Advanced Modulation & Coding

EEE8003/8104

Stéphane Le Goff

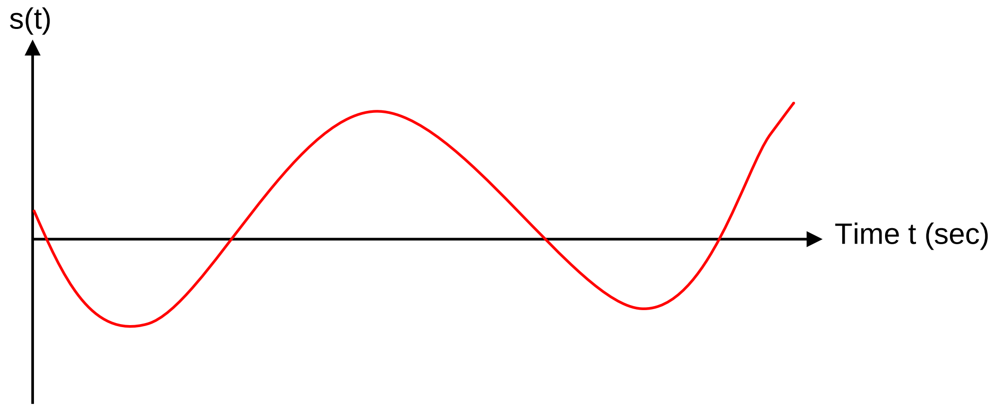
School of Electrical and Electronic Engineering

Newcastle University

1. Basic Definitions

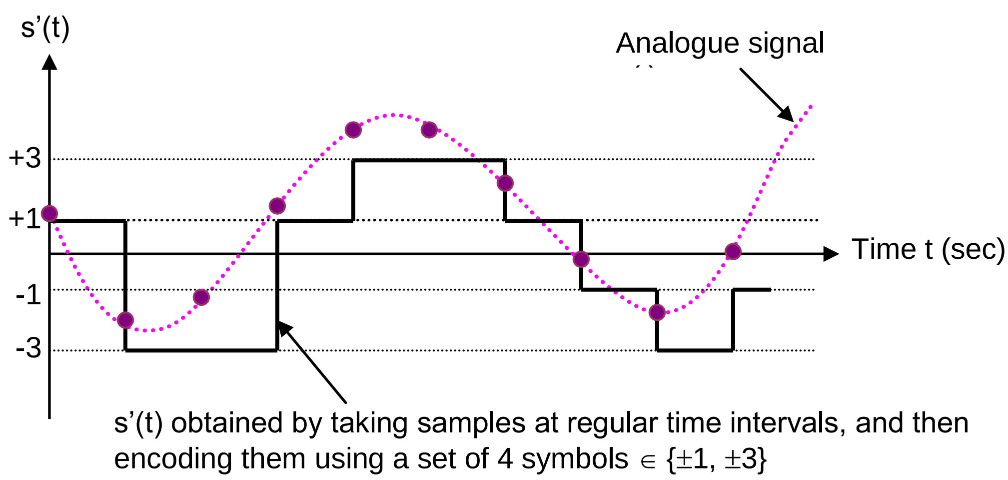
• Analogue signal

An analogue signal s(t) is defined as a physical time-varying quantity and is usually smooth and continuous, e.g. acoustic pressure variation when speaking.

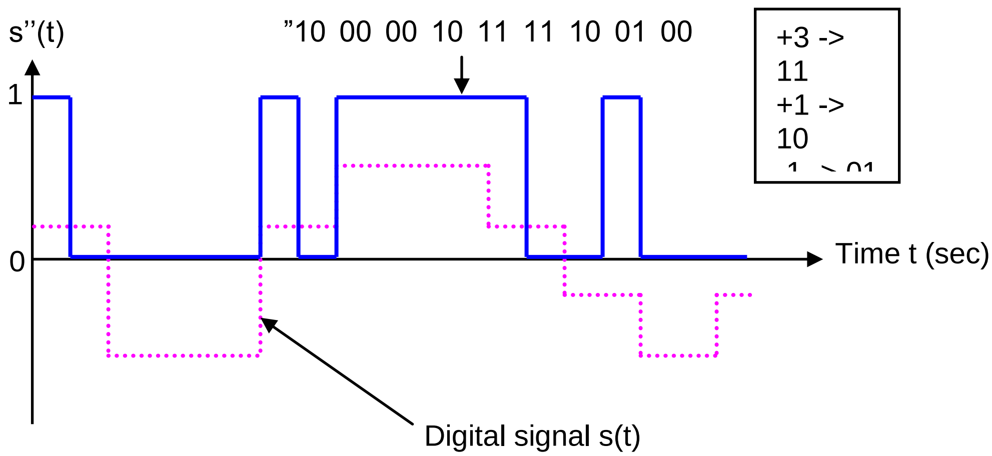


• Digital signal

A digital signal on the other hand is made up of discrete symbols selected from a finite set, e.g. letters from the alphabet or binary data. For example, the previous analogue signal s(t) can be converted into a 4-level digital signal s'(t) as follows:



In many systems, the digital signal is actually a binary signal in the sense that it can take only 2 values: 0 or 1. For example, the digital signal s'(t) can be converted into a binary digital s''(t) signal as shown below.



The main advantage of digital systems over analogue systems is that they are far more resistant to noise. In addition, digital systems allow for the use of powerful digital processing techniques such as error correction/detection, data compression, easier data multiplexing, etc.

- **Transmitter**

The transmitter element in a communication system processes the message signal in order to produce a signal most likely to pass reliably and efficiently through the channel. This usually involves modulation of a carrier signal by the message signal, coding of the signal to help correct for transmission errors, filtering of the message or modulated signal to limit the occupied bandwidth, and power amplification to overcome channel losses.

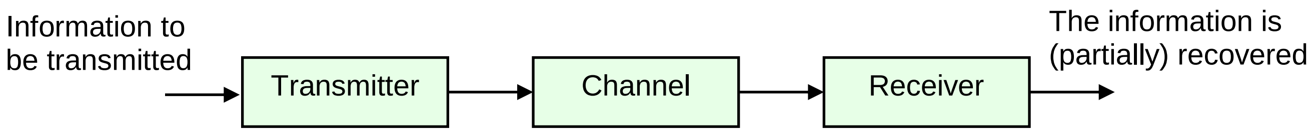
- **Transmission Channel**

The channel is defined as the electrical medium between source and destination, e.g. cable, optical fiber, free space, etc. Any channel is characterized by its loss/attenuation, bandwidth, noise/interference and distortion.

- **Receiver**

The receiver function in a communication system is principally to reverse the modulation processing of the transmitter in order to recover the message signal, attempting to compensate for any signal degradation introduced by the channel. This will normally involve amplification, filtering, demodulation and decoding, and, in general, is a more complex task than the transmit processing.

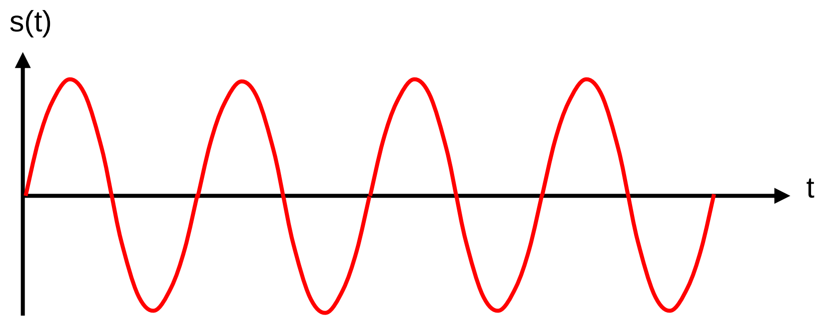
• **General structure of a communication scheme**



2. Deterministic Signals

A signal is defined as any sign, gesture, token, etc. that serves to communicate information.

A signal is said to be deterministic if its future values can be predicted. Therefore, deterministic signals do not carry information, but they are used by transmitters to carry information. An example of deterministic signal is $s(t) = A.\cos(2\pi ft + \theta)$.



• **Periodic signals**

A deterministic signal s(t) is said to be periodic if $s(t) = s(t \pm n.T)$, where n is an integer and T (in sec) is the period of the signal. Periodic signals are eternal. The frequency f of the signal is given by $f = \frac{1}{T}$ (Hz). The mean value, m, of a periodic signal is

$$m = \frac{1}{T} \cdot \int_{t}^{t+T} s(t)\, dt,$$

while its power, P, is given by

$$P = \frac{1}{T} \cdot \int_{t}^{t+T} [s(t)]^2 dt.$$

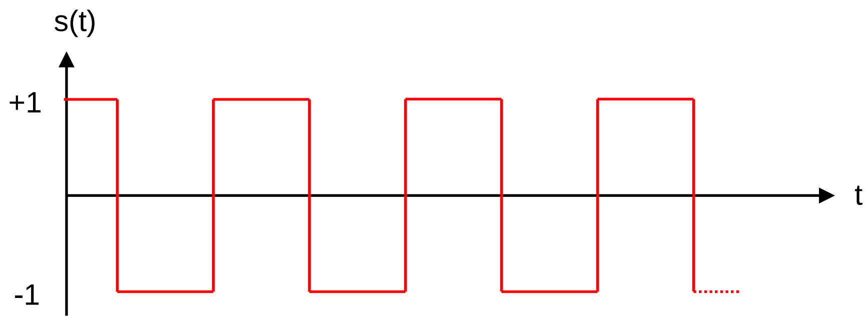
The total energy of a periodic signal is infinite. Periodic signals are sometimes called **power signals**.

Example ---

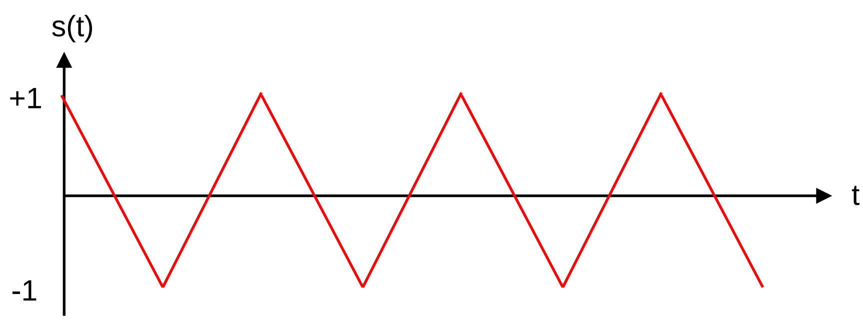
Show that the power of the periodic signal $s(t) = A \cdot \cos(2\pi f t + \theta)$ is given by $P = \frac{A^2}{2}$. Note that this power is proportional to the square of the signal amplitude. Does it make sense?

Examples of periodic signals ---

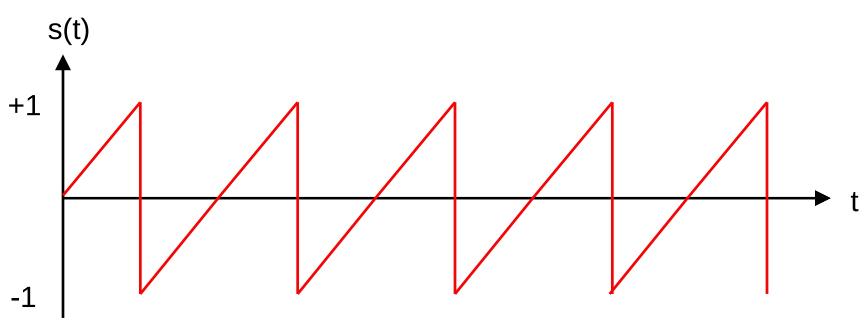
Square signal:



Triangular signal:



Saw-tooth signal:



- **Finite-energy (or energy) signals**

A deterministic signal is said to be finite-energy if it is time-limited. Pulses are a good example of such signals. The energy, E, of a finite-energy signal s(t) is given by

$$E = \int_{-\infty}^{+\infty} [s(t)]^2 dt .$$

Example --

Show that the energy of a square pulse of amplitude A and duration T is given by $E = A^2 \cdot T$. Note that this energy is proportional to the duration of the pulse and the square of its amplitude. Does it make sense?

--

3. Time/Frequency Representations of Deterministic Signals

To understand the operation of any communication link, it is essential to have a good grounding in the relationship between the shape of a waveform in the **time domain** and its corresponding spectral content in the **frequency domain**.

Any signal, s(t), represented in the time domain, has an equivalent representation, S(f), in the frequency domain. S(f) is often referred to as the **spectrum** of s(t). The spectrum shows the distribution of the signal energy/power as a function of frequency.

• **Spectrum of a periodic signal: Fourier series**

Assume that s(t) is a periodic signal with period T. Fourier showed that such signal can be seen as an infinite sum of sine and cosine terms:

$$s(t) = a_0 + 2 \cdot \sum_{n=1}^{+\infty} \left[a_n \cdot \cos\left(\frac{2\pi n}{T} t\right) + b_n \cdot \sin\left(\frac{2\pi n}{T} t\right) \right],$$

where

- $a_0 = \frac{1}{T} \cdot \int_{-T/2}^{T/2} s(t)\, dt$ is the mean value of the periodic signal,

- $a_n = \frac{1}{T} \cdot \int_{-T/2}^{T/2} s(t) \cdot \cos\left(\frac{2\pi n}{T} t\right) dt$,

- $b_n = \frac{1}{T} \cdot \int_{-T/2}^{T/2} s(t) \cdot \sin\left(\frac{2\pi n}{T} t\right) dt$.

The term $f_1 = \frac{1}{T}$ is the fundamental frequency of the periodic signal, and the quantity $f_n = \frac{n}{T} = n \cdot f_1$ represents its (n-1)-th harmonic. It is possible to re-write the equations in a more compact way:

$$s(t) = a_0 + 2 \cdot \sum_{n=1}^{+\infty} \mathrm{Re}\left[c_n \cdot \exp\{ -j 2 \pi f_n t \} \right],$$

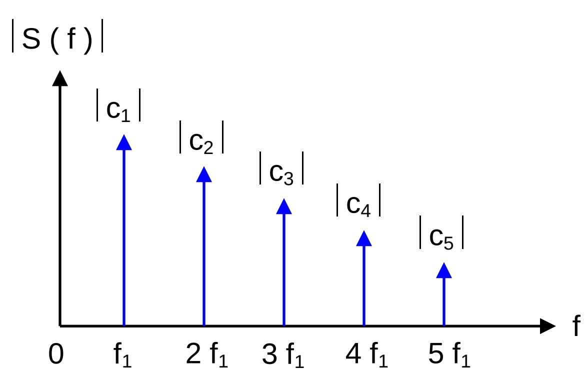
where $c_n = \frac{1}{T} \cdot \int_{-T/2}^{T/2} s(t) \cdot \exp\{+j 2\pi f_n t\} dt$.

The complex coefficient $c_n = a_n + jb_n$ is the complex amplitude of the (n-1)-th harmonic.

The representation of a periodic signal by a Fourier series is equivalent to the resolution of the signal into its various harmonic components. We see that a signal s(t) with period T is composed of sine waves with frequencies 0, f_1, $2f_1$, $3f_1$ and so on. The frequency-domain description, i.e. the spectrum, of the signal therefore consists of components at frequencies 0, f_1, $2f_1$, $3f_1$ and so on. The spectrum of periodic signals is discrete.

If we specify the spectrum S(f), we can determine the corresponding time-domain periodic signal s(t), and vice-versa. It is important to understand that s(t) and S(f) are two different representations of a same signal.

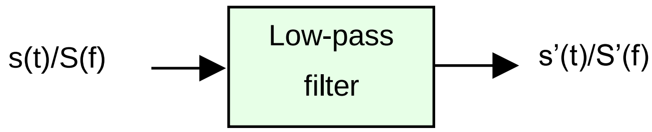
It is common practice to represent only the magnitude of each component c_n in the spectrum. However, one must not forget that the terms c_n in the Fourier series expansion are, generally, complex numbers, and hence both magnitude and phase of c_n are required for a complete representation of the signal.



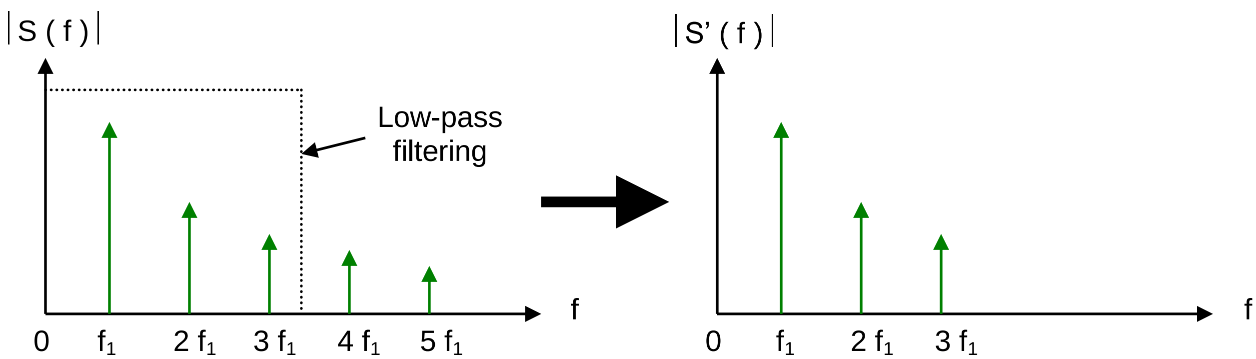
This discrete spectrum shows the distribution of the signal energy on the frequency axis.

If the signal is transmitted through a filter, some frequency components in the spectrum will be suppressed. This will introduce some distortion in the time-domain signal.

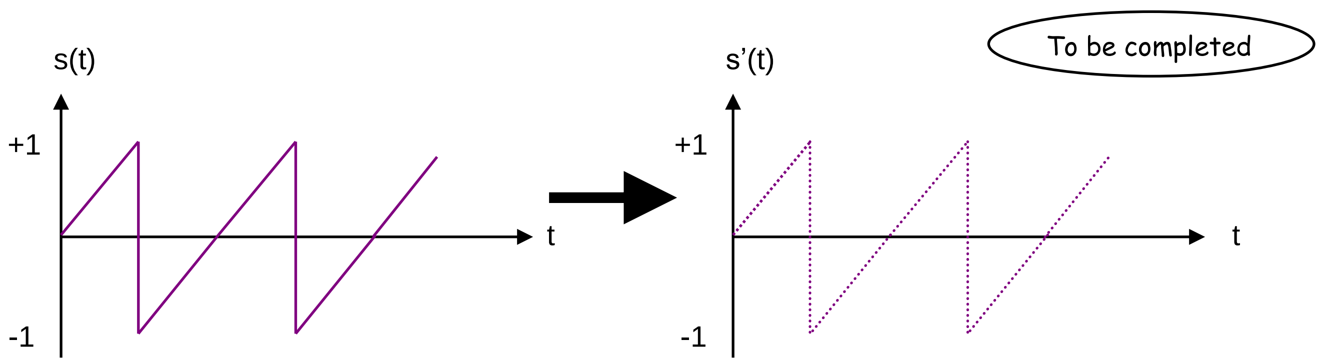
Example: effect of low-pass filtering on a saw-tooth wave



In the frequency domain:



In the time domain:

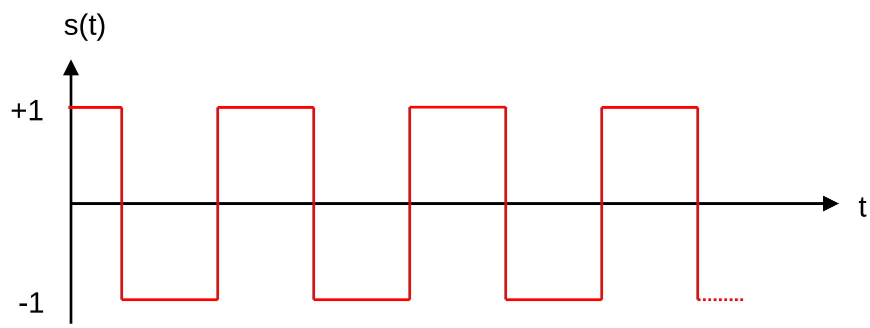


The bandwidth of a signal is defined as the frequency band occupied by its spectrum. Depending on their shape in the time domain, certain types of signals have a finite bandwidth (typically, signals with smooth variations, e.g. sine waves), while some others have an infinite bandwidth (typically, signals with infinitely sharp variations, e.g. square waves).

Examples of periodic signals and their Fourier series

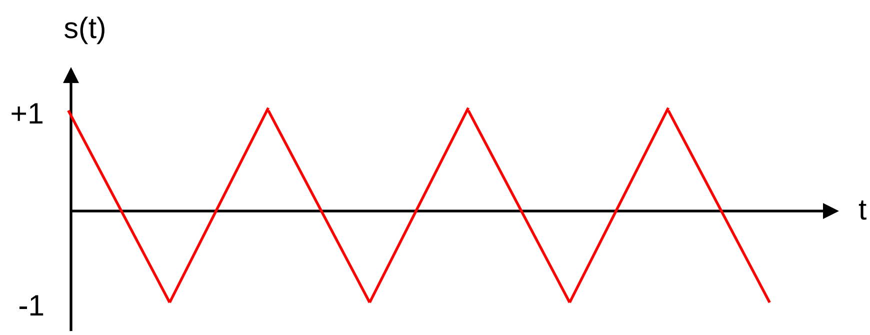
$\omega = 2\pi f_1 = 2\pi/T$ → $$s(t) = a_0 + 2\cdot\sum_{n=1}^{+\infty}\left[a_n\cdot\cos(n\omega t) + b_n\cdot\sin(n\omega t)\right]$$

Square signal:



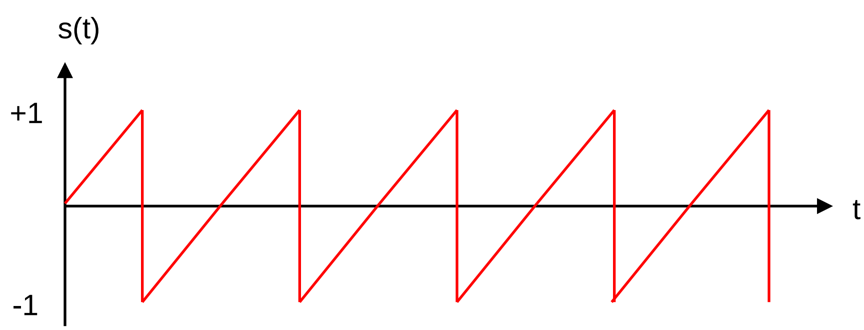
$$s(t) = \frac{4}{\pi}\cdot\left(\cos\omega t - \frac{1}{3}\cos 3\omega t + \frac{1}{5}\cos 5\omega t - \frac{1}{7}\cos 7\omega t + \ldots\right)$$

Triangular signal:



$$s(t) = \frac{8}{\pi^2}\cdot\left(\cos\omega t + \frac{1}{9}\cos 3\omega t + \frac{1}{25}\cos 5\omega t + \frac{1}{49}\cos 7\omega t + \ldots\right)$$

Saw-tooth signal:



$$s(t) = \frac{2}{\pi} \cdot \left(\sin\omega t - \frac{1}{2}\sin 2\omega t + \frac{1}{3}\sin 3\omega t - \frac{1}{4}\sin 4\omega t + \ldots \right)$$

It is worthwhile noting that the spectrum of a constant (DC) signal is composed of a single component located at the zero frequency.

• Spectrum of a finite-energy signal: Fourier transform

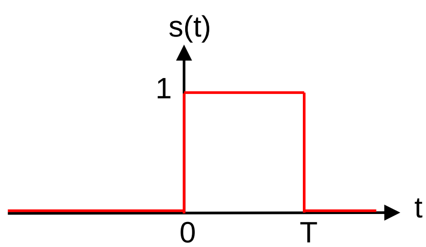
The frequency content of a deterministic finite-energy signal cannot be obtained by evaluating its Fourier series expansion because such signal is not periodic. The mechanism for obtaining the spectrum of a finite-energy signal is to calculate the Fourier transform of its time-domain representation as follows:

$$S(f) = \int_{-\infty}^{+\infty} s(t) \cdot e^{-j2\pi f t}\, dt\,.$$

This equation cannot be evaluated for infinite-energy signals, such as periodic signals for example. Unlike the spectrum obtained for periodic signals, the function S(f) is now continuous.

Example ---

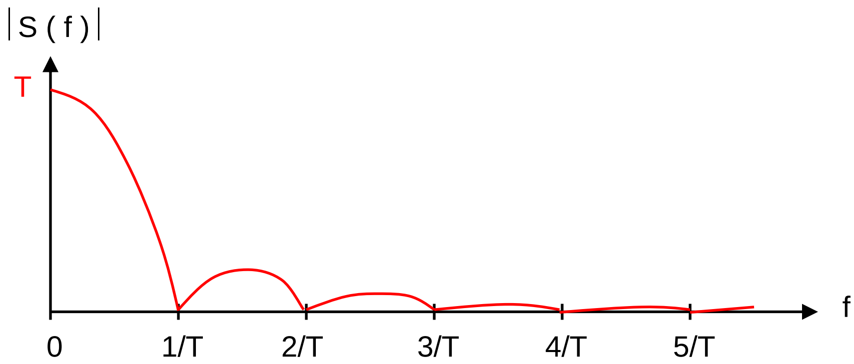
Square pulse



Show that $S(f) = T \cdot \mathrm{sinc}(\pi f T) \cdot e^{-j\pi f T}$.

In general, we are more particularly interested in the magnitude of the complex function S(f) since it contains the information related to the energy distribution versus frequency. Here, the magnitude of S(f) is given by

$$|S(f)| = T \cdot |\mathrm{sinc}(\pi f T)|\,.$$



Strictly speaking, the bandwidth of the square pulse is infinite. This means that, if one attempts to transmit such pulse through a practical communication system, some (high) frequency components will be filtered out, which will result in distortion in the time-domain.

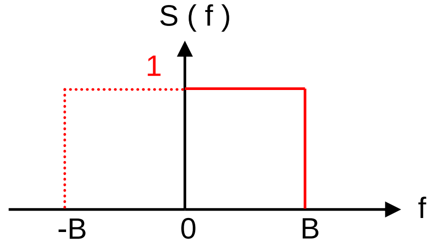
The factors affecting the bandwidth of a signal are both its duration and its shape. In particular, a smooth, slow-varying signal will always have a small bandwidth, while a sharp, fast-varying signal will have a larger bandwidth.

If we know the spectrum of a signal, it is possible to obtain its time-domain representation by computing the inverse Fourier transform as follows:

$$s(t) = \int_{-\infty}^{+\infty} S(f) \cdot e^{+j2\pi f t} \, df \, .$$

Example ---

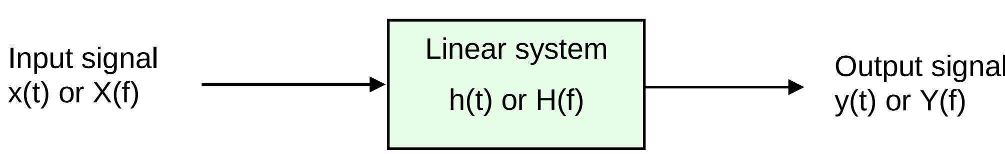
Signal with bandwidth B



Show that $s(t) = 2 \cdot B \cdot \mathrm{sinc}(2\pi B t)$.

The shape of this “sinc” pulse is very smooth, which explains why its bandwidth is not infinite, unlike the square pulse studied previously. In case the sinc pulse is “spread”, i.e. made even smoother, this will result in a smaller bandwidth B.

In signal processing, the Fourier transform is often used to evaluate the response of a linear system to an input signal. To illustrate this, let us consider a system characterized by its **impulse response** h(t) or, equivalently, its **transfer function** H(f). The transfer function H(f) is the Fourier transform of the impulse response h(t).



The output signal is obtained by applying:

$$y(t) = h(t) * x(t) = \int_{-\infty}^{+\infty} h(\tau) \cdot x(t-\tau)\, d\tau \,,$$

or

$$Y(f) = H(f) \cdot X(f).$$

The last expression is much easier to evaluate than the previous one since it only involves a multiplication instead of a convolution. This is why it is often preferable to deal with frequency-domain representations rather than time-domain representations.

Example ---

Consider a signal s(t) and its Fourier transform S(f). Show that the Fourier transform of s(-t) is S*(f).

Example ---

Consider a signal s(t) and its Fourier transform S(f). Show that the Fourier transform of s(t-T), where T is a constant, is S(f).exp{j2πfT}.

Example ---

Consider two signals g(t) and h(t) defined so that $g(t) = h(t) * h(-t)$. Show that the energy of h(t) is equal to g(0).

