{2} \cos(\frac{\pi k}{L_\mathrm{w} + 1} - \pi) + \frac{1}{2} \right)^{1/2}, \quad k = 1, \ldots, L_\mathrm{w}, \tag{14}$$

for which all derivatives are continuous. While the root raised-cosine window keeps the average per-sample power of $t(n)$ constant, the linear-slope window might allow a more efficient implementation for certain values of L_w. Figure 8 compares the resulting PSDs of a single subcarrier with and without intersymbol windowing. The two exemplary window shapes perform similarly in terms of spectral leakage. Clearly, there is a tradeoff between spectral leakage and window length.

Intersymbol windowing requires joint-optimization of both window function $\boldsymbol{u}$ and power values $\boldsymbol{p}$ with R as objective function, *i.e.*,

$$\{\boldsymbol{u}^{(\mathrm{opt})}, \boldsymbol{p}^{(\mathrm{opt})}\} = \begin{array}{c} \arg\max\limits_{\boldsymbol{u},\boldsymbol{p}} R \\ \text{subject to } \boldsymbol{q} \leq \boldsymbol{m} \end{array} ,$$

which is a non-convex problem. All subcarriers are used for transmission of information ($\mathcal{S}_\mathrm{i} = \{1, \ldots, N\}$). Note that L_w is an optimization parameter concealed in $\boldsymbol{u}$. It turns out that simple window shapes, such as the linear-slope window or the root raised-cosine window, often yield results that are close to those achieved with the optimal window shape [11]. In the sequel, the linear-slope window of optimal length is used together with optimal power loading (in the sense of maximizing R/N) according to (11) for comparison with other methods.

Intrasymbol windowing

In contrast to intersymbol windowing, an intrasymbol window does not extend the symbol but weights the length-N symbol before cyclic extension [14]. A transmit symbol can be written as vector

$$\boldsymbol{t} = \boldsymbol{Z}_\mathrm{add}\, \mathrm{diag}\{\boldsymbol{u}\}\, \boldsymbol{W}^\mathrm{H} \mathrm{diag}\{\sqrt{\boldsymbol{p}}\} \boldsymbol{x}, \tag{15}$$

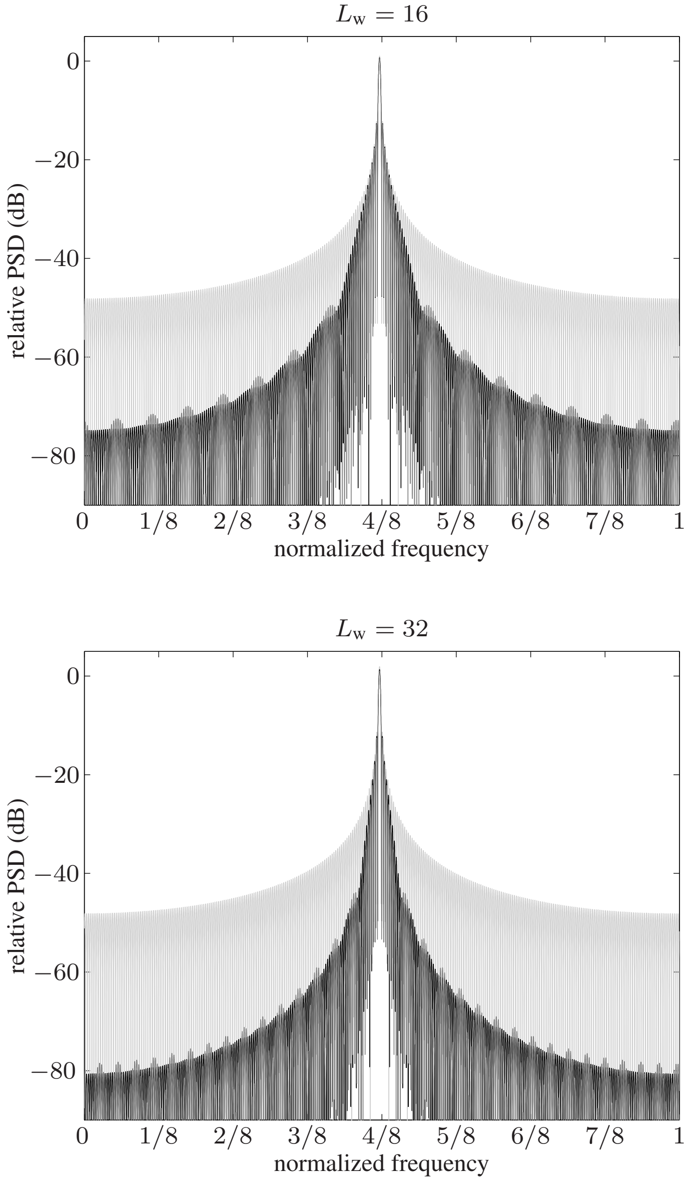


Figure 8: PSDs of single subcarriers without windowing (light gray) and intersymbol windowing: linear-slope window given by (13) (gray), root raised cosine window given by (14) (black). $N = 256$, $L_w = 16$ (top), $L_w = 32$ (bottom).

where $\boldsymbol{u} \in \mathbb{C}^{N\times 1}$ is the intrasymbol window. After cyclic extension, the extended window $\boldsymbol{Z}_{\text{add}}\boldsymbol{u}$ itself has the cyclic-prefixed property (cf. Figure 9). This type of window is referred to as intrasymbol window—its extent is limited to a single symbol and no overlap occurs ($L_w = 0$) and all subcarriers are used for transmission of information ($\mathcal{S}_{\text{i}} = \{1, \ldots, N\}$).

Equivalently to (15), a transmit block can also be written as

$$\boldsymbol{t} = \underbrace{\boldsymbol{Z}_{\text{add}}\,\boldsymbol{W}^{\text{H}}\,\boldsymbol{U}\text{diag}\{\sqrt{\boldsymbol{p}}\}}_{\boldsymbol{A}}\,\boldsymbol{x},$$

where the transmit windowing is performed in DFT-domain by the circulant matrix $\boldsymbol{U} \in \mathbb{C}^{N\times N}$ defined

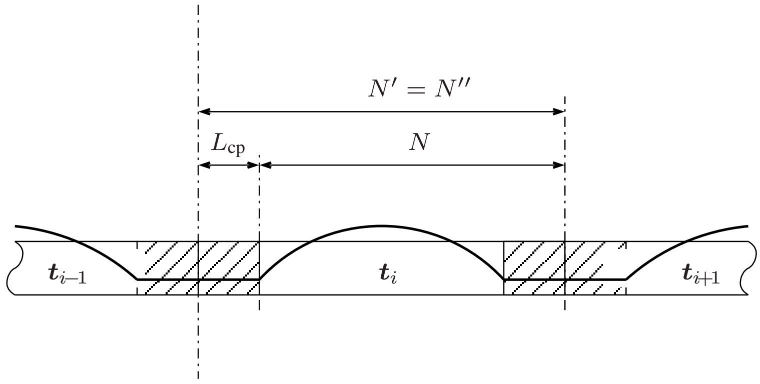


Figure 9: Intrasymbol windowing: the entire block is windowed and consecutive blocks do not overlap. Windowing-related receive processing according to (16) is required to restore orthogonality among subcarriers. (For simplicity, $L_{\mathrm{cs}} = 0$.)

as

$$\boldsymbol{U}[n,k] = \frac{1}{\sqrt{N}}(\boldsymbol{W}^{\mathrm{H}}\boldsymbol{u})[((k-n) \bmod N)+1], \;\; n,k = 1,\dots,N.$$

Intrasymbol windowing destroys the orthogonality of the received basis functions and thus necessitates shaping-related receive processing, which in turn amplifies the noise and thus reduces the signal-to-noise power ratio (SNR). In order to restore the orthogonality among subcarriers, the receiver needs to invert $\boldsymbol{U}$, which yields the data estimates

$$\widehat{\boldsymbol{x}} = \underbrace{\boldsymbol{U}^{-1}\boldsymbol{G}\boldsymbol{W}\boldsymbol{Z}_{\mathrm{rem}}}_{\widehat{=}\boldsymbol{B}}\boldsymbol{r} \tag{16}$$

Note that the shaping related receive processing is channel independent ($\boldsymbol{U}^{-1}$ depends only on the window).

Joint optimization of both window function $\boldsymbol{u}$ and power values $\boldsymbol{p}$ with the normalized information rate as objective function can be formulated as

$$\{\boldsymbol{u}^{(\mathrm{opt})}, \boldsymbol{p}^{(\mathrm{opt})}\} = \begin{array}{l} \arg\max\limits_{\boldsymbol{u},\boldsymbol{p}} R \\ \text{subject to} \;\; \boldsymbol{q} \le \boldsymbol{m} \end{array},$$

where all subcarriers are used for transmission of information ($\mathcal{S}_{\mathrm{i}} = \{1,\dots,N\}$). The dependence of $\boldsymbol{U}^{-1}$ (and thus of $\boldsymbol{B}$ in R) on $\boldsymbol{u}$ renders the problem non-convex [15]. An alternative design method that is optimal in the sense of maximizing the mainlobe energy of the transmit basis functions' spectra was suggested in [16] and a performance comparison was presented in [17]. Hereinafter, the maximum mainlobe-energy window of optimal length is employed with optimal power loading (in the sense of maximizing R/N) according to (11) for comparison. Finally, it should be noted that intrasymbol windowing can be regarded as a special form of pulse shaping yielding non-orthogonal pulses that can, however, be restored at the receiver with comparably low complexity.

Nyquist windowing

Nyquist windowing at the transmitter [18] is a variation of intrasymbol windowing applied together with zero padding. It allows intrasymbol shaping while preserving subcarrier orthogonality at the cost of very simple receive processing. A transmit symbol of length $N' = N + 2L_{\mathrm{nyq}} + L_{\mathrm{cp}}$ can be written as

$$\boldsymbol{t} = \underbrace{\mathrm{diag}\{\boldsymbol{u}\}\boldsymbol{Z}_{\mathrm{add}}\boldsymbol{W}^{\mathrm{H}}\mathrm{diag}\{\sqrt{\boldsymbol{p}}\}}_{\widehat{=}\boldsymbol{A}}\boldsymbol{x},$$

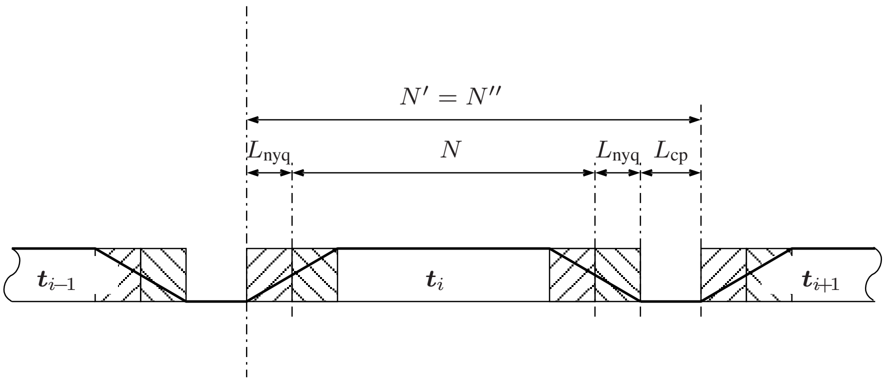


Figure 10: Nyquist windowing: a block is cyclically extended and shaped by a Nyquist window. Zero padding together with simple receive processing (adding transients according to (17)) maintains orthogonality among subcarriers.

where $\boldsymbol{Z}_{\rm add}$ given by (2) introduces cyclic extensions of length $L = L_{\rm nyq}$ and $L' = L_{\rm nyq} + L_{\rm cp}$. The window $\boldsymbol{u} \in \mathbb{C}^{N' \times 1}$ fulfills

$$\boldsymbol{u}[k] = \begin{cases} 1 - \boldsymbol{u}[k+N], & k = 1, \dots, 2L_{\rm nyq} \\ 1 & k = 2L_{\rm nyq} + 1, \dots, N \\ 0 & k = N + 2L_{\rm nyq} + 1, \dots, N' \end{cases}$$

Note that the last $L_{\rm cp}$ samples are nulled by the window, which corresponds to zero padding with $L_{\rm cp}$ samples. No overlap of adjacent symbols occurs ($L_{\rm w} = 0$), as illustrated in Figure 10, and all subcarriers are used for transmission of information ($\mathcal{S}_{\rm i} = \{1, \dots, N\}$).

Instead of purging samples, the receiver sums leading and trailing transients according to

$$\widehat{\boldsymbol{x}} = \underbrace{\boldsymbol{G}\boldsymbol{W}\left[\begin{bmatrix}\boldsymbol{0}_{N-L_{\rm nyq},L_{\rm nyq}} \\ \boldsymbol{I}_{L_{\rm nyq}}\end{bmatrix} \boldsymbol{I}_N \begin{bmatrix}\boldsymbol{I}_{L_{\rm nyq}+L_{\rm cp}} \\ \boldsymbol{0}_{N-L_{\rm nyq}-L_{\rm cp},L_{\rm nyq}+L_{\rm cp}}\end{bmatrix}\right]}_{\widehat{=}\boldsymbol{B}} \boldsymbol{r}. \tag{17}$$

It can be shown that $\boldsymbol{BHA} = \boldsymbol{I}$, *i.e.*, the orthogonality among subcarriers in time-dispersive channels is preserved in time-dispersive channels.

Nyquist windowing allows weighting of the entire symbol, which should result in low spectral leakage. On the other hand, each symbol needs to be extended cyclically by $2L_{\rm nyq}$ samples. Furthermore, zero padding makes each symbol $L_{\rm cp}$ samples longer. The transient-adding procedure at the receiver described by (17) collects more noise compared to CP-removal. On the other hand, zero padding allows for a higher average transmit power since, in contrast to cyclic prefixing, no power is wasted when padding zeros. To summarize, it is *a priori* not clear whether Nyquist windowing has an advantage over other windowing techniques, which motivates casting all schemes into a framework for comparison.

