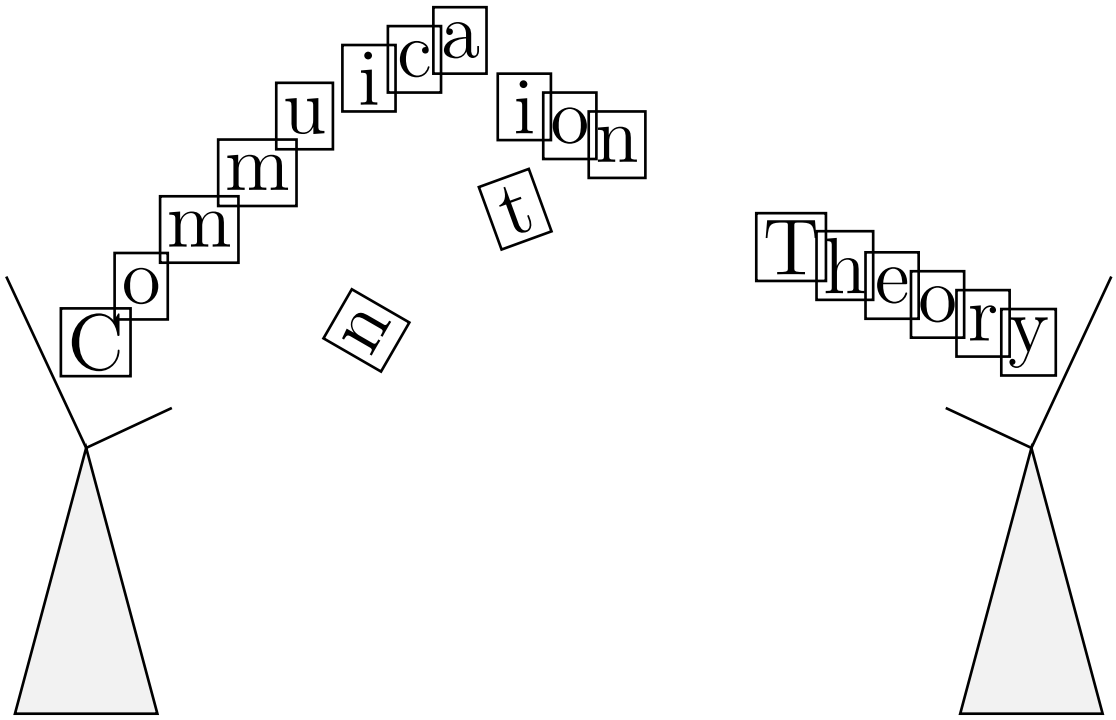


Manuel A. Vázquez

January 30, 2024

Contents

Acknowledgments

This book is in part derived from class notes by Matilde Sánchez and Marcelino Lázaro, so kudos to them.

Errata

If you find any error, typo,...I'd appreciate it if you drop me a line at manuvazquez@protonmail.com.

Notation

$\underline{\underline{A}}$ Matrix A

$\underline{a}$ Vector a

BSC binary symmetric channel

DMC discrete memoryless channel

$\langle \underline{a}, \underline{b} \rangle$ inner product between vectors $\underline{a}$ and $\underline{b}$

pdf probability density function

pmf probability mass function

PSD power spectral density

r.v. random variable

SNR signal-to-noise ratio

WSS wide-sense stationary

Chapter 0

Introduction

The book is articulated around four big blocks:

1. Noise in communications systems: stochastic processes, white noise, SNR
2. Modulation and detection in Gaussian channels: information modulation, demodulation and detection, error probability
3. Fundamental limits in communications
4. Analog modulations

We will deal with each one at its due time in a different chapter, but before that we spend a few pages here to give some context.