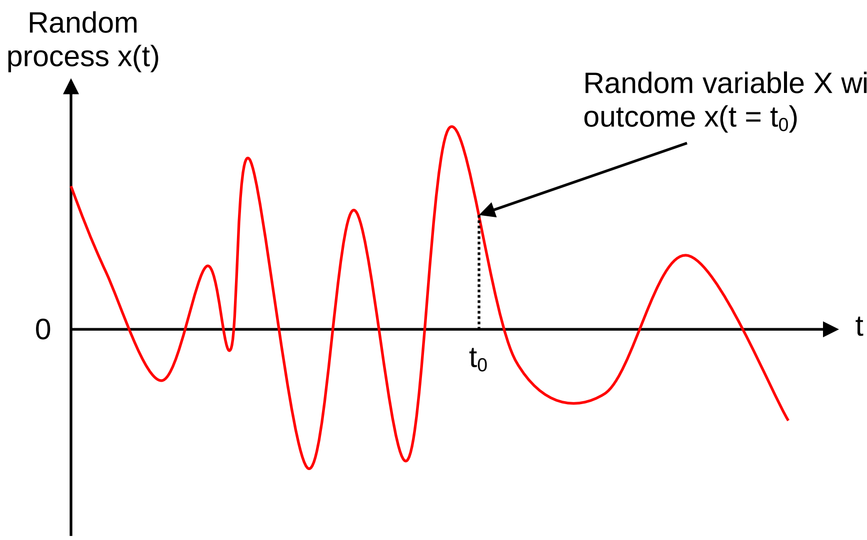
Example ---

Consider a signal s(t) and its Fourier transform S(f). Show that $s(0) = \int_{-\infty}^{+\infty} S(f)\,df$ and $S(0) = \int_{-\infty}^{+\infty} s(t)\,dt$.

----

4. Random Variables and Signals

In communication engineering, one must often deal with **random processes**, i.e. time-varying random "signals" (voice, noise…). A random process is not deterministic, meaning that it is impossible to predict what values such "signal" will take in the future. One can just have an "idea" about its behavior. In general, communication engineers have the knowledge of some of the statistical properties of the random process.



The random process x(t), at any time t = t_0, can be seen as a **random variable** X which has some statistical properties, such as a **mean** (average value of the outcomes of X) and a **variance** (indicates the variance of the outcomes of X around their average value). The outcome of this random variable X at time t = t_0 is equal to x(t_0).

If X can take an infinite number of values, X is said to be **continuous**. If X can only take a limited number of values, X is said to be **discrete**.

• Discrete random variables

Consider a random variable X that can take m values a_i with probabilities p_i. We must have:

$$\sum_{i=1}^{m} p_i = 1$$

If we plot the probability p_i as a function of values a_i, we obtain a discrete distribution (histogram, for example).

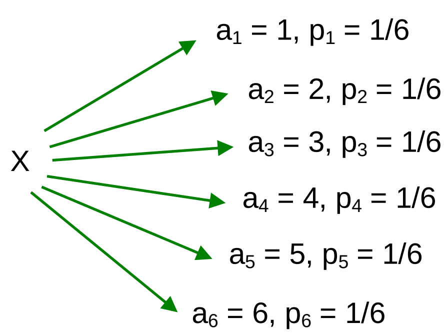
The mean m of X, also called expectancy E{X} of X, is given by

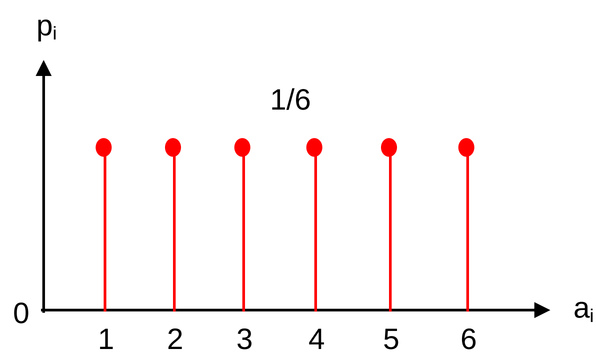
$$m = E\{X\} = \sum_{i=1}^{m} a_i \cdot p_i \,.$$

The variance σ^2 of X is given by

$$\sigma^2 = E\{X^2\} - \left(E\{X\}\right)^2 = \left[\sum_{i=1}^{m} (a_i)^2 \cdot p_i\right] - m^2$$

Example: cast of a dice ---





The distribution of X is uniform. Show that its mean m is equal to 3.5, and its variance σ^2 is approximately equal to 2.92.

• Continuous random variables

Since a continuous random variable X can take an infinite number of values, we need to use a mathematical function to specify the distribution of X. This function is called the probability density function (PDF), $P_X(x)$, of X.

The mean m of X is then given by

$$m = E\{X\} = \int_{-\infty}^{+\infty} x \cdot P_X(x)dx\ ,$$

while its variance σ^2 is expressed as

$$\sigma^2 = E\{X^2\} - (E\{X\})^2 = \left[\int_{-\infty}^{+\infty} x^2 \cdot P_X(x)dx\right] - m^2 .$$

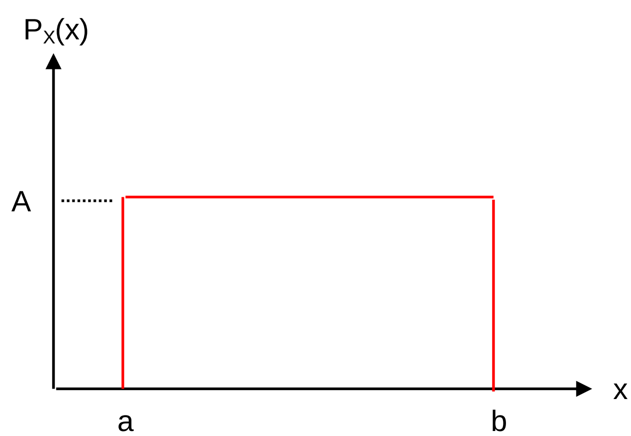
A very important property of PDFs is:

$$\Pr\{\ A < x(t_0) < B\ \} = \int_A^B P_X(x)\, dx\ ,$$

which implies that we must have:

$$\int_{-\infty}^{+\infty} P_X(x)\, dx = 1 .$$

Example: uniform PDF --



Show that:

(1) The amplitude A should be equal to $\frac{1}{b-a}$;

(2) The mean of the random variable X is $m = \frac{a+b}{2}$;

(3) The variance of X is given by $\sigma^2 = \frac{(b-a)^2}{12}$;

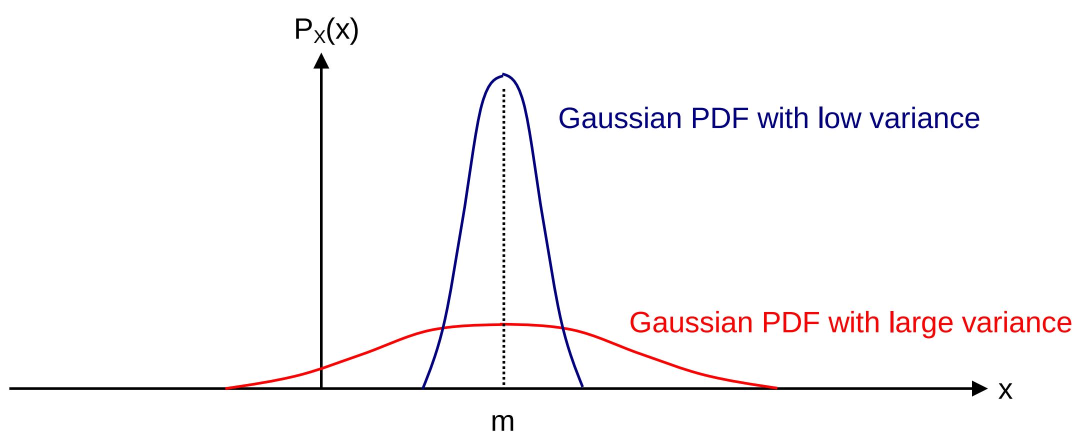
(4) We can write $\Pr\{ a < x(t_0) < c \} = \frac{c-a}{b-a}$ for c < b.

Example: Gaussian random variable --

A random variable X is said to be Gaussian if it has a Gaussian PDF given by

$$P_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left\{ -\frac{(x-m)^2}{2\sigma^2} \right\},$$

where m is the mean of X and σ^2 is its variance.



----

5. Power Spectral Density and Autocorrelation Function of Random Processes

A random process is a non-periodic infinite-energy “signal”. Hence, both its Fourier series expansion and Fourier transform cannot be evaluated. The spectral characteristic of a random process is obtained by computing its **power spectral density** (PSD), $\Phi(f)$, which represents the distribution of power with frequency.

The PSD of a given process can be used to evaluate its power P in a frequency band ranging from f_1 to f_2 by computing the area under $\Phi(f)$:

$$P = \int_{f_1}^{f_2} \Phi(f)\, df \,.$$

The total power of the process is thus given by $P_{total} = \int_{-\infty}^{+\infty} \Phi(f)\, df$.

The PSD $\Phi(f)$ is actually the Fourier transform of the **autocorrelation function** $\Gamma(t)$ of the random process x(t) defined as follows:

$$\Gamma(t) = E_\tau\{\, x(\tau) \cdot x(\tau - t) \,\}.$$

We can therefore write:

$$\Phi(f) = \int_{-\infty}^{+\infty} \Gamma(t) \cdot e^{-j2\pi f t}\, dt \,,$$

and

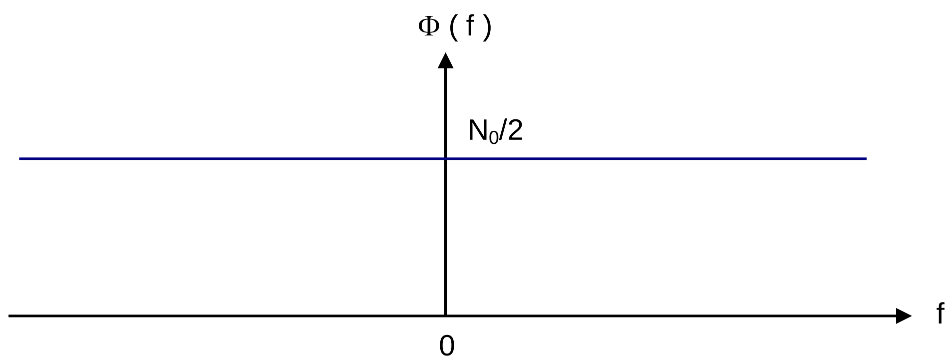
$$\Gamma(t) = \int_{-\infty}^{+\infty} \Phi(f) \cdot e^{+j2\pi f t}\, df \,.$$

The autocorrelation function informs us about the degree of correlation between successive samples of the random process.

Note that $P_{total} = \int_{-\infty}^{+\infty} \Phi(f)\, df = \Gamma(t=0) = E_\tau\left\{ [\, x(\tau)\,]^2 \right\}$. What does such result mean? And what does such result imply for a white random process?

Example --

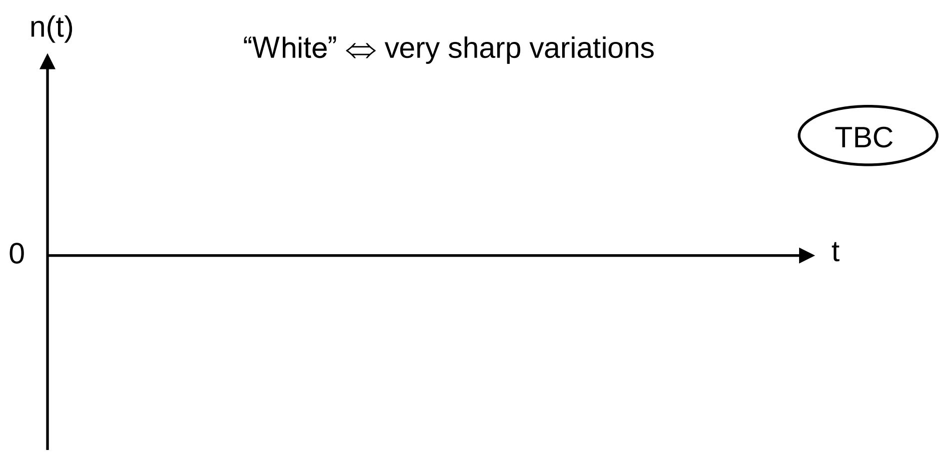
A **white noise** process n(t) is defined to have a flat PSD over the entire frequency range. This model is, however, unrealistic since it would mean that such signal has an infinite power P.



Since we have $\Phi(f) = \frac{N_0}{2}$, the autocorrelation function of the random process n(t) is $\Gamma(t) = \frac{N_0}{2} \cdot \int_{-\infty}^{+\infty} e^{+j2\pi f t}\, df = \frac{N_0}{2} \cdot \delta(t)$, where $\delta(t)$ is a function called the unit impulse and defined as equal to 0 for any $t \neq 0$ and having unit area, i.e. $\int_{-\infty}^{+\infty} \delta(t)\, dt = 1$.

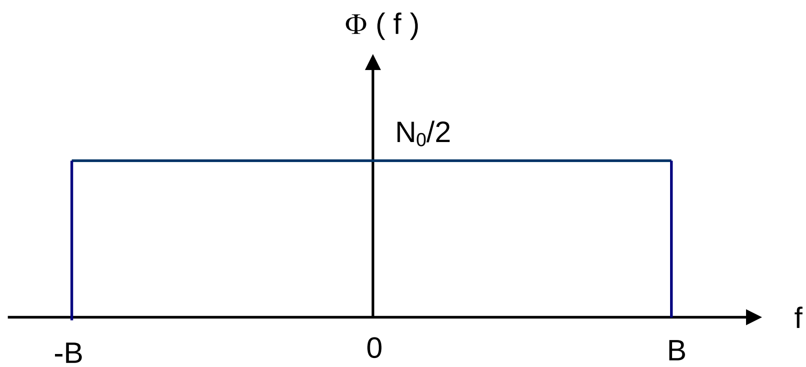
This result implies that two white noise successive samples are never correlated, no matter how close they are (unrealistic once again).

In practice, a noise process can be considered as white if its PSD is constant in the bandwidth of interest.



As a practical noise process cannot be white strictly speaking, there is always in practice some correlation between successive time-domain samples. For instance, if the PSD of the noise process is flat inside a [-B, B] frequency range and equal to zero elsewhere, the autocorrelation function is given by

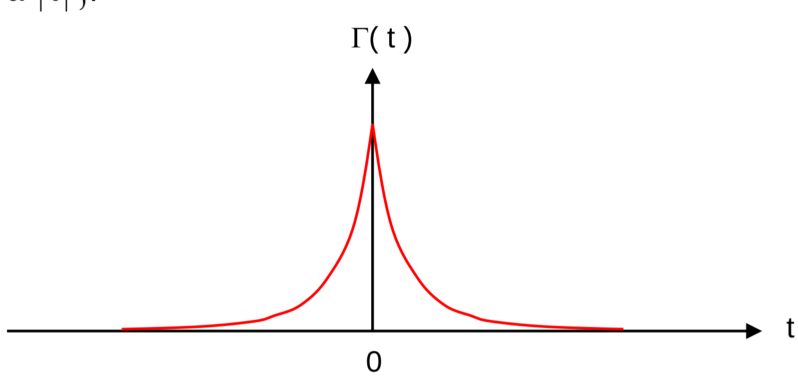
$$\Gamma_n(t) = N_0 \cdot B \cdot \mathrm{sinc}(2\pi B t).$$



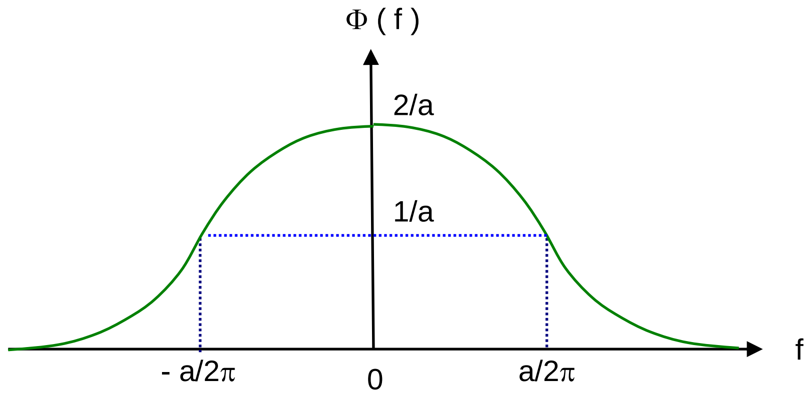
In practice, the white noise is often **Gaussian.** However, it is important to note that white and Gaussian are two adjectives that have nothing to do with each other. As an example, a noise process can be Gaussian without being white, and vice-versa. For instance, if a white Gaussian noise process is filtered, the output of the filter will be another noise process which is still Gaussian, but no longer white.

Example --

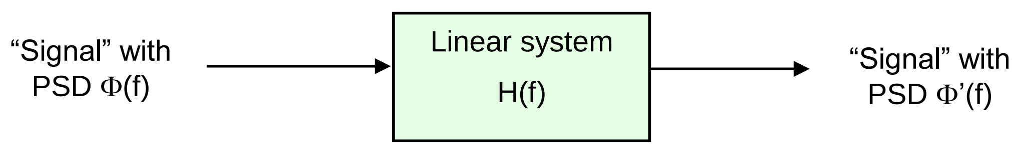
Assume $\Gamma(t) = \exp\{-a \cdot |t|\}$.



Show that $\Phi(f) = \dfrac{2 \cdot a}{a^2 + (2\pi f)^2}$.



Finally, we can also have a quick look at how a PSD is modified by a linear system characterized by its transfer function H(f).



The output PSD is obtained by applying:

$$\Phi'(f) = \Phi(f) \cdot |H(f)|^2 .$$

6. Transmission of Signals over Wireless Channels

In telecommunications, the signal to be transmitted is generally a low-frequency signal, called the **message**, which contains all frequencies from 0 to B, where B is the **bandwidth**. The direct wireless transmission of such signal would require the use of a very long antenna. In fact, a basic principle of wireless communications is that the minimum length, L_{min}, of the antenna is inversely proportional to the signal frequency:

$$L_{min} \approx \frac{c}{10f},$$

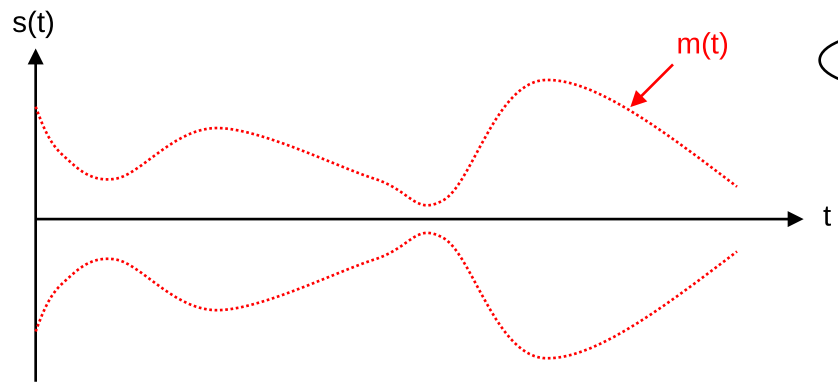
where c denotes the speed of light ($c = 3\times10^8$ m/s) and f is the electromagnetic (EM) wave frequency.

To be able to use an antenna of reasonable length, it is therefore necessary to increase the frequency of the message in an artificial way ⇒ **Modulation** Technique.

The modulation technique simply consists of mixing the message with a high-frequency sine wave called the **carrier**. The resulting signal is called **modulated signal**.

- **Time-domain representation (example of amplitude modulation)**

The amplitude modulation operation consists of multiplying the message m(t) with the carrier $\cos(2\pi f_0 t)$. The expression of the modulated signal is then $s(t) = m(t).\cos(2\pi f_0 t)$.



TBC

• **Frequency-domain representation (example of amplitude modulation)**

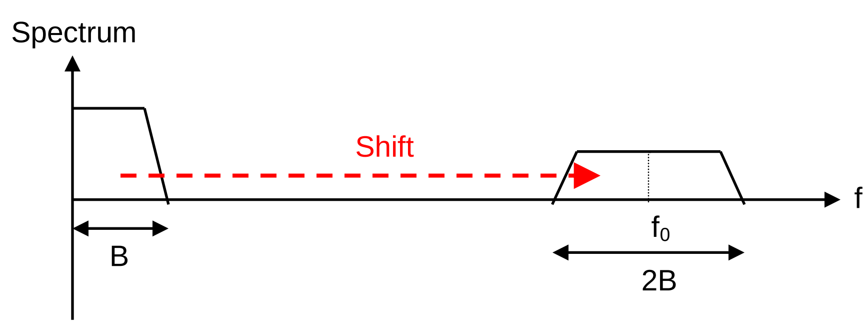
In the frequency domain, the effect of amplitude modulation is to shift the power spectral density, i.e. spectrum, M(f) of the message from the frequency 0 to the frequency f_0.

To demonstrate this, consider a particular frequency component f_1 in M(f), i.e. a particular signal $\cos(2\pi f_1 t)$ among all those constituting the message m(t). When mixed with the carrier $\cos(2\pi f_0 t)$, the component $\cos(2\pi f_1 t)$ becomes:

$$\cos(2\pi f_1 t)\cdot\cos(2\pi f_0 t) = \frac{1}{2}\left[\cos[2\pi(f_0 - f_1)t] + \cos[2\pi(f_0 + f_1)t]\right].$$

This result indicates that the frequency component f_1 is shifted to frequencies $(f_0 - f_1)$ and $(f_0 + f_1)$, but its amplitude has been halved.

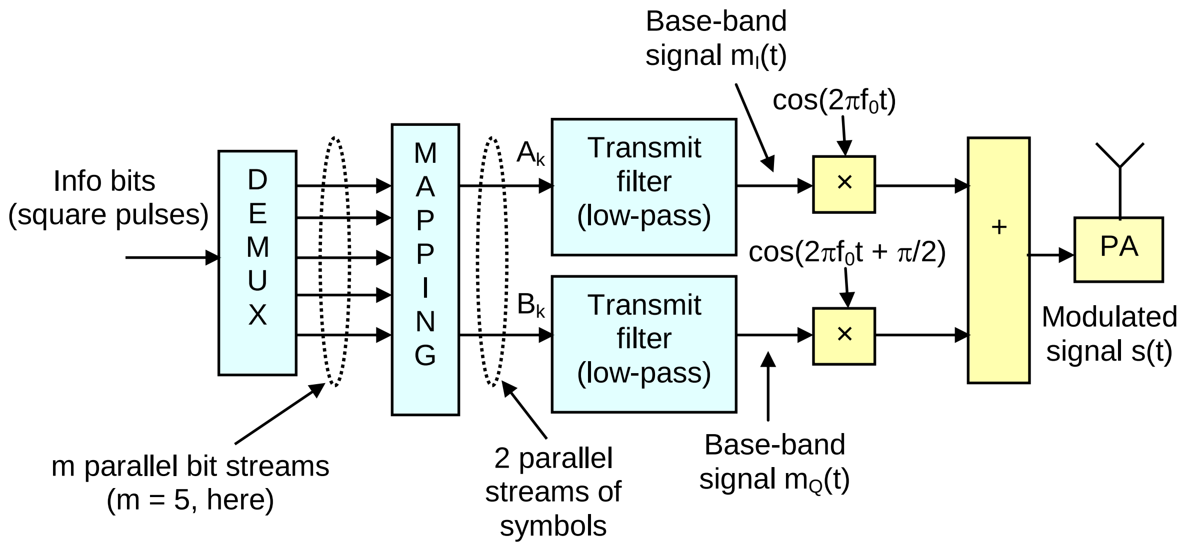
If we apply this simple calculation to all frequency components of M(f), we can easily obtain the spectrum S(f) of the modulated signal. After modulation, the energy of the signal is thus located around the carrier frequency f_0. The spectrum is symmetrical around the carrier frequency. We remark that the bandwidth occupied by the modulated signal is twice that occupied by the message.



In addition to making the transmission possible using an antenna of reasonable size, the modulation operation allows for an equitable sharing of the spectrum by all users according to the international regulations regarding bandwidth allocations (user 1: $0 \rightarrow f_{01}$, user 2: $0 \rightarrow f_{02}$, user 3: $0 \rightarrow f_{03}$, user 4: $0 \rightarrow f_{04}$…).

7. Digital Wireless Communication Transmitters

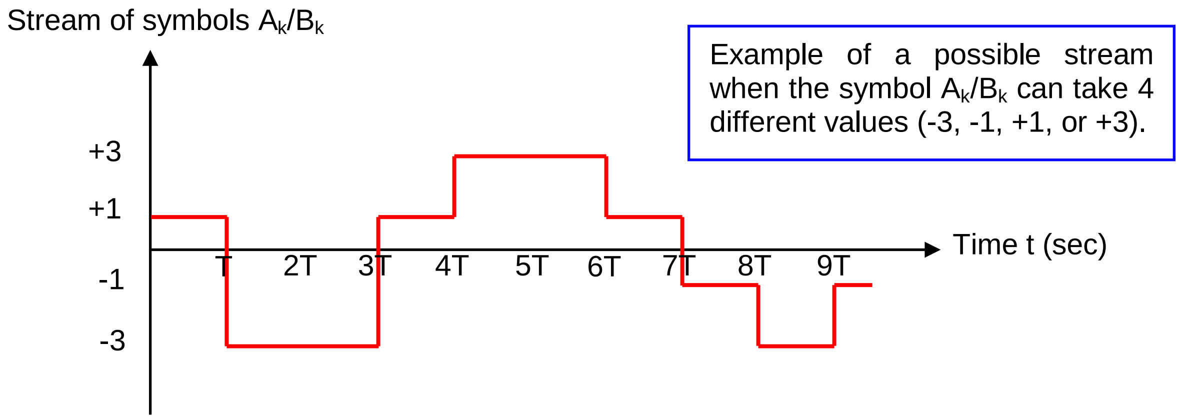
We have represented below the generic structure of a digital wireless communication transmitter.



The sequence of information bits to be transmitted is de-multiplexed into m parallel bit streams. Let $(x_{1,k}, x_{2,k}, \ldots, x_{m,k})$ denote the vector of m parallel bits at time kT, where k is an integer and T is the duration of a symbol.

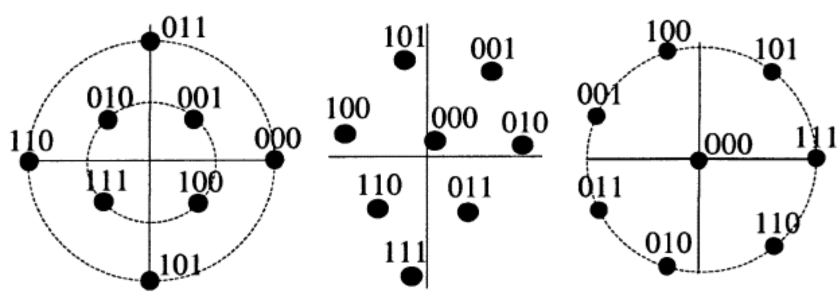
The mapping operation converts this m-bit vector $(x_{1,k}, x_{2,k}, \ldots, x_{m,k})$ into a pair of real symbols denoted as A_k and B_k, or equivalently a complex symbol $C_k = A_k + jB_k$. The complex symbol C_k can take $M = 2^m$ different values.

At the mapping block output, we thus have two streams composed at time kT of symbols A_k and B_k that can be either binary or non-binary (depending on the value of m).

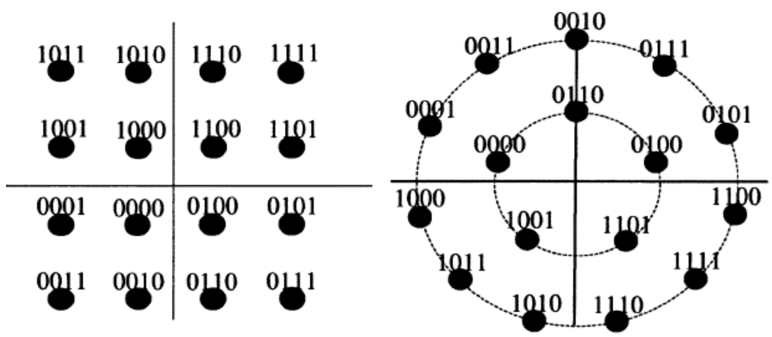


To completely describe the operation of the mapping function, it is necessary to specify the possible values of C_k as well as the one-to-one correspondence between a m-bit vector ($x_{1,k}$, $x_{2,k}$, …, $x_{m,k}$) and a M-ary complex symbol C_k → **Signal constellation**.

Example: M = 8 (m = 3) --



Example: M = 16 (m = 4) --



Filtering is mandatory within the transmitter to constrain the bandwidth of the signal to that dictated by regulation or by the practical necessity to co-habit with other users on adjacent channel frequencies. This implies, in particular, that we cannot use simple square pulses to transmit symbols A_k and B_k because, as previously seen, square pulses have an infinite bandwidth.