4.4 Spectral compensation

An advanced approach, hereinafter referred to as spectral compensation, divides the N subcarriers into two sets: a set $\mathcal{S}_{\rm c}$ of compensation subcarriers and a set $\mathcal{S}_{\rm i} = \{1, \dots, N\} \backslash \mathcal{S}_{\rm c}$ of information-carrying subcarriers. The transmitter modulates each compensation subcarrier $n \in \mathcal{S}_{\rm c}$ with a linear combination of the data transmitted over a set $\mathcal{S}_{\rm r}(n) \subseteq \mathcal{S}_{\rm i}$ of reference subcarriers. Using properly chosen subcarriers as compensation subcarriers leads to a better exploitation of the spectral mask in the sense that the power on some of the information subcarriers can be increased. Spectral compensation does not require any shaping-related processing at the receiver, *i.e.*, simple subcarrier-wise equalization is possible since the orthogonality of the received basis functions is preserved. Consequently, the technique conforms with any standard.

A transmit block of N' samples can be written as vector

$$\boldsymbol{t} = \underbrace{\boldsymbol{Z}_{\rm add} \boldsymbol{W}^{\rm H} \boldsymbol{S}}_{\boldsymbol{A}} \boldsymbol{x},$$

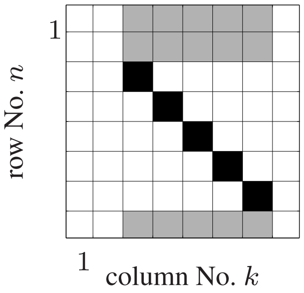


Figure 11: Structure of spectral compensation matrix $\boldsymbol{S}$ given by (18). Black: information-subcarrier power-scaling coefficients; gray: spectral compensation coefficients; white: zero. Example for $N = 8$; $\mathcal{S}_{\mathrm{c}} = \{1, 2, 8\}$; $\mathcal{S}_{\mathrm{r}}(n) = \mathcal{S}_{\mathrm{i}}, n \in \mathcal{S}_{\mathrm{c}}$.

where the shaping matrix $\boldsymbol{S} \in \mathbb{C}^{N \times N}$ defined as

$$\boldsymbol{S}[n,k] = \begin{cases} \sqrt{\boldsymbol{p}[n]}, & n = k \in \mathcal{S}_{\mathrm{i}} \\ s_{n,k}, & n \in \mathcal{S}_{\mathrm{c}}, k \in \mathcal{S}_{\mathrm{r}}(n) \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

for $n, k = 1, \ldots, N$ performs both the power loading for the subcarriers in $\mathcal{S}_{\mathrm{i}}$ and the linear combination of the data for the subcarriers in $\mathcal{S}_{\mathrm{c}}$. Figure 11 illustrates the structure of an exemplary shaping matrix. Apart from CP and CS, no additional extension is required, *i.e.*, $L = L_{\mathrm{cp}}$ and $L' = L'' = L_{\mathrm{cs}}$. Generally, the receiver processes only the subcarriers in $\mathcal{S}_{\mathrm{i}}$ (a more complex receiver could exploit the redundancy available through the subcarriers in $\mathcal{S}_{\mathrm{c}}$, however, this would imply shaping-related receive processing, which is not considered here).

To be best of the author's knowledge, the idea of spectral compensation was first mentioned in [19][20] followed by a considerable research effort to gain performance and insight [21]-[27]. Spectral compensation techniques can be classified as *active* or *passive*. Active methods re-compute the coefficients in $\mathbf{S}_i$ for each block based on the transmit data $\boldsymbol{x}_i$ [21]-[25]. Thus, active spectral compensation is very similar to tone reservation used for reducing the transmit signal's peak-to-average power ratio [28]. All active methods are based on criteria of more or less *ad-hoc* nature, which makes it difficult to include them in any kind of structured framework. In [21]-[23], measures proportional to the spectral energy in specified out-of-band regions are minimized using essentially the least squares method. [23] additionally introduces some feasibility constraints on the magnitude of the compensation coefficients. [1] suggests a combination of windowing and spectral compensation. In [25], an interesting time-domain criterion is proposed, which aims at improving the transmit signal's continuity at symbol borders. Passive methods use the same coefficients $\mathbf{S}$ once computed based on a given PSD mask and on second-order statistics of the data for many consecutive blocks and are updated only rarely, for example, to react to severely changing channel conditions [26][27].

Before focusing on the spectral compensation technique, two frequency-domain interpretations are presented that aim at supporting the intuitive understanding.

Interpretation 1: The first interpretation has in fact coined the term spectral compensation. The subcarriers in $\mathcal{S}_{\mathrm{i}}$ remain untouched and function like in an ordinary multicarrier system. Assume for a moment that we only transmit information over the subcarriers in $\mathcal{S}_{\mathrm{i}}$ with the optimal power values for a given PSD mask. Figure 12 illustrates the resulting transmit PSD for an example scenario (thin solid line). This uncompensated PSD exhibits considerable violations of the mask (bold solid line). Next, additionally the set $\mathcal{S}_{\mathrm{c}}$ of compensation subcarriers is modulated with data correlated with the information sent over the subcarriers in $\mathcal{S}_{\mathrm{i}}$, such that spectral violations are compensated for (dashed-dotted line). Since the data transmitted over the subcarriers is correlated, the spectral components of individual subcarriers may interfere in a constructive or destructive way and thus remove the mask violations and yield a better exploitation of the spectrum. In mathematical terms, data characterized by the covariance matrix $\boldsymbol{S}\boldsymbol{S}^{\mathrm{H}}$ modulates the N basis functions which are complex exponentials defined by the columns of $\boldsymbol{Z}_{\mathrm{add}}\,\boldsymbol{W}^{\mathrm{H}}$.

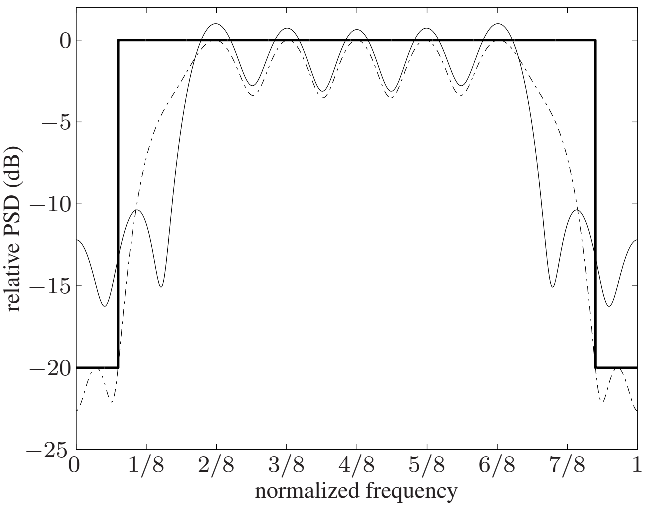


Figure 12: Interpretation 1: The uncompensated PSD (solid line), which is the sum of the PSDs of the subcarriers in $\mathcal{S}_\mathrm{i}$, exhibits violations of the mask (bold-solid line). Proper modulation of the subcarriers in $\mathcal{S}_\mathrm{c}$ yields the compensated PSD (dashed-dotted line). Example for $N = 8$; $L_\mathrm{cp} = 2$; $\mathcal{S}_\mathrm{c} = \{1, 2, 8\}$.

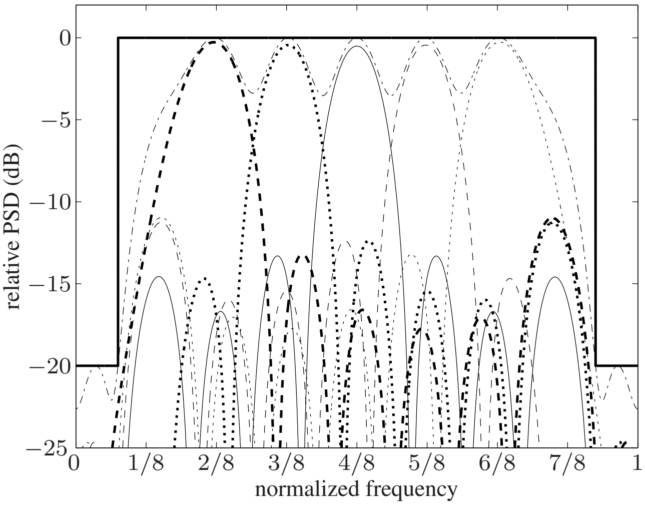


Figure 13: Interpretation 2: The columns No. $i, i \in \mathcal{S}_\mathrm{i}$ of $\boldsymbol{Z}_\mathrm{add}\, \boldsymbol{W}^\mathrm{H} \boldsymbol{S}$ are the $|\mathcal{S}_\mathrm{i}|$ transmit basis functions. Their individual PSDs ($i = 3$: bold-dashed line; $i = 4$: bold-dotted line; $i = 5$: solid line; $i = 6$: dashed line; $i = 7$: dotted line) sum up to the aggregate PSD (dashed-dotted line), which obeys the mask (bold-solid line). Example for $N = 8$; $L_\mathrm{cp} = 2$; $\mathcal{S}_\mathrm{c} = \{1, 2, 8\}$.

Interpretation 2: The second interpretation is not based on the concept of subcarriers but uses a set of $|\mathcal{S}_\mathrm{i}|$ basis functions, which are linear combinations of complex exponentials, specified by the kth columns of $\boldsymbol{Z}_\mathrm{add}\, \boldsymbol{W}^\mathrm{H} \boldsymbol{S}$, where $k \in \mathcal{S}_\mathrm{i}$. This set of basis functions is modulated with uncorrelated data such that the individual spectral components can be added. Figure 13 illustrates the individual PSDs corresponding to the information sent via the individual basis functions and the corresponding aggregate PSD (dashed-dotted line). The PSDs of the individual basis functions are neither of identical shape nor symmetric with respect to their center of mass. The particular shape of an individual basis function is the result of the optimization described in the following.

Hereinafter, the focus is on passive spectral compensation, which is a less complex technique than active compensation since the weights are determined only once in a while. The parameters of the optimal spectral-compensation transmitter—optimal in the sense of maximizing (5)—are given by

$$\{\mathcal{S}_\mathrm{c}^{(\mathrm{opt})}, \boldsymbol{S}^{(\mathrm{opt})}\} = \begin{array}{c} \arg\max\limits_{\mathcal{S}_\mathrm{c}, \boldsymbol{S}} R \\ \text{subject to } \ \boldsymbol{q} \leq \boldsymbol{m} \end{array}, \tag{19}$$

which is a non-convex problem. In general, (19) can be split into two subproblems: Finding the spectral compensation matrix $\boldsymbol{S}$ for a given subcarrier-set split and finding the subcarrier-set split itself. The latter is a problem general to all techniques using subcarriers for some purpose other than straightforward information transmission (including active spectral compensation and peak-to-average power ratio reduction).

In [26], a problem formulation using the data's autocorrelation matrix $\boldsymbol{S}\boldsymbol{S}^\mathrm{H}$ instead of $\boldsymbol{S}$ is proposed, which leads to a tractable solution for a given set $\mathcal{S}_\mathrm{c}$ via solving a semidefinite program. Furthermore, a heuristics to find the set $\mathcal{S}_\mathrm{c}$ of compensation subcarriers is suggested. These results are applied for the comparison with other techniques presented in the next section.

5 Comparison

This section presents a brief comparison of the techniques discussed in this chapter. The case study assumes a multicarrier system with $N = 64$ subcarriers, no CS ($L_\mathrm{cs} = 0$), and different cyclic prefix lengths. In order to allow for a straightforward interpretation of the results, an additive white Gaussian noise channel (*i.e.*, the channel has no memory and its transfer function is constant) is chosen. The ratio between the receive signal PSD and the noise PSD is thus constant. A moderate signal-to-noise power ratio of $SNR = 10\,\mathrm{dB}$ is assumed. As PSD limit $\boldsymbol{m}$, the exemplary mask shown in Figures 12 and 13 is employed (bold-solid line). The convex optimization problems are solved numerically, using dedicated software described in [29]-[32].

A simple upper bound for the achievable spectral efficiency R/N is given by

$$R/N \leq \frac{1}{N + L_\mathrm{cp}} \sum_k \log_2(1 + SNR\, \boldsymbol{m}[kQ/N]), \tag{20}$$

which assumes "sidelobe-free" basis functions. Consequently, there is no out-of-band emission and all subcarriers can be loaded with power values proportional to the PSD mask.

Figure 14 depicts the spectral efficiency R/N in bit per second per Hertz of the different techniques versus L_cp. With increasing L_cp, the oscillations of the in-band PSD become more pronounced and cause a decay in spectral efficiency R/N. The SNR-penalty of intrasymbol windowing caused by the receive processing (16) is severe and renders intrasymbol windowing inferior compared to the other techniques. Nyquist windowing has an advantage over other schemes for large L_cp since longer zero-padded portions of the transmit sequence allow higher power values in a PSD-constrained channel. Spectral compensation outperforms all other methods for small L_cp.

In general, it should be noted that the different in performance of various methods is more pronounced for a low number of subcarriers. As the number of subcarriers increases while the sampling frequency of the system remains constant, both mainlobe-width and sidelobe-widths of the basis functions' spectra decrease which improves the inherent spectral confinement of the basis functions.