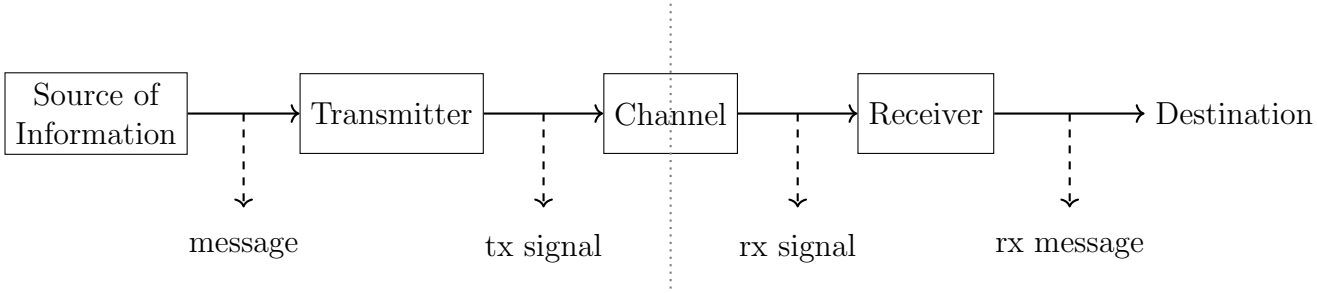
0.1 Communications systems

The goal in any communications systems is the transmission of information between two points that are somehow connected by any physical structure (either natural or artificial)[1]. This physical structure might be a cable, air, empty space (satellite). As we all know, communications systems have plenty of applications, e.g.,

- cellphone - base station
- base station - TV
- peer-to-peer
- radio
- streaming
- ...plenty more

When focusing on functionality, the block diagram of a general communications system is

[1]Transmission is the process of sending information from one point, the origin, to another, the destination by means of a transmission medium or channel.



The *message* (physical manifestation of information) may be a signal (waveform) or a symbol.

We'll study every block in the diagram separately.

0.1.1 Source of information

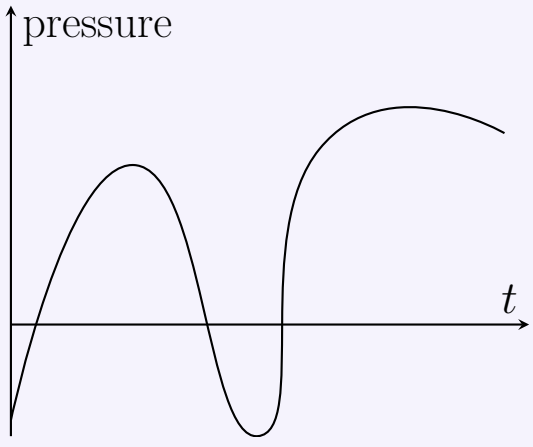
It aims at communicating or reporting something (news). Information lies in the messages and, for us, a message is the physical manifestation of the information produced by the source. Information can be in many different formats: voice, text, images... According to the format of the information they produce, sources can be classified in analog and digital.

Analog source

It produces messages (recall that a message is something tangible, something physical) that are modeled as a continuous waveform.

Some examples

- variation in air pressure when we talk (that is, the voice)



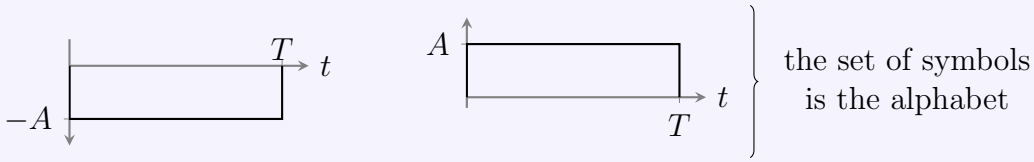
- temperature variation in a sensor

The communications system should transmit that waveform with a certain accuracy/**fidelity** (later on, we will see a way of measuring this).

Digital source

Information lies in *symbols* belonging to a **finite** set called the alphabet, and they are sent in a finite time interval.