At the output of both transmit filters, the expressions of the signals are:

$$m_I(t) = \sum_{k=-\infty}^{+\infty} A_k \cdot h(t - kT),$$

and

$$m_Q(t) = \sum_{k=-\infty}^{+\infty} B_k \cdot h(t - kT).$$

This can be written in a more compact way as a single complex signal m(t) given by:

$$m(t) = \sum_{k=-\infty}^{+\infty} C_k \cdot h(t - kT),$$

where C_k is the M-ary symbol transmitted at time kT and h(t) is the pulse shape. The signal m(t) is said to be **base-band** since its spectrum is centered on the zero frequency.

The **symbol rate** D_s is given by $D_s = \frac{1}{T}$ symbols/sec. However, what really matters is not the symbol rate, but the **bit rate** D_b given by $D_b = \frac{m}{T}$ bits/sec.

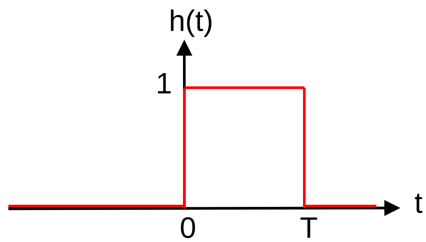
It is interesting to determine the PSD of the base-band signal. In the context of these lecture notes, we can use the following expression:

$$M(f) = K \cdot |H(f)|^2,$$

where K is a constant and H(f) is the Fourier transform of the pulse shape h(t). This equation indicates that the bandwidth B_{bb} of the base-band signal is equal to that of H(f).

Example ---

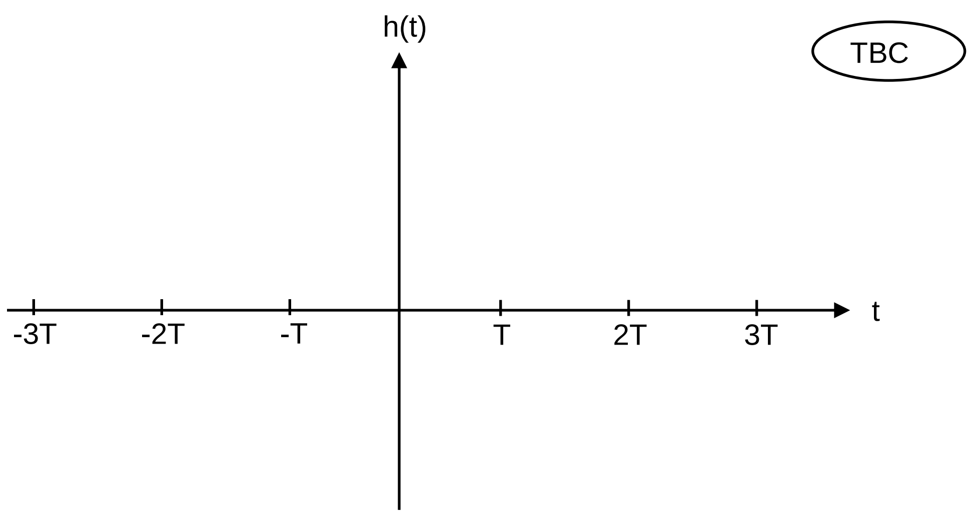
The pulse h(t) is a square function (implying that no transmit filter has actually been employed).



Show that the PSD of m(t) is centered on the zero frequency and the bandwidth of this signal is infinite.

Example ---

The pulse h(t) is a "sinc" function, i.e. $h(t) = \mathrm{sinc}\left(\frac{\pi t}{T}\right)$.



Show that the PSD of m(t) is centered on the zero frequency and the bandwidth of this signal is given by

$$B_{bb} = \frac{1}{2T}.$$

From these examples, we can see that the bandwidth B_{bb} of the base-band signal depends on the shape of the pulse (the smoother the pulse, the smaller the bandwidth). In addition, B_{bb} is inversely proportional to the symbol duration T.

=> The bandwidth B_{bb} is proportional to the symbol rate D_s and the data rate D_b.

Think about the fashionable term **broadband**.

With the kind of pulse shapes that are used in practical systems, the bandwidth B_{bb} required for the transmission of m(t) is:

$$B_{bb} = \frac{1+\alpha}{2T},$$

where α is a fixed system parameter, called roll-off factor, ranging from 0 to 1. This crucial result will be demonstrated later.

We can also write:

$$B_{bb} = \frac{(1+\alpha)\cdot D_b}{2m}.$$

In other words, the amount of bandwidth required for transmitting the base-band signal m(t) is proportional to the bit rate.

The equation of the modulated signal s(t) is finally given by

$$s(t) = \sum_{k=-\infty}^{+\infty} A_k \cdot h(t-kT)\cdot \cos(2\pi f_0 t) + B_k \cdot h(t-kT)\cdot \cos\left(2\pi f_0 t + \frac{\pi}{2}\right),$$

where f_0 designates the carrier frequency, $\cos(2\pi f_0 t)$ is the in-phase (I) carrier, and $\cos(2\pi f_0 t + \pi/2)$ is the quadrature (Q) carrier.

This equation can be written in several different ways which are as follows:

$$s(t) = m_I(t)\cdot \cos(2\pi f_0 t) + m_Q(t)\cdot \cos\left(2\pi f_0 t + \frac{\pi}{2}\right),$$

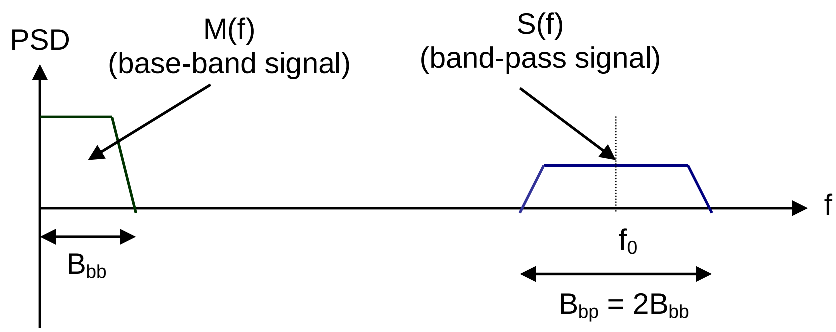
$$s(t) = m_I(t)\cdot \cos(2\pi f_0 t) - m_Q(t)\cdot \sin(2\pi f_0 t),$$

$$s(t) = \mathrm{Re}\{m(t)\cdot \exp\{j2\pi f_0 t\}\},$$

and $$s(t) = \mathrm{Re}\left\{ \sum_{k=-\infty}^{+\infty} C_k \cdot h(t - kT) \cdot \exp\{j2\pi f_0 t\} \right\},$$

where $\exp\{j2\pi f_0 t\}$ designates the complex carrier.

As previously seen, the mixing operation shifts the spectrum of m(t) from the frequency 0 Hz to the frequency f_0. The signal s(t) is also called the **band-pass signal**.



The bandwidth B_{bp} occupied by the band-pass signal is twice that occupied by the base-band signal. With the transmit filters used in practice, the bandwidth needed for the transmission of the modulated signal is therefore equal to $B_{bp} = 2B_{bb} = \dfrac{(1+\alpha) \cdot D_b}{m}$.

At this stage, we need to introduce an important parameter called **spectral efficiency** and defined as

$$\eta = \frac{D_b}{B_{bp}}.$$

The value of η (expressed in bits/sec/Hz) measures the bit rate that can be transmitted per Hz of bandwidth. A higher value of η indicates that the transmission system can transmit a higher bit rate while keeping the required bandwidth unchanged, which is obviously a good thing. We can easily show that, in practice, the spectral efficiency is simply given by

$$\eta = \frac{m}{1+\alpha}.$$

This equation indicates that η depends on both parameters m and α. Generally, communication systems engineers assume that the system is designed with $\alpha = 0$ (the most optimistic and unrealistic scenario, as we will later see) and thus consider that η = m bits/sec/Hz.

We are now going to introduce three different types of digital modulation techniques that are commonly used in practice.

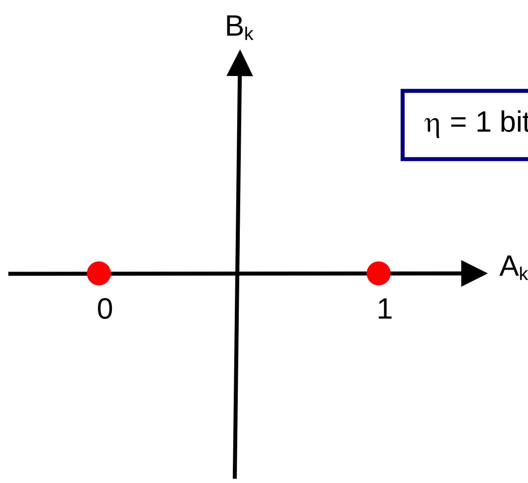
- **M-state amplitude shift keying (M-ASK) modulation**

For such modulation schemes, the symbol C_k is a real symbol that can take M possible values: $C_k = A_k \in \{\pm 1, \pm 3, \pm 5, \ldots\}$. Therefore, we can write:

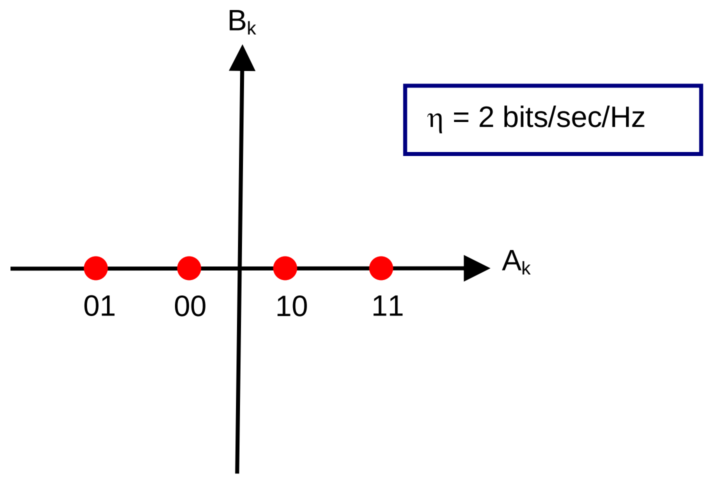
$$s(t) = \sum_{k=-\infty}^{+\infty} A_k \cdot h(t - kT) \cdot \cos\{2\pi f_0 t\},$$

which corresponds to a classical amplitude modulation for which the message is a digital base-band signal.

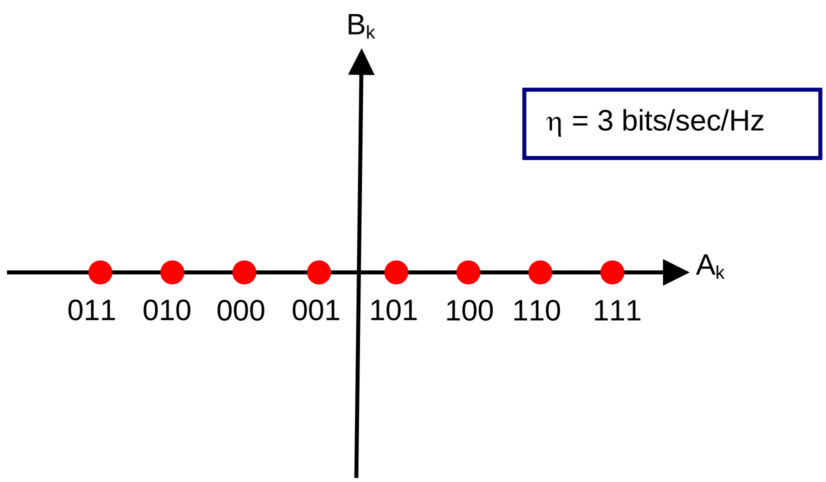
Constellation of 2-ASK:



Constellation of 4-ASK (with Gray mapping):



Constellation of 8-ASK (with Gray mapping):



• **M-state phase shift keying (M-PSK) modulation**

For such modulation schemes, the complex symbol C_k has a constant modulus, i.e. $C_k = \exp\{j\Phi_k\}$ with $\Phi_k \in \{0, 2\pi/M, 4\pi/M, 6\pi/M, \ldots, 2(M-1)\pi/M\}$. Therefore, we can write:

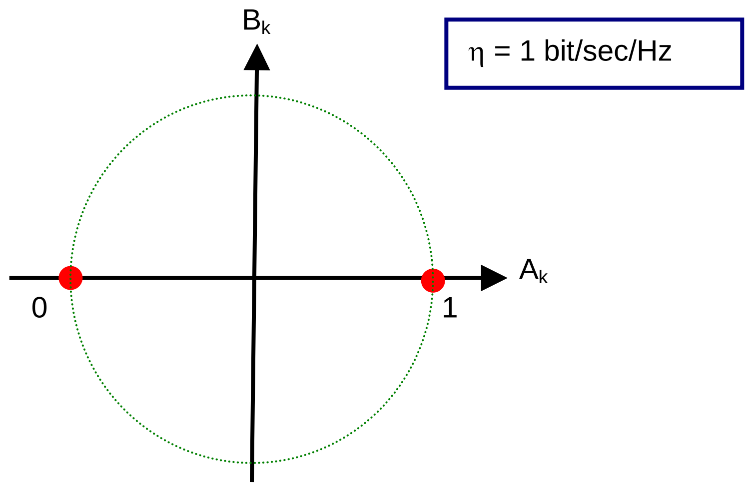
$$s(t) = \mathrm{Re}\left\{\sum_{k=-\infty}^{+\infty} h(t-kT)\cdot \exp\{j2\pi f_0 t + \Phi_k\}\right\},$$

or, equivalently,

$$s(t) = \sum_{k=-\infty}^{+\infty} h(t-kT)\cdot \cos(2\pi f_0 t + \Phi_k),$$

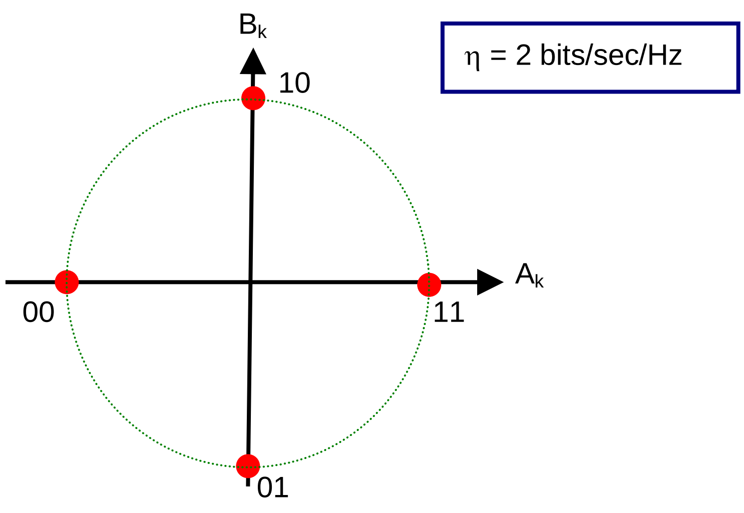
which corresponds to a classical phase modulation for which the phase of the carrier can only take discrete values.

Constellation of 2-PSK:



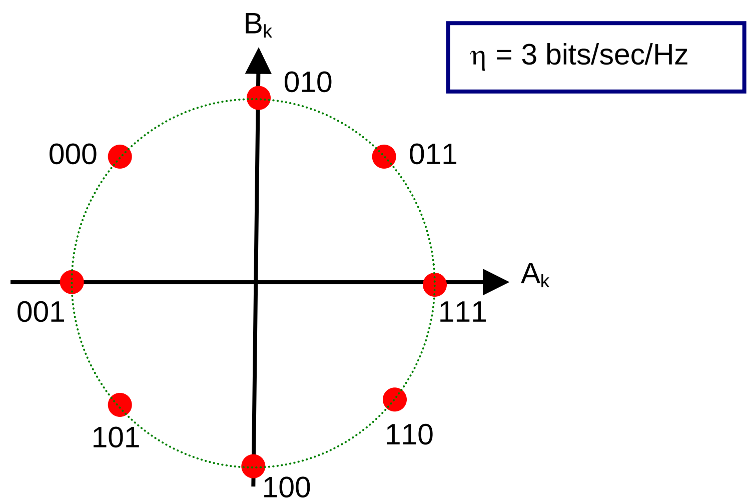
2-PSK is also known as binary PSK (BPSK). It is identical to 2-ASK.

Constellation of 4-PSK (with Gray mapping):



4-PSK is also known as quaternary PSK (QPSK).

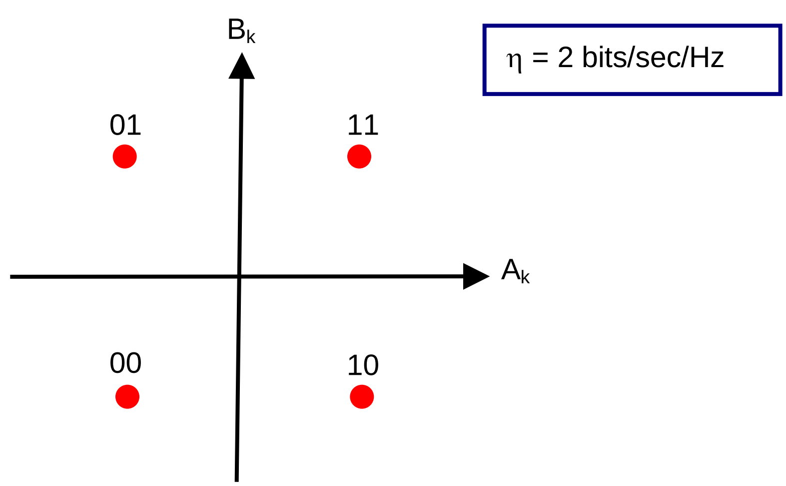
Constellation of 8-PSK (with Gray mapping):



• **M-state quadrature amplitude modulation (M-QAM)**

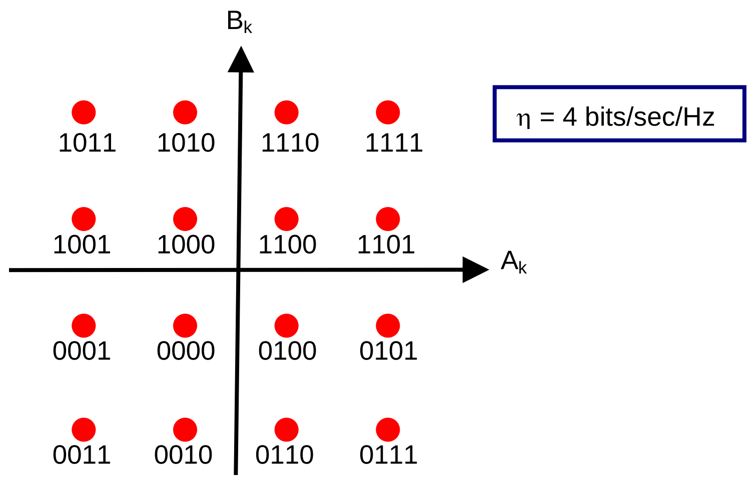
For such modulation schemes, the symbol $C_k = A_k + jB_k$ is a complex symbol that can take M possible values, with A_k and $B_k \in \{\pm1, \pm3, \pm5, \ldots\}$.

Constellation of 4-QAM (with Gray mapping):

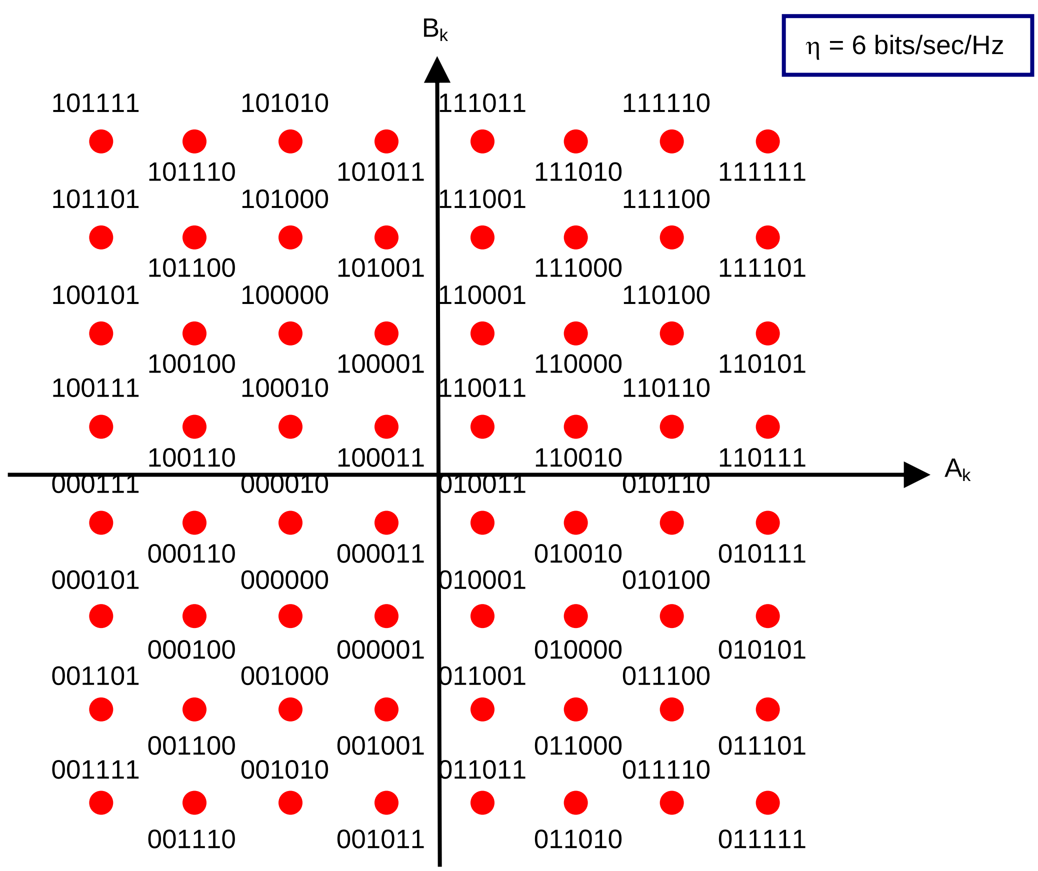


4-QAM is identical to QPSK.

Constellation of 16-QAM (with Gray mapping):

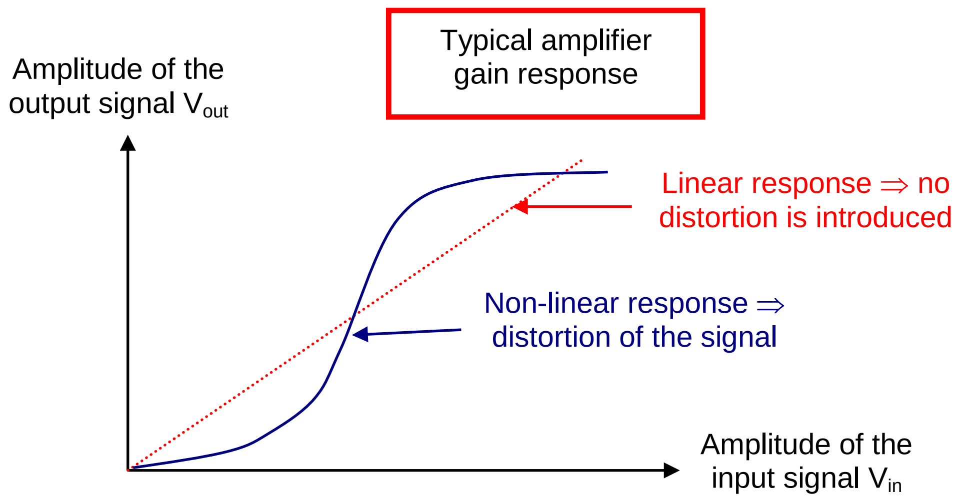


Constellation of 64-QAM (with Gray mapping):



The power of the modulated signal needs to be amplified so as to generate an EM wave that has sufficient power to reach the receiver. Ideally, the RF power amplifier should be as linear as possible in order to introduce no significant distortion on the RF signal.

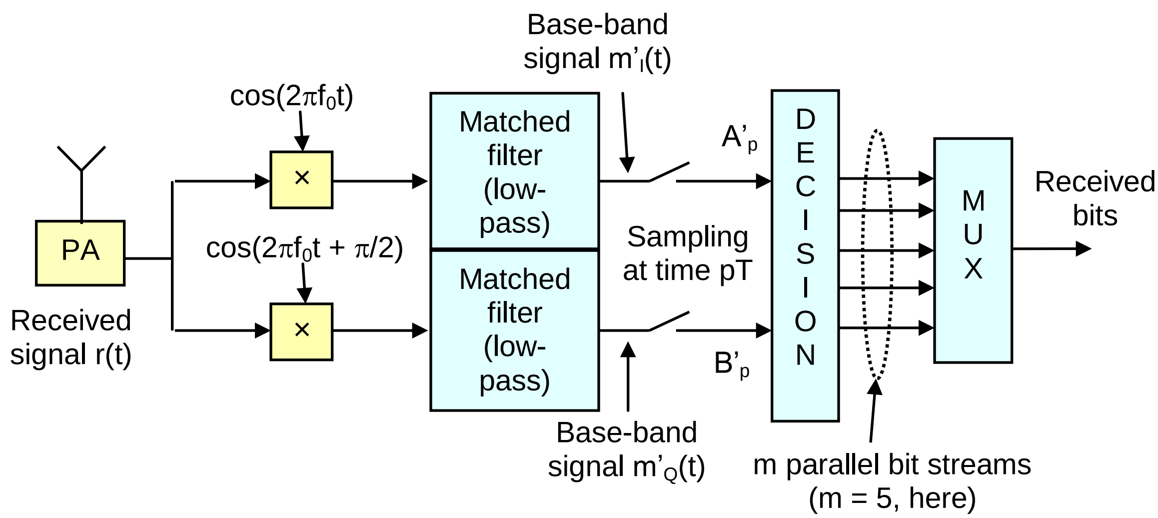
On the other hand, in portable devices operating from batteries (cellular phone handsets for example), the amount a power wasted as heat in the RF amplifier should be minimized. In other words, the RF amplifier should have a power-efficiency close to 100%, meaning that most of the power drawn from the battery should be delivered to the load. Unfortunately, the most power-efficient amplifiers tend to be highly non-linear.



Finally, the amplifier output is connected to the antenna whose task is to radiate the RF signal as EM waves into free space.

8. Digital Wireless Communication Receivers

We have represented below the generic structure of a digital wireless communication receiver.



The received signal must be amplified since the power received by the antenna is usually quite low. A significant amount of noise is added to the received signal at this stage. This noise originates from the movement of electrons inside the connections and components of the RF power amplifier.

Such noise is modeled as "**additive white Gaussian noise**" (AWGN). Its temporal and spectral characteristics were explained earlier. The noise power strongly affects the performance of the system in terms of error probability at the receiver output.

In radio channels, the AWGN is not the only disturbance that affects the transmitted signal s(t). In fact, many wireless channels are subject to interference that is even more detrimental to communications integrity than the AWGN:

(1) Interference among different users → **Multi-user interference, co-channel interference.**

(2) Interference among several (randomly-attenuated and randomly-delayed) versions of the same signal → **Fading channels.**

(3) Interference among successive symbols carried by the same signal → **Inter-symbol interference.**

Throughout this chapter, we will assume that the channel is an AWGN channel, i.e. the received signal r(t) is given by

$$r(t) = s(t) + n(t),$$

where s(t) is the transmitted band-pass signal and n(t) is a zero-mean Gaussian random process with constant power spectral density (PSD), i.e. $\Phi_n(f) = \frac{N_0}{2}$ for all frequency values.

We are first going to focus on the in-phase component of the receiver, knowing that the equations for the quadrature component can be obtained in a similar manner.

To obtain the base-band signal $m'_I(t)$, which is an estimate of the transmitted base-band signal $m_I(t)$, one has first to demodulate the received signal r(t). Demodulation is performed by mixing r(t) with a locally-generated carrier signal $\cos(2\pi f_0 t)$.

The expression of the received signal can be written as follows:

$$r(t) = \sum_{k=-\infty}^{+\infty} A_k \cdot h(t-kT) \cdot \cos(2\pi f_0 t) - B_k \cdot h(t-kT) \cdot \sin(2\pi f_0 t) \quad + \quad n(t).$$

The signal at the mixer output is then expressed as:

$$r(t) \cdot \cos(2\pi f_0 t) = \sum_{k=-\infty}^{+\infty} A_k \cdot h(t-kT) \cdot \cos^2(2\pi f_0 t) - B_k \cdot h(t-kT) \cdot \sin(2\pi f_0 t) \cdot \cos(2\pi f_0 t) \quad + \quad n'(t).$$

We know that

$$\cos^2(2\pi f_0 t) = \frac{1}{2} \cdot [1 + \cos(4\pi f_0 t)],$$

and

$$\cos(2\pi f_0 t) \cdot \sin(2\pi f_0 t) = \frac{1}{2} \cdot \sin(4\pi f_0 t).$$

Therefore, we can write

$$r(t)\cdot\cos(2\pi f_0 t) = \sum_{k=-\infty}^{+\infty}\frac{A_k}{2}\cdot h(t-kT) + \sum_{k=-\infty}^{+\infty}\frac{A_k}{2}\cdot h(t-kT)\cdot\cos(4\pi f_0 t) - \quad \ldots$$

$$\ldots \quad -\sum_{k=-\infty}^{+\infty}\frac{B_k}{2}\cdot h(t-kT)\cdot\sin(4\pi f_0 t) \quad + \quad n'(t).$$

We need to find the characteristics of the noise $n'(t) = n(t).\cos(2\pi f_0 t)$ present at the mixer output.