Table 1 summarizes the approximate[2] run-time complexity of different methods in terms of complex-valued multiply-and-add operations per multicarrier symbol. The matrix multiplication required at the receiver to restore orthogonality, as described by (16), renders intrasymbol windowing most complex among all methods. Furthermore, the complexity scales with the number of subcarriers N. Spectral compensation allows a tradeoff between performance and complexity—both $\mathcal{S}_\mathrm{c}$ and $\mathcal{S}_\mathrm{r}(n), n \in \mathcal{S}_\mathrm{c}$ can be reduced at the cost of performance. The complexity does thus not scale with N but rather with the number of sharp edges in the PSD mask since each edge requires a couple of compensation subcarriers in order to achieve a shaping effect. In general, spectral compensation is less complex than intrasymbol windowing

[2] For simplicity, no explicit distinction between additions and multiplications is made. Plain additions are thus sometimes counted as multiply-and-add operations. Furthermore, certain parameters and parameter combinations allow a reduction of the nominal complexity stated in Table 1.

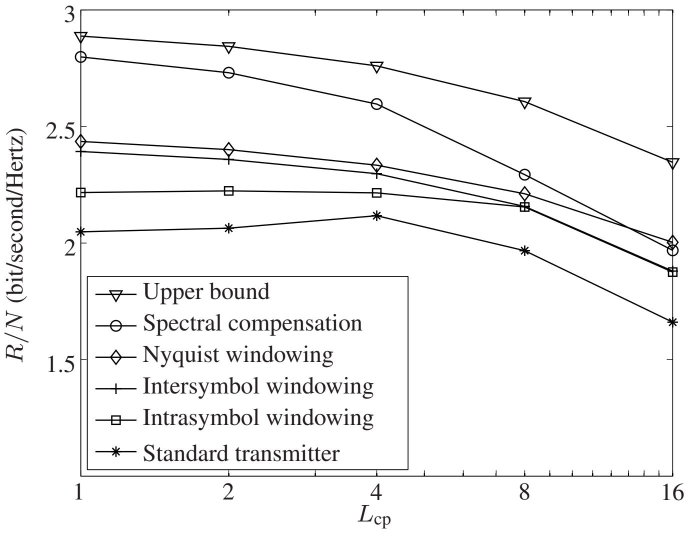


Figure 14: Case study: spectral efficiency R/N versus CP-length L_{cp} of different egress-reduction techniques for the PSD mask shown in Figure 12 and a frequency-flat channel. Optimal power loading according to (11) (star), maximum mainlobe-energy intrasymbol windowing [16] (square), linear-slope intersymbol windowing (plus), linear-slope Nyquist windowing [18] (diamond), spectral compensation [26] (circle), and upper bound given by (20) (triangle). All windows have optimal length and are applied together with optimal power loading (optimal in the sense of maximizing R/N). ($N = 64$, $SNR = 10\,\mathrm{dB}$).

Table 1: Total run-time complexity (of transmitter and receiver processing) of different methods in terms of complex-valued multiply-and-add operations per multicarrier symbol.

method	complex multiply-and-add operations per symbol	
	general	case study, $L_{\mathrm{cp}} = 16$
Spectral compensation	$\sum_{n \in \mathcal{S}_{\mathrm{c}}} \lvert\mathcal{S}_{\mathrm{r}}(n)\rvert$	212
Nyquist windowing	$6L_{\mathrm{nyq}} + L_{\mathrm{cp}}$	34
Intersymbol windowing	$2L_{\mathrm{w}}$	4
Intrasymbol windowing	$2N + N\log_2 N$	512

but more costly than the other two windowing techniques. The complexity of Nyquist windowing scales not only with the window-function parameter L_{nyq} but also with L_{cp}. Note that Nyquist windowing requires dedicated processing at the receiver ($2L_{\mathrm{w}} + L_{\mathrm{cp}}$ additions). Intersymbol windowing is clearly the cheapest technique and its complexity scales exclusively with the window length L_{w}.

6 Conclusion and open issues

In any setup where bandwidth is costly, high spectral efficiency is desirable. Most practical OFDMA systems are FFT-based and use rectangular pulse shapes, which emphasizes the tradeoff between transmit power and out-of-band emissions on subcarriers close to band edges. The chapter has tackled the tradeoff from an optimization perspective with the PSD as constraint and the throughput as objective function—an approach that allows a fair comparison of techniques and can also serve as a basis for system design.

In the following, some open issues in the context of spectrum efficiency are summarized.

- The framework presented in this chapter is based on maximizing the throughput under a PSD constraint while preserving orthogonality in time-dispersive channels. It may be worthwhile to explicitly include the effect of spectral leakage due to frequency offsets in the optimization process in order to enhance the robustness against frequency dispersion.

- A rather general problem that has also applications outside the scope of spectrum efficiency is finding “good” subcarrier sets $\mathcal{S}_\mathrm{i}$ and $\mathcal{S}_\mathrm{c}$. If the objective function does not introduce structure, the problem requires testing all possible splits and is thus NP hard. Finding the optimal split may not be of vital importance for practical applications in communication, however, low-complexity solutions to quickly finding an acceptable split are clearly of interest. Good ideas and deeper insights may also be used for other applications, like the design of tone-reservation schemes or precoders aiming at mitigating intercarrier interference.
- Spectral compensation appears to be a promising technique. A conclusive comparison of active and passive methods including both performance and complexity is yet to be done.

References

[1] U. Berthold, F. Jondral, S. Brandes, and M. Schnell, “OFDM-based overlay systems: A promising approach for enhancing spectral efficiency [topics in radio communications],” *IEEE Communications Magazine*, vol. 45, no. 12, pp. 52–58, December 2007.

[2] W. Yu, D. Toumpakaris, J. Cioffi, D. Gardan, and F. Gauthier, “Performance of asymmetric digital subscriber lines in an impulse noise environment,” *IEEE Transactions on Communications*, vol. 51, no. 10, pp. 1653–1657, Oct. 2003.

[3] A. Vahlin and N. Holte, “Optimal finite duration pulses for OFDM,” *IEEE Transactions on Communications*, vol. 44, no. 1, pp. 10–14, Jan. 1996.

[4] H. Bölcskei, P. Duhamel, and R. Hleiss, “Design of pulse shaping OFDM/OQAM systems for high data-rate transmission over wireless channels,” in *Proc. IEEE Intl. Conference on Communications (ICC ’99)*, vol. 1, Vancouver, Canada, June 1999, pp. 559–564.

[5] W. Kozek and A. Molisch, “Nonorthogonal pulseshapes for multicarrier communications in doubly dispersive channels,” *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 8, pp. 1579–1589, Oct. 1998.

[6] R. Chang, “Synthesis of band-limited orthogonal signals for multi-channel data transmission,” *The Bell System Technical Journal*, vol. 45, pp. 1775–1796, 1966.

[7] B. Maham and A. Hjorungnes, “ICI reduction in OFDM by using maximally flat windowing,” *Proc. IEEE International Conference on Signal Processing and Communications (ICSPC’07)*, pp. 1039–1042, Nov. 2007.

[8] P. Tan and N. Beaulieu, “Reduced ICI in OFDM systems using the ”better than” raised-cosine pulse,” *IEEE Communications Letters*, vol. 8, no. 3, pp. 135–137, March 2004.

[9] V. P. Sathe and P. P. Vaidyanathan, “Effects of multirate systems on the statistical properties of random signals,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 131–146, Jan. 1993.

[10] W. A. Gardner, “Exploitation of spectral redundancy in cyclostationary signals,” *IEEE Signal Processing Magazine*, vol. 8, pp. 14–36, Apr. 1991.

[11] T. Magesacher, P. Ödling, and P. O. Börjesson, “Optimal intersymbol transmit windowing for multicarrier modulation,” in *Proc. Nordic Signal Processing Symp. NORSIG 2006*, Reykjavik, Iceland, June 2006.

[12] J. M. Cioffi, *Advanced Digital Communication.* class reader EE379C, Stanford University, 2005, class web page `http://www.stanford.edu/class/ee379c/`.

[13] ETSI TM6, “Transmission and multiplexing (TM); access transmission systems on metallic access cables; Very high speed Digital Subscriber Line (VDSL); Part 2: Transceiver specification,” *TS 101 270-2, Version 1.1.5*, Dec. 2000.

[14] G. Cuypers, K. Vanbleu, G. Ysebaert, and M. Moonen, “Intra-symbol windowing for egress reduction in DMT transmitters,” *EURASIP Journal on Applied Signal Processing*, no. 1, pp. 87–87, 2006.

[15] T. Magesacher, P. Ödling, and P. O. Börjesson, “Optimal intersymbol transmit windowing for multicarrier modulation,” in *Proc. 7th Nordic Signal Processing Symposium (NORSIG ’06)*, June 2006, pp. 70–73.

[16] Y.-P. Lin and S.-M. Phoong, “Window designs for DFT-based multicarrier systems,” *IEEE Transactions on Signal Processing*, vol. 53, pp. 1015–1024, Mar. 2005.

[17] T. Magesacher, “Optimal intra-symbol transmit windowing for multicarrier modulation,” in *Proc. Intl. Symp. on Communications, Control and Signal Processing (ISCCSP ’06)*, Marrakech, Morocco, Mar. 2006.

[18] M. Sebeck and G. Bumiller, “Effective configurable suppression of narrow frequency bands in multicarrier modulation transmission,” in *Proc. IEEE International Symposium on Power Line Communications and Its Applications*, 2006, pp. 128–133.

[19] J. Bingham and M. Mallory, “RFI egress suppression for SDMT,” *ANSI Contribution T1E1.4/96-085*, 1996.

[20] J. Bingham, “RFI suppression in multicarrier transmission systems,” in *Proc. Global Telecommunications Conference (GLOBECOM’96)*, vol. 2, Nov 1996, pp. 1026–1030.

[21] R. Baldemair, “Suppression of narrow frequency bands in multicarrier transmission systems,” in *Proc. European Signal Processing Conference (EUSIPCO ’00)*, Tampere, Finland, Sept. 2000, pp. 553–556.

[22] H. Yamaguchi, “Active interference cancellation technique for MB-OFDM cognitive radio,” in *Proc. 34th European Microwave Conference*, vol. 2, Oct. 2004, pp. 1105–1108.

[23] I. Cosovic, S. Brandes, and M. Schnell, “Subcarrier weighting: a method for sidelobe suppression in OFDM systems,” *IEEE Communications Letters*, vol. 10, no. 6, pp. 444–446, June 2006.

[24] S. Brandes, I. Cosovic, and M. Schnell, “Reduction of out-of-band radiation in OFDM systems by insertion of cancellation carriers,” *IEEE Communications Letters*, vol. 10, no. 6, pp. 420–422, June 2006.

[25] J.-J. van de Beek and F. Berggren, “Out-of-band power suppression in OFDM,” *IEEE Communications Letters*, vol. 12, no. 9, pp. 609–611, September 2008.

[26] T. Magesacher, P. Ödling, and P. O. Börjesson, “Optimal intra-symbol spectral compensation for multicarrier modulation,” in *Proc. Intl. Zurich Seminar on Broadband Communications (IZS ’06)*, Zurich, Switzerland, Feb. 2006, pp. 138–141.

[27] T. Magesacher, “Spectral compensation for multicarrier communication,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3366–3379, July 2007.

[28] J. Tellado and J. Cioffi, “Efficient algorithms for reducing PAR in multicarrier systems,” in *Proc. IEEE International Symposium on Information Theory*, Aug. 1998, p. 191.

[29] B. Borchers, “CSDP, a C library for semidefinite programming,” *Optimization Methods & Software*, vol. 11-2, pp. 613–623, 1999, available from `http://infohost.nmt.edu/~borchers/csdp.html`.

[30] J. Löfberg, “YALMIP : A toolbox for modeling and optimization in MATLAB,” in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004, available from `http://control.ee.ethz.ch/~joloef/yalmip.php`.

[31] J. F. Sturm, “Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones,” in *Optimization Methods and Software*, vol. 11-12, 1999, pp. 625–653, available from http://sedumi.mcmaster.ca.

[32] Mathworks, “Matlab optimization toolbox (online documentation),” 2005, http://www.mathworks.com.

Part IX

MIMO OFDM and Multiuser OFDM

This chapter deals with a few aspects of OFDM applied in MIMO settings and multiuser scenarios.

MIMO glossary

- Spatial multiplexing: split of the transmit bitstream into several substreams that are transmitted in parallel through the transmit antennae/ports yielding a throughput gain compared to SISO channel. (Note: there is a tradeoff between spatial multiplexing gain and diversity gain.)
- Diversity gain: improvement achieved via combining independently fading paths. (Note: there is a tradeoff between diversity gain and spatial multiplexing gain.)
- Diversity order: number of independently fading paths.
- Array gain: average improvement (e.g., in terms of SNR) achieved through transmitting/receiving multiple copies of the transmit bitstream via coherent combining (SIMO), precoding (MISO), or both (MIMO).
- Interference reduction: reduction of co-channel interference (caused by frequency reuse) via exploiting spatial signatures of signal and co-channel signal (interference).
- Delay diversity: improvement achieved via combining independently fading paths obtained through repeating symbols in time (repeat code).

1 Motivation: capacity of MIMO channels

A Gaussian multiple-input multiple-output channel is described by the matrix model

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{s} \in \mathbb{C}^S$ is the channel input, $\boldsymbol{r} \in \mathbb{C}^R$ is the channel output, $\boldsymbol{H} \in \mathbb{C}^{R \times S}$ is the channel matrix (S input ports, R output ports), and $\boldsymbol{n} \in \mathcal{CN}^R$ is a zero-mean circularly symmetric Gaussian random vector modelling the noise. The capacity of a MIMO channel depends on $\boldsymbol{H}$ (deterministic, ergodic, non-ergodic) and on the channel state information (CSI) available at transmitter and receiver.

Considering the structure of the capacity formula for Gaussian channels ($\log(1 + SNR)$), it is not surprising that a factor outside the logarithm (obtained through parallel paths in MIMO channels) achieves a higher leverage than a factor inside the logarithm (obtained through higher SNR in SISO channels).