Example of an an alphabet encompassing two symbols



Transmitting the symbol on the left means one thing (e.g. *yes* or bit "1"), and transmitting the one on the right another thing (e.g. *no* or bit "0").

In this course a symbol is simply a finite-duration signal.

The (digital) communications system must transmit the messages with a certain accuracy or **probability of error**. We'll come back to this later.

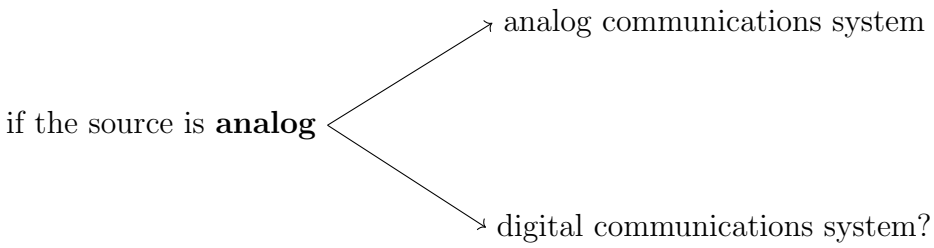
The difference between the messages produced by analog and digital sources suggest that there are two kinds of communications systems:

- analog: old TV, radio (probably not for long...already shut down in Norway)
- digital: cell phones, ADSL (internet connections)...mostly everything else

They have different goals and this shows in the design

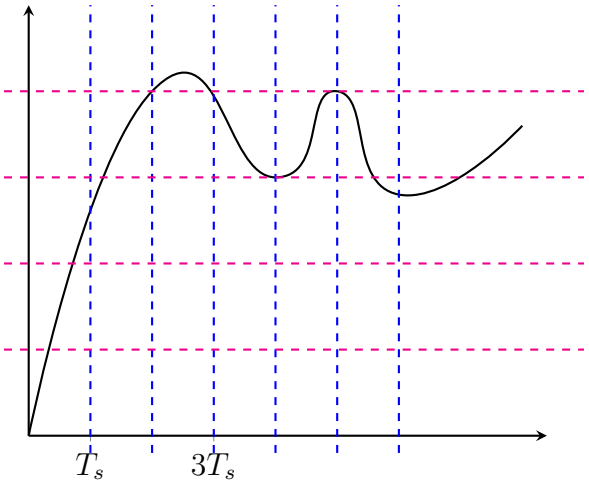
if the source is **digital** ⟶ digital communications system

(no other possibility, we cannot *make up* a continuous waveform from a sequence of symbols)



Can we use a digital system to transmit information from an analog source?

Yes, we can!! by digitizing the information



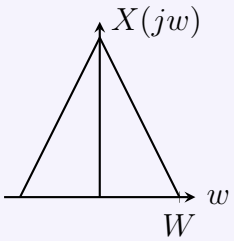
We discretize both

- the time axis
  - this is called *sampling*
  - it is reversible if the Nyquist condition holds, in which case no information is lost
- the amplitude axis
  - this is called *quantization*
  - it is **not** reversible, i.e., **information is lost**

0.1.2 Transmitter

The transmitter is the device/element that shapes up the information coming from the source so that it can traverse the channel. Is this thing necessary? Can't we transmit the symbols (digital communications system) or the waveform (analog communications system) as they are? In general we can't: signals must have certain properties in order to traverse the channel. Hence, **the transmitter must have some knowledge about the channel**. For starters, the transmitter needs to know whether the system is analog or digital but, moreover, a *modulator* is usually necessary.