We recall that n(t) is white and Gaussian with a mean equal to 0. Therefore, n'(t) is also Gaussian with mean

$$m = E\{\, n'(t)\,\} = E\{\, n(t)\,\}\cdot E\{\, \cos(2\pi f_0 t)\,\} = 0.$$

The autocorrelation function of this noise process n'(t) is computed as follows:

$$\Gamma_{n'}(t) = E_\tau\{\, n'(\tau)\cdot n'(\tau - t)\,\} = E_\tau\{\, n(\tau)\cdot n(\tau - t)\,\}\cdot E_\tau\{\, \cos(2\pi f_0 \tau)\cdot\cos(2\pi f_0 [\tau - t])\,\}$$

=> $$\Gamma_{n'}(t) = \frac{1}{2}\cdot\Gamma_n(t)\cdot\left[E_\tau\{\, \cos(2\pi f_0 t)\,\} + E_\tau\{\, \cos(2\pi f_0 [2\tau - t])\,\}\right] = \frac{1}{2}\cdot\Gamma_n(t)\cdot\cos(2\pi f_0 t)$$

=> $$\Gamma_{n'}(t) = \frac{N_0}{4}\cdot\delta(t).$$

This result simply means that n'(t) is also a white random process.

The signal at the mixer output goes through a low-pass filter with an impulse response q(t) and a transfer function Q(f). At the filter output, we obtain the base-band signal $m'_I(t)$ given by

$$m'_I(t) = \sum_{k=-\infty}^{+\infty}\frac{A_k}{2}\cdot h(t-kT) * q(t) \quad + \quad n'(t) * q(t).$$

Let us focus on the first term in this expression. It can easily be shown that the Fourier transform of the term $h(t-kT) * q(t)$ is

$$H(f)\cdot\exp\{-j2\pi f k T\}\cdot Q(f) = H(f)\cdot Q(f)\cdot\exp\{-j2\pi f k T\} = G(f)\cdot\exp\{-j2\pi f k T\}.$$

The result is actually the Fourier transform of a term $g(t-kT)$, with $g(t)=h(t)*q(t)$.

We then obtain:
$$m'_I(t)=\sum_{k=-\infty}^{+\infty}\frac{A_k}{2}\cdot g(t-kT) \quad + \quad n'(t)*q(t).$$

In order to maximize the signal-to-noise ratio (SNR) at each time pT, p being an integer, we could show that the impulse response $q(t)$ is actually matched to the pulse shape. In other words, the impulse response $q(t)$ is chosen so that $q(t)=h(-t)$. The use of a **matched filter** at the receiver side guarantees optimal error performance at the receiver output.

We can thus write:

$$m'_I(t)=\sum_{k=-\infty}^{+\infty}\frac{A_k}{2}\cdot g(t-kT) \quad + \quad n''(t),$$

with $g(t)=h(t)*h(-t)$ and $n''(t)=n'(t)*h(-t)$.

Let us determine the characteristics of the noise n''(t) present at the filter output. First, it is clear that the noise process n''(t) is not white because of the low-pass filtering. However, n''(t) is Gaussian with zero-mean since linear filtering of a zero-mean Gaussian process produces another zero-mean Gaussian process.

We need to evaluate the variance of n''(t). It can be done as follows:

$$\sigma^2=E\left\{[n''(t)]^2\right\}=\Gamma_{n''}(0)=\int_{-\infty}^{+\infty}\Phi_{n''}(f)\,df=\int_{-\infty}^{+\infty}\Phi_{n'}(f)\cdot|Q(f)|^2\,df.$$

Since $q(t)=h(-t)$, we have $Q(f)=H^*(f)$, and thus $|Q(f)|^2=|H(f)|^2$. Hence, we can write:

$$\sigma^2=\frac{N_0}{4}\cdot\int_{-\infty}^{+\infty}|\,H(f)\,|^2\,df.$$

Since $g(t)=h(t)*h(-t)$, we can also write

$$G(f) = H(f) \cdot H^*(f) = |H(f)|^2,$$

which finally yields:
$$\sigma^2 = \frac{N_0}{4} \cdot \int_{-\infty}^{+\infty} G(f)\, df = \frac{N_0}{4} \cdot g(0).$$

The base-band signal $m'_I(t)$ is sampled every pT to obtain an estimate A_p' of the symbol A_p that was transmitted at time pT. At time pT, we have:

$$m'_I(pT) = A_p' = \sum_{k=-\infty}^{+\infty} \frac{A_k}{2} \cdot g((p-k)T) \quad + \quad N_p$$

=>
$$A_p' = \ldots + \frac{A_{p-1}}{2} \cdot g(T) + \frac{A_p}{2} \cdot g(0) + \frac{A_{p+1}}{2} \cdot g(-T) + \ldots \quad + \quad N_p,$$

where N_p is a Gaussian noise sample with a mean equal to 0 and a variance given by $\sigma^2 = \frac{N_0}{4} \cdot g(0)$.

This equation indicates that a sample A_p' may depend not only on the symbol A_p, but also on the other symbols A_k, $k \neq p$, in the sequence. When it happens, we say that there is **inter-symbol interference** (ISI). The presence of ISI usually leads to very severe, and probably unacceptable, degradation of the error performance at the receiver output.

In practice, ISI must thus be avoided at any cost. If ISI cannot be avoided (such as in the case of frequency-selective fading channels), then it has to be cancelled at the receiver side using techniques such as equalization or orthogonal-frequency division-multiplexing (OFDM) modulation.

Here, in the simple case of an AWGN channel, if g(t) is chosen so that $g((p-k)T) = 0$ for any $p \neq k$, then ISI is suppressed, and we obtain:

$$A_p' = \frac{A_p}{2} \cdot g(0) + N_p,$$

which only depends on symbol A_p. The previous condition is termed **Nyquist criterion**. The filters g(t), defined so that $g((p-k)T)=0$ for any $p \neq k$, constitute a family of filters called **raised-cosine filters**. In other words, raised-cosine filters satisfy the Nyquist criterion, and therefore guarantee ISI-free transmission (in the case of the simple AWGN channel).

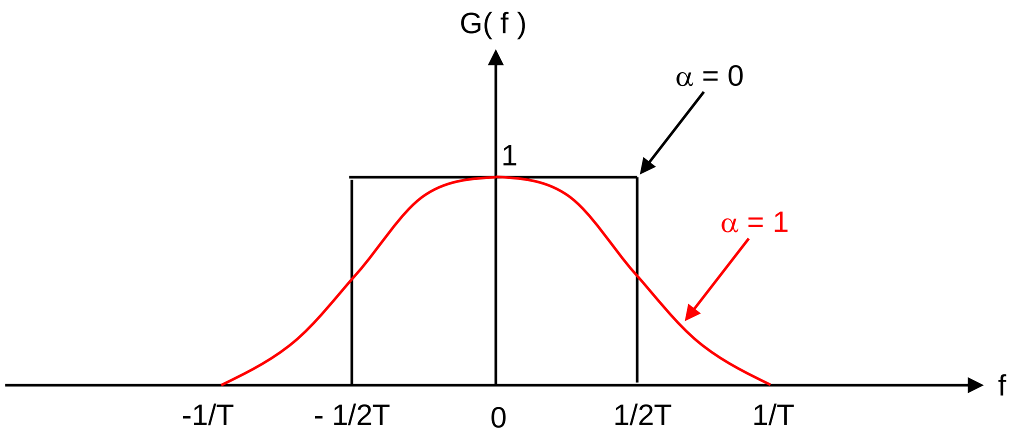
The impulse response of raised cosine filters is in the form:

$$g(t)=\frac{1}{T}\cdot \mathrm{sinc}\left(\frac{\pi t}{T}\right)\cdot\frac{\cos\left(\frac{\pi\alpha t}{T}\right)}{1-\frac{4\alpha^2 t^2}{T^2}},$$

where the parameter α ranges from 0 to 1, and is termed "roll-off factor". This is the system parameter that was introduced earlier. In the frequency-domain, such impulse response corresponds to the following transfer function:

$$G(f)=\begin{cases} 1 & \text{for} \quad f\le\frac{1-\alpha}{2T} \\ \frac{1}{2}\cdot\left[1+\cos\left(\frac{\pi T}{\alpha}\left(f-\frac{1-\alpha}{2T}\right)\right)\right] & \text{for} \quad \frac{1-\alpha}{2T}<f\le\frac{1+\alpha}{2T} \\ 0 & \text{for} \quad f>\frac{1+\alpha}{2T} \end{cases}.$$

This equation shows that the bandwidth of a raised cosine filter is equal to $\frac{1+\alpha}{2T}$ and thus ranges from $\frac{1}{2T}$ (when α = 0) to $\frac{1}{T}$ (when α = 1). We can easily demonstrate that the bandwidth of this raised cosine filter is actually the bandwidth B_{bb} necessary to transmit the base-band signal m(t). To do so, we just have to remember that $G(f)=|H(f)|^2$, i.e. $G(f)=\frac{1}{K}\cdot M(f)$, where $M(f)$ is the PSD of the base-band signal. Since the constant K does not affect the bandwidth, we conclude that the bandwidth B_{bb} of $M(f)$ is equal to the bandwidth of $G(f)$, i.e. $B_{bb}=\frac{1+\alpha}{2T}$.



The case "α = 0" is optimal in terms of bandwidth but not realistic (brick-wall filters are not realistic). The case "α = 1" is the most realistic configuration, but it requires the largest bandwidth.

We conclude that ISI-free transmission requires a bandwidth of, at least, $\frac{1}{2T}$ for base-band signals and $\frac{1}{T}$ for modulated signals.

With such filters, the expression of the sample $A_p^{'}$ is

$$A_p^{'} = \frac{A_p}{2} \cdot g(0) + N_p .$$

If we normalize for simplicity sake, we obtain:

$$A_p^{'} = A_p + N_p ,$$

where N_p is now a Gaussian noise sample with zero mean and a variance given by

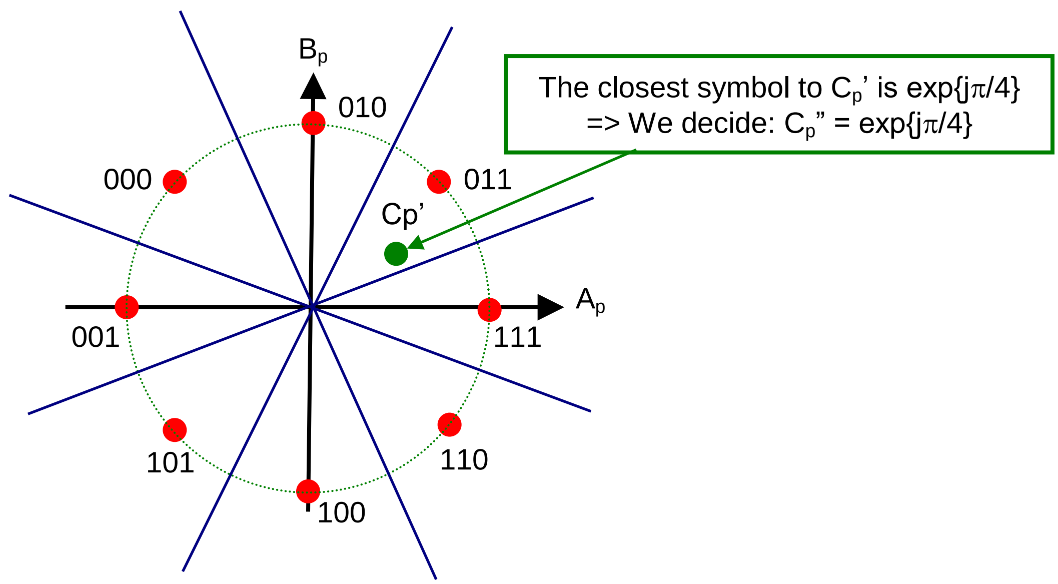
$$\sigma^2 = \frac{N_0}{4} \cdot g(0) \cdot \left(\frac{2}{g(0)} \right)^2 = \frac{N_0}{g(0)} .$$

In a similar way, we could show that the sample B_p' is expressed as

$$B_p^{'} = B_p + N_p^{'},$$

where N_p‘ is a Gaussian noise sample with zero-mean and variance $\sigma^2 = \frac{N_0}{g(0)}$. Note that the noise samples N_p and N_p‘ are independent.

The next task simply consists of taking a decision regarding the value the complex symbol C_p transmitted at time pT. To this end, a decision block uses the available sample $C_p^{'} = A_p^{'} + jB_p^{'}$ and compares it with some thresholds, as illustrated by the example below (8-PSK constellation).



Such process, although optimal in terms of symbol detection, sometimes fails to properly detect the value of the transmitted symbol, especially when the noise samples N_p and N_p‘ have a large magnitude compared to that of the transmitted symbols A_p and B_p.

This simply means that transmission errors occur from time to time. This means that it is now time to introduce the concept of **error probability** in a digital communication system.

9. Error Probability at the Receiver Output for Digital Modulations in AWGN Channels

Let us recall that, over an AWGN channel, the samples available at the decision block input are given by

$$A_p^{'} = A_p + N_p,$$

and

$$B_p^{'} = B_p + N_p^{'},$$

where N_p and N_p' are independent Gaussian noise samples with zero mean and variance $\sigma^2 = \frac{N_0}{g(0)}$.

The reliability of these samples depends on the average magnitude of the noise samples, which is measured by their variance, with respect to the average magnitude of the transmitted symbols A_p and B_p. Hence, the reliability of samples A_p' and B_p' ultimately depends of the signal-to-noise ratio (SNR) over the channel.

The error performance of a digital communications scheme can be assessed by evaluating (via calculations or computer simulations) the **symbol error probability** and/or the **bit error probability** at the receiver output.

Let us first focus our attention on the symbol error probability, P_{es}, defined as the probability that the decision block chooses the wrong symbol, i.e. $P_{es} = \Pr\{C_p'' \neq C_p\}$.

• **Expression of the symbol error probability for any modulation scheme**

Let us drop the time index p (since the time is not relevant throughout the following calculation) and replace it with an index i which indicates a particular symbol in the constellation.

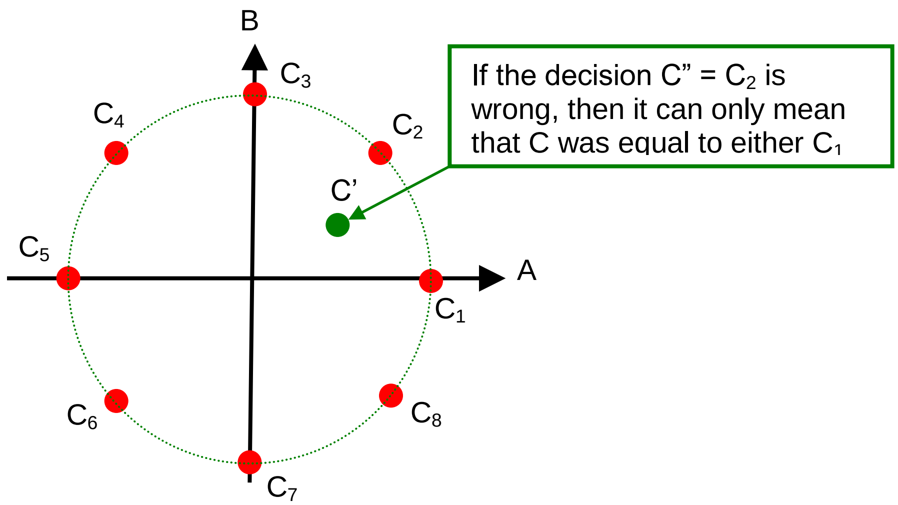
We start by noticing that

$$P_{es} = \Pr\left\{ C_1 \text{ is transmitted and } C_1 \text{ is not detected} \quad \text{OR} \quad C_2 \text{ is transmitted and } C_2 \text{ is not detected} \quad \text{OR} \ldots \right\}.$$

As the M events “C_i is transmitted and C_i is not detected”, $i \in \{1, 2, \ldots, M\}$, are mutually exclusive, the ORs can be written as a sum of probabilities:

$$P_{es} = \sum_{i=1}^{M} \Pr\{ C_i \text{ is transmitted and } C_i \text{ is not detected} \}.$$

At this stage, we are going to use the following assumption which is valid at all SNRs of practical interest: when taking an erroneous decision, the decision block always chooses a symbol that is a nearest neighbor to the symbol actually transmitted.



With such assumption, we can write:

$$P_{es} = \sum_{i=1}^{M} \Pr\Big\{ C_i \text{ is transmitted and } C_{i,1} \text{ is detected} \quad \text{OR} \quad C_i \text{ is transmitted and } C_{i,2} \text{ is detected} \quad \text{OR} \ldots \Big\},$$

where $C_{i,j}$ is the j-th nearest neighbor of C_i.

Since we have once again to consider the OR of events that are mutually exclusive, the previous expression can be written as a sum of probabilities:

$$P_{es} \approx \sum_{i=1}^{M} \sum_{C_j \in S_i} \Pr\Big\{ C_i \text{ is transmitted and } C_j \text{ is detected} \Big\},$$

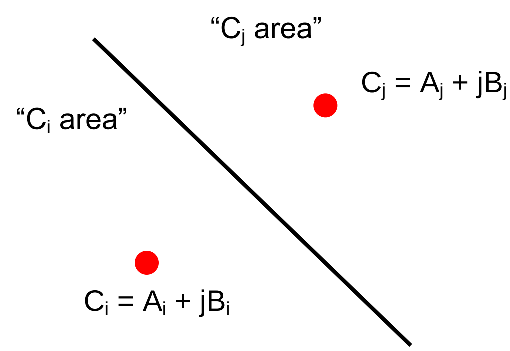
where S_i is the set containing all nearest neighbor symbols of C_i. Using Bayes' rule, we obtain:

$$P_{es} \approx \sum_{i=1}^{M} \sum_{C_j \in S_i} \Pr\left\{ C_i \to C_j \right\} \cdot \Pr\left\{ C_i \text{ is transmitted} \right\},$$

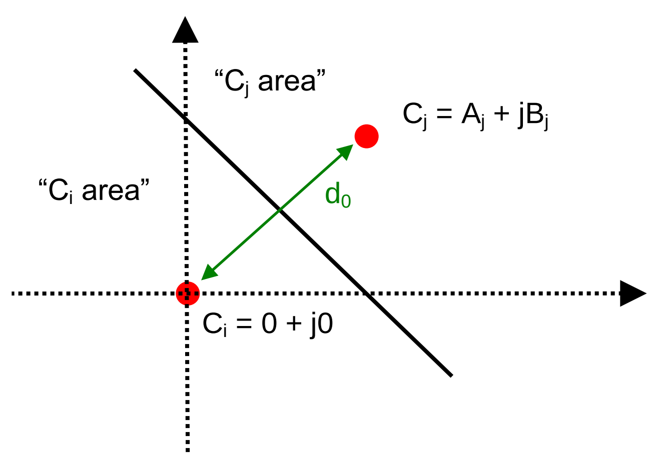
where $\Pr\left\{ C_i \to C_j \right\}$ denotes the probability to detect C_j given that C_i was transmitted. Since all symbols are transmitted with equal probabilities, i.e. $\Pr\left\{ C_i \text{ is transmitted} \right\} = \frac{1}{M}$, we have:

$$P_{es} \approx \frac{1}{M} \cdot \sum_{i=1}^{M} \sum_{C_j \in S_i} \Pr\left\{ C_i \to C_j \right\}.$$

We are now going to find an expression for the term $\Pr\left\{ C_i \to C_j \right\}$.



To simplify the calculations, we can assume without any loss of generality that $A_i = B_i = 0$.



The Euclidean distance between C_i and C_j is given by $d_0 = \sqrt{A_j^2 + B_j^2}$.

Note that d_0 is actually the minimum distance between symbols in the constellation since C_i and C_j are two nearest neighbor symbols.

The equation of the boundary line between the "C_i area" and the "C_j area" is given by

$$y = -\left(\frac{A_j}{B_j}\right) \cdot x + \frac{d_0^2}{2B_j}$$. (Never forget your high school mathematics)

It is therefore easy to see that

$$\Pr\left\{ C_i \to C_j \right\} = \Pr\left\{ B' > -\left(\frac{A_j}{B_j}\right) \cdot A' + \frac{d_0^2}{2B_j} \right\},$$

where C' = A' + jB' is the sample used by the decision block. Since the transmitted symbol was C = 0 + j0, we can see that: C' = N + jN', where N and N' are two independent Gaussian noise samples with zero-mean and variance $\sigma^2 = N_0.T$. Thus, we have:

$$\Pr\left\{ C_i \to C_j \right\} = \Pr\left\{ N' + \left(\frac{A_j}{B_j}\right) \cdot N > \frac{d_0^2}{2B_j} \right\} = \Pr\left\{ N'' > \frac{d_0^2}{2B_j} \right\}.$$

The noise sample $N'' = N' + \left(\frac{A_j}{B_j}\right) \cdot N$ is also a Gaussian sample with a mean given by

$$m = E\{N''\} = E\{N'\} + \left(\frac{A_j}{B_j}\right) \cdot E\{N\} = 0,$$

and a variance equal to

$$E\left\{ (N'')^2 \right\} = E\left\{ (N')^2 \right\} + \left(\frac{A_j}{B_j}\right)^2 \cdot E\left\{ N^2 \right\} + 2 \cdot \left(\frac{A_j}{B_j}\right) \cdot E\left\{ N' \right\} \cdot E\left\{ N \right\}$$

$$= \sigma^2 + \left(\frac{A_j}{B_j}\right)^2 \cdot \sigma^2 = \sigma^2 \cdot \left[1 + \left(\frac{A_j}{B_j}\right)^2\right] = \frac{N_0}{g(0)} \cdot \left[1 + \left(\frac{A_j}{B_j}\right)^2\right] = \frac{N_0 \cdot d_0^2}{g(0) \cdot B_j^2}.$$

At this stage, we need to exploit our knowledge about the Gaussian noise sample N''. Its probability density function (PDF) is given by

$$P_{N''}(x) = \frac{B_j\sqrt{g(0)}}{d_0\sqrt{2\pi N_0}} \cdot \exp\left\{-\frac{B_j^2 g(0)}{2N_0 d_0^2} \cdot x^2\right\}.$$

By applying a basic property of PDFs which says that

$$\Pr\{\, a < N'' < b \,\} = \int_a^b P_{N''}(x)\, dx,$$

it can be easily shown that

$$\Pr\{\, C_i \to C_j \,\} = \frac{B_j\sqrt{g(0)}}{d_0\sqrt{2\pi N_0}} \cdot \int_{\frac{d_0^2}{2B_j}}^{+\infty} \exp\left\{-\frac{B_j^2 g(0)}{2N_0 d_0^2} \cdot x^2\right\} dx.$$

By introducing the well-known **complementary error function**:

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \cdot \int_x^{+\infty} e^{-u^2}\, du,$$

and performing the change of variable $u = x \cdot \frac{B_j\sqrt{g(0)}}{d_0\sqrt{2N_0}}$, we finally obtain the expression:

$$\Pr\{\, C_i \to C_j \,\} = \frac{1}{2} \cdot \mathrm{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right).$$

A table giving the value of erfc(x) as a function of x is shown in the next page.

		Complementary Error Function Table											
x	**erfc(x)**	**x**	**erfc(x)**	**x**	**erfc(x)**	**x**	**erfc(x)**	**x**	**erfc(x)**	**x**	**erfc(x)**	**x**	**erfc(x)**
0	1.000000	0.5	0.479500	1	0.157299	1.5	0.033895	2	0.004678	2.5	0.000407	3	0.00002209
0.01	0.988717	0.51	0.470756	1.01	0.153190	1.51	0.032723	2.01	0.004475	2.51	0.000386	3.01	0.00002074
0.02	0.977435	0.52	0.462101	1.02	0.149162	1.52	0.031587	2.02	0.004281	2.52	0.000365	3.02	0.00001947
0.03	0.966159	0.53	0.453536	1.03	0.145216	1.53	0.030484	2.03	0.004094	2.53	0.000346	3.03	0.00001827
0.04	0.954889	0.54	0.445061	1.04	0.141350	1.54	0.029414	2.04	0.003914	2.54	0.000328	3.04	0.00001714
0.05	0.943628	0.55	0.436677	1.05	0.137564	1.55	0.028377	2.05	0.003742	2.55	0.000311	3.05	0.00001608
0.06	0.932378	0.56	0.428384	1.06	0.133856	1.56	0.027372	2.06	0.003577	2.56	0.000294	3.06	0.00001508
0.07	0.921142	0.57	0.420184	1.07	0.130227	1.57	0.026397	2.07	0.003418	2.57	0.000278	3.07	0.00001414
0.08	0.909922	0.58	0.412077	1.08	0.126674	1.58	0.025453	2.08	0.003266	2.58	0.000264	3.08	0.00001326
0.09	0.898719	0.59	0.404064	1.09	0.123197	1.59	0.024538	2.09	0.003120	2.59	0.000249	3.09	0.00001243
0.1	0.887537	0.6	0.396144	1.1	0.119795	1.6	0.023652	2.1	0.002979	2.6	0.000236	3.1	0.00001165
0.11	0.876377	0.61	0.388319	1.11	0.116467	1.61	0.022793	2.11	0.002845	2.61	0.000223	3.11	0.00001092
0.12	0.865242	0.62	0.380589	1.12	0.113212	1.62	0.021962	2.12	0.002716	2.62	0.000211	3.12	0.00001023
0.13	0.854133	0.63	0.372954	1.13	0.110029	1.63	0.021157	2.13	0.002593	2.63	0.000200	3.13	0.00000958
0.14	0.843053	0.64	0.365414	1.14	0.106918	1.64	0.020378	2.14	0.002475	2.64	0.000189	3.14	0.00000897
0.15	0.832004	0.65	0.357971	1.15	0.103876	1.65	0.019624	2.15	0.002361	2.65	0.000178	3.15	0.00000840
0.16	0.820988	0.66	0.350623	1.16	0.100904	1.66	0.018895	2.16	0.002253	2.66	0.000169	3.16	0.00000786
0.17	0.810008	0.67	0.343372	1.17	0.098000	1.67	0.018190	2.17	0.002149	2.67	0.000159	3.17	0.00000736
0.18	0.799064	0.68	0.336218	1.18	0.095163	1.68	0.017507	2.18	0.002049	2.68	0.000151	3.18	0.00000689
0.19	0.788160	0.69	0.329160	1.19	0.092392	1.69	0.016847	2.19	0.001954	2.69	0.000142	3.19	0.00000644
0.2	0.777297	0.7	0.322199	1.2	0.089686	1.7	0.016210	2.2	0.001863	2.7	0.000134	3.2	0.00000603
0.21	0.766478	0.71	0.315335	1.21	0.087045	1.71	0.015593	2.21	0.001776	2.71	0.000127	3.21	0.00000564
0.22	0.755704	0.72	0.308567	1.22	0.084466	1.72	0.014997	2.22	0.001692	2.72	0.000120	3.22	0.00000527
0.23	0.744977	0.73	0.301896	1.23	0.081950	1.73	0.014422	2.23	0.001612	2.73	0.000113	3.23	0.00000493
0.24	0.734300	0.74	0.295322	1.24	0.079495	1.74	0.013865	2.24	0.001536	2.74	0.000107	3.24	0.00000460
0.25	0.723674	0.75	0.288845	1.25	0.077100	1.75	0.013328	2.25	0.001463	2.75	0.000101	3.25	0.00000430
0.26	0.713100	0.76	0.282463	1.26	0.074764	1.76	0.012810	2.26	0.001393	2.76	0.000095	3.26	0.00000402
0.27	0.702582	0.77	0.276179	1.27	0.072486	1.77	0.012309	2.27	0.001326	2.77	0.000090	3.27	0.00000376
0.28	0.692120	0.78	0.269990	1.28	0.070266	1.78	0.011826	2.28	0.001262	2.78	0.000084	3.28	0.00000351
0.29	0.681717	0.79	0.263897	1.29	0.068101	1.79	0.011359	2.29	0.001201	2.79	0.000080	3.29	0.00000328
0.3	0.671373	0.8	0.257899	1.3	0.065992	1.8	0.010909	2.3	0.001143	2.8	0.000075	3.3	0.00000306
0.31	0.661092	0.81	0.251997	1.31	0.063937	1.81	0.010475	2.31	0.001088	2.81	0.000071	3.31	0.00000285
0.32	0.650874	0.82	0.246189	1.32	0.061935	1.82	0.010057	2.32	0.001034	2.82	0.000067	3.32	0.00000266
0.33	0.640721	0.83	0.240476	1.33	0.059985	1.83	0.009653	2.33	0.000984	2.83	0.000063	3.33	0.00000249
0.34	0.630635	0.84	0.234857	1.34	0.058086	1.84	0.009264	2.34	0.000935	2.84	0.000059	3.34	0.00000232
0.35	0.620618	0.85	0.229332	1.35	0.056238	1.85	0.008889	2.35	0.000889	2.85	0.000056	3.35	0.00000216
0.36	0.610670	0.86	0.223900	1.36	0.054439	1.86	0.008528	2.36	0.000845	2.86	0.000052	3.36	0.00000202
0.37	0.600794	0.87	0.218560	1.37	0.052688	1.87	0.008179	2.37	0.000803	2.87	0.000049	3.37	0.00000188
0.38	0.590991	0.88	0.213313	1.38	0.050984	1.88	0.007844	2.38	0.000763	2.88	0.000046	3.38	0.00000175
0.39	0.581261	0.89	0.208157	1.39	0.049327	1.89	0.007521	2.39	0.000725	2.89	0.000044	3.39	0.00000163
0.4	0.571608	0.9	0.203092	1.4	0.047715	1.9	0.007210	2.4	0.000689	2.9	0.000041	3.4	0.00000152
0.41	0.562031	0.91	0.198117	1.41	0.046148	1.91	0.006910	2.41	0.000654	2.91	0.000039	3.41	0.00000142
0.42	0.552532	0.92	0.193232	1.42	0.044624	1.92	0.006622	2.42	0.000621	2.92	0.000036	3.42	0.00000132
0.43	0.543113	0.93	0.188437	1.43	0.043143	1.93	0.006344	2.43	0.000589	2.93	0.000034	3.43	0.00000123
0.44	0.533775	0.94	0.183729	1.44	0.041703	1.94	0.006077	2.44	0.000559	2.94	0.000032	3.44	0.00000115
0.45	0.524518	0.95	0.179109	1.45	0.040305	1.95	0.005821	2.45	0.000531	2.95	0.000030	3.45	0.00000107
0.46	0.515345	0.96	0.174576	1.46	0.038946	1.96	0.005574	2.46	0.000503	2.96	0.000028	3.46	0.00000099
0.47	0.506255	0.97	0.170130	1.47	0.037627	1.97	0.005336	2.47	0.000477	2.97	0.000027	3.47	0.00000092
0.48	0.497250	0.98	0.165769	1.48	0.036346	1.98	0.005108	2.48	0.000453	2.98	0.000025	3.48	0.00000086
0.49	0.488332	0.99	0.161492	1.49	0.035102	1.99	0.004889	2.49	0.000429	2.99	0.000024	3.49	0.00000080

The symbol error probability can then be approximated as

$$P_{es} \approx \frac{1}{2M} \cdot \sum_{i=1}^{M} \sum_{C_j \in S_i} \operatorname{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right).$$

Remarkably, the term $\operatorname{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right)$ does not depend on the actual values of symbols C_i and C_j, but on the Euclidean distance between these symbols. For all symbols $C_j \in S_i$, the terms $\operatorname{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right)$ are identical since all these symbols are at the same distance d_0 from C_i.