In order to demonstrate the advantage of MIMO over SISO in terms of capacity, we focus on the simplest case: deterministic $\boldsymbol{H} = \boldsymbol{I}$, where $R = S$. The capacity of the scalar channel ($R = S = 1$) is $\log_2(1 + SNR)$ bits per channel-use per two dimensions. The MIMO channel $\boldsymbol{H} = \boldsymbol{I}$ is diagonal. Since there is no spatial interference, we have a set of $R = S$ independent parallel subchannels. The resulting channel capacity is $R \log_2(1 + SNR/R)$. Figure 1 depicts the capacity versus $R = S$. The factor R outside

Chapter written by T. Magesacher.

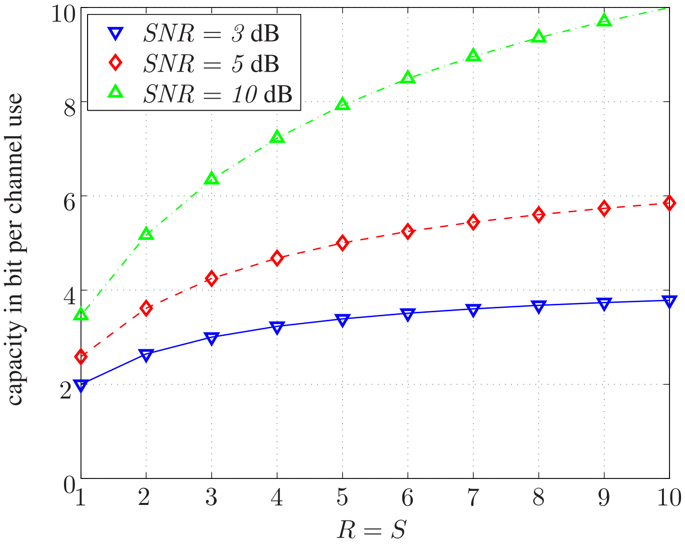


Figure 1: MIMO channel capacity $R\log_2(1 + SNR/R)$ versus R.

the logarithm yields a considerable benefit. Note that the *total transmit power* is the *same* for the MIMO channel as for the scalar channel. Another way to view the benefit is the following: Consider a MIMO channel with $\boldsymbol{H} = \boldsymbol{I}$ and transmit power P. The transmit power required to achieve the same capacity over a scalar channel $\boldsymbol{H} = 1$ is proportional to P^S (exponential in $R = S$). Similar results for fading channels (ergodic and non-ergodic) motivate the interest in MIMO communication.

2 MIMO OFDM

OFDM used for a SISO channel partitions the given bandwidth into N subbands that are narrow enough to justify the assumption that these subbands are frequency-flat. As a result, we obtain N parallel *independent* subchannels:

$$\boldsymbol{y} = \boldsymbol{\Lambda}\boldsymbol{x}.$$

OFDM applied in a *synchronized* manner both at the transmitter and at the receiver to a MIMO channel partitions the given bandwidth on each antenna/port in frequency domain. We model a frequency-selective MIMO channel with S inputs and R outputs in time-domain using $M+1$ matrices $\boldsymbol{H}[m] \in \mathbb{C}^{R\times S}, m = 0, \dots, M$, where the element in row No. k and column No. ℓ is the mth coefficient of the impulse response $h_{k,\ell}[m]$ of the path from antenna (or port) ℓ to antenna (port) k. The Fourier transforms

$$\boldsymbol{\mathcal{H}}[k] = \sum_m \boldsymbol{H}[m]\mathrm{e}^{-j2\pi mk/N}, \qquad k = 0, \dots, N-1$$

yields N frequency-flat MIMO channels $\boldsymbol{\mathcal{H}}[k] \in \mathbb{C}^{R\times S}, k = 0, \dots, N-1$. As a result, in the ideal case there is neither ISI nor ICI, *i.e.*, there is no interference among subchannels. However, in each subchannel, spatial interference occurs:

$$\begin{bmatrix} y_k^{(1)} \\ \vdots \\ y_k^{(R)} \end{bmatrix} = \boldsymbol{\mathcal{H}}[k] \begin{bmatrix} x_k^{(1)} \\ \vdots \\ x_k^{(S)} \end{bmatrix}, \qquad k = 0, \dots, N-1, \tag{2}$$

where $x_k^{(s)}$ denotes the sth antenna's transmit symbol and $y_k^{(r)}$ denotes the rth antenna's receive symbol on subchannel No. k. Recall that (2) holds iff the multicarrier symbols on all transmit antennae/ports and on all receive antennae/ports are aligned in time (perfect synchronisation). A longer cyclic-prefix or guard-interval can help to mitigate the impact of the latency spread among antennae while increasing the inherent throughput loss (cf. single-user multicarrier synchronization).

Consequently, SISO space-time techniques can be applied on a subchannel level. To summarize, OFDM turns the rather complicated frequency-selective MIMO channel $H_{k,\ell}[m]$ into N frequency-flat MIMO channels via decoupling of subbands (frequency domain). If CSI is available at the transmitter, waterfilling can be applied both in the frequency domain (over tones) and in the spatial domain (over antennae/ports).

3 Space-time techniques

Virtually all space-time techniques developed for MIMO channels can be applied to MIMO OFDM exploiting the frequency domain. Examples are:

- Alamouti over subchannels (instead of time): Replacing the time index by the frequency index, the Alamouti scheme (cf. Example 58) can be applied over tones to yield spatial diversity gain in a MIMO OFDM scheme. The assumption is that the coherence bandwidth is large enough so that neighboring subchannels exhibit identical channel coefficients.

- Spatial multiplexing: NS scalar data symbols can be transmitted over one OFDM symbol using standard MIMO techniques (optimal: ML vector receiver. suboptimal: MMSE, ZF receiver), which require $R > S$, for each subchannel.

- Exploiting frequency diversity: Spreading data suitably across frequency (and hereby using a tone allocation that ensures a subchannel spacing exceeding the coherence bandwidth), yields additional diversity compared to exploiting spatial diversity only.

4 OFDMA

When the inherent orthogonality of OFDM subcarriers is exploited to separate users, we speak of OFDM Access (OFDMA). Different allocation variants include

- fixed in frequency, varying time (time hopping)

- varying in frequency and time (frequency hopping)

Forward error correction coding over hops yields diversity gain. Furthermore, hopping results in an averaging effect in the presence of interference (for example from an adjacent cell in a cell-based system) since the paths gains between the interference source and different time/frequency slots vary as a result of fading. Compared to a static allocation without hops, the performance with hopping is determined by the average interference rather than the worst-case interference. Note that the orthogonality exploited through OFDMA holds only if all transmitters and all receivers of all users are synchronized.

5 Multicarrier CDMA

Multicarrier code division multiple access (MC-CDMA) is a combination of the CDMA multiplexing technique and the OFDM modulation technique. Instead of spreading data in time like in CDMA, MC-CDMA spreads data across subcarriers. A general model of the kth user's transmit block is

$$\boldsymbol{t}_k = \boldsymbol{W}^{\mathrm{H}} \boldsymbol{S}_k \boldsymbol{x}_k$$

where $\boldsymbol{S}_k \in \mathbb{C}^{N \times P}$ spreads the P data symbols of user k across the N subcarriers. Note:

- $P = N$ and $\boldsymbol{S}_k = \boldsymbol{I}$ corresponds to standard OFDM

- $P = 1$ and $\boldsymbol{S}_k = \boldsymbol{s}_k$ spreads the kth user's data across all N subcarriers using the signature $\boldsymbol{s}_k$

The kth user's receiver signal is

$$\boldsymbol{r}_k = \sum_{m=1}^{K} \boldsymbol{H}_m \boldsymbol{S}_m \boldsymbol{x}_m$$

Compared to OFDMA, MC-CDMA offers frequency diversity for each user. Compared to DS-CDMA, MC-CDMA offers the potential to handle frequency selectivity more efficiently.

Given a bandwidth B, the choice of N and L for a MC-CDMA system involves the same tradeoff as for an OFDM system. A larger N implies a smaller subcarrier spacing and thus lower robustness to Doppler spread. On the other hand, a smaller subcarrier spacing decreases the frequency selectivity within a subband (which is frequency flat in the ideal case). Furthermore, a larger N implies longer multicarrier symbols and thus a lower datarate loss caused by a cyclic prefix not shorter than the channel delay spread the systems is expected to handle. Consequently, for a given frequency selectivity and a given time selectivity, there is an optimal choice for N and L in the sense of minimizing the resulting bit error rate.

If the users are not synchronized (i.e., the symbols received from different users are not aligned in time), the optimal multiuser detector (MUD) is a bank of single-user matched filter detectors followed by joint maximum-likelihood sequence detection of all users.

If the users are synchronized, it is sufficient to observe one symbol. The equivalent discrete-time model is given by

$$\boldsymbol{r} = \boldsymbol{R}\mathrm{diag}\{h_1 \dots h_K\}\boldsymbol{b} + \boldsymbol{n}, \tag{3}$$

where $\boldsymbol{b}$ denotes the transmitted symbols and $\boldsymbol{R}$ contains the cross-correlations between the spreading codes. The optimal MUD is again the ML detector. A number of suboptimal MUD have been proposed. The most important linear MUDs are

- Decorrelator
- MMSE detector

The most important nonlinear MUDs are

- Multistage detector
- DFE
- Interference canceller

6 Duplexing in DMT: Zipper

The term *duplexing* denotes simultaneous transmission between two users, which essentially requires the separation of the two signals in time, frequency or space. Multicarrier transmission offers large flexibility for duplexing through a method referred to as *zipper*. The principle idea is to divide the N subcarriers into two distinct sets, one for each direction. This allows an arbitrary division of the total capacity among the two directions. The possibility of interleaving the carriers in alternating fashion gave the methods its name. The application where zipper is used is wireline communication, in particular VDSL.

A necessary requirement for this principle to work in practice, is that the subcarriers of a receive symbol are not only mutually orthogonal (i.e., the cyclic prefix L is not shorter than the dispersion M of the channel) but that they are also orthogonal to the subcarriers of interfering symbols (crosstalk) or echo symbols.

Figure 2 illustrates the symbol timing. All transmitters and all receivers need to be synchronised, i.e., they must operate using a common time basis. The transmit symbols of all the transmitters involved are sent at the same time (beginning at relative time instant 0 in Figure 2). Thus, near-end interfering symbols and echo symbols arrive at the same time, in general D_n samples later. All far-end signals arrive at the same time, in general D_f samples later. Processing the received symbols at relative time instant $D_\mathrm{f} + L$ together with a cyclic suffix of D_f samples ensures orthogonality.

To summarise, the zipper method requires two necessary conditions:

- addition of cyclic suffix of length D_f, where D_f is the delay of the channel
- synchronisation of all transmitters and receivers

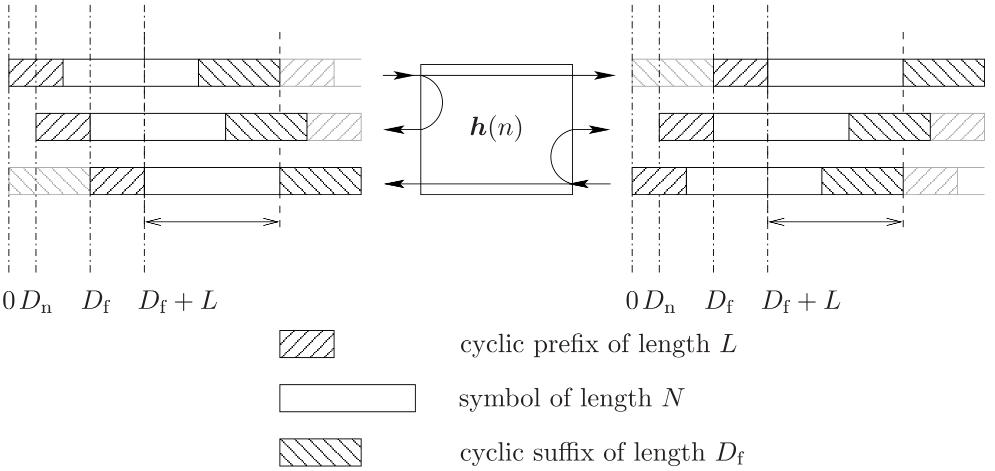


Figure 2: The two prerequisites for zipper: 1. every symbol is extended by a cyclic suffix of length D_f samples; 2. all transmitters and all receivers are synchronised, each transmitter sends its symbol at relative time instant 0 (beginning with the cyclic prefix), each receiver processes the corresponding symbol D_f samples later.

Problems

Please note: Both the level of difficulty and the degree of detail regarding the assignment description vary strongly without warning. Some examples are intended for illustration and deepening of key concepts (and are typically presented in class) while others are for training.

Example 1: Spectra of $\underline{r}_k(n)$ and $\underline{s}_k(n)$ (a) Determine the discrete-time Fourier transform (DTFT) of the complex-valued receive signal components $\underline{r}_k(n) = r_k^{(\mathrm{i})}(n) + jr_k^{(\mathrm{q})}(n)$, where

$$\begin{aligned} r_k^{(\mathrm{i})}(n) &= \frac{1}{\sqrt{N}}\cos(2\pi\frac{k}{N}n), & k &= 0,\ldots,\left\lfloor\frac{N}{2}\right\rfloor, \\ r_k^{(\mathrm{q})}(n) &= -\frac{1}{\sqrt{N}}\sin(2\pi\frac{k}{N}n), & k &= 1,\ldots,\left\lfloor\frac{N-1}{2}\right\rfloor, \end{aligned} \qquad n = 0,\ldots,N-1.$$

(b) Determine the discrete-time Fourier transform (DTFT) of the complex-valued transmit signal components $\underline{s}_k(n) = s_k^{(\mathrm{i})}(n) + js_k^{(\mathrm{q})}(n)$, where

$$\begin{aligned} s_k^{(\mathrm{i})}(n) &= \frac{1}{\sqrt{N}}\cos(2\pi\frac{k}{N}n), & k &= 0,\ldots,\left\lfloor\frac{N}{2}\right\rfloor, \\ s_k^{(\mathrm{q})}(n) &= -\frac{1}{\sqrt{N}}\sin(2\pi\frac{k}{N}n), & k &= 1,\ldots,\left\lfloor\frac{N-1}{2}\right\rfloor, \end{aligned} \qquad n = -L,\ldots,N-1.$$

- Cf. `aliasing.m`, `rxspectra.m`, `txspectra.m`

Example 2: Orthogonality of receive signal components
Prove that the receive signal components

$$\begin{aligned} r_k^{(\mathrm{i})}(n) &= \frac{1}{\sqrt{N}}\cos(2\pi\frac{k}{N}n), & k &= 0,\ldots,\left\lfloor\frac{N}{2}\right\rfloor, \\ r_k^{(\mathrm{q})}(n) &= -\frac{1}{\sqrt{N}}\sin(2\pi\frac{k}{N}n), & k &= 1,\ldots,\left\lfloor\frac{N-1}{2}\right\rfloor, \end{aligned} \qquad n = 0,\ldots,N-1,$$

are orthogonal.