Example

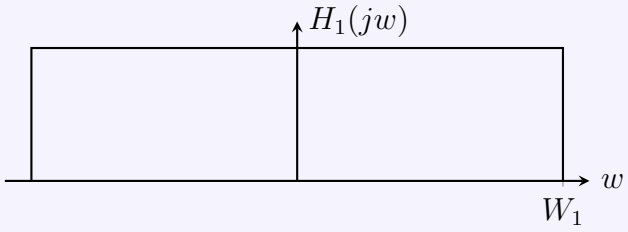


- baseband signal
- bandwidth[a] is W

[a] length of the interval of positive frequencies associated with non-zero values

Let us consider two different transmission channels

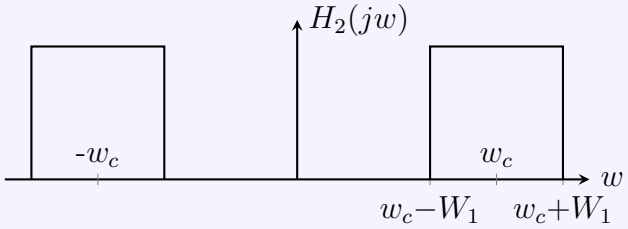
- Channel #1



– baseband channel

– bandwidth is W_1

- Channel #2



– passband channel

– bandwidth is $2W_1$

– w_c is the center frequency

The channel is modeled as an LTI system

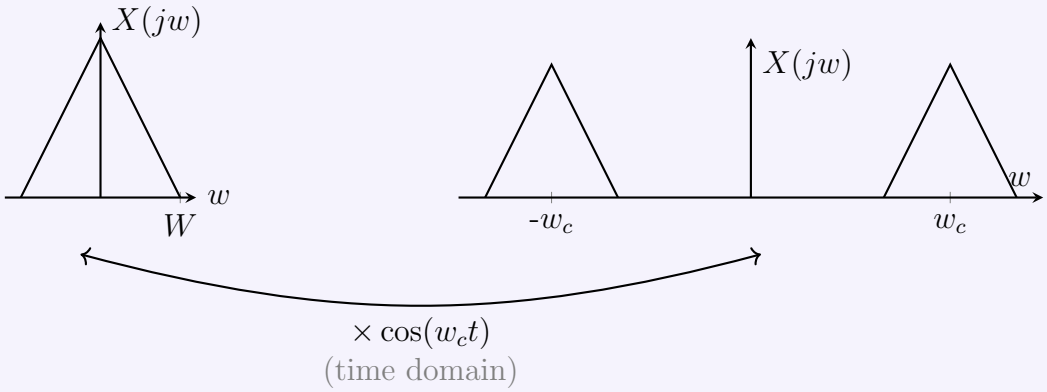
$$\xrightarrow[X(jw)]{x(t)} \boxed{\begin{matrix} h(t) \\ H(jw) \end{matrix}} \xrightarrow[Y(jw)=X(jw)H(jw)]{y(t)=x(t)*h(t)}$$

Can signal $x(t)$ go through the baseband channel? and through the passband channel?

- it can travel the baseband channel
  - without distortion if $W_1 > W$
  - **with** distortion if $W_1 < W$ (information loss)

  ...and we have *baseband transmission*, which means, transmitting the signal keeping its original frequency band (as it was)

- signal $x(t)$ **cannot** travel the passband channel as it is...we have to use an auxiliar signal to allow the transmission...the so-called carrier signal (sinusoid)



...and we have *passband transmission*

This multiplication by a cosine (or sine) to shift the signal from one frequency band to another is called *modulation*. Besides serving the purpose of moving the modulated signal to another frequency band better fit for transmission, it is also useful to

- reduce the noise and interference and, especially,
- share the channel among users: frequency division multiple access (FDMA)

0.1.3 Channel

The channel is the physical medium though which information propagates. It may cause disturbances and distortions:

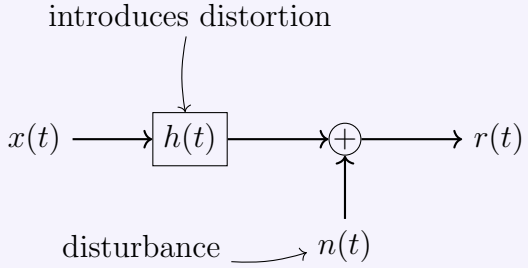
disturbances: noise and/or interfering signals contaminate (get *added* to) the transmitted signal