Therefore, we can write:

$$P_{es} \approx \frac{1}{2M} \cdot \sum_{i=1}^{M} \operatorname{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right) \sum_{C_j \in S_i} 1 = \frac{1}{2M} \cdot \sum_{i=1}^{M} \operatorname{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right) \cdot N_i ,$$

where N_i is the number of nearest neighbor symbols of C_i.

This expression can be further simplified as follows:

$$P_{es} \approx \frac{1}{2M} \cdot \operatorname{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right) \sum_{i=1}^{M} N_i .$$

If we now define the average number of nearest neighbor symbols in the constellation as

$$\overline{N} = \frac{1}{M} \cdot \sum_{i=1}^{M} N_i ,$$

we finally obtain:

$$P_{es} \approx \frac{\overline{N}}{2} \cdot \operatorname{erfc}\left(\sqrt{\frac{d_0^2 g(0)}{8N_0}}\right).$$

Such expression is not very informative since it does not tell us much about the way the error probability can be increased or decreased.

We would like to introduce in this equation the average energy, E_s, per transmitted signal. This energy is computed as the average energy of the pulse that carries a constellation symbol at the transmitter output:

$$E_s = \underset{C}{E}\left\{ \int_{-\infty}^{+\infty} [\, A \cdot h(t) \cdot \cos(2\pi f_0\, t) - B \cdot h(t) \cdot \sin(2\pi f_0\, t)\,]^2 \, dt \right\},$$

where the notation $\underset{C}{E}\{\cdot\}$ denotes the expected value of "." over all possible constellation symbols C = A + jB.

A few calculations lead to $E_s = \underset{C}{E}\left\{ \int_{-\infty}^{+\infty} \left[\frac{A}{2} \cdot [h(t)]^2 + \frac{B}{2} \cdot [h(t)]^2 \right] dt \right\}$, and finally to

$$E_s = \frac{1}{2} \cdot \int_{-\infty}^{+\infty} [\, h(t)\,]^2 \, dt \cdot \underset{C}{E}\{ A^2 + B^2 \} = \frac{1}{2} \cdot \int_{-\infty}^{+\infty} [\, h(t)\,]^2 \, dt \cdot \underset{C}{E}\{ |\, C\,|^2 \}.$$

The term $\int_{-\infty}^{+\infty} [\, h(t)\,]^2 \, dt$, which is simply the energy of the finite-energy signal (pulse) used to carry the transmitted symbols, is equal to g(0) since $g(t) = h(t) * h(-t)$.

The term $\gamma = \underset{C}{E}\{ A^2 + B^2 \} = \underset{C}{E}\{ |\, C\,|^2 \}$ only depends on the shape of the signal constellation. It can be easily computed for any constellation.

Finally, we obtain $E_s = \frac{\gamma}{2} \cdot g(0)$, which yields:

$$P_{es} \approx \frac{\overline{N}}{2} \cdot \mathrm{erfc}\left(\sqrt{\frac{d_0^2}{4\gamma} \cdot \frac{E_s}{N_0}} \right).$$

The ratio E_s/N_0, where E_s is the average energy per transmitted symbol and N_0 refers to the PSD of the white Gaussian noise process, designates the **SNR per transmitted symbol**.

In practice, most engineers prefer using another definition of the SNR in which E_s is replaced by E_b, where E_b is the average energy per transmitted bit. Since each symbol carries m bits, we have $E_s = m.E_b$. Finally, we can write

$$P_{es} \approx \frac{\overline{N}}{2} \cdot \operatorname{erfc}\left(\sqrt{\frac{m d_0^2}{4\gamma} \cdot \frac{E_b}{N_0}}\right),$$

where E_b/N_0 designates the **SNR per transmitted bit**.

We can now use this generic expression to determine the expression of P_{es} for the modulation schemes introduced earlier.

• **M-state amplitude shift keying (M-ASK) modulation**

For such modulation schemes, we always have $d_0^2 = 4$, $\overline{N} = \frac{2(M-1)}{M}$, and $\gamma = \frac{M^2-1}{3}$. We can thus write:

$$P_{es} \approx \frac{M-1}{M} \cdot \operatorname{erfc}\left(\sqrt{\frac{3m}{M^2-1} \cdot \frac{E_b}{N_0}}\right).$$

=> $P_{es} \approx \frac{1}{2} \cdot \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$ for 2-ASK.

Note that, in this particular case, this expression does not only provide an approximation, but the exact value of the symbol error probability.

=> $P_{es} \approx \frac{3}{4} \cdot \operatorname{erfc}\left(\sqrt{\frac{2}{5} \cdot \frac{E_b}{N_0}}\right)$ for 4-ASK.

=> $P_{es} \approx \frac{7}{8} \cdot \mathrm{erfc}\left(\sqrt{\frac{1}{7} \cdot \frac{E_b}{N_0}}\right)$ for 8-ASK.

• **M-state phase shift keying (M-PSK) modulation**

For such modulation schemes with M > 2, we always have $d_0^2 = 4 \cdot \sin^2\left(\frac{\pi}{M}\right)$, $\overline{N} = 2$, and $\gamma = 1$. We can thus write:

$$P_{es} \approx \mathrm{erfc}\left(\sqrt{m \cdot \sin^2\left(\frac{\pi}{M}\right) \cdot \frac{E_b}{N_0}}\right).$$

Note that, in the case M = 2, we have $\overline{N} = 1$ instead of $\overline{N} = 2$.

=> $P_{es} \approx \frac{1}{2} \cdot \mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$ for 2-PSK (BPSK).

This is identical to the result obtained for 2-ASK (it was expected). Once again, note that, in this case, this expression does not only provide an approximation, but the exact value of the symbol error probability.

=> $P_{es} \approx \mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$ for 4-PSK (QPSK).

=> $P_{es} \approx \mathrm{erfc}\left(\sqrt{3 \cdot \sin^2\left(\frac{\pi}{8}\right) \cdot \frac{E_b}{N_0}}\right)$ for 8-PSK.

• **M-state quadrature amplitude modulation (M-QAM)**

For such modulation schemes, we always have $d_0^2 = 4$, $\overline{N} = 4 \cdot \frac{\sqrt{M} - 1}{\sqrt{M}}$, and $\gamma = \frac{2(M-1)}{3}$. We can thus write:

$$P_{es} \approx 2 \cdot \frac{\sqrt{M} - 1}{\sqrt{M}} \cdot \mathrm{erfc}\left(\sqrt{\frac{3m}{2(M-1)} \cdot \frac{E_b}{N_0}}\right).$$

=> $P_{es} \approx \mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$ for 4-QAM.

This is identical to the result obtained for QPSK (it was expected).

=> $P_{es} \approx \frac{3}{2} \cdot \mathrm{erfc}\left(\sqrt{\frac{2}{5} \cdot \frac{E_b}{N_0}}\right)$ for 16-QAM.

=> $P_{es} \approx \frac{7}{4} \cdot \mathrm{erfc}\left(\sqrt{\frac{1}{7} \cdot \frac{E_b}{N_0}}\right)$ for 64-QAM.

• Expression of the bit error probability for any modulation scheme

In practice, engineers may be more interested in the bit error probability, P_{eb}, at the receiver output rather than the symbol error probability. After all, the original goal is to transmit bits as reliably as possible.

It is very simple to obtain an expression for P_{eb} starting from:

$$P_{eb} = \Pr\{\text{ a bit is detected erroneously }\}.$$

Using a small "trick" of probability theory, we can also write:

$$P_{eb} = \Pr\{\text{ a symbol is detected erroneously and a bit is detected erroneously }\},$$

which is, according to Bayes' rule, equivalent to

$$P_{eb} = \Pr\{\text{ a bit is detected erroneously} \mid \text{ symbol is detected erroneously }\} \cdot P_{es}.$$

If the association between bits and symbols is done using Gray mapping, we can write:

$$P_{eb} \approx \frac{P_{es}}{m} .$$

Finally, we obtain the final expression of P_{eb} versus E_b/N_0 for all modulation schemes considered throughout these notes:

=> $P_{eb} = / \approx \frac{1}{2} \cdot \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$ for BPSK and QPSK.

So, my advice is to use QPSK instead of BPSK as you double the spectral efficiency at no cost in terms of error performance.

=> $P_{eb} \approx \frac{3}{8} \cdot \text{erfc}\left(\sqrt{\frac{2}{5} \cdot \frac{E_b}{N_0}}\right)$ for 4-ASK and 16-QAM.

So, my advice is to use 16-QAM instead of 4-ASK for the same reason as above.

=> $P_{es} \approx \frac{7}{24} \cdot \text{erfc}\left(\sqrt{\frac{1}{7} \cdot \frac{E_b}{N_0}}\right)$ for 8-ASK and 64-QAM.

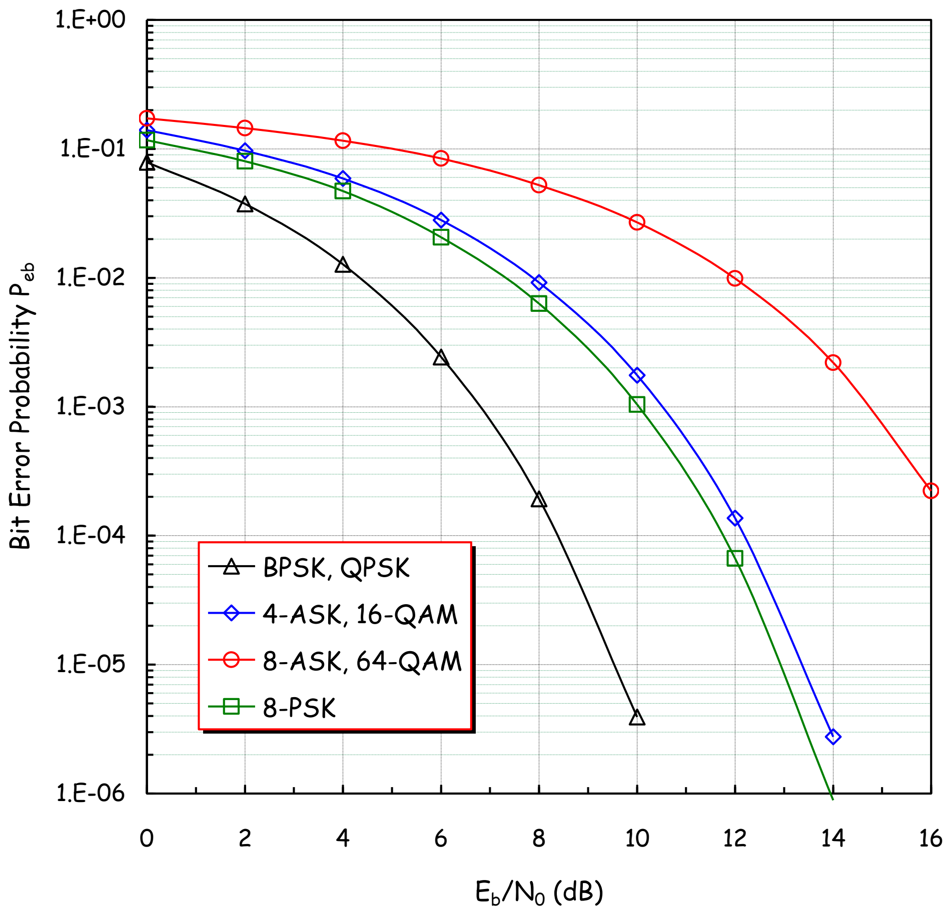
So, my advice is to use 64-QAM instead of 8-ASK for the same reason as above.

=> $P_{es} \approx \frac{1}{3} \cdot \text{erfc}\left(\sqrt{3 \cdot \sin^2\left(\frac{\pi}{8}\right) \cdot \frac{E_b}{N_0}}\right)$ for 8-PSK.

At high SNR, the parameter governing the error performance of a modulation scheme is the argument of the function $\text{erfc}(\cdot)$. Those among you who took the module EEE2004 (Information Theory & Coding) know it very well.

Therefore, QPSK is $10.\log_{10}(2.5) \approx 3.98$ dB more power-efficient (but twice less spectral efficient) than 16-QAM. In addition, QPSK is $10.\log_{10}(7) \approx 8.45$ dB more power-efficient (but three times less spectral efficient) than 64-QAM. As for 8-PSK, it is approximately 3.57 dB less power-efficient (but one and a half times more spectral efficient) than QPSK.

Conclusion: there is a trade-off between spectral efficiency and error performance (those who have heard about Claude E. Shannon and his celebrated theorems will not be surprised).



Using QAM modulations, the SNR increase necessary for incrementing the spectral efficiency by 2 bits/sec/Hz without degrading the error probability is $10 \cdot \log_{10}\left[\frac{4M-1}{M-1} \cdot \frac{m}{m+2}\right]$ dB.

QPSK → 16-QAM: ≈ 3.98 dB,

16-QAM → 64-QAM: ≈ 4.47 dB,

64-QAM → 256-QAM: ≈ 4.82 dB,

256-QAM → 1024-QAM: ≈ 5.06 dB.

For a given spectral efficiency, M-QAM is $-10 \cdot \log_{10}\left[\frac{2}{3}(M-1)\cdot \sin^2\left(\frac{\pi}{M}\right)\right]$ dB more power-efficient than M-PSK:

4-QAM Vs. 4-PSK: 0 dB (are you surprised?),

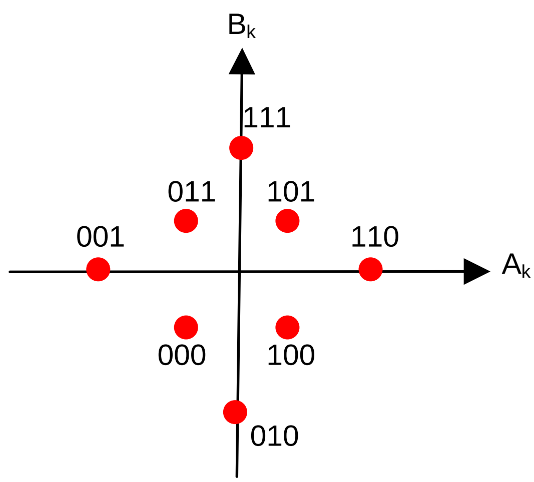
16-QAM Vs. 16-PSK: $\approx$ 4.20 dB,

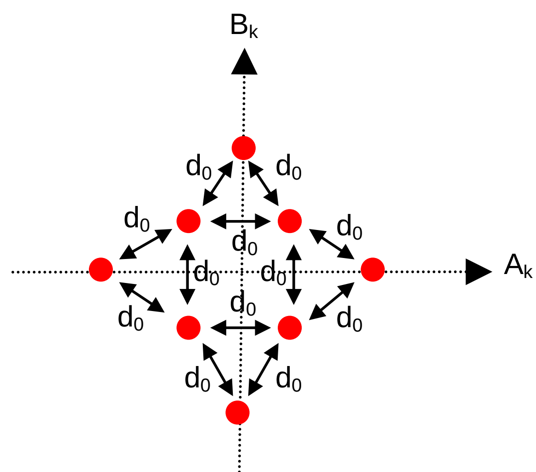
64-QAM Vs. 64-PSK: $\approx$ 9.95 dB.

Therefore, my advice is to always use M-QAM rather than M-PSK, unless you have no choice (presence of a non-linear power amplifier).

Example ---

Consider the 3-bit/s/Hz constellation and mapping depicted below. The signal points are positioned symmetrically around the constellation centre in a way that the Euclidean distance between any two neighbors is equal to a constant d_0 representing the minimal distance between signal points in the constellation.





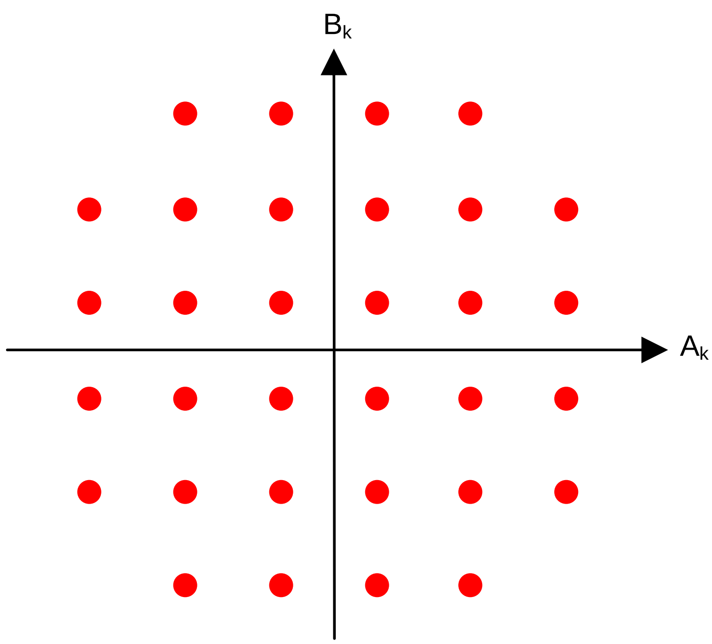
Is the mapping shown above a Gray mapping?

Show that, for such constellation, we have $\overline{N} = 3$ and $\gamma = d_0^2 \frac{3+\sqrt{3}}{4}$, thus leading to $P_{es} \approx \frac{3}{2}\cdot \mathrm{erfc}\left(\sqrt{\frac{3}{3+\sqrt{3}}\cdot\frac{E_b}{N_0}}\right)$.

Also show that this constellation outperforms 8-PSK by 1.6 dB at high SNR.

Example ---

Consider the 5-bit/s/Hz constellation depicted below. For such constellation, called cross 32-QAM, the complex symbol $C_k = A_k + jB_k$ can take 32 possible values, i.e. A_k and $B_k \in \{\pm 1, \pm 3, \pm 5\}$ without allowing for the four particular symbols $C_k = \pm 5 \pm j5$ to be generated.



Can we find a Gray mapping for this constellation?

Show that we have $\overline{N} = \frac{13}{4}$ and $\gamma = 5d_0^2$, thus leading to $P_{es} \approx \frac{13}{8} \cdot \mathrm{erfc}\left(\sqrt{\frac{1}{4} \cdot \frac{E_b}{N_0}}\right)$.

Compare the asymptotic error performance of cross 32-QAM with those of 16–QAM and 64-QAM.

---

10. Assessing the Error Performance of Uncoded or Coded Digital Modulations using Computer Simulations

In the previous chapter, we have seen that it is possible to find a mathematical expression for approximating the error probability for any modulation scheme. This expression has been obtained assuming transmission over an AWGN channel in the absence of any error-correcting (channel) code.

However, there are plenty of practical situations where finding an accurate approximation of the error probability at the receiver output is very difficult, or even impossible. For instance, this is the case with systems for which:

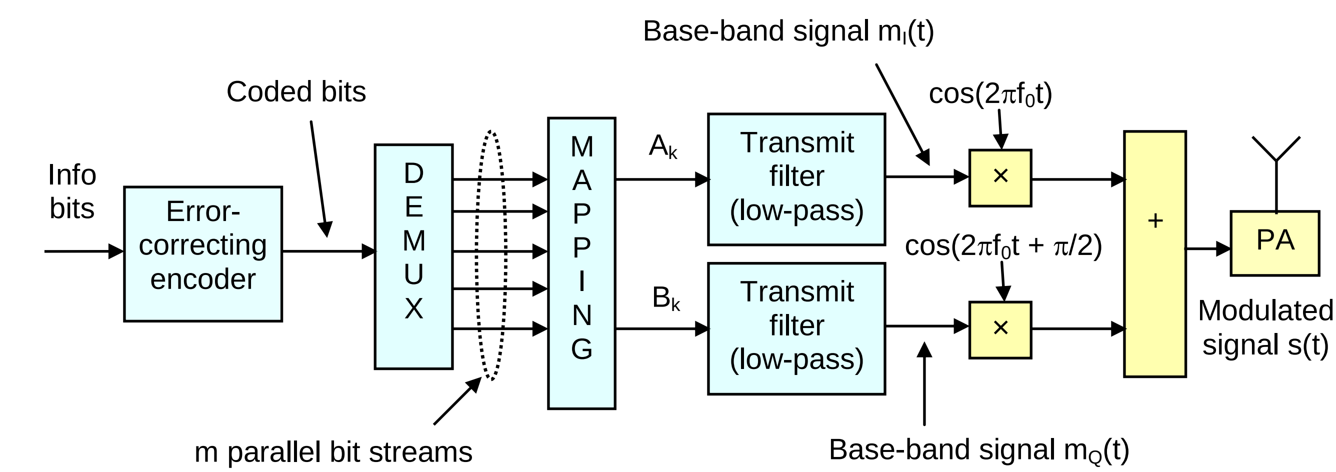
(1) Channel coding is used in association with the modulation scheme in order to achieve a better error performance.

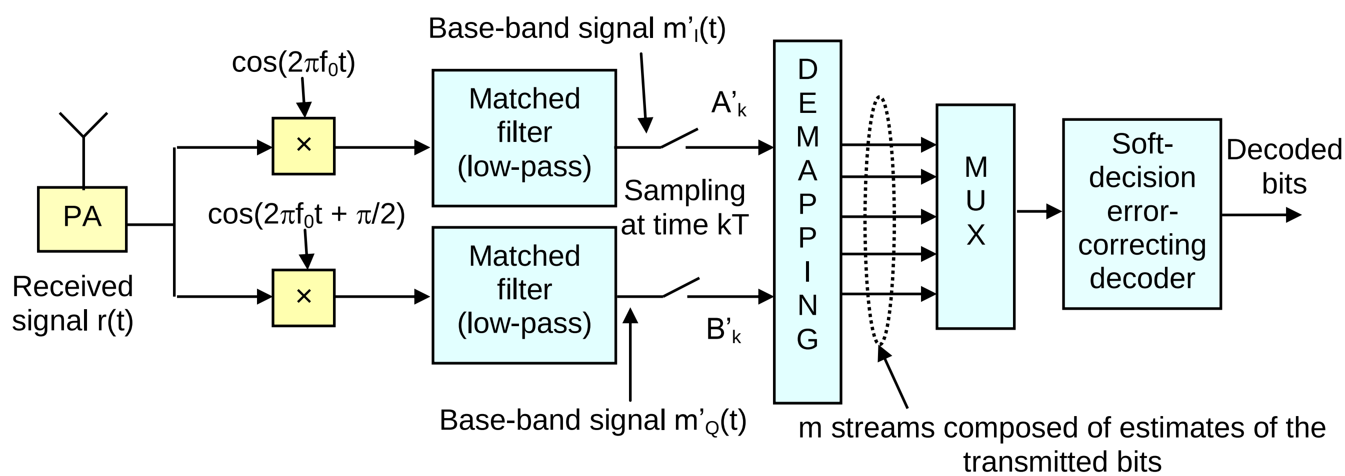
(2) The channel is not AWGN. For instance, the channel could introduce ISI or some fading of the transmitted signal.

For many of these systems, the only way to obtain an accurate approximation of the error performance at the receiver output consists of using computer simulations.

• **Coded modulation schemes**

We have represented below the generic block diagram of a coded modulation scheme, i.e. a modulation scheme combined with a binary error-correcting code (binary ≡ operating on bits).





This block diagram is very similar to that of an uncoded modulation scheme. However, in coded modulation systems, the information bits are encoded by an error-correcting encoder, which can typically be a convolutional, turbo, or low-density parity-check encoder. The coded bits thus generated are then transmitted using the classical modulator structure.

We recall that the coded bit stream is de-multiplexed into m parallel bit streams. Let $(x_{1,k}, x_{2,k}, \ldots, x_{m,k})$ denote the vector of m parallel coded bits at time kT, where k is an integer and T is the duration of a symbol. The mapping operation converts this m-bit vector $(x_{1,k}, x_{2,k}, \ldots, x_{m,k})$ into a pair of real symbols denoted as A_k and B_k, or equivalently a complex symbol $C_k = A_k + jB_k$, which can take $M = 2^m$ different values. In classical systems, we employ Gray mapping since such labeling technique minimizes the bit error probability for a given symbol error probability.

At the mapping block output, we thus have two streams composed at time kT of symbols A_k and B_k that can be either binary or non-binary: In BPSK and QPSK, A_k and B_k are binary, whereas with other modulation schemes, A_k and B_k are not binary.

At the receiver side, we recall that, at time kT, an estimate $C_k^{'}$ of the corresponding transmitted symbol C_k is available at the sampling device output. In a coded modulation system, the estimate $C_k^{'}$ is processed by a "de-mapping" block whose task is to convert $C_k^{'}$ into a vector ($x'_{1,k}$, $x'_{2,k}$, …, $x'_{m,k}$) of analogue samples that estimate the transmitted m-bit vector ($x_{1,k}$, $x_{2,k}$, …, $x_{m,k}$). Such de-mapping block does not exist in uncoded communication systems where it is replaced by a simple decision block.

Why do we need to use such de-mapping block in coded systems?

The answer is simple. A basic principle of information theory (see the late EEE2004 – Information Theory & Coding module) is that channel decoders perform better when operating using soft decisions, i.e. analogue estimates of the transmitted coded bits, rather than hard decisions (bits).

To optimize the error-correction capabilities of the channel decoder, we must therefore make sure to feed this decoder with estimates of the transmitted coded bits that are unfortunately not available at the sampling device output (except in the case of BPSK and QPSK). As a result, we need to compute these estimates using the sequence of samples $C_k^{'}$.