Example 3: PSD, autocorrelation, spectra
Consider a colored stationary process. Show that the power spectral density (PSD), which is the Fourier transform of the autocorrelation sequence, is equal to the squared magnitude of the coloring filter's Fourier transform.

Example 4: MATLAB warm-up
This is a warm-up exercise dealing with some basic signal processing operations. Open an editor, create a file (with extension `.m`) and write your code into that file (begin with `close all; clear all` for a clean start). Consider the two sequences $\boldsymbol{s} = \begin{bmatrix} s(0) & s(1) & s(2) & s(3) \end{bmatrix} = \begin{bmatrix} 1 & 2 & -1 & 2 \end{bmatrix}$ and $\boldsymbol{h} = \begin{bmatrix} h(0) & h(1) & h(2) \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$.

(a) Compute the result $r_{\mathrm{lin}}(n)$ of the linear convolution of $s(n)$ and $h(n)$. (Linear convolution: $r_{\mathrm{lin}}(n) = s(n) * h(n) = \sum_k s(k)h(n-k), n = 0,\ldots,\mathrm{length}(s(n)) + \mathrm{length}(h(n)) - 2$).

(b) Compute the result $r_{\mathrm{cir}}(n)$ of the 4-point circular convolution of $s(n)$ and $h(n)$. (N-point circular convolution[1]: $r_{\mathrm{cir}}(n) = s(n) \circledast_N h(n) = \sum_k s(k)h((n-k) \bmod N), n = 0,\ldots,N-1$).

[1] Modulo operator: $n \bmod N = n - N\lfloor\frac{n}{N}\rfloor$ for $N > 0$. Use `mod` in MATLAB

(c) Compute the 4-point discrete Fourier transform (DFT) $S[k]$ of $s(n)$.
(N-point DFT: $S[k] = \sum_n s(n)e^{-j2\pi kn/N}, k = 0, \ldots, N-1$).

(d) Compute the 4-point DFT $H[k]$ of $h(n)$.

(e) Compute the 4-point inverse DFT (IDFT) $r_1(n)$ of $R_1[k] = H[k]S[k]$.
(N-point IDFT: $s(n) = \frac{1}{N}\sum_k S[k]e^{j2\pi kn/N}, n = 0, \ldots, N-1$).

Example 5: Correlation, Matlab
Correlation is an essential operation that you will encounter often in communications. The following examples should help you convince yourself, that correlation is a measure for similarity. Consider the sequences $s_1(n) = \frac{1}{\sqrt{16}}e^{j2\pi\frac{1}{16}n}, n = 0, \ldots, 15$ and $s_2(n) = \frac{1}{\sqrt{16}}e^{j2\pi\frac{2}{16}n}, n = 0, \ldots, 15$.

(a) Compute the correlation $\langle s_1(n), s_2(n)\rangle$ between the signals $s_1(n)$ and $s_2(n)$.
(Correlation: $\langle s_1(n), s_2(n)\rangle = \sum_n s_1(n)s_2^*(n)$).

(b) Compute the correlation $\langle s_1(n), -s_1(n)\rangle$ between the signals $s_1(n)$ and $-s_1(n)$.

(c) Compute the correlation $\langle s_3(n), s_3(n)\rangle$, *i.e.*, the autocorrelation of $s_3(n) = 3s_1(n), n = 0, \ldots, 15$.

(d) Compute the energy E_{s_3} of the sequence $s_3(n)$ (energy: $E_{s_3} = \sum_n |s_3(n)|^2$).

Example 6: OFDM system parameters
Compute the symbol rate τ, quantify the ISI, and provide a recommendation regarding singlecarrier versus multicarrier transmission:

- System 1: max. delay spread $T_{\text{multi}} = 8\,\mu$s, data rate $R = 100\,$kbit/s, uncoded QPSK
- System 2: max. delay spread $T_{\text{multi}} = 8\,\mu$s, data rate $R = 10\,$Mbit/s, uncoded BPSK

Example 7: OFDM system parameters, channel delay spread, coherence time
The migration from analogue to digital TV broadcasting has been completed recently. By November 2007, all analogue TV broadcasts are expected to be switched off. The physical layer of DVB-T (Digital Video Broadcast - Terrestrial) is based on OFDM (ETSI EN 300 744 standard).

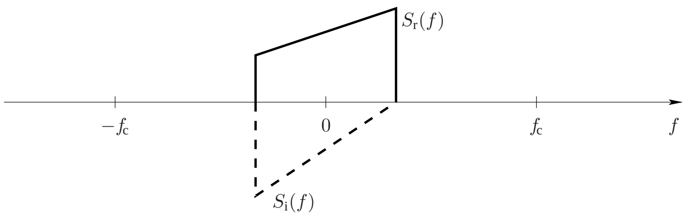
In the so-called 8k-mode, DVB-T uses 6817 consecutive carriers out of $N = 8192$ (FFT size). The lowest and the highest used carrier are 5705357 Hz apart. The TV program is broadcast in the band 790–798 MHz (*i.e.*, the RF modulator runs at frequency $F_c = 794\,$MHz).

(a) Determine the carrier spacing Δf, the OFDM-symbol duration T_{MC} (excluding the cyclic prefix) and the sampling period T_{S} of the discrete-time baseband transmit multiplex.

(b) Assume a maximum delay spread of $50\,\mu s$. The length T_{CP} of the cyclic prefix for DVB-T can be $T_{\text{MC}}/4$, $T_{\text{MC}}/8$, $T_{\text{MC}}/16$, or $T_{\text{MC}}/32$. Choose T_{CP} such that the bit rate is maximised and there is neither ISI nor ICI. Compute the bit rate R if all carriers use 16-QAM and a channel code of rate 2/3.

(c) Assume that the system specification requires a coherence time (assume that the propagation velocity of the radio wave is $c = 3 \cdot 10^8\,$m/s) that is at least 20 times larger than the multicarrier symbol duration ($T_{\text{MC}} + T_{\text{CP}}$). What is the maximum velocity at which we can still watch TV?

Example 8: RF-modulation ($\sin/\cos$ modulator/demodulator), spectra
Consider the complex-valued signal $s(n), n = -\infty, \ldots, \infty$. Its DTFT $S(f) = S_{\text{r}}(f) + jS_{\text{i}}(f)$ is depicted below:



(a) Determine and sketch the DTFT $S_1(f)$ of $s_1(n) = s(n) \cdot e^{j2\pi f_c n}$.

(b) Determine and sketch the DTFT $S_2(f)$ of $s_2(n) = s(n) \cdot \cos(2\pi f_c n)$.

(c) Determine and sketch the DTFT $S_3(f)$ of $s_3(n) = s(n) \cdot \sin(2\pi f_c n)$.

(d) Remember that $s(n)$ is complex-valued: $s(n) = s^{(r)}(n) + js^{(i)}(n)$. The DTFTs of $s^{(r)}(n)$ and $s^{(i)}(n)$ are given by:

$$\begin{aligned} s^{(r)}(n) &\longleftrightarrow S^{(r)}(f) = S_r^{(r)}(f) + jS_i^{(r)}(f) \\ s^{(i)}(n) &\longleftrightarrow S^{(i)}(f) = S_r^{(i)}(f) + jS_i^{(i)}(f) \end{aligned}$$

Express the DTFT of $s(n)$ in terms of $S_r^{(r)}(f)$, $S_i^{(r)}(f)$, $S_r^{(i)}(f)$, $S_i^{(i)}(f)$. Express the DTFT $S_4(f)$ of $s_4(n) = s^*(n)$ in terms of $S_r^{(r)}(f)$, $S_i^{(r)}(f)$, $S_r^{(i)}(f)$, $S_i^{(i)}(f)$.

(e) Determine the DTFT $S_5(f)$ of $s_5(n) = \mathrm{Re}\left\{s(n)e^{j2\pi f_c n}\right\}$ (Hint: $\mathrm{Re}\{x\} = \frac{1}{2}(x + x^*)$).

Example 9: BER AWGN channel
Determine the bit error rate (BER) of uncoded equiprobable binary phase shift keying (BPSK) over the additive white Gaussian noise (AWGN) channel as a function of E_b/N_0 (E_b is the energy per bit, $N_0/2$ is the noise variance).

Example 10: Q-function
MATLAB provides the function $\texttt{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_{t=x}^{\infty} e^{-t^2} \mathrm{d}t$. Express $Q(x)$ as a function of $\texttt{erfc}(x)$ ($Q(x)$ is the "tail" probability $\Pr(X > x)$ of a zero-mean unit-variance normal random variable X).

Example 11: QAM constellation, BER
Consider equiprobable 4-PAM (pulse amplitude modulation) transmission over the AWGN channel with optimal detection. The data bits are mapped onto the modulation alphabet $\{\pm1, \pm3\}$ according to the so-called natural mapping:

$$\begin{aligned} 00 &\longrightarrow -3 \\ 01 &\longrightarrow -1 \\ 10 &\longrightarrow 1 \\ 11 &\longrightarrow 3 \end{aligned}$$

(a) Compute the resulting BER as a function of E_b/N_0.

(b) Plot the BER as a function of E_b/N_0 in dB using MATLAB (use the command `semilogy`) and compare it with the simulation result.

(c) Consider the following mapping (Gray mapping):

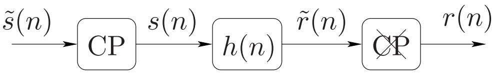
$$\begin{aligned} 00 &\longrightarrow -3 \\ 01 &\longrightarrow -1 \\ 11 &\longrightarrow 1 \\ 10 &\longrightarrow 3 \end{aligned}$$

Which of the two mappings do you expect to perform better at high E_b/N_0? Why?

(d) Consider a 16-QAM constellation that uses natural mapping in both dimensions. What is its BER?

Example 12: Correlation
Consider the sequences $r_k(n) = \frac{1}{\sqrt{N}} e^{j2\pi \frac{k}{N} n}$, $n = 0, \ldots, N-1$, $k = 0, \ldots, N-1$. Determine the auto/crosscorrelations $\langle r_k(n), r_m(n) \rangle$ of $r_k(n)$ and $r_m(n)$ for $k, m \in \{0, \ldots, N-1\}$. What do you conclude?

Example 13: Linear versus circular convolution
Consider the following system:



The first block in the chain (labeled CP) performs a cyclic extension of the transmit signal $\tilde{s}(n), n = 0, \ldots, N-1$, which yields

$$s(n) = \begin{cases} \tilde{s}(n), & n = 0, \ldots, N-1 \\ \tilde{s}(N+n), & n = -L, \ldots, -1 \end{cases}.$$

The channel with impulse response $h(n) \neq 0, n = 0, \ldots, M$, performs linear convolution resulting in $\tilde{r}(n) = s(n) * h(n)$. The last block takes in $N + L$ samples, discards the first L samples and delivers the remaining N samples at the output, i.e., $r(n) = \tilde{r}(n), n = 0, \ldots, N-1$.

(a) Determine the relation between $r(n)$ and $\tilde{s}(n)$.

(b) Determine the relation between $R[k]$ and $\tilde{S}[k]$, which are the N-point discrete Fourier transforms (DFTs) of $r(n)$ and $\tilde{s}(n)$, respectively.

Example 14: OFDM system design
Consider a HiperLAN2 system, which operates at $f = 5.2\,\text{GHz}$ with a bandwidth of $B = 20\,\text{MHz}$ and $N = 64$ carriers. The design specification requires pedestrian mobility: $v \leq 15\,\text{km/h}$. Assume a maximum delay spread of $T_{\text{multi}} = 250\,\text{ns}$.

(a) Determine the (multicarrier) symbol duration T_{sym}.

(b) Determine the coherence time T_{coh} (assume a propagation velocity $c = 3 \cdot 10^8\,\text{m/s}$).

(c) Compare T_{multi}, T_{coh}, and T_{sym}.

Example 15: Matrix representation, spectra
H, a concept engineer working on an OFDM-based air interface for a point-to-point radio system, realises that he has made a mistake: instead of a 64-point IFFT, he has implemented a 64-point FFT at the transmitter. The tape-out deadline has already passed—the integrated circuit with the modulator is in production. Luckily, it is a custom design, *i.e.*, the transmitter is not standardised. H has the chance to modify his receiver design—however, if he fails to make the system work, he will get fired. Help H by answering the following questions:

(a) Determine the receive basis functions and prove that the system allows ISI/ICI-free transmission over a time-dispersive channel. (Use the following notation; $\boldsymbol{W}$: normalised DFT matrix; $\boldsymbol{S}_{\text{H}}$: H's transmit matrix; $\boldsymbol{R}_{\text{H}}$: H's receive matrix)

The system is allowed to use subcarriers within $[2400, 2421]$ MHz of the ISM band. The baseband processing clock is 24 MHz and the mixer frequency (sin/cos modulator) is 2412 MHz.