- noise is a random signal (so, we don't know its behavior) that gets added to the transmitted signal (additive noise)
- interference occurs when signals coming from another communications system (or a different user in ours!!) get mixed with our signal (for us, they are a nuisance, undesirable...but they are of interest for someone else)

distortions: linear or non-linear modification of the waveform; they depend on the signal itself and are caused by a non-ideal channel

- linear: $\exists h(t)$ that serves to characterize it $\rightarrow$ channel is LTI
- time-varying linear: a different $h(t)$ for every time instant
- non-linear: $\nexists h(t)$ characterizing it

Example: channel introducing distortions and disturbances



Noise is always added at the end.

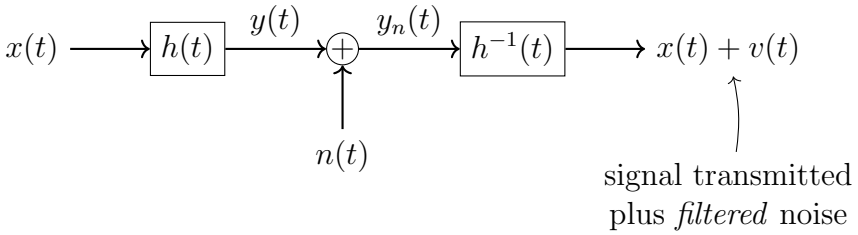
In this course we will assume that there are no distortions, and we will only have to deal with disturbances that are Additive White Gaussian noise (AWGN).

0.1.4 Receiver

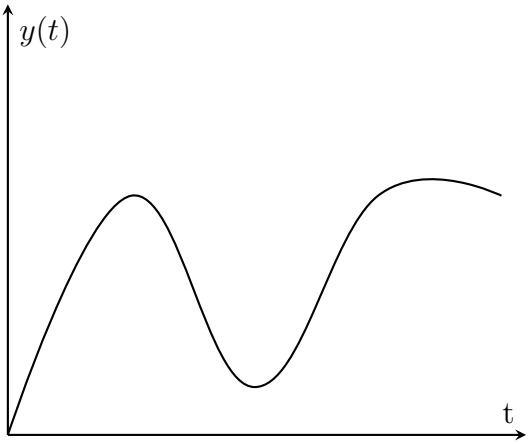
It must recover the information transmitted as faithfully as possible. Tasks:

1. Demodulate, i.e., carry the signal back to its original frequency band
2. Reject disturbances
3. Fix channel distortions whenever possible

Ideally, we would like to find $h^{-1}(t)$ such that



2 and 3 are challenges in an analog system...why? If we receive the signal

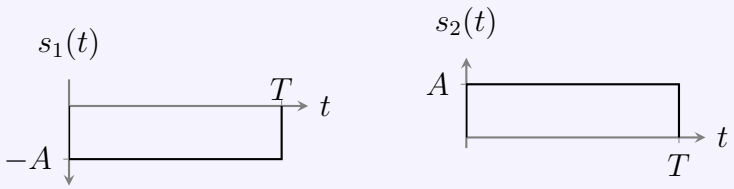


Is this the signal transmitted? **We don't know!!** In an analog system we can transmit *any* signal, and hence whatever we receive might well be the signal transmitted (no matter how weird or twisted it is).

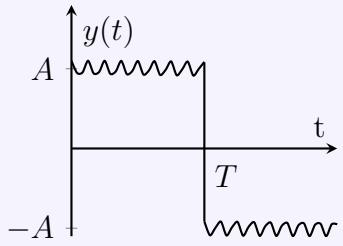
On the other hand, in a digital signal we cannot transmit whatever we want.

Digital communications system with two symbols in the alphabet

We have to choose the signal to transmit among a set of possible *symbols* (the alphabet).