For instance, over an AWGN channel with a given noise variance σ^2, the estimate $x'_{i,k}$, i $\in$ {1, 2, …, m}, of the coded bit $x_{i,k}$ is computed using

$$x_{i,k}^{'} = \frac{\sigma^2}{2} \cdot \ln\left[\frac{\Pr\left\{ x_{i,k} = 0 \,\middle|\, C_k^{'} \right\}}{\Pr\left\{ x_{i,k} = 1 \,\middle|\, C_k^{'} \right\}}\right],$$

which, after a few mathematical manipulations, leads to

$$x'_{i,k} = \frac{\sigma^2}{2} \cdot \ln\left[\frac{\sum_{C \in \Delta_{0,i}} \exp\left\{-\frac{d^2\left(C'_k, C\right)}{2\sigma^2}\right\}}{\sum_{C \in \Delta_{1,i}} \exp\left\{-\frac{d^2\left(C'_k, C\right)}{2\sigma^2}\right\}}\right],$$

where $d^2\left(C'_k, C\right) = \left(A_k - A\right)^2 + \left(B_k - B\right)^2$ designates the Euclidean distance between the received channel sample C'_k and a given possible symbol C (no time index needed here), whereas $\Delta_{j,i}$, j = 0 or 1 and i $\in$ {1, 2, …, m}, represents the set of all symbols C whose label is j in the i-th position. In those expressions, we assume that the soft-decision decoder uses the following convention: a negative received estimate is indicative of a coded bit equal to 1, whereas a positive received estimate is indicative of a coded bit equal to 0. This convention is used by the soft-decision Viterbi decoder available in Matlab.

For instance, for the simplest case corresponding to BPSK modulation, the de-mapper output is given by

$$x'_{1,k} = \frac{\sigma^2}{2} \cdot \ln\left[\frac{\exp\left\{-\frac{\left(A'_k - 1\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k + 1\right)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2} \cdot \left[-\frac{\left(A'_k - 1\right)^2}{2\sigma^2} + \frac{\left(A'_k + 1\right)^2}{2\sigma^2}\right] = A'_k,$$

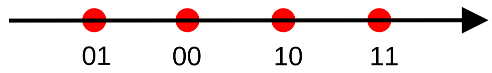
assuming the use of a (0 -> +1 and 1 -> -1) mapping. This result shows that, for BPSK, no de-mapping is actually needed since the estimate of the bit $x_{1,k}$ is already available at the output of the sampling device.

Example ---

Show that, for QPSK with Gray mapping, we can write $x'_{1,k} = A'_k$ and $x'_{2,k} = B'_k$. What conclusion can you draw from such result?

Example ---

Consider a 4-ASK constellation with the Gray mapping indicated below.



The two de-mapper outputs are therefore given by

$$x'_{1k}=\frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{(A'_k+3)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{(A'_k+1)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{(A'_k-3)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{(A'_k-1)^2}{2\sigma^2}\right\}}\right],$$

and

$$x'_{2,k}=\frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{(A'_k-1)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{(A'_k+1)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{(A'_k-3)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{(A'_k+3)^2}{2\sigma^2}\right\}}\right].$$

Note that those expressions can actually be simplified as follows:

If $A'_k>+2$, then $x'_{1k}\approx\frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{(A'_k+1)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{(A'_k-3)^2}{2\sigma^2}\right\}}\right]=\frac{\sigma^2}{2}\cdot\left[-\frac{(A'_k+1)^2}{2\sigma^2}+\frac{(A'_k-3)^2}{2\sigma^2}\right]=-2A'_k+2$.

If $-2\le A'_k\le+2$, then $x'_{1k}\approx\frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{(A'_k+1)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{(A'_k-1)^2}{2\sigma^2}\right\}}\right]=\frac{\sigma^2}{2}\cdot\left[-\frac{(A'_k+1)^2}{2\sigma^2}+\frac{(A'_k-1)^2}{2\sigma^2}\right]=-A'_k$.

If $A'_k < -2$, then $x'_{1k} \approx \frac{\sigma^2}{2} \cdot \ln\left[\frac{\exp\left\{-\frac{(A'_k+3)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{(A'_k-1)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2} \cdot \left[-\frac{(A'_k+3)^2}{2\sigma^2} + \frac{(A'_k-1)^2}{2\sigma^2}\right] = -2A'_k - 2$.

If $A'_k > 0$, then $x'_{2,k} \approx \frac{\sigma^2}{2} \cdot \ln\left[\frac{\exp\left\{-\frac{(A'_k-1)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{(A'_k-3)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2} \cdot \left[-\frac{(A'_k-1)^2}{2\sigma^2} + \frac{(A'_k-3)^2}{2\sigma^2}\right] = -A'_k + 2$.

If $A'_k \le 0$, then $x'_{2,k} \approx \frac{\sigma^2}{2} \cdot \ln\left[\frac{\exp\left\{-\frac{(A'_k+1)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{(A'_k+3)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2} \cdot \left[-\frac{(A'_k+1)^2}{2\sigma^2} + \frac{(A'_k+3)^2}{2\sigma^2}\right] = A'_k + 2$.

The last two equations can be combined to form the following expression:

$$x'_{2,k} \approx 2 - \left| A'_k \right|.$$

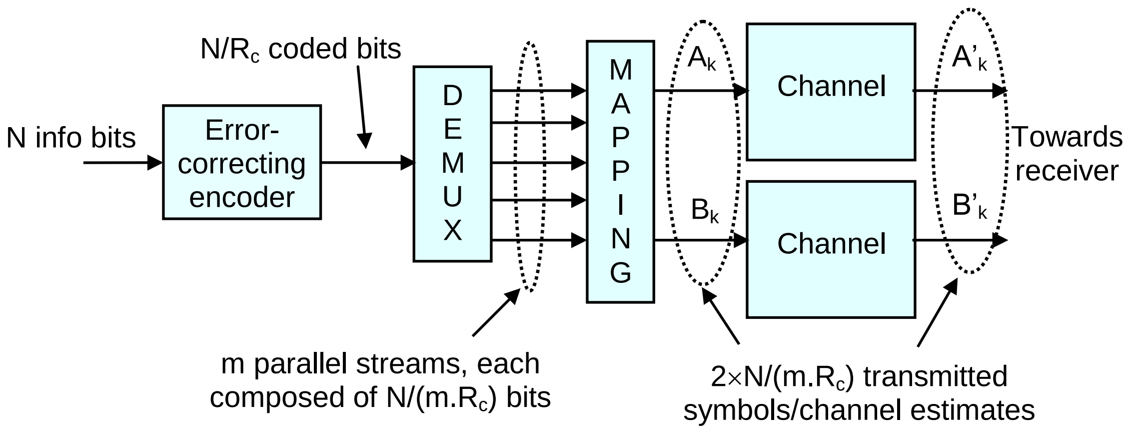
- **Computer simulations of coded modulation schemes**

Computer simulations can be used to assess the error performance of a coded modulation scheme with great accuracy. The first stage consists of generating a stream of N information bits with Pr{0} = Pr{1} = 1/2. Then, this stream is encoded using the algorithm that you have decided to use. At the output of the encoder, the length of the encoded sequence is N/R_c (> N) bits where R_c is the coding rate of the error-correcting code.

After de-multiplexing of the coded bit stream, the k–th vector composed of m coded bits is mapped onto a constellation symbol C_k, i.e. a pair of real-valued symbols A_k and B_k.

Note that each transmitted constellation symbol C_k carries the information corresponding to m coded bits, i.e. only $m.R_c$ information bits. Thus, the spectral efficiency of the whole communication system is no longer m bits/s/Hz as in the uncoded case, but reduced to $m.R_c$ bits/s/Hz => **The use of channel coding reduces the spectral efficiency by a factor equal to the coding rate of the channel code**.

When comparing the error performance of different coded modulation systems, you will have to take this into account as all schemes must have identical spectral efficiencies.



Then, transmission of constellation symbols $C_k = A_k + jB_k$ over the desired channel is simulated by computing the channel outputs A'_k and B'_k as a function of A_k and B_k. For instance, in the case of an AWGN channel, we have already seen that:

$$A_k^{'} = A_k + N_{k,1},$$

and

$$B_k^{'} = B_k + N_{k,2},$$

where $N_{k,1}$ and $N_{k,2}$ are two independent Gaussian noise samples with zero-mean and variance σ^2.

Since the error performance of the system is ultimately a function of the SNR per information bit, E_b/N_0, we need to specify in the computer program the relation between σ^2 and the SNR. This can be done by using a few precious equations derived in the two previous chapters:

(1) $\sigma^2 = \frac{N_0}{g(0)}$,

(2) $E_s = \frac{\gamma}{2} \cdot g(0)$,

(3) $E_b = \frac{E_s}{m \cdot R_c}$ (not $E_b = \frac{E_s}{m}$ since this is a coded system!).

By combining (1), (2), and (3), we find the generic expression: $\sigma^2 = \frac{\gamma}{2m \cdot R_c} \cdot \left(\frac{E_b}{N_0}\right)^{-1}$.

If the SNR is expressed in decibels (dB), we use the following equation instead:

$$\sigma^2 = \frac{\gamma}{2m \cdot R_c} \cdot 10^{-\frac{1}{10} \cdot \left(\frac{E_b}{N_0}\right)_{dB}}.$$

Example 1 ---

Coded QPSK with R_c = ½ (η = 1 bit/s/Hz) and symbols A_k and $B_k \in \{\pm 1\}$:

$\gamma = 2$ and m = 2 => $\sigma^2 = 10^{-0.1 \cdot \left(\frac{E_b}{N_0}\right)_{dB}}$.

Example 2 ---

Coded QPSK with R_c = ½ (η = 1 bit/s/Hz) and symbols A_k and B_k located on the unit-energy circle, i.e. A_k and $B_k \in \{\pm 1/\text{sqrt}(2)\}$:

$\gamma = 1$ and m = 2 => $\sigma^2 = \frac{1}{2} \times 10^{-0.1 \cdot \left(\frac{E_b}{N_0}\right)_{dB}}$.

Example 3 ---

Uncoded QPSK (η = 2 bits/s/Hz) with symbols A_k and $B_k \in \{\pm 1\}$:

R_c = 1, $\gamma = 2$ and m = 2 => $\sigma^2 = \frac{1}{2} \times 10^{-0.1 \cdot \left(\frac{E_b}{N_0}\right)_{dB}}$.

Example 4 ---

Coded 8-PSK with R_c = 2/3 (η = 2 bits/s/Hz) and symbols A_k and B_k located on the unit-energy circle:

$\gamma = 1$ and m = 3 => $\sigma^2 = \frac{1}{4} \times 10^{-0.1 \cdot \left(\frac{E_b}{N_0}\right)_{dB}}$.

Example 5 ---

Uncoded 8-PSK (η = 3 bits/s/Hz) with symbols A_k and B_k located on the unit-energy circle:

R_c = 1, $\gamma = 1$ and m = 3 => $\sigma^2 = \frac{1}{6} \times 10^{-0.1 \cdot \left(\frac{E_b}{N_0}\right)_{dB}}$.

Example 6 ---

Coded 16-QAM with R_c = 3/4 (η = 3 bits/s/Hz) and symbols A_k and $B_k \in \{\pm 1, \pm 3\}$:

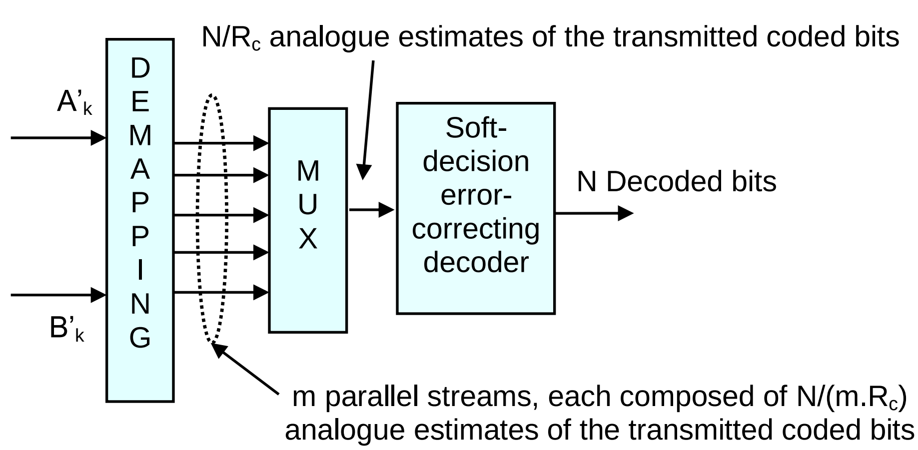
$\gamma = 10$ and m = 4 => $\sigma^2 = \frac{5}{3} \times 10^{-0.1 \cdot \left(\frac{E_b}{N_0}\right)_{dB}}$.

Example 7 ---

Uncoded 16-QAM (η = 4 bits/s/Hz) with symbols A_k and $B_k \in \{\pm 1, \pm 3\}$:

R_c = 1, $\gamma = 10$ and m = 4 => $\sigma^2 = \frac{5}{4} \times 10^{-0.1 \cdot \left(\frac{E_b}{N_0}\right)_{dB}}$.

At the receiver side, the k-th channel outputs A'_k and B'_k are processed by the de-mapping block so as to produce the k-th vector composed of m analogue samples. These samples estimate the m coded bits that were associated with the transmitted symbols A_k and B_k at the transmitter side. Finally, after multiplexing, the samples are fed into the soft-decision channel decoder.



At this stage, we are able to compute the bit error rate (BER) which is a measure of the bit error probability P_{eb}. To do so, the sequence of information bits is compared with the decoded sequence: Every time a decoded bit is different from the corresponding transmitted information bit, an error is counted. Finally, the BER is obtained by dividing the total number of errors by the number N of transmitted information bits.

- **A few examples of error performance results obtained by computer simulations**

To illustrate the kind of error performance that can be achieved with today's communication schemes, we have plotted below several "BER Vs. SNR" curves obtained by computer simulations. For all systems, the spectral efficiency is equal to 3 bits/s/Hz. Here, such spectral efficiency has been achieved using either uncoded 8-PSK or coded 16-QAM based on a rate-3/4 error-correcting code. In all cases, Gray mapping is employed with both 8-PSK and 16-QAM.

For a change, we have not assumed transmission over an AWGN channel. Instead, we have assumed transmission over a **flat Rayleigh fading channel with perfect channel interleaving** which is a better model than the AWGN channel for many communication systems, such as wireless mobile communications for instance.

For a flat Rayleigh fading channel with perfect channel interleaving, the channel output is given by:

$$A_k^{'} = h_{k,1} \cdot A_k + N_{k,1},$$

and

$$B_k^{'} = h_{k,2} \cdot B_k + N_{k,2},$$

where $N_{k,1}$ and $N_{k,2}$ are two independent additive Gaussian noise samples with zero-mean and variance σ^2. The multiplicative noise samples $h_{k,1}$ and $h_{k,2}$ represent the fading affecting the I and Q components of the modulated signal, respectively. Owing to channel interleaving, both samples $h_{k,1}$ and $h_{k,2}$ are assumed to be independent. Without channel interleaving, we would have $h_{k,1} = h_{k,2} = h_k$.

In order to understand how channel interleaving operates, we can consider the example below. Assume the sequence to be transmitted is $\{C_0, C_1, C_2, C_3, C_4, C_5\} = \{A_0, B_0, A_1, B_1, A_2, B_2, A_3, B_3, A_4, B_4, A_5, B_5\}$.

If this sequence is transmitted through the fading channel without prior interleaving, the received sequence will be $\{h_0C_0, h_1C_1, h_2C_2, h_3C_3, h_4C_4, h_5C_5\} = \{h_0A_0, h_0B_0, h_1A_1, h_1B_1, h_2A_2, h_2B_2, h_3A_3, h_3B_3, h_4A_4, h_4B_4, h_5A_5, h_5B_5\}$.

Note that, for simplicity sake, we assume here no presence of additive noise. It is also worthwhile mentioning that two successive fading samples h_{k-1} and h_k can be strongly correlated since the channel parameters can change very slowly in practice.

Assume now the use of an interleaving function before transmission of the symbols so that the transmitted sequence is $\{A_1, B_2, A_3, B_4, A_5, B_0, A_0, B_1, A_2, B_3, A_4, B_5\}$. In this case, the received sequence will be $\{h_0A_1, h_0B_2, h_1A_3, h_1B_4, h_2A_5, h_2B_0, h_3A_0, h_3B_1, h_4A_2, h_4B_3, h_5A_4, h_5B_5\}$. After de-interleaving, this sequence becomes $\{h_3A_0, h_2B_0, h_0A_1, h_3B_1, h_4A_2, h_0B_2, h_1A_3, h_4B_3, h_5A_4, h_1B_4, h_2A_5, h_5B_5\}$.

We can see that, thanks to the interleaving function, the I and Q components of a given complex symbol are no longer affected by the same fading sample. In addition, if the size of the interleaving function is large enough (which is not the case in this example), we can ensure that two successive fading samples in the de-interleaved sequence are independent.

For a Rayleigh fading channel, both fading samples $h_{k,1}$ and $h_{k,2}$ follow a Rayleigh distribution which, for simplicity sake, is assumed to be normalized, i.e. defined so that $E\{(h_{k,1})^2\} = E\{(h_{k,2})^2\} = 1$. To generate a normalized Rayleigh fading sample h, you must first generate two independent Gaussian samples n_1 and n_2 with zero-mean and variance 1/2, and then use the following equation:

$$h = \sqrt{(n_1)^2 + (n_2)^2}\ .$$

It is easy to check that $E\{h^2\} = E\{n_1^2\} + E\{n_2^2\} = ½ + ½ = 1$.

We also assume that the receiver knows the values of samples $h_{k,1}$ and $h_{k,2}$. In technical terms, we say that "perfect channel state information" is assumed. This assumption is not as unrealistic as it seems because there are practical techniques ("channel estimation algorithms") that can be used to accurately calculate in real time the values of samples $h_{k,1}$ and $h_{k,2}$. You will see during the assignment that the communication scheme could not work properly without the knowledge of $h_{k,1}$ and $h_{k,2}$, thus implying that channel state information is a must.

Finally, the Rayleigh fading channel is said to be flat because the only effect of the channel (apart from the addition of Gaussian noise) is to amplify/attenuate the transmitted symbols.

It is worthwhile mentioning that, over a Rayleigh fading channel with perfect channel state information, the de-mapping expressions are different from those employed over an AWGN channel. Those expressions must take into account the fact that the fading samples are known at the receiver side.

Over a Rayleigh fading channel with perfect channel state information and a given noise variance σ^2, the estimate $x'_{i,k}$, $i \in \{1, 2, \ldots, m\}$, of the coded bit $x_{i,k}$ is computed using

$$x'_{i,k} = \frac{\sigma^2}{2} \cdot \ln\left[\frac{\Pr\left\{x_{i,k} = 0 \,\middle|\, h_{k,1}, h_{k,2}, C'_k\right\}}{\Pr\left\{x_{i,k} = 1 \,\middle|\, h_{k,1}, h_{k,2}, C'_k\right\}}\right],$$

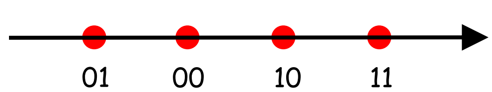
which, after a few mathematical manipulations, leads to

$$x'_{i,k} = \frac{\sigma^2}{2} \cdot \ln\left[\frac{\sum_{C \in \Delta_{0,i}} \exp\left\{-\frac{(A_k - h_{k,1} \cdot A)^2 + (B_k - h_{k,2} \cdot B)^2}{2\sigma^2}\right\}}{\sum_{C \in \Delta_{1,i}} \exp\left\{-\frac{(A_k - h_{k,1} \cdot A)^2 + (B_k - h_{k,2} \cdot B)^2}{2\sigma^2}\right\}}\right],$$

where $\Delta_{j,i}$, j = 0 or 1 and $i \in \{1, 2, \ldots, m\}$, represents the set of all symbols C whose label is j in the i-th position. In those expressions, we recall that the soft-decision decoder uses the following convention: a negative received estimate is indicative of a coded bit equal to 1, whereas a positive received estimate is indicative of a coded bit equal to 0.

Example ---

Consider once again a 4-ASK constellation with the Gray mapping indicated below.



The de-mapper outputs are therefore given by

$$x'_{1,k} = \frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{\left(A'_k+3h_{k,1}\right)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{\left(A'_k+h_{k,1}\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k-3h_{k,1}\right)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{\left(A'_k-h_{k,1}\right)^2}{2\sigma^2}\right\}}\right],$$

and

$$x'_{2,k} = \frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{\left(A'_k-h_{k,1}\right)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{\left(A'_k+h_{k,1}\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k-3h_{k,1}\right)^2}{2\sigma^2}\right\}+\exp\left\{-\frac{\left(A'_k+3h_{k,1}\right)^2}{2\sigma^2}\right\}}\right].$$

Note that those expressions can actually be simplified as follows:

If $A'_k > +2h_{k,1}$, then

$$x'_{1,k} \approx \frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{\left(A'_k+h_{k,1}\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k-3h_{k,1}\right)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2}\cdot\left[-\frac{\left(A'_k+h_{k,1}\right)^2}{2\sigma^2}+\frac{\left(A'_k-3h_{k,1}\right)^2}{2\sigma^2}\right] = -2h_{k,1}\cdot\left(A'_k-h_{k,1}\right).$$

If $\left|A'_k\right| \le +2h_{k,1}$, then

$$x'_{1,k} \approx \frac{\sigma^2}{2}\cdot\ln\left[\frac{\exp\left\{-\frac{\left(A'_k+h_{k,1}\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k-h_{k,1}\right)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2}\cdot\left[-\frac{\left(A'_k+h_{k,1}\right)^2}{2\sigma^2}+\frac{\left(A'_k-h_{k,1}\right)^2}{2\sigma^2}\right] = -h_{k,1}\cdot A'_k.$$

If $A'_k < -2h_{k,1}$, then

$$x'_{1,k} \approx \frac{\sigma^2}{2} \cdot \ln\left[\frac{\exp\left\{-\frac{\left(A'_k + 3h_{k,1}\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k - h_{k,1}\right)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2} \cdot \left[-\frac{\left(A'_k + 3h_{k,1}\right)^2}{2\sigma^2} + \frac{\left(A'_k - h_{k,1}\right)^2}{2\sigma^2}\right] = -2h_{k,1} \cdot \left(A'_k + h_{k,1}\right).$$

If $A'_k > 0$, then

$$x'_{2,k} \approx \frac{\sigma^2}{2} \cdot \ln\left[\frac{\exp\left\{-\frac{\left(A'_k - h_{k,1}\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k - 3h_{k,1}\right)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2} \cdot \left[-\frac{\left(A'_k - h_{k,1}\right)^2}{2\sigma^2} + \frac{\left(A'_k - 3h_{k,1}\right)^2}{2\sigma^2}\right] = -h_{k,1} \cdot \left(A'_k - 2h_{k,1}\right).$$

If $A'_k \le 0$, then

$$x'_{2,k} \approx \frac{\sigma^2}{2} \cdot \ln\left[\frac{\exp\left\{-\frac{\left(A'_k + h_{k,1}\right)^2}{2\sigma^2}\right\}}{\exp\left\{-\frac{\left(A'_k + 3h_{k,1}\right)^2}{2\sigma^2}\right\}}\right] = \frac{\sigma^2}{2} \cdot \left[-\frac{\left(A'_k + h_{k,1}\right)^2}{2\sigma^2} + \frac{\left(A'_k + 3h_{k,1}\right)^2}{2\sigma^2}\right] = -h_{k,1} \cdot \left(-A'_k - 2h_{k,1}\right).$$

The last two equations can be combined into a single expression which is as follows:

$$x'_{2,k} \approx -h_{k,1} \cdot \left(\left|A'_k\right| - 2h_{k,1}\right).$$

If the Rayleigh fading channel was frequency-selective rather than flat, it would also introduce the dreaded ISI among channel output samples. For a frequency-selective fading channel (without perfect channel interleaving), the channel output is given by:

$$A'_k = \sum_{i=0}^{L-1} h_i \cdot A_{k-i} + N_{k,1},$$

and $$B_k^{'} = \sum_{i=0}^{L-1} h_i \cdot B_{k-i} + N_{k,2} \, ,$$

where L is the number of paths. Generally, the L fading samples h_i, $i \in \{0, 1, \ldots, L-1\}$, are independent and follow a Rayleigh distribution with normalized power, i.e. defined so that $\sum_{i=0}^{L-1} E\left\{(h_i)^2\right\} = 1$.

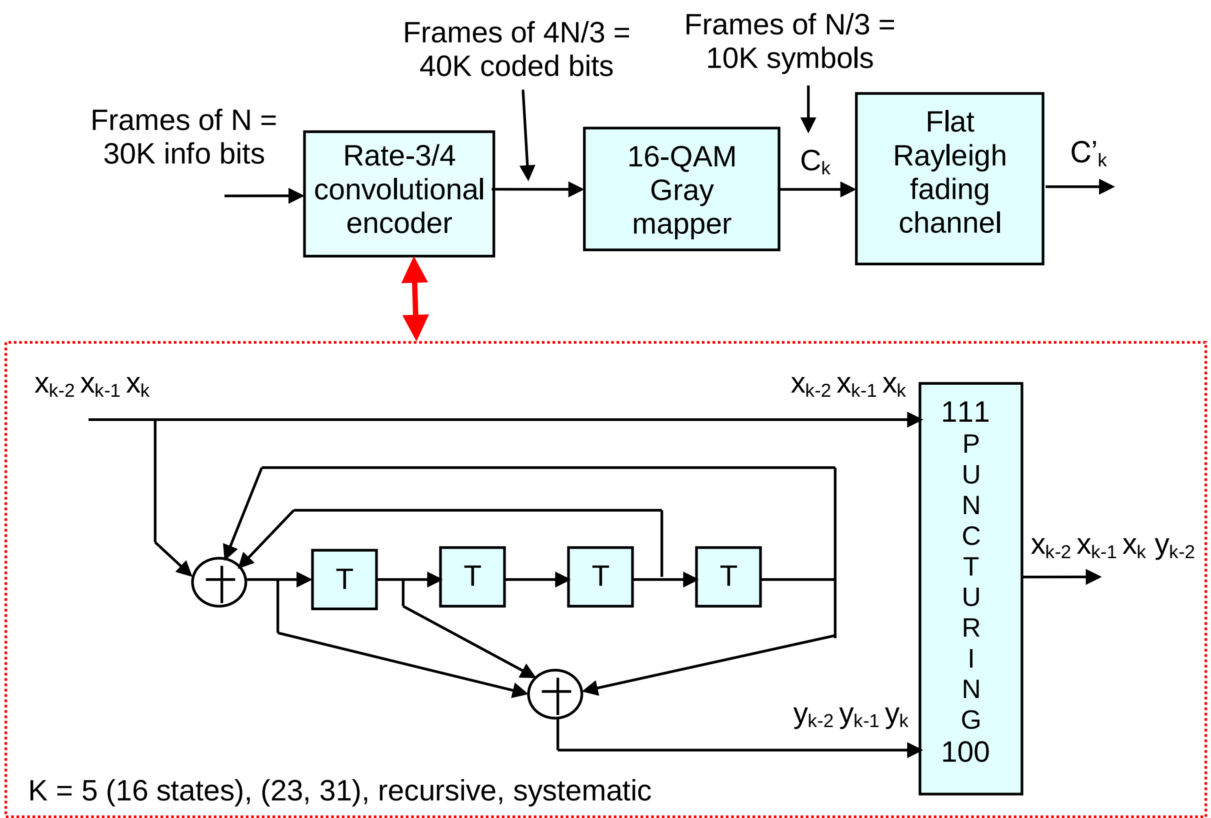
Nowadays, due to the ever-increasing bit rates, wireless communication channels can almost always be considered as frequency-selective fading channels. To suppress the ISI inherent to such channels, one must use equalization techniques at the receiver side and/or OFDM (orthogonal frequency division multiplexing) architectures. However, such topics are beyond the scope of these lecture notes.