(b) Determine the set $\mathcal{S}_{\text{inband}}$ of subcarriers that can be used ($\mathcal{S}_{\text{inband}} \subset \{-32, \ldots, 31\}$).

(c) Determine the set $\mathcal{S}'_{\text{inband}}$ of subcarriers that can be used with a "proper" transmitter employing an IFFT ($\mathcal{S}'_{\text{inband}} \subset \{-32, \ldots, 31\}$).

Eventually, poor H has to learn that his company has already acquired the "proper" receiver design (using an FFT) through the merge with another company.

(d) Is there a simple pre-processing that could be applied before the modulator to save H? In other words, can you find a matrix $\boldsymbol{P}_{\mathrm{H}}$ consisting only of zeros and ones such that $\boldsymbol{S}_{\mathrm{H}}\boldsymbol{P}_{\mathrm{H}}$ realises the "proper" transmitter? If yes, determine $\boldsymbol{P}_{\mathrm{H}}$

Example 16: Response of LTI channel (a) Determine the response of the discrete-time LTI channel $h(n)$ with dispersion M to $\cos(2\pi\frac{k}{N}n)$ and $-\sin(2\pi\frac{k}{N}n)$ as a function of

$$A_k = \sum_{m=0}^{M} h(m)\cos(2\pi\frac{k}{N}m) \quad\text{and}\quad B_k = -\sum_{m=0}^{M} h(m)\sin(2\pi\frac{k}{N}m).$$

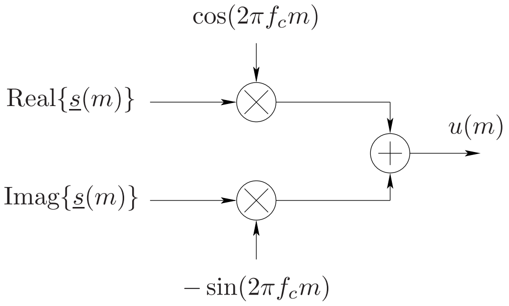
(b) Now we transmit $s(n) = \sum_k (x_k^{(\mathrm{i})} s_k^{(\mathrm{i})}(n) + x_k^{(\mathrm{q})} s_k^{(\mathrm{q})}(n))$. Determine and interpret the relation between $\underline{y}_k \triangleq \langle r, r_k^{(\mathrm{i})}\rangle + j\langle r, r_k^{(\mathrm{q})}\rangle$ and $\underline{x}_k \triangleq x_k^{(\mathrm{i})} + jx_k^{(\mathrm{q})}$ as a function of $\underline{H}_k \triangleq A_k + jB_k$.

Example 17: Up-converter, RF-modulator
Show that

$$u(m) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \left(x_{[k]_N}^{(\mathrm{i})} \cos(2\pi \left(f_{\mathrm{c}} + \frac{k}{Np}\right) m) - x_{[k]_N}^{(\mathrm{q})} \sin(2\pi \left(f_{\mathrm{c}} + \frac{k}{Np}\right) m) \right)$$

is the result of the operation depicted in the block diagram below, where

$$\underline{s}(m) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \underline{x}_{[k]_N} e^{j2\pi\frac{k}{Np}m}, \qquad \underline{x}_{[k]_N} = x_{[k]_N}^{(\mathrm{i})} + jx_{[k]_N}^{(\mathrm{q})}.$$



Example 18: Down-converter (sin/cos demodulator, RF (radio frequency) demodulator)
Show that sin/cos multiplication $(2e^{-j2\pi f_c m})$, filtering by $h_{\mathrm{LP}}(m)$ and sampling-rate reduction by factor p of

$$v(m) = \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} \Bigg((A_{[k]_N} x_{[k]_N}^{(\mathrm{i})} - B_{[k]_N} x_{[k]_N}^{(\mathrm{q})}) \cos(2\pi \left(f_{\mathrm{c}} + \frac{k}{pN}\right) m) - \\ (B_{[k]_N} x_{[k]_N}^{(\mathrm{i})} + A_{[k]_N} x_{[k]_N}^{(\mathrm{q})}) \sin(2\pi \left(f_{\mathrm{c}} + \frac{k}{pN}\right) m) \Bigg)$$

yield

$$\underline{r}(n) = \frac{1}{\sqrt{N}} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \Big((A_{[k]_N} x_{[k]_N}^{(\mathrm{i})} - B_{[k]_N} x_{[k]_N}^{(\mathrm{q})}) + j(B_{[k]_N} x_{[k]_N}^{(\mathrm{i})} + A_{[k]_N} x_{[k]_N}^{(\mathrm{q})}) \Big) e^{j2\pi\frac{k}{N}n}.$$

Example 19: Channel partitioning with SVD
Consider the channel $H(z) = 1 + z^{-2}$.

(a) Determine the circulant convolution matrix $\tilde{\boldsymbol{H}}$ of size 3×3 for this channel such that there is neither inter-symbol interference nor inter-carrier (intra-symbol) interference.

(b) Find a solution to the channel partitioning problem using the SVD.

Example 20: Matched filter, correlation and sufficient statistics
Matched filter:

(a) Show that the correlation operation $\langle r, r_k \rangle$ can be described by an equivalent filter operation (with impulse response $h_{\mathrm{MF}}(n)$) followed by a sampler (sampled matched filter matched to the pulse $r_k(n), n = 0, \ldots, N-1$).

(b) Show that the output of the matched filter is a sufficient statistic (*i.e.*, that it contains all the relevant information) to estimate x_k.
Hint: A function $T(\boldsymbol{r})$ is a sufficient statistic for $\boldsymbol{x}$, iff the density $f_{\boldsymbol{r}}(\boldsymbol{r};\boldsymbol{x})$ can be factored as $f_{\boldsymbol{r}}(\boldsymbol{r};\boldsymbol{x}) = g(T(\boldsymbol{r}),\boldsymbol{x})h(\boldsymbol{r})$, where $g(\cdot)$ depends on $\boldsymbol{r}$ only through $T(\boldsymbol{r})$ and $h(\cdot)$ depends only on $\boldsymbol{r}$ (Neyman-Fisher factorisation theorem).

Example 21: Analysis of ICI and ISI
Compute the distortion caused by inter-symbol interference (ISI) and inter-carrier interference (ICI) that arises when the cyclic prefix is shorter than the channel dispersion.

Example 22: Linear matrix channel model $\boldsymbol{r} = \boldsymbol{H}\boldsymbol{s}$
Consider time-dispersive block transmission (block length N) over a MIMO channel with $S = 2$ inputs and $R = 3$ outputs. We stack the samples of the input signals $s_n^{(k)}, n = 1, \ldots, N, k = 1, \ldots, S = 2$ and the samples of the output signals $r_n^{(k)}, n = 1, \ldots, N, k = 1, \ldots, R = 3$ to obtain the vectors

$$\boldsymbol{s} = \begin{bmatrix} s_1^{(1)} & s_2^{(1)} & \ldots & s_N^{(1)} & s_1^{(2)} & \ldots & s_N^{(2)} \end{bmatrix}^{\mathrm{T}}$$

and

$$\boldsymbol{r} = \begin{bmatrix} r_1^{(1)} & \ldots & r_N^{(1)} & r_1^{(2)} & \ldots & r_N^{(2)} & r_1^{(3)} & \ldots & r_N^{(3)} \end{bmatrix}^{\mathrm{T}}.$$

The path from input $k \in \{1, 2\}$ to output $\ell \in 1, \ldots, 3$ is described by the impulse response $h_{\ell,k}(n),\ n = 0, \ldots, M$. Construct $\boldsymbol{H}$ for the channel model $\boldsymbol{r} = \boldsymbol{H}\boldsymbol{s}$ and $N = 3$, $M = 2$.

Example 23: Modelling of AWGN channel and slicer
Find a model of a binary-input AWGN channel (real-valued, mean zero, variance σ^2) followed by a hard-decision device (slicer).

Example 24: Rayleigh distribution
Determine the distribution of $Y_1 = \sqrt{X_1^2 + X_2^2}$ and $Y_2 = \arctan(X_2/X_1)$ if $X_1 \sim \mathcal{N}(0, \sigma^2)$ and $X_2 \sim \mathcal{N}(0, \sigma^2)$ are uncorrelated.

Example 25: Jakes/Clarke spectrum
Assume the FT emits a tone of wavelength λ. Determine the power spectral density of the received signal under "Rayleigh conditions" when the receiver is moving with velocity v.

Example 26: Coherence bandwidth, Doppler bandwidth
Consider a system using bandwidth B and a channel for which the following holds: Doppler bandwidth $B_{\mathrm{doppler}} \ll$ coherence bandwidth B_{coh}. Characterize the channel (*slow*/*fast* fading, *frequency-selective*/*frequency-flat*) for the following situations:

(a) $B < B_{\mathrm{doppler}}$

(b) $B_{\mathrm{doppler}} < B < B_{\mathrm{coh}}$

(c) $B_{\mathrm{coh}} < B$

Example 27: Circularly symmetric complex noise
Circularly symmetric complex noise (or sometimes also referred to as noise with independent real and imaginary parts) ...

Example 28: Frequency-flat slowly-fading system
Consider an indoor wireless system that transmits at $f = 2\,\mathrm{GHz}$. The delay spread is $T_{\mathrm{multi}} = 10\,\mathrm{ns}$ the maximum mobile speed is $v = 5.4\,\mathrm{km/h}$. Determine the range for the system bandwidth B such that the channel is frequency-flat and slowly fading (assume that $a < b$ if b is at least 10 times larger than a).

Example 29: Channel coherence, SNR
The performance of both absolute and differential modulation depends on the coherence of the channel. The complex-valued channel coefficient h_k, where k denotes either a time index (diff. modulation in time) or a subcarrier index (diff. modulation in frequency), can be written as

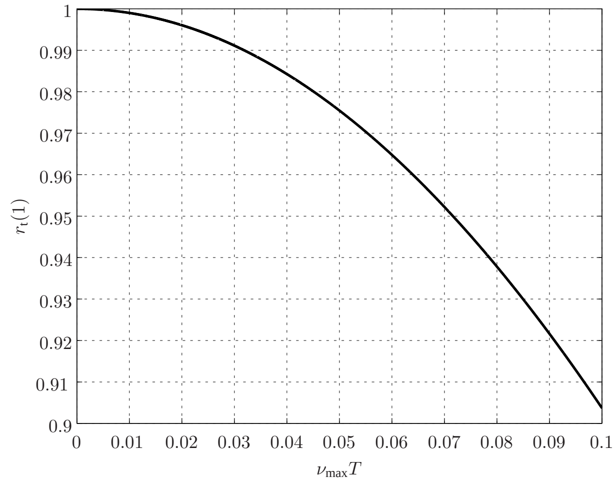
$$h_k = r \cdot h_{k-1} + \sqrt{1 - |r|^2} \cdot q$$

where $r = \mathsf{E}\left\{h_k h_{k-1}^*\right\}$ is the correlation coefficient and q is a complex-valued random variable modelling the innovation ($\mathsf{E}\left\{|h_k|^2\right\} = \mathsf{E}\left\{|h_{k-1}|^2\right\} = \mathsf{E}\left\{|q|^2\right\}$, $\mathsf{E}\left\{h_k\right\} = \mathsf{E}\left\{h_{k-1}\right\} = \mathsf{E}\left\{q\right\} = 0$). (Note that correlation and covariance are identical since we assume all the random variables to have zero mean.)

(a) Express the signal-to-distortion power ratio $\mathrm{SDR} = \frac{\mathsf{E}\left\{|h_k|^2\right\}}{\mathsf{E}\left\{|h_k - h_{k-1}|^2\right\}}$ as a function of r.

(b) Assume the channel signal-to-noise power ratio (SNR) is 12.04 dB. Determine the allowed SDR caused by channel variations to keep the total signal-to-noise-and-distortion power ratio (SNDR) at 10 dB.

Example 30: Differential modulation
Consider OFDM-based transmission using differential modulation in time (No. of subcarriers $N = 64$, cyclic-prefix length $L = 4$, baseband sampling frequency $F_{\mathrm{s}} = 20\,\mathrm{MHz}$, carrier frequency $F_{\mathrm{c}} = 5\,\mathrm{GHz}$). The correlation of the channel in time domain is characterised by the correlation between two consecutive symbols denoted $r_{\mathrm{t}}(1) = J_0(2\pi\nu_{\max}T)$ and depicted below (assuming Jakes spectrum; $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind). The distortion introduced by channel variations in time should be 20 dB below the signal power. Compute the maximum velocity $v_{\max}$.



Example 31: Absolute modulation
Consider OFDM-based packet transmission using coherent detection (No. of subcarriers $N = 64$, cyclic-prefix length $L = 8$, time-domain window length $W = 4$, subcarrier spacing $\Delta f = 312.5\,\mathrm{kHz}$, carrier frequency $F_{\mathrm{c}} = 2.4\,\mathrm{GHz}$). A pilot symbol (a symbol where the data on all subcarriers is known to the receiver) is transmitted at the beginning of each packet. The variation of the channel over time together with the tolerable distortion introduced by channel estimation errors determine the maximum packet length. Determine the maximum packet lengths $L_{\mathrm{packet}}(v)$ (in multicarrier symbols) to keep the Doppler-induced distortion 15.05 dB below the signal level for velocities v up to 5.4 km/h (walking speeds) and 180 km/h (vehicle speeds).

Example 32: MMSE equaliser
Derive the MMSE equaliser (use the orthogonality principle).

Example 33: LS, system of linear equations
Consider

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a coefficient matrix, $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ is given and $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ is unknown. When determining the solution $\boldsymbol{x}$ of such a system of linear equations, we distinguish three cases:

1. there is a unique solution (we know how to handle that: `x=inv(A)*y` or better `x=A\y`)
2. there are infinitely many solutions: in that case we may be interested in the solution $\bar{\boldsymbol{x}}$ with the minimum Euclidean norm $\|\bar{\boldsymbol{x}}\|_2 = \sqrt{\bar{\boldsymbol{x}}^{\mathrm{H}}\bar{\boldsymbol{x}}}$ (minimum energy)
3. there is no solution: in that case we may be interested in the "best" approximation $\hat{\boldsymbol{x}}$ in the sense of minimising the Euclidean norm $\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2$ of the error $\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}$

Regarding existence and uniqueness of solutions to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ we recall the following results from linear algebra (we refer to $\boldsymbol{A}' = \begin{bmatrix}\boldsymbol{A} & \boldsymbol{y}\end{bmatrix}$ as augmented matrix (augmented by the left side of $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$)):

- If $\operatorname{rank}\{\boldsymbol{A}\} = \operatorname{rank}\{\boldsymbol{A}'\}$, then $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ can be solved (exactly), *i.e.*, there is either a unique solution or there are infinitely many solutions.
- If there is a solution and $\operatorname{rank}\{\boldsymbol{A}\} = n$ then the solution is unique.

The function `linsysexample` returns example systems:

```
[A y]=linsysexample(1)    % example system No. 1 (there are 10 in total)
```

Determine existence and uniqueness of the solution for each case.

Example 34: LS, least norm
Begin with Example 33. An important case is the following: suppose $\boldsymbol{A}$ is skinny (or square), *i.e.*, $m \geq n$ ($m > n$ corresponds to an overdetermined equation system), and has full rank.

(a) What is the rank of $\boldsymbol{A}$ in this case? What do you know about existence and uniqueness of the solution in this case?

The unique $\boldsymbol{x}$ that minimises

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2$$

is given by

$$\boldsymbol{x}_{\mathrm{LS}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2 = \underbrace{(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{H}}}_{\boldsymbol{G}_{\mathrm{LS}}}\boldsymbol{y}$$

and it is referred to as least-squares solution[2]. The least squares solution has an insightful geometric interpretation: among all points $\boldsymbol{A}\boldsymbol{x}$ in the signal subspace $\mathcal{R}(\boldsymbol{A})$ spanned by the columns of $\boldsymbol{A}$, the least-squares solution $\boldsymbol{x}_{\mathrm{LS}}$ is closest to $\boldsymbol{y}$ and it is thus the orthogonal projection of $\boldsymbol{y}$ onto the signal subspace $\mathcal{R}(\boldsymbol{A})$.

(b) Determine the least-squares solution for example system No. 7.

Another important case is the following: $\boldsymbol{A}$ is a strictly fat matrix ($m < n$) of full rank. The solution with the minimum Euclidean norm is given by $\boldsymbol{A}^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}})^{-1}\boldsymbol{y}$ (minimum norm solution).

(c) Determine the least-norm solution for example system No. 9.

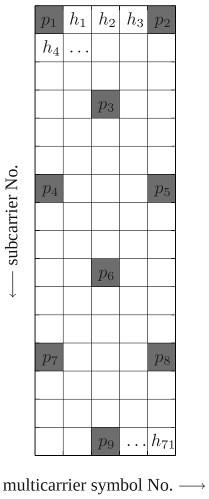
In general, minimum-norm solutions can be found using the Moore-Penrose pseudoinverse `pinv`.

[2] The term "least squares solution" is actually misleading in both cases: either there is a unique solution (then there is no error and no ambiguity, so there is no need for least squares) or there is no solution (then we should be talking about an approximation rather than about a solution). Nevertheless, we will adopt the common terminology.

Example 35: TEQ equaliser design using least-squares
Derive the solution for the time-domain equaliser using the method of least-squares (use the orthogonality principle).

Example 36: Estimation of *pilot channel coefficients*
Consider an OFDM system with $N = 16$ subcarriers, baseband sampling frequency $F_\mathrm{s} = 5\,\mathrm{MHz}$ and carrier frequency $F_\mathrm{c} = 5\,\mathrm{GHz}$. The system is designed for a maximum delay spread of $\tau_\mathrm{max} = 72\,\mathrm{ns}$ and a Doppler spread ν_max not exceeding 6 % of the subcarrier spacing.

In order to facilitate channel estimation, the system uses pilots with an allocation as illustrated below (for the sake of simplicity, the pilot symbols (which are known to the receiver) are all 1):



Assuming an exponentially decaying power delay profile and a Jakes Doppler spectrum, the correlation between two channel coefficients spaced by k subcarriers and ℓ multicarrier symbols is given by

$$r(k,\ell) = \frac{J_0(2\pi\ell\nu_\mathrm{max}T_\mathrm{sym})}{1 + j2\pi k\tau_\mathrm{max}/T_\mathrm{sym}}$$

where $J_0(\cdot)$ is the zeroth order Bessel function of the first kind (in MATLAB: `besselj(0,·)`). Channel estimation is done in two steps:

We stack the 9 channel coefficients on pilot positions in the vector $\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 & \dots & p_9 \end{bmatrix}^\mathrm{T}$.

(a) Derive the least-squares (LS) channel estimator $\boldsymbol{p}_\mathrm{LS}$.

(b) Derive the minimum mean square error (MMSE) channel estimator $\boldsymbol{p}_\mathrm{MMSE}$.

(c) Determine the mean channel estimation errors $E_{\boldsymbol{p}_\mathrm{LS}}$ and $E_{\boldsymbol{p}_\mathrm{MMSE}}$ of the LS and the MMSE estimator, respectively. Use the data in `chdata.mat`, which contains the three matrices `Hobs3dB`, `Hobs6dB` and `Hobs10dB` of size 80×1000 with 1000 observations of the 80 coefficients and three matrices `Htrue3dB`, `Htrue6dB` and `Htrue10dB` with the corresponding true values. Compare and comment on the results for signal-to-noise-power ratios of 3 dB, 6 dB and 10 dB.

Note: The channel at frequency instant $f \in \mathbb{Z}$ and time instant $t \in \mathbb{Z}$ is described by a complex-valued random variable $h(f,t)$ (which is the DFT of the channel impulse response). The spacing of the coefficients in frequency corresponds to the subcarrier spacing $\Delta f = 1/T_\mathrm{sym}$. The spacing of the coefficients in time corresponds to the multicarrier symbol length T_sym (to be more precise, the spacing should actually correspond to $T_\mathrm{sym} + \tau_\mathrm{max}$; consequently, we should replace T_sym in the Bessel function by $(T_\mathrm{sym} + \tau_\mathrm{max})$). Assuming stationarity in time and frequency, the correlation between two channel coefficients is defined as

$$r(k,\ell) \hat{=} \mathsf{E}\left[h(f,t)\,h^*(f-k,t-\ell)\right] = \mathsf{E}\left[h(f'+k,t'+\ell)\,h^*(f',t')\right].$$

Note that for positive lags in frequency ($k > 0$) and in time ($\ell > 0$) in the above expression, we take the complex conjugate of the coefficient that appears earlier time instant and at lower frequency. In order to compute the elements of the correlation matrix $\boldsymbol{R_{pp}} = \mathsf{E}\left[\boldsymbol{p}\boldsymbol{p}^{\mathrm{H}}\right]$, we need to find the correct lag-values k and ℓ. Consider, for example, the element in row No. 1 and column No. 6:

$$\mathsf{E}\left[p_1 p_6^*\right] = \mathsf{E}\left[h(f,t)\, h^*(f+9, t+2)\right] = \mathsf{E}\left[h(f,t)\, h^*(f-(-9), t-(-2))\right] = r(-9,-2).$$

For the element in row No. 6 and column No. 1, on the other hand, we have

$$\mathsf{E}\left[p_6 p_1^*\right] = \mathsf{E}\left[h(f+9,t+2)\, h^*(f,t)\right] = \mathsf{E}\left[h(f',t')\, h^*(f'-9, t'-2)\right] = r(9,2) = r^*(-9,-2),$$

which ensures Hermitian symmetry of $\boldsymbol{R_{pp}}$.

Example 37: Two-dimensional estimation of *remaining channel coefficients*
Begin with Example 36. Now we estimate the remaining channel coefficients from the coefficients $\boldsymbol{p}_{\mathrm{MMSE}}$ on the pilot grid. We stack the remaining 71 channel coefficients in the vector $\boldsymbol{h} = \begin{bmatrix} h_1 & h_2 & \dots & h_{71}\end{bmatrix}^{\mathrm{T}}$. Estimating $\boldsymbol{h}$ from $\boldsymbol{p}_{\mathrm{MMSE}}$ can also be interpreted as interpolation.

(a) Apply the linear MMSE estimator $\boldsymbol{h}_{\mathrm{MMSE}}$ for the remaining channel coefficients using $\boldsymbol{p}_{\mathrm{MMSE}}$ as observations.

(b) Determine the mean channel estimation error $E_{\boldsymbol{h}_{\mathrm{MMSE}}}$. Use the data in `chdata.mat`.

Example 38: SNR gap
Calculate the symbol error rate for BPSK. Compute the SNR gap Γ to achieve a symbol error rate $P_{\mathrm{s}} = 10^{-6}$.

Example 39: Bit- and power-loading
Consider a multicarrier system used to transmit over a white Gaussian noise channel with z-transform $H(z) = 1 + \frac{9}{10}z^{-1}$ and noise power $P_{Zi} = 0.1$. For $N = 8$ subcarriers, determine ...

(a) ... the achievable datarate for symbol error probability $P_{\mathrm{s}} = 10^{-6}$ and corresponding power/bit-loading.

(b) ... the achievable datarate for symbol error probability $P_{\mathrm{s}} = 10^{-6}$ and integer bit-loading and corresponding power/bit-loading.

Repeat for $N = 64$. Conclusion?

Example 40: Power-minimizing (margin-maximizing) loading
Modify the rate-maximizing loading Consider a multicarrier system used to transmit over a white Gaussian noise channel with z-transform $H(z) = 1 + \frac{9}{10}z^{-1}$ and noise power $P_{Zi} = 0.1$. Determine for $N = 8$ subcarriers

(a) achievable datarate for symbol error probability $P_{\mathrm{s}} = 10^{-6}$ and corresponding power/bit-loading

(b) achievable datarate for symbol error probability $P_{\mathrm{s}} = 10^{-6}$ and integer bit-loading and corresponding power/bit-loading

Repeat for $N = 64$.

Example 41: Waterfilling
Consider a multicarrier system with complex-valued baseband multiplex and $N = 4$. The channel is frequency-selective and time-invariant with coloured additive Gaussian noise. The total transmit power must not exceed 1.

(a) Determine $\{P_{Xi}\}$ such that the achievable information rate is maximised.

(b) Assume that the receiver performs channel identification (estimation of channel coefficients and noise variances) and reports its results to the transmitter:

$$\begin{aligned}\{\underline{H}_i\} =& \{\sqrt{\frac{4}{10}}e^{j\pi/3},\ \sqrt{\frac{3}{10}}e^{j\pi/4},\ \sqrt{\frac{2}{10}}e^{-j\pi/6},\ \sqrt{\frac{1}{10}}e^{j\pi/2}\}\\ \{P_{Zi}\} =& \{0.2,\ 0.1,\ 0.05,\ 0.1\}\end{aligned}$$

Determine $\{P_{Xi}\}$ such that the achievable information rate is maximised.

(c) In addition to the total power constraint, there is a PSD constraint formulated as $\{P_{Xi}\} \leq 1/2$. Determine $\{P_{Xi}\}$ such that the achievable information rate is maximised.

Example 42: Reed Solomon code—an example
Consider a 2-error correcting RS code over $GF(2^3)$ field. The field is generated by the polynomial $p(x) = x^3 + x + 1$.

(a) Determine the code length N, information block length K and minimum distance $d_{\min}$.

(b) Find the field elements and represent them in binary notation.

(c) Determine the generator polynomial $g(D)$ and find its binary representation.

Example 43: Reed Solomon code from DVB-T standard
Analyse the following specifications of the Reed Solomon code used in the DVB-T standard:

(a) Systematic (255,239) $t = 8$ RS code

(b) Shortened (204,188) $t = 8$ RS code

(c) Field generator polynomial $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

Example 44: Convolutional code from DVB-T standard
The DVB-T standard uses a rate $R = 1/2$ convolutional code specified by the generator matrix $(171, 133)_8$.

(a) Write down the generator matrix $\boldsymbol{G}(D)$. Determine the memory m.

(b) Draw the encoder realisation in controller canonical form.

Example 45: Puncturing of convolutional code
Puncturing of a convolutional code is defined as deleting specified code symbols of the output code sequence. As a result, a new code is obtained, whose rate is *higher* than the rate of the original *mother code*. Puncturing is specified by the puncturing pattern, where the zeros mark the code symbols that should be omitted. A punctured code is decoded using the trellis of the mother code.

The DVB-T standard specifies (among others) the following puncturing sequence for the mother code of Example 44.

(a) Puncturing pattern for the first output: $P_1 = 1\ 0\ 1\ 0\ 1$

(b) Puncturing pattern for the second output: $P_2 = 1\ 1\ 0\ 1\ 0$

Determine the rate of the punctured code.

Example 46: Interleaver
Consider a multicarrier system with a data rate of 8 Mbit/s, a multicarrier-symbol rate of 4000 multicarrier-symbols per second and a shortened GF(256) RS code that can correct 8 code symbols. Codewords are locked to the multicarrier-symbol rate: each codeword corresponds to one DMT symbol. A block interleaver is used to spread errors such that a completely destroyed DMT symbol can be recovered.