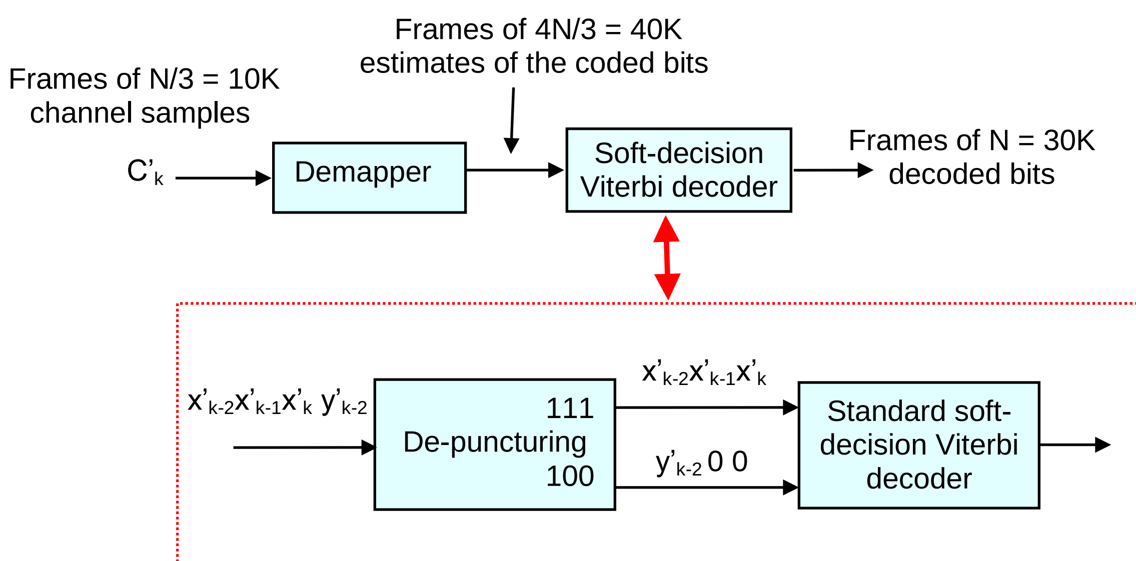
For the coded 16-QAM, we have basically considered four different systems.

- **System # 1**

Transmission is performed using successive frames of 30,000 (30K) information bits. 16-QAM is combined with a rate-3/4 convolutional code (which happens to be recursive and systematic, for no particular reason).

The constraint length of this convolutional code is K = 5 (corresponding to a 16-state code) and its generator polynomials are (23, 31) in octal notation, or alternatively (010 011, 011 001) in binary notation. The coding rate R_c = ¾ is obtained by puncturing the rate-1/2 mother convolutional encoder using a periodic pattern that keeps the code systematic.

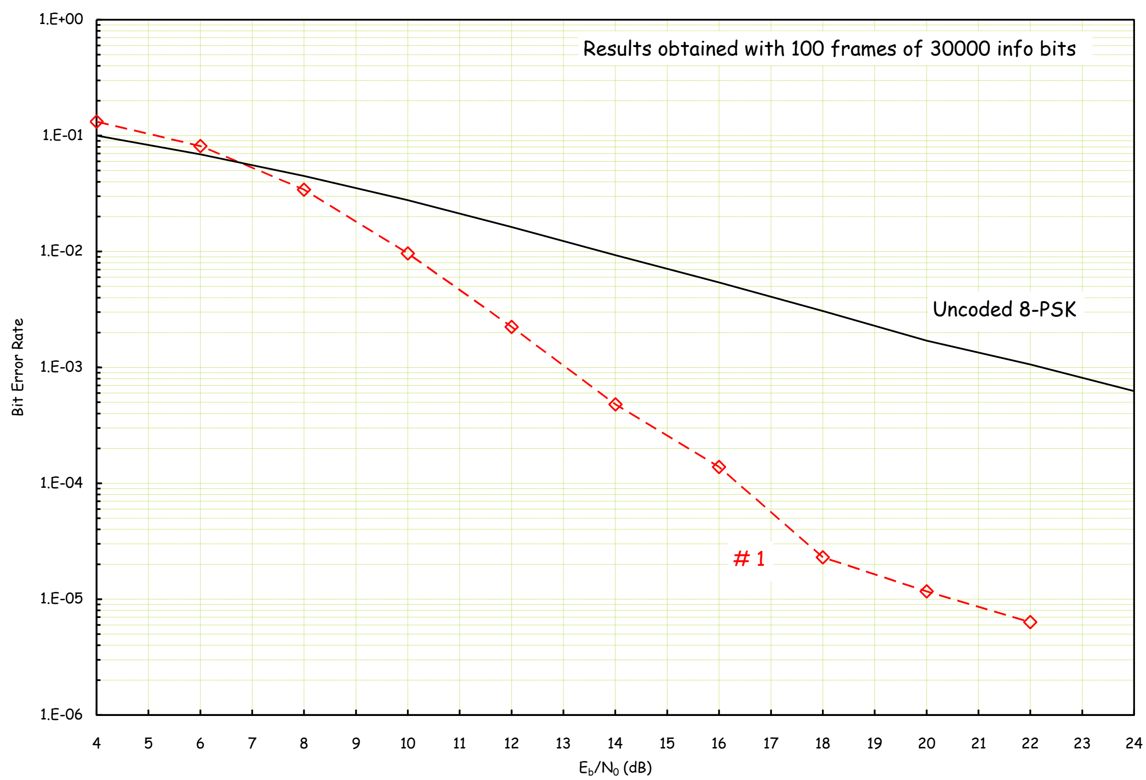

Frames of N =
30K info bits
Frames of 4N/3 =
40K coded bits
Frames of N/3 =
10K symbols
Rate-3/4
convolutional
encoder
16-QAM
Gray
mapper
C_k
Flat
Rayleigh
fading
channel
C'_k
x_{k-2} x_{k-1} x_k
x_{k-2} x_{k-1} x_k
T
T
T
T
111
PUNCTURING
100
x_{k-2} x_{k-1} x_k y_{k-2}
y_{k-2} y_{k-1} y_k
K = 5 (16 states), (23, 31), recursive, systematic


Frames of N/3 = 10K
channel samples
Frames of 4N/3 = 40K
estimates of the coded bits
C'_k
Demapper
Soft-decision
Viterbi decoder
Frames of N = 30K
decoded bits
$x'_{k-2}x'_{k-1}x'_k$ y'_{k-2}
111
De-puncturing
100
$x'_{k-2}x'_{k-1}x'_k$
y'_{k-2} 0 0
Standard soft-
decision Viterbi
decoder

At the receiver side, the decoder uses the soft-decision Viterbi algorithm associated with the rate-1/2 mother convolutional encoder (often referred to as "standard Viterbi decoder").

A de-puncturing block must be used prior to Viterbi decoding in order to replace the missing estimates with neutral values (= 0 since the Viterbi decoder is a binary decoder using the representation +1/-1).

If we compare the error performance of system # 1 with that of uncoded 8-PSK, it appears that very large coding gains are achieved by using channel coding. For instance, at BER = 10^{-3}, coded 16-QAM outperforms uncoded 8-PSK by approximately 9 dB. At lower BER values, the coding gains are so large that they cannot even be measured using our graph.



• **System # 2**

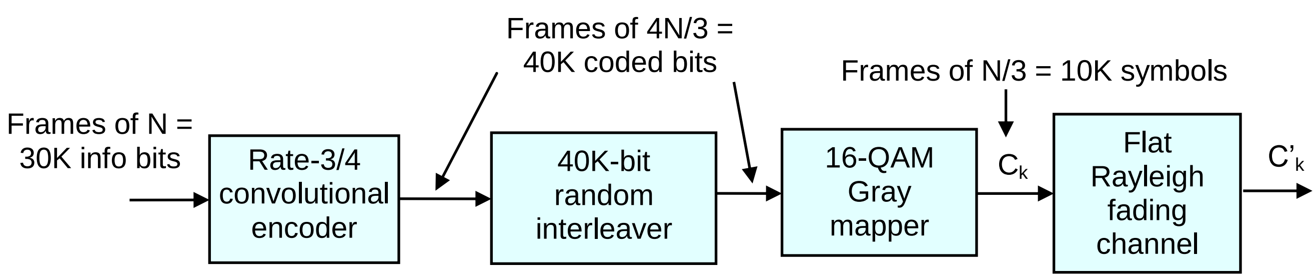
It is identical to system # 1, except that a 40K-bit random interleaving function is inserted between the convolutional code and the mapping block. It is observed from the graph that the use of a 40K-bit random interleaving function slightly improves the error performance of the system (compared to # 1). You could ask yourself the following questions:

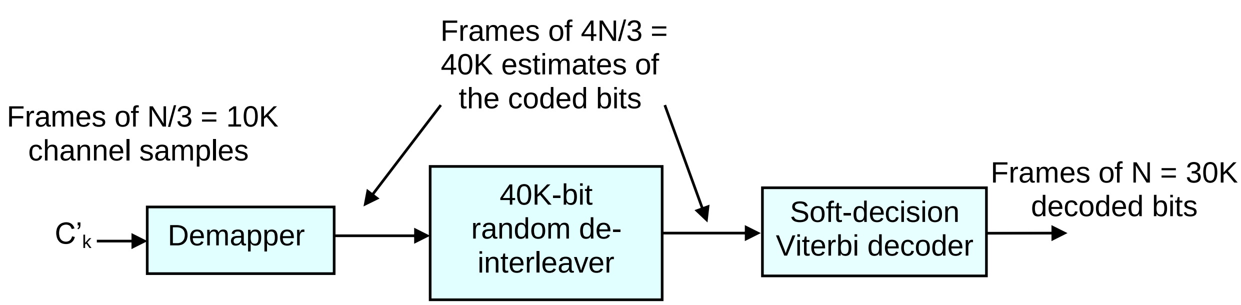
(1) How do we explain such performance improvement?

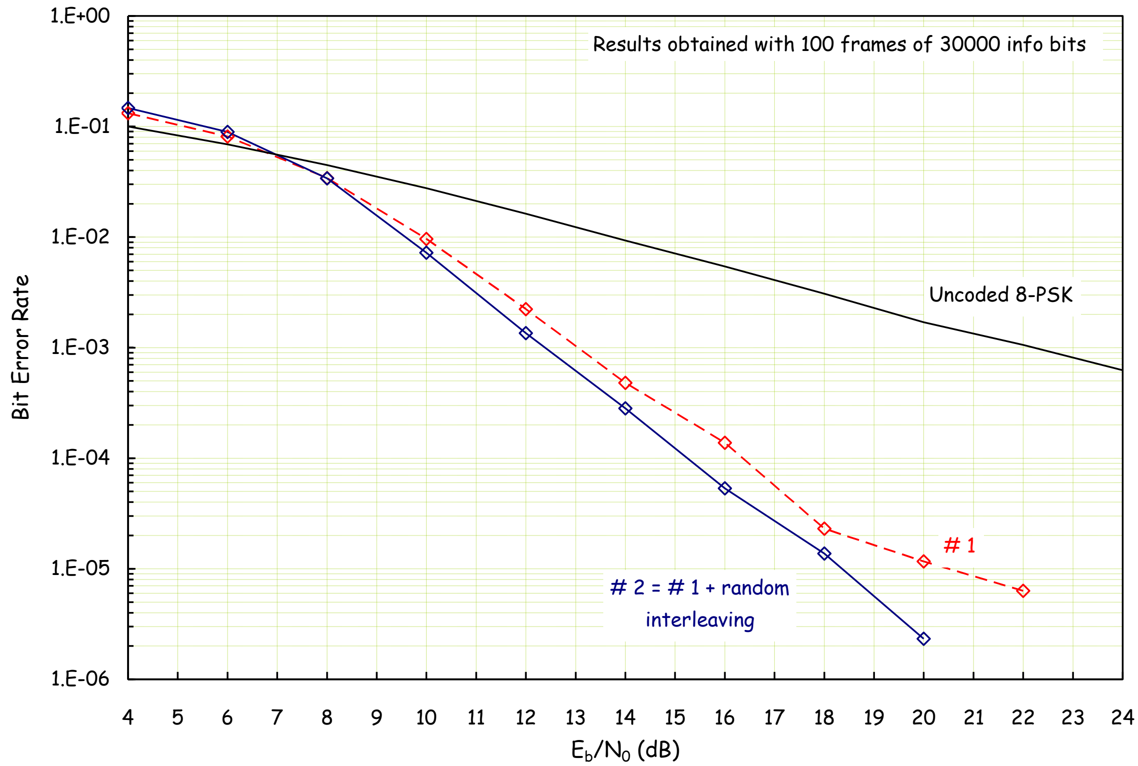
(2) Would an increase in the interleaver size result in even better performance?

(3) Would a deterministic interleaver perform better than a random one?

(4) Would the use of such interleaving function also be beneficial on AWGN channels?





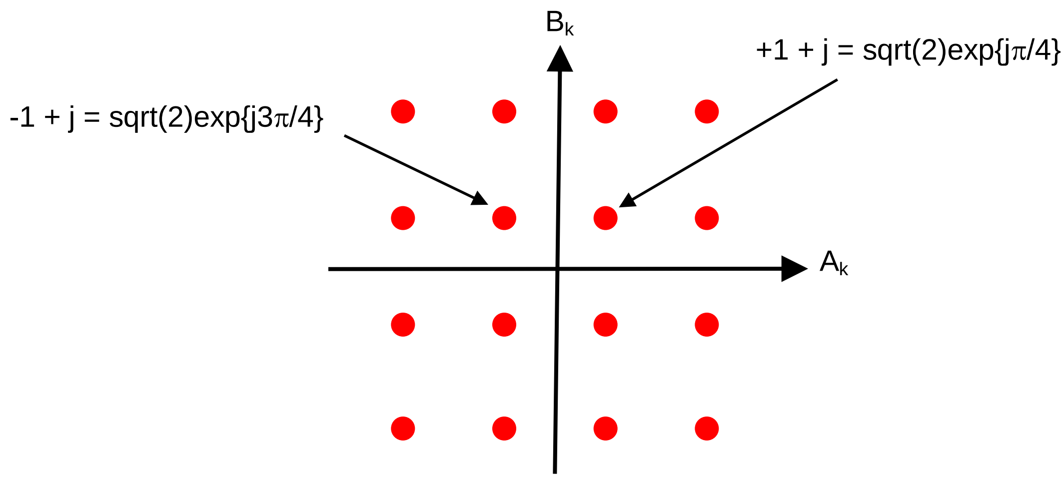


• **System # 3**

It is identical to system # 2, except that the 16-QAM constellation is rotated by an angle $\pi/8$.

Let us explain briefly why rotating the constellation can improve the error performance of the whole communication system over fading channels.

Consider the two constellation symbols below as an example (no loss of generality).



The distance between these two signal points is equal to $d_0 = 2$. Assume now that the I component A_k undergoes a "deep fading", meaning that the fading sample $h_{k,1}$ associated with A_k is very small. In this case, both signal points come much closer to each other: The distance between both signal points is no longer $d_0 = 2$ but equal to $d_1 = d_0.h_{k,1} = 2.h_{k,1} << 2$. Therefore, it becomes very easy to make a symbol error even in presence of a very small additive noise sample. This is the reason behind the poor BER performance of uncoded systems over Rayleigh fading channels.

Now assume that the constellation symbols have all been rotated by an angle θ prior to transmission. In other words, our two signal points now correspond to the constellation symbols $\mathrm{sqrt}(2).\exp\{j(3\pi/4+\theta)\}$ and $\mathrm{sqrt}(2).\exp\{j(\pi/4+\theta)\}$. Obviously, the distance between these two signal points remains equal to d_0 = 2 as for the non-rotated constellation.

However, in the presence of fading, the distance between both symbols becomes equal to $d_2 = 2.\mathrm{sqrt}[(h_{k,1}.\cos\theta)^2 + (h_{k,2}.\sin\theta)^2]$, where $h_{k,1}$ and $h_{k,2}$ are the fading samples affecting the I and Q components, respectively. The distance d_2 can be very small (<< 2) only if $h_{k,1}$ and $h_{k,2}$ are both very small. Having $h_{k,1}$ and $h_{k,2}$ very small at the same time is less likely to happen than having only $h_{k,1}$ very small.

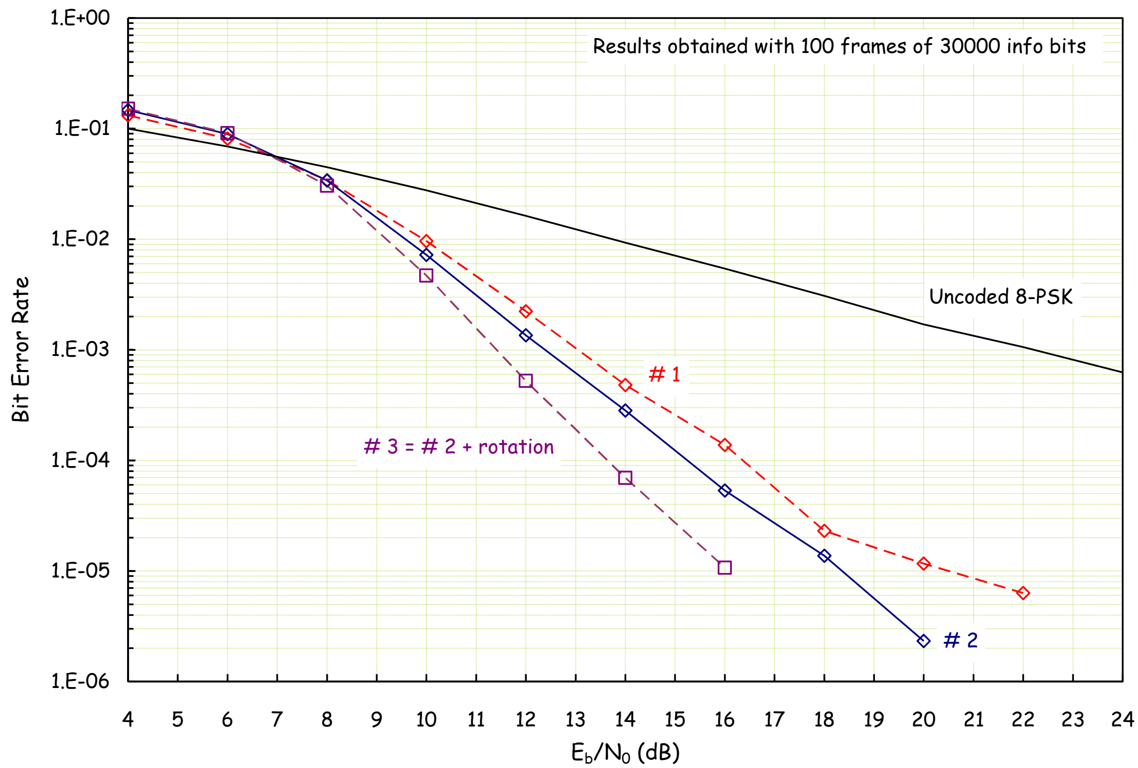
Therefore, rotating the constellation improves the BER performance over fading channel. In technical terms, we say that such rotation generates a "**diversity of order 2**" whereas a non-rotated 16-QAM constellation only has a diversity of order 1.

The BER curves shown below clearly confirm that rotating the constellation by an angle of $\pi/8$ significantly improves the error performance of the system.

You could ask yourself the following questions:

(1) Would constellation rotation be beneficial over AWGN channel?

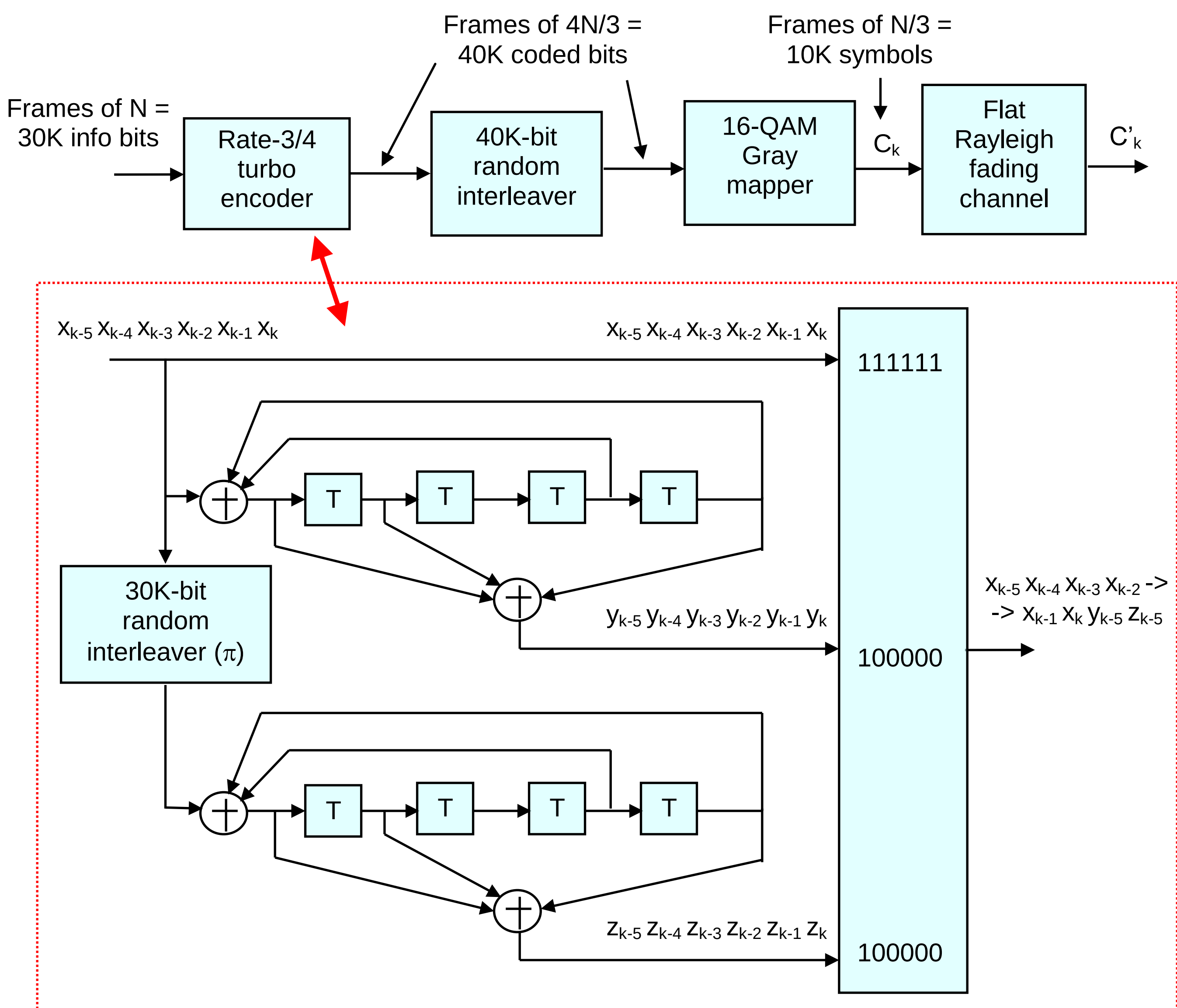
(2) Would constellation rotation be beneficial over a Raleigh fading channel with no channel interleaving ($h_{k,1} = h_{k,2}$)?



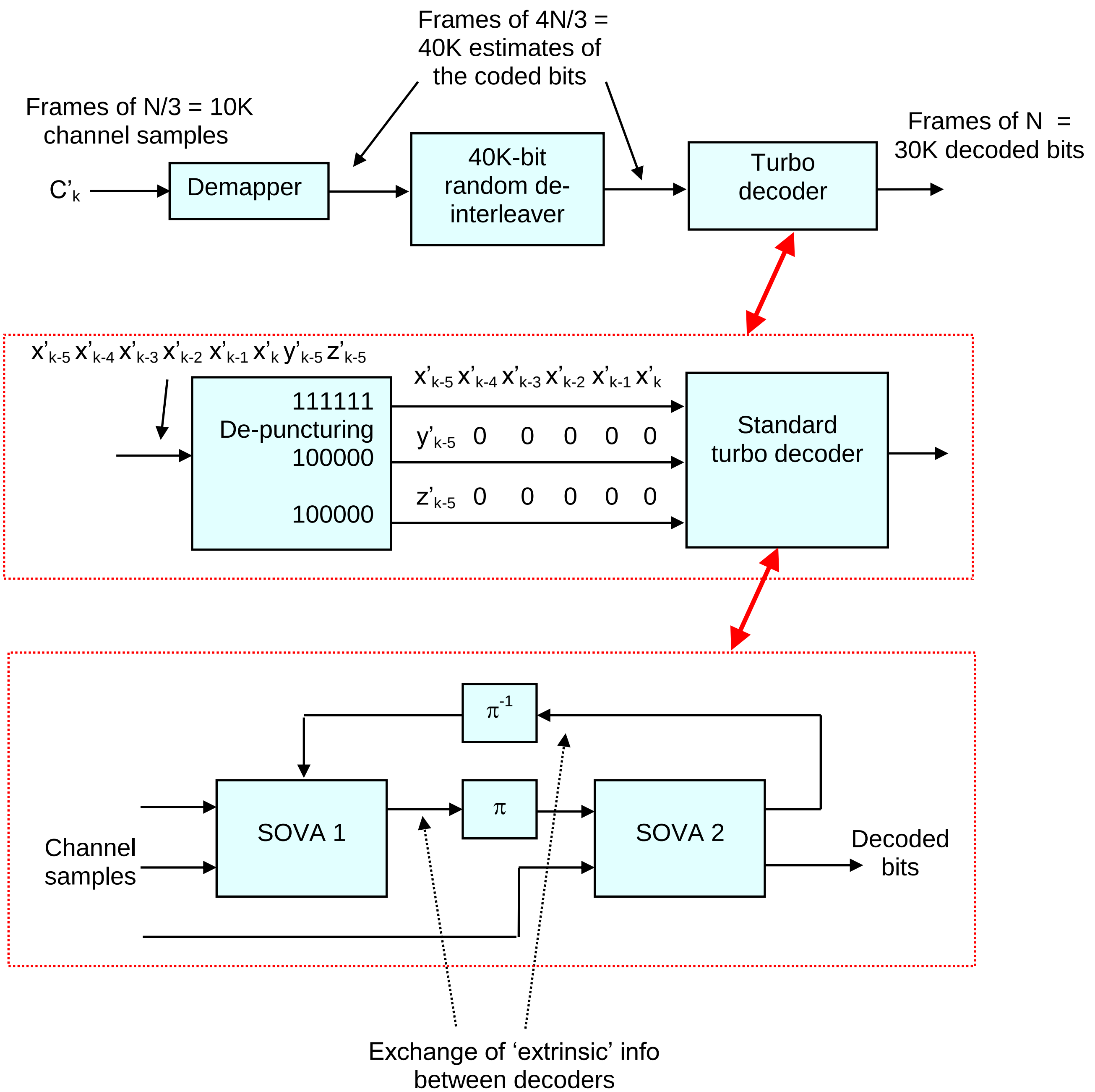
- **System # 4**

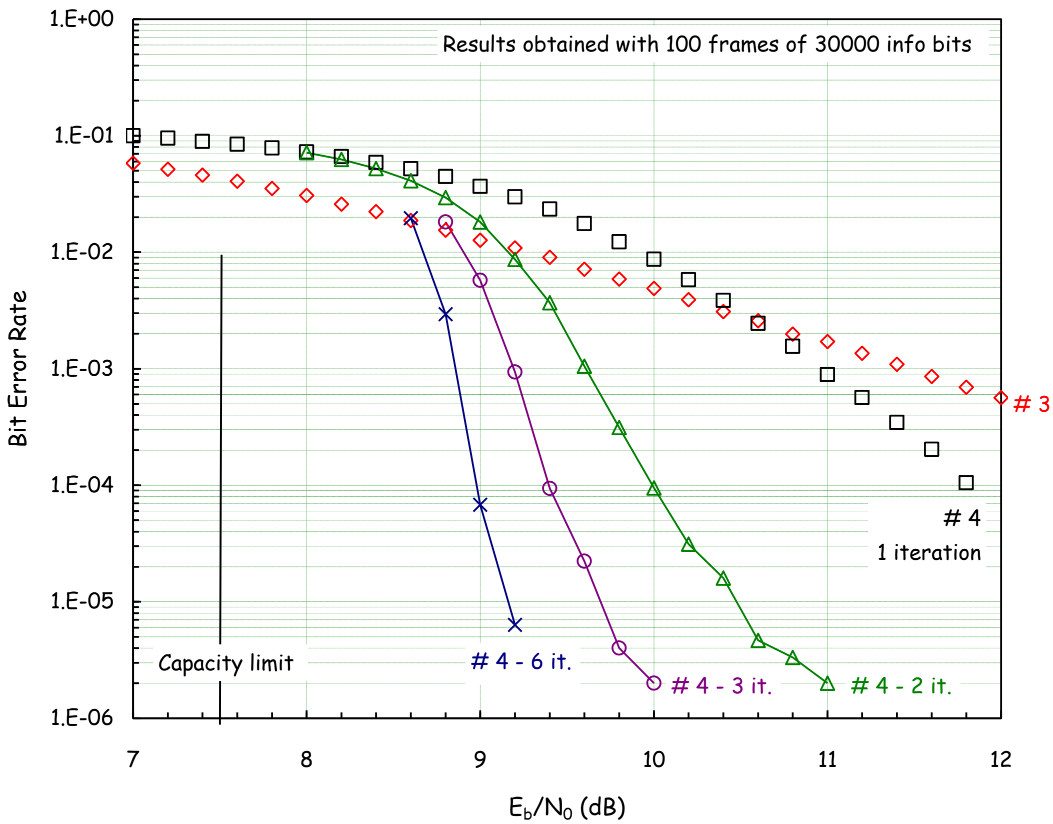
It is identical to system # 3, except that the convolutional code is replaced by a rate-3/4 turbo code generated by puncturing a rate-1/3 mother turbo encoder.