(a) Determine the required interleaver depth

(b) Determine the resulting latency in ms

Hint: block interleaver (4 code symbols per codeword, depth 3)

write row-wise: $1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12;$
read column-wise: $1, 5, 9, 2; 6, 10, 3, 7; 11, 4, 8, 12;$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

Example 47: Synchronization, clock/carrier frequency offset
Consider an OFDM-based wireless LAN transceiver with the following parameters: No. of carriers $N = 64$, sampling frequency $F_\mathrm{s} = 20\,\mathrm{MHz} \pm 20\,\mathrm{ppm}$, carrier frequency $F_\mathrm{c} = 5\,\mathrm{GHz} \pm 50\,\mathrm{ppm}$

(a) Determine the maximum relative sampling (clock) frequency offset with respect to the carrier spacing in %.

(b) Determine the maximum relative carrier frequency offset with respect to the carrier spacing in %.

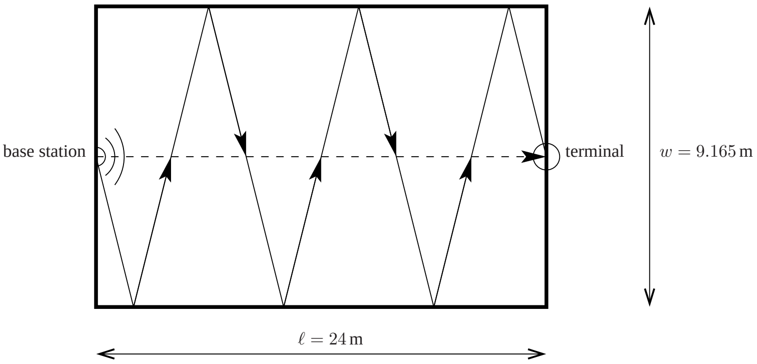
(c) After how many samples is the timing one sample off the correct timing?

(d) After how many multicarrier symbols is the timing one sample off?

(e) After how many seconds is the timing one sample off?

(f) After 1 second of transmission, how many samples and how many multicarrier symbols is the timing off with respect to the correct timing?

Example 48: Synchronization, clock/carrier frequency offset
Consider an indoor WiFi downlink transmission. The propagation scenario depicted below represents the situation with the largest delay spread. (There may very well be distinct propagation paths that are longer but we consider their resulting signal strength to be below the terminal's sensitivity.)



The system parameters are: No. of carriers $N = 64$, sampling frequency $F_\mathrm{s} = 20\,\mathrm{MHz}$.

(a) Determine the delay spread T_multi and the required CP-length L to avoid ISI and ICI.

(b) Assume $L = 8$. Let $r'(n) = r(n - d)$ denote the time-domain receive signal with a symbol-timing offset of d samples. Determine the values d that allow a simple post-FFT symbol-timing correction. Let $R'(k), k = 0, \ldots, N-1$ denote the FFT of $r'(n)$. Determine the required post-FFT processing to obtain $R(k)$ (FFT of the receive signal $r(n)$ with correct timing) from $R'(k)$.

Example 49: Synchronisation symbol, elementary FFT/IFFT computation
Consider an arbitrary pseudo-noise sequence $Z[k] \in \mathbb{C}, k = 0, \ldots, N/2 - 1$, where N is even.

(a) Show that the IFFT of the (frequency-domain) sequence

$$X[2k] = \begin{cases} Z[k], & k = 0, \ldots, N/2 - 1 \\ 0, & \text{otherwise} \end{cases}$$

yields a (time-domain) sequence with two identical halves.

(a) Show that the IFFT of the (frequency-domain) sequence

$$X[2k+1] = \begin{cases} Z[k], & k = 0, \ldots, N/2 - 1 \\ 0, & \text{otherwise} \end{cases}$$

yields a (time-domain) sequence with two negated but otherwise identical halves.

Example 50: Synchronisation algorithm example—demo and analysis
Run the demo script `synch.m` and convince yourself of the following (load the corresponding settings, *e.g.*, `load s1.mat`, before running `synch.m`):

(a) Settings: `s1.mat`. Convince yourself that $M_{\text{corr}}(d)$ has a unique maximum if $L = M$.

(b) Settings: `s2.mat`. Convince yourself that $M_{\text{corr}}(d)$ has a nonunique maximum ("maxima-plateau") for $L > M$ and argue why the last value of that plateau should be chosen as $\hat{d}$.

(c) Settings: `c1.mat`, `c2.mat`. Convince yourself that a residual carrier-phase error can yield to erroneous decisions and determine the condition for correct decisions.

Example 51: PAR of OFDM signal
Let $\Pr(|X| > p)$ and $\Pr(|X| \leq p)$, where $X = R + jI$, $R \sim \mathcal{N}(0, \sigma^2/2)$, $I \sim \mathcal{N}(0, \sigma^2/2)$, and

$$\mathsf{E}\left\{\begin{bmatrix} R \\ I \end{bmatrix}\begin{bmatrix} R & I \end{bmatrix}^{\mathrm{H}}\right\} = \begin{bmatrix} \sigma^2/2 & 0 \\ 0 & \sigma^2/2 \end{bmatrix},$$

denote the cumulative distribution function (CDF) and the complementary cumulative distribution function (CCDF), respectively, of an OFDM transmit signal's PAR with infinitely many subcarriers.

(a) Determine $\Pr(|X| > p)$ and $\Pr(|X| \leq p)$. Plot $\Pr(|X| > p)$ and $\Pr(|X| \leq p)$ versus $20\log_{10}(p/\sigma)$.

(b) Verify (a) via simulation.

Example 52: Block-PAR of OFDM signal
Let PAR_N denote the probability that the magnitude of all N samples of an OFDM transmit block is below p, *i.e.*, PAR_N is the CDF of the per-block PAR or of the peak PAR.

(a) Determine the CDF PAR_N and the CCDF $1 - PAR_N$ (assume independent samples). Plot PAR_N and $1 - PAR_N$ versus $20\log_{10}(p/\sigma)$ for $N = 64, 128, 512, 1024, 4096$.

(b) Verify (a) via simulation.

Example 53: Gradient-based tone reservation algorithm
Derive the gradient-based tone reservation algorithm for PAR reduction:

(a) Determine an expression for the residual peak, which is the signal $d(n)$ that is clipped if a reduced signal $s(n) + c(n)$ is hard-limited to $-s_{\max} \leq s(n) + c(n) \leq s_{\max}$ (assume real-valued baseband multiplex).

(b) Determine the power P_{clip} of all residual peaks in a multicarrier symbol.

(c) Determine the gradient $\boldsymbol{\nabla}_{x_k, k\in\mathcal{T}_{\text{tr}}} P_{\text{clip}}$ of the clip-noise power P_{clip} with respect to the data $x_k, k \in \mathcal{T}_{\text{tr}}$ of the reserved tones.

(d) Identify the terms in the recursive update-equation for the counter-peak:

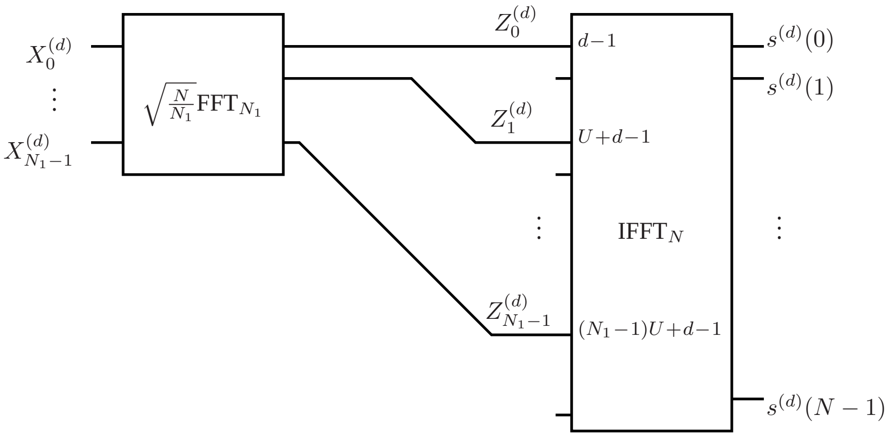
$$\boldsymbol{c}^{(i+1)} = \boldsymbol{c}^{(i)} - \mu \sum_{n:|s(n)+c^{(i)}(n)|>s_{\max}} \underbrace{\operatorname{sign}(s(n) + c^{(i)}(n))}_{\text{direction}} \underbrace{(|s(n) + c^{(i)}(n)| - s_{\max})}_{\text{magnitude}} \boldsymbol{p}(n)$$

Example 54: LTE uplink PAR-reduction proposal: FFT spreading
Beyond-3G cellular mobile communication systems (long term evolution (LTE)) are based on OFDM. The peak-to-average power ratio is an important issue—in particular for the uplink.

An approach sometimes referred to as FFT-spreading is the following. Consider an OFDMA system with U users. Each user occupies N_1 of the $N = N_1U$ subcarriers. User No. $d \in \{1, \ldots, U\}$ occupies subcarriers $\{d-1, U+d-1, 2U+d-1, \ldots, (N_1-1)U+d-1\}$. The transmit processing for user No. d is depicted below (FFT_N and IFFT_N denote normalised size-N DFT and IDFT, respectively).

(a) Compute the transmit signal $s^{(d)}(n), n = 0, \ldots, N-1$ of user No. d.

(b) Assume 16QAM modulation for the subcarrier data $X_k^{(d)}, k = 0, \ldots, N_1 - 1$. Compute the resulting worst-case PAR with FFT-spreading. Compare the result to the worst-case PAR of a system without FFT-spreading (using an N_1-point IFFT).



Example 55: Out-of-band egress, zero padding

An alternative to the cyclic-prefix method is zero-padding:

- at the transmitter, each symbol $s(n)$ of length N (output of the IFFT) is extended by L trailing zeros yielding a block $s'(n)$ of length $N+L$:

$$s'(n) = \begin{cases} s(n), & n = 0, \dots, N-1 \\ 0, & n = N, \dots, N+L-1 \end{cases}$$

- at the receiver, the trailing L samples of each block $r'(n)$ of length $N+L$ (output of the channel, assuming perfect symbol-timing synchronisation) are added element-wise to the first L samples, which yields $r(n)$ (input to FFT):

$$r(n) = \begin{cases} r'(n) + r'(n+N), & n = 0, \dots, L-1 \\ r'(n), & n = L+1, \dots, N-1 \end{cases}$$

(a) Express zero padding in a matrix-based notation (define suitable matrices $\boldsymbol{Z}_{\mathrm{add}}$ and $\boldsymbol{Z}_{\mathrm{rem}}$) and show that zero-padding diagonalises any dispersive channel as long as $L \geq M$.

(b) Compute (analytically) the DTFT $S(f)$ of the resulting transmit basis functions when zero padding is used.

(c) Verify your result on (b) in MATLAB (approximate the DTFT of a basis function `s` using the command `fft(s,Nfft)`, where `Nfft`$\gg$`length(s)`).

(d) Consider a system with $N = 64$ and $L = 8$. Compare the spectral containment of cyclic prefixing and zero padding:

 – We define the in-band power P_{in} of the kth basis function as the integral power in a band of width $1/N$ around the kth subcarrier frequency:

$$P_{\mathrm{in}} = \int\limits_{f=k/N-1/(2N)}^{k/N+1/(2N)} |S(f)|^2 \mathrm{d}f$$

 We define the out-of-band power P_{out} as the power in the remaining frequency range.

 (a) Compute the ratio $P_{\mathrm{in}}/P_{\mathrm{out}}$ for zero padding.

 (b) Compute the ratio $P_{\mathrm{in}}/P_{\mathrm{out}}$ for cyclic prefixing.

 (c) Compare the two methods with respect to spectral containment.

Example 56: MIMO capacity
Let P_T denote the transmit power used to communicate over a deterministic $T \times T$ MIMO channel $\boldsymbol{H} = \boldsymbol{I}_T$ with additive, (spatially) white, circularly symmetric, Gaussian noise of variance P_{noise}. Compute the power P_1 necessary so that a SISO channel $\boldsymbol{H} = 1$ with P_1 yields the same capacity as a MIMO channel $\boldsymbol{H} = \boldsymbol{I}_T$ with P_T. Plot P_1/P_T versus T and convince yourself that the required power increase is dramatic.

Example 57: Diversity gain, spatial multiplexing gain
A dumb diversity scheme: $S = 2$ transmit antennae, $R = 1$ receive antenna

$$\begin{bmatrix} x_1 & x_2 & x_3 & \dots \\ 0 & x_1 & x_2 & \dots \end{bmatrix}$$

Determine the number of transmitted symbols per time slot and the diversity gain.

Example 58: Diversity gain, spatial multiplexing gain
Begin with Example 57. Now consider a smarter diversity scheme: $S = 2$ transmit antennae, $R = 1$ receive antenna. The transmitter has no CSI. Assumption: the channel coefficients h_1 and h_2 remain constant for two consecutive time slots (coherence time of the channel is large enough to justify this assumption).

$$\begin{bmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{bmatrix}$$

The receiver has perfect CSI and combines

$$\begin{aligned} \tilde{x}_1 &= h_1 x_1 + h_2 x_2 \\ \tilde{x}_2 &= -h_1 x_2^* + h_2 x_1^* \end{aligned}$$

Determine the number of transmitted symbols per time slot and the diversity gain.

Example 59: Classical receive beamforming (SIMO case)
Consider the SIMO channel $\boldsymbol{y} = \boldsymbol{h}x + \boldsymbol{n}$, $\boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, $\mathsf{E}\left\{|x|^2\right\} = 1$. Design a linear receiver $\hat{x} = \boldsymbol{r}^{\mathrm{H}}\boldsymbol{y}$ that maximizes the SNR.

Example 60: Classical transmit beamforming (MISO case)
Consider the MISO channel $y = \boldsymbol{h}^{\mathrm{H}}\boldsymbol{x} + n$, $n \sim \mathcal{CN}(0, \sigma^2)$. Design a linear transmitter $\boldsymbol{x} = \boldsymbol{t}x$, $\mathsf{E}\left\{|x|^2\right\} = 1$, $\boldsymbol{t}^{\mathrm{H}}\boldsymbol{t} \leq 1$ that maximizes the SNR.