The mother turbo code is obtained by parallel concatenation of two identical rate-1/2 recursive and systematic convolutional (RSC) codes with 16 states (corresponding to a constraint length K = 5) and generator polynomials equal to (23, 31). Both RSC codes are separated by a 30K-bit random interleaving function (π).



Inside the turbo decoder, both RSC decoders use the soft-input soft-output Viterbi algorithm (SOVA) associated with the rate-1/2 encoders. A de-puncturing block is used prior to turbo decoding. Turbo decoding is performed using 1, 2, 3 or 6 iterations.





We observe from the BER plots that using a turbo code instead of a convolutional code results in a very significant performance improvement. For instance, at BER = 10^{-3}, system # 4 with 6 decoding iterations outperforms system # 3 by approximately 2.6 dB. With system # 4 using 6 decoding iterations, error performance within 1.7 dB of the channel capacity limit can be achieved (if BER = 10^{-5} is taken as a reference). With a larger number of decoding iterations, we could show that a BER = 10^{-5} is obtained for $E_b/N_0 \approx 8.6$ dB, i.e. ≈ 1.1 dB away from the capacity limit, which is an impressive performance indeed.

Matlab Assignment

Simulations of Wireless Communication Systems

The communication link shown in Fig. 1 is to be simulated in Matlab in order to investigate its bit error rate (BER) performance for several configurations.

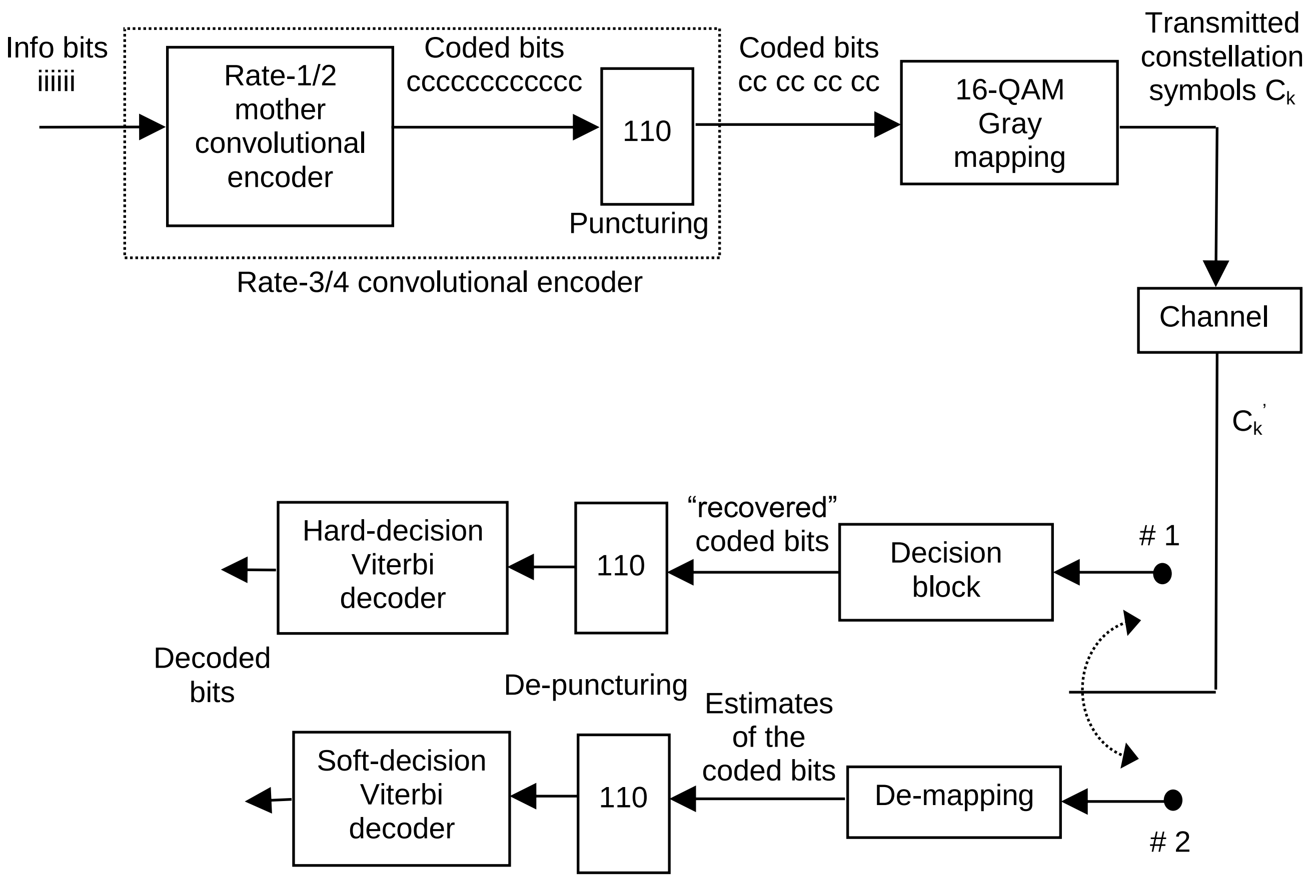


Fig. 1 – Generic structure of the communication link to be studied.

The sequence of information bits is first encoded by a convolutional encoder with a coding rate R_c = 3/4. The rate-3/4 encoder is obtained by puncturing a rate-1/2 mother convolutional encoder using the periodic mask [110] where 1 means "transmission of the coded bit" and 0 means "deletion of the coded bit". The generator polynomials of the rate-1/2 mother convolutional code are (7, 5) in octal representation, corresponding to a constraint length K = 3.

The coded bits are then modulated using 16-QAM modulation with Gray mapping. Fig. 2 shows a possible Gray mapping for 16-QAM.

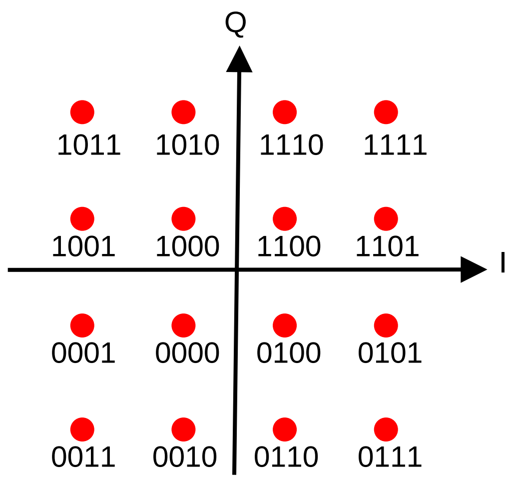


Fig. 2 – Possible Gray mapping for 16-QAM.

The spectral efficiency of the system is η = 3 bits/s/Hz, and is therefore identical to that achieved with an uncoded 8-PSK modulation scheme. The resulting sequence of 16-QAM complex symbols is then transmitted over the channel. Hereafter, the k-th transmitted symbol is denoted as C_k (like in my lecture notes).

The receiver can operate according to two configurations: # 1 and # 2 (see Fig. 1). In both configurations, each channel sample C'_k is processed in order to produce an estimate of the four bits that were associated with the transmitted symbol C_k.

In configuration # 1, a “decision block” generates a binary sequence (“hard decisions”) whereas, in configuration # 2, a “de-mapping” block produces a sequence of analogue samples (“soft decisions”).

In any case, the sequence thus obtained is then de-punctured, which means that the estimates corresponding to the coded bits deleted by the puncturing function are replaced with neutral estimates (= 0, i.e. “halfway” between -1 and +1), and finally decoded by the Viterbi decoder associated with the rate-1/2 mother convolutional encoder used in the transmitter.

Question 1 --

Simulate the BER performance of the system with the following parameters:

(1) The channel is additive white Gaussian noise (AWGN): The k-th channel sample C_k' is given by $C_k' = C_k + N_k$, where C_k denotes the corresponding transmitted 16-QAM constellation symbol and N_k is a complex Gaussian noise sample with zero mean and variance σ^2 that is a function of the SNR per information bit (E_b/N_0).

(2) The receiver operates using configuration # 1.

For reference sake, you should also determine the BER performance of an uncoded 8-PSK modulation scheme over AWGN channel. Remember that it is good practice to compare the error performance of a coded modulation scheme with that of an uncoded scheme with identical spectral efficiency.

In summary, you have to plot TWO "BER vs. E_b/N_0" curves on the same graph. Note that you should plot your results for E_b/N_0 ranging from 2 to 12 dB (by steps of 1 dB) and BER ranging from 10^0 to 10^{-5}.

It is worthwhile mentioning that, in order to obtain a reasonably accurate BER value via computer simulations, it is necessary to count at least ≈ 30 error events at the receiver output. Throughout this assignment, you will consider the transmission of 1000 frames, each of them composed of 3000 information bits. In other words, your computer simulations will be performed assuming the transmission of 3 million bits. With such number of information bits, you should be able to accurately measure any BER values higher than approximately 10^{-5}.

Comment on your results.

A bit of help: Matlab program for uncoded 8-PSK (with a few missing bits and pieces)

```
% This program simulates the BER performance of uncoded 8-PSK modulation over either AWGN or flat
% Rayleigh fading channel with perfect channel interleaving and CSI.

clear all;
close all;
clc;

% Range of SNR to be tested, (snr is the SNR per info bit (Eb/N0) expressed
% in dB)
snr_dB=2:12;
```

```
% Number of information bits per frame
length_frame=3000;

% Number of frames
number_frames=1000;

% Total number of information bits
number_bits=???*???;

fprintf(' BER performance of uncoded 8PSK\n');
fprintf(' Number of bits per frame = %d\n',length_frame);
fprintf(' Number of frames = %d\n',number_frames);
fprintf(' Number of info bits = %d\n',number_bits);

% Constellation and Gray mapping
[signal,bit]=constellation();

% Initial result vector
BER=zeros(1,length(snr_dB));

for i=1:length(snr_dB)
   snr=snr_dB(i);
   fprintf('\n');
   fprintf('=============== SNR (in dB) = %d\n',snr_dB(i));
   fprintf('\n');

   for frame=1: ???

 % Random vector of information bits
       msg = round(rand(1,???));

 % Mapping and transmission through the AWGN or Rayleigh fading channel
       [r1,r2,h1,h2] = transmission(length_frame,signal,bit,snr,msg);

 % Decision block
       demod = demodulation(length_frame,signal,bit,r1,r2,h1,h2);

 % Error count
       [number_errors(frame),ratio] = biterr(???,???);
   end

 % Computation of the total number of errors and BER
   sum_errors = sum(number_errors);
   BER(i) = ???/ ???;
   fprintf('Total number of errors = %d\n',sum_errors);
   fprintf('BER = %12.8f\n',BER(i));

end

figure(1);
semilogy(snr_dB,BER,'r-');
axis([2 16 10^-4 1]);
legend('8-PSK over AWGN (hard-decision)');
xlabel('EbN0 - dB');
```

```
ylabel ('BER');
grid on;



function [signal,bit]=constellation()

% Definition of the constellation and mapping

signal(1)=1;
bit(1,1)=1;
bit(2,1)=1;
bit(3,1)=1;

signal(2)=0.7071+1i*0.7071;
bit(1,2)=0;
bit(2,2)=1;
bit(3,2)=1;

signal(3)=1i;
bit(1,3)=0;
bit(2,3)=1;
bit(3,3)=0;

signal(4)=-0.7071+1i*0.7071;
bit(1,4)=0;
bit(2,4)=0;
bit(3,4)=0;

signal(5)=-1;
bit(1,5)=0;
bit(2,5)=0;
bit(3,5)=1;

signal(6)=-0.7071-1i*0.7071;
bit(1,6)=1;
bit(2,6)=0;
bit(3,6)=1;

signal(7)=-1i;
bit(1,7)=1;
bit(2,7)=0;
bit(3,7)=0;

signal(8)=0.7071-1i*0.7071;
bit(1,8)=1;
bit(2,8)=1;
bit(3,8)=0;
```

```
function [r1,r2,h1,h2]=transmission(length_frame,signal,bit,snr,msg)

sd=sqrt(???);
r1=zeros(1,length_frame/3);
r2=zeros(1,length_frame/3);
h1=zeros(1,length_frame/3);
h2=zeros(1,length_frame/3);

% Serial-to-parallel conversion (1:3)
for ii=1:length_frame/3
   a=msg(3*(ii-1)+1);
   b=msg(3*(ii-1)+2);
   c=msg(3*(ii-1)+3);

% Selection of the transmitted signal (mapping)
   for jj=1:8
      e=bit(1,jj);
      f=bit(2,jj);
      g=bit(3,jj);
      if ((a==e)&&(b==f)&&(c==g))
        transmitted=???;
      end
   end

% Generation of the complex AWGN samples
   noise=sd*randn(1)+1i*sd*randn(1);

% Generation of the fading samples h1 and h2
% (1) For AWGN channels
   h1(ii)=1;
   h2(ii)=1;

% (2) For Rayleigh fading channels
%   noise1=sqrt(0.5)*randn(1);
%   noise2=sqrt(0.5)*randn(1);
%   h1(ii)=sqrt(noise1^2 + noise2^2);
%   noise1=sqrt(0.5)*randn(1);
%   noise2=sqrt(0.5)*randn(1);
%   h2(ii)=sqrt(noise1^2 + noise2^2);

% Expression of the received signal as a function of the transmitted signal
   r1(ii)=h1(ii)*real(???) + real(???);
   r2(ii)=h2(ii)*imag(??? ) + imag(???);

end
```

```
function demod=demodulation(length_frame,signal,bit,r1,r2,h1,h2)

demod=zeros(1,length_frame);

% Computation of the Euclidean distance between the received signal and the 8
% possible 8-PSK signals. Note that the receiver knows the value of the fading
% samples h1 and h2.

for ii=1:length_frame/3
   a=r1(ii);
   b=r2(ii);
   for jj=1:8
      c=h1(ii)*real(signal(jj));
      d=h2(ii)*imag(signal(jj));
      distance(jj)=(???)^2+(???)^2;
   end

% The signal closest to the received signal is chosen
   closest=find(distance==min(distance));

% The corresponding bits are "extracted"
   demod(1,3*(ii-1)+1)=bit(1,???);
   demod(1,3*(ii-1)+2)=bit(2,???);
   demod(1,3*(ii-1)+3)=bit(3,???);

end
```

More help: Matlab program for coded 16-QAM using configuration # 1 (with even more missing bits and pieces).

```
% The program below simulates the BER performance of a 16-QAM modulation scheme
% combined with a rate-3/4 convolutional code. The channel is AWGN. The convolu-
% tional code is obtained by puncturing the coded bits generated by a rate-1/2
% mother encoder with 4 states and polynomials (7, 5). The spectral efficiency of
% the system is equal to 3 bits/s/Hz and thus identical to that of uncoded 8-PSK.
% The decision block generates bits to be decoded by the hard-decision Viterbi
% decoder after depuncturing.

clear all;
close all;
clc;

% Range of SNR to be tested, (snr is the SNR per info bit (Eb/N0) expressed
% in dB)
snr_dB=2:12;

% Number of information bits per frame
length_frame = 3000;

% Number of frames
number_frames = 1000;
```

```
% Total number of information bits
number_bits=???*???;

fprintf(' BER performance of a 3-bit/s/Hz coded 16-QAM\n');
fprintf(' Number of bits per frame = %d\n',length_frame);
fprintf(' Number of frames = %d\n',number_frames);
fprintf(' Number of info bits = %d\n',number_bits);

% Constellation and Gray mapping
[signal,bit] = constellation();
 BER=zeros(1,length(snr_dB));

for i=1:length(snr_dB)
    snr=snr_dB(i);

    fprintf('\n');
    fprintf('=============== SNR (in dB) = %d\n',snr);
    fprintf('\n');

    for frame=1:???

 % Random vector of information bits
        msg = round(rand(1,???));

 % Definition of the convolutional code
        t = poly2trellis(3, [???,???]);

 % Encoding of the information bits and puncturing
        coded = convenc(???, t, [???]);

 % Modulation and transmission through the AWGN channel
        [r1,r2] = transmission(length_frame,signal,bit,snr,coded);

 % Decision block
        demod = demodulation(length_frame,signal,bit,r1,r2);

 % Depuncturing and Viterbi decoding
        decoded = vitdec(???, t, 100, 'trunc', 'hard', [???]);

 % Error count
        [number_errors(frame),ratio] = biterr(???,???);
    end

 % Computation of the total number of errors and BER
    sum_errors = sum(number_errors);
    BER(i) =???/???;
    fprintf('Total number of errors = %d\n',sum_errors);
    fprintf('BER = %12.8f\n',BER(i));

end

figure(1);
semilogy(snr_dB,BER,'r-');
axis([2 12 10^-4 1]);
```

```
legend('16-QAM(soft-decision)')
xlabel('EbN0 - dB');
ylabel ('BER');
grid on;



function[signal,bit]=constellation()

% Definition of the constellation and mapping

signal(1)=1+1i*1;
bit(1,1)=1;
bit(2,1)=1;
bit(3,1)=0;
bit(4,1)=0;

signal(2)=3+1i*1;
bit(1,2)=?;
bit(2,2)=?;
bit(3,2)=?;
bit(4,2)=?;

signal(3)=1+1i*3;
bit(1,3)=1;
bit(2,3)=1;
bit(3,3)=1;
bit(4,3)=0;

signal(4)=3+1i*3;
bit(1,4)=?;
bit(2,4)=?;
bit(3,4)=?;
bit(4,4)=?;

signal(5)=-1+1i*1;
bit(1,5)=1;
bit(2,5)=0;
bit(3,5)=0;
bit(4,5)=0;

signal(6)=-3+1i*1;
bit(1,6)=1;
bit(2,6)=0;
bit(3,6)=0;
bit(4,6)=1;

signal(7)=-1+1i*3;
bit(1,7)=1;
bit(2,7)=0;
bit(3,7)=1;
bit(4,7)=0;

signal(8)=-3+1i*3;
bit(1,8)=1;
```

```
bit(2,8)=0;
bit(3,8)=1;
bit(4,8)=1;

signal(9)=-1-1i*1;
bit(1,9)=?;
bit(2,9)=?;
bit(3,9)=?;
bit(4,9)=?;

signal(10)=-3-1i*1;
bit(1,10)=0;
bit(2,10)=0;
bit(3,10)=0;
bit(4,10)=1;

signal(11)=-1-1i*3;
bit(1,11)=?;
bit(2,11)=?;
bit(3,11)=?;
bit(4,11)=?;

signal(12)=-3-1i*3;
bit(1,12)=0;
bit(2,12)=0;
bit(3,12)=1;
bit(4,12)=1;

signal(13)=1-1i*1;
bit(1,13)=0;
bit(2,13)=1;
bit(3,13)=0;
bit(4,13)=0;

signal(14)=3-1i*1;
bit(1,14)=0;
bit(2,14)=1;
bit(3,14)=0;
bit(4,14)=1;

signal(15)=1-1i*3;
bit(1,15)=0;
bit(2,15)=1;
bit(3,15)=1;
bit(4,15)=0;

signal(16)=3-1i*3;
bit(1,16)=?;
bit(2,16)=?;
bit(3,16)=?;
bit(4,16)=?;
```

```
function[r1,r2] = transmission(length_frame,signal,bit,snr,coded)

sd=sqrt((???)*(10^(-snr/10)));
r1=zeros(1,length_frame/3);
r2=zeros(1,length_frame/3);

% Serial-to-parallel conversion (1:4)
for ii=1:length_frame/3
   a=coded(4*(ii-1)+1);
   b=coded(4*(ii-1)+2);
   c=coded(4*(ii-1)+3);
   d=coded(4*(ii-1)+4);

% Selection of the transmitted signal (mapping)
   for jj=1:16
      e=bit(1,jj);
      f=bit(2,jj);
      g=bit(3,jj);
      h=bit(4,jj);
      if((a==e)&&(b==f)&&(c==g)&&(d==h))
         transmitted=signal(jj);
      end
   end

% Generation of complex AWGN samples
   noise=sd*randn(1)+1i*sd*randn(1);

% Expression of the received signal as a function of the transmitted signal
   r1(ii)=real(???)+real(???);
   r2(ii)=imag(???)+imag(???);

end



function demod = demodulation(length_frame,signal,bit,r1,r2)

demod=zeros(1,length_frame);

% Computation of the Euclidean distance between the received signal and the 16 possible 16-QAM signals

for ii=1:length_frame/3
   a=r1(ii);
   b=r2(ii);
   for jj=1:16
      c=real(signal(jj));
      d=imag(signal(jj));
      distance(jj)=(???)^2+(???)^2;
   end

% The signal closest to the received signal is chosen.
   closest=find(distance==min(distance));

% The corresponding bits are "extracted".
   demod(1,4*(ii-1)+1)=bit(1,???);
```

```
   demod(1,4*(ii-1)+2)=bit(2,???);
   demod(1,4*(ii-1)+3)=bit(3,???);
   demod(1,4*(ii-1)+4)=bit(4,???);

end
```

Question 2 ---

Simulate the BER performance of the system with the same parameters as in Question 1, with one major exception: the receiver now operates using configuration # 2.

This basically allows the Viterbi algorithm to operate from soft decisions instead of hard decisions, and should thus improve its performance in terms of error correction. Remember that information theory tells us that a channel decoder always performs better when using soft decisions rather than hard estimates (bits).

In this question, the main challenge is to devise an algorithm that can convert the estimate C_k' of the transmitted 16-QAM complex signal C_k into estimates of the four bits that are associated with such signal. The reason for this is that the Viterbi decoder is a binary decoder and must thus operate from estimates of bits. Unfortunately, those estimates are not available at the channel output, and must therefore be computed using the received sample C_k'.

It is interesting to compare the results obtained in Questions 1 and 2. Hence, make sure to plot the new “BER vs. E_b/N_0” curve and the curves obtained in Question 1 on the same graph. Thus, the graph for Question 2 should therefore show THREE BER curves.

Make sure to plot your results for E_b/N_0 values ranging from 2 to 12 dB (by steps of 1 dB) and BER ranging from 10^0 to 10^{-5}.

Comment on your results.

Question 3 ---

Consider from now on (i.e., for Questions 3, 4, 5, and 6) that the receiver always operate using configuration # 2.

Simulate the BER performance of the system studied in Question 2 by now assuming transmission over a flat Rayleigh fading channel with perfect channel interleaving instead of an AWGN channel.

Note that you do not need to plot any BER curve for Question 3.

Comment on your results.

Question 4 ---

We are now going to try to improve the error performance of the scheme considered in Question 3. The way to do it is to assume that the 16-QAM de-mapping block knows the value of the fading samples $h_{k,1}$ and $h_{k,2}$ associated with each channel sample C_k'. In technical terms, we say that we have perfect “channel state information”. Note that this is actually a realistic and common assumption since various techniques can be used by a practical receiver to estimate the value of $h_{k,1}$ and $h_{k,2}$ (“channel estimation techniques”).

In this question, your task is to devise a de-mapping algorithm able to exploit the knowledge of $h_{k,1}$ and $h_{k,2}$ to convert the sample C_k' into estimates of the four bits that are associated with C_k.

Once it is done, simulate the error performance of the whole scheme and plot the resulting “BER vs. E_b/N_0” curve. As a reference, you must also display on the same graph the curve corresponding to the uncoded 8-PSK system (also assuming Rayleigh fading channel with knowledge of $h_{k,1}$ and $h_{k,2}$, for fair comparison).

The graph for Question 4 should show TWO curves. Note that you should plot your results for E_b/N_0 ranging from 6 to 20 dB by steps of 1 dB and BER ranging from 10^0 to 10^{-5}.

Comment on your results.

Question 5 --

The scheme studied in Question 4 should work fine, but it is possible to further improve its error performance by employing an interleaving function (denoted as π) between channel encoding and modulation, as shown in Fig. 3 below.

Such interleaving function "breaks" the correlation between successive estimates at the demapper output, and thus allows for optimal operation of the Viterbi decoder. In fact, it is well known that the error correction capabilities of Viterbi decoders can be significantly degraded when they are fed with correlated estimates.

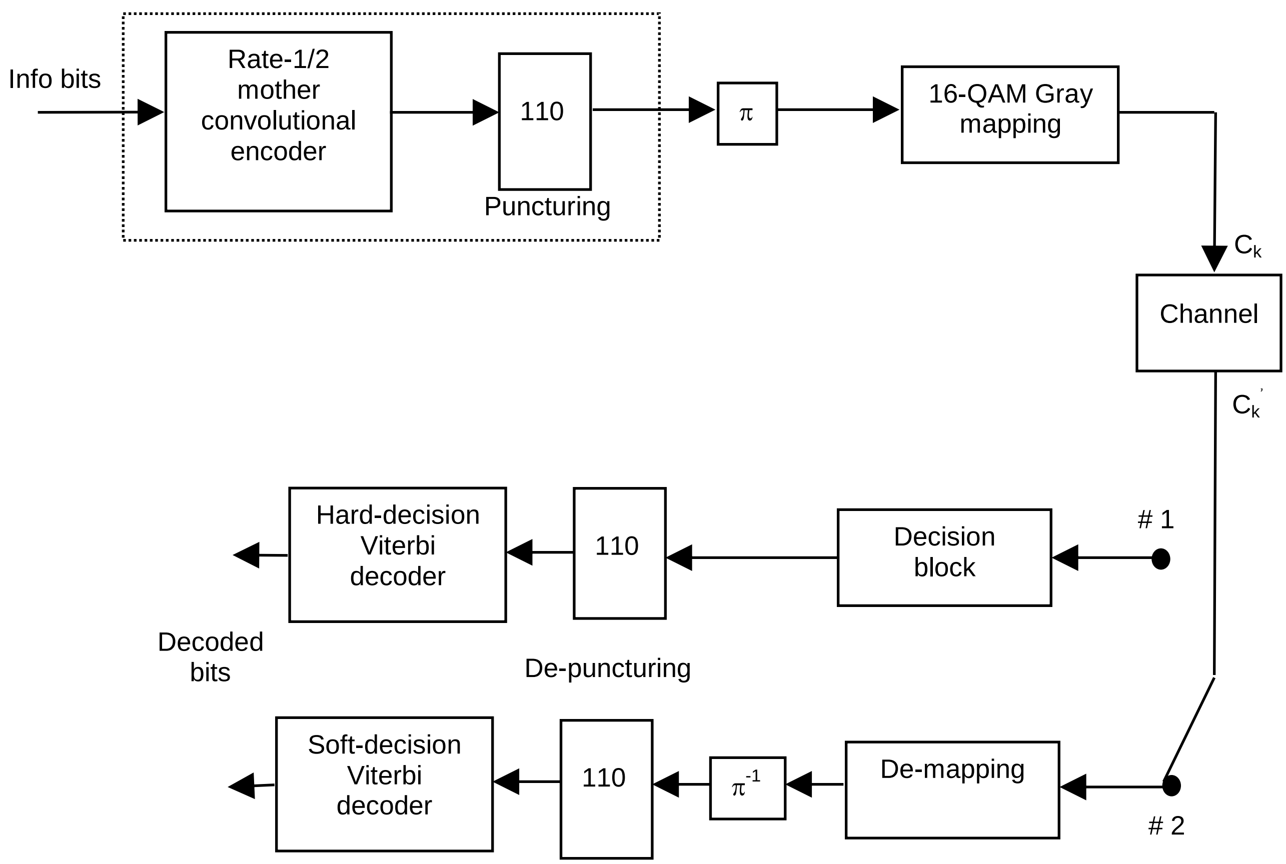


Fig. 3 – Improved structure of the communication link.

You can choose to use a random interleaving function as it is the simplest choice, at least in terms of Matlab programming effort.

Simulate the error performance of the "improved" scheme, and plot the resulting "BER vs. E_b/N_0" curve. As references, you must also display on the same graph both curves obtained in Question 4 corresponding to the same system without interleaving as well as uncoded 8-PSK system. The graph for Question 5 should show THREE curves.

Note that you should plot your results for E_b/N_0 ranging from 6 to 20 dB by steps of 1 dB and BER ranging from 10^0 to 10^{-5}.

Comment on your results.

Question 6 --

Question 6 is a "free-style" exercise.

Assume that our initial target was to design a 3-bit/s/Hz communication system that can achieve a BER of 10^{-5} for $E_b/N_0 \leq 14$ dB over a flat Rayleigh fading channel with perfect channel interleaving and perfect channel state information. Using our best communication system so far (hopefully, the scheme simulated in Question 5), we have unfortunately not been able to reach our goal.

Modify the system in Question 5 so as to meet the initial target (BER = 10^{-5} for $E_b/N_0 \leq 14$ dB). You are allowed to change the error-correcting code, modulation scheme, puncturing pattern, etc., and even add a few tail bits to each frame. Obviously, the channel remains unchanged, i.e. flat Rayleigh fading channel with perfect channel interleaving and channel state information.

Finally, make sure to reach the target at a minimal cost in terms of complexity increase.

- END -