If we receive



we know that the transmitted signal has been affected by disturbances/distortions (errors occurred) because the transmitter is not allowed to transmit that kind of signal. However, it is easy to guess what was transmitted: first signal $s_2(t)$ and then $s_1(t)$.

The point is: in a digital communications system not every signal is possible.

0.2 Design of a system: quality metrics

When designing a system, there are several aspects to take into account (some of them are limitations)

- Quality
- Available technologies
- Cost
- Resources consumption

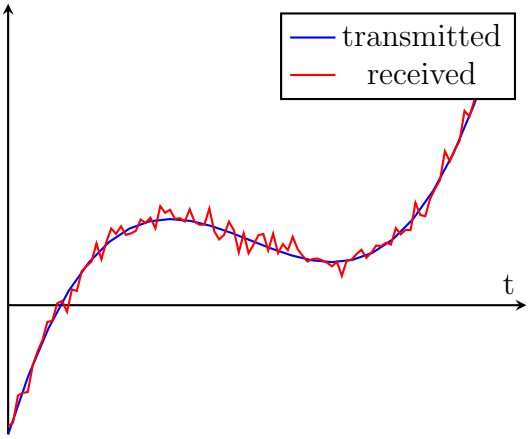
We briefly talk about every one of them.

0.2.1 Quality

We need a measure of the quality of a system so that we can design it properly and compare it against others. This metric is different depending on whether the system is analog or digital.

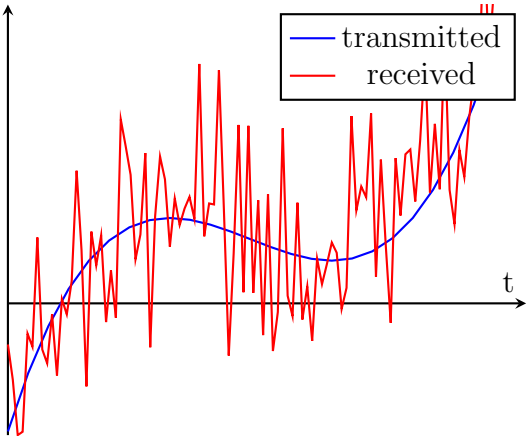
analog systems

The metric is fidelity: it measures how the received signal resembles the original one



$$y(t) = x(t) + n(t)$$

fidelity is fine: the received signal resembles the transmitted one



$$y(t) = x(t) + n(t)$$

fidelity is **not** fine: the received signal does not resemble the transmitted one (although you can see the trend)

On the left there is little noise whereas on the right, the noise nearly hides the signal.

In general, the higher the power of the signal as compared to the power of the noise, the more closely the received signal will resemble the original. In order to have a quantitative measure of this, we define the signal-to-noise ratio (SNR) as

$$\frac{\mathbf{S}}{\mathbf{N}} \quad \begin{matrix} \rightarrow \text{ power of the signal} \\ \rightarrow \text{ power of the noise} \end{matrix}$$

In an analog system it is important to analytically compute this parameter.

There are other parameters impacting the quality of an analog system, such as the bandwidth of the transmitted signal

- ↑ bandwidth ⇒ ↑ quality

telephone AM radio FM radio HiFi system

− quality → + quality

digital systems

The metric is the accuracy or **error probability**. The receiver knows which symbols (signals) are possible/feasible: we can tally up how many symbols were correctly received.

- ↑ quality ⇒ ↓ probability of error (P_e)

Just like the fidelity for analog systems, the SNR affects the quality of the received signal, and hence it is important. The bandwidth is also important

- ↑ bandwidth ⇒ ↑ quality (ADSL, HDTV)

0.2.2 Resources consumption

We need to monitor the resources we are using because they don't come for free

- can we take up as much bandwidth as we like?
- how much transmission power is too much? (health factors, other systems deployed in the same space)

Chapter 1

Noise in communication systems

1.1 Cyclostationarity

Cyclostationary processes are **non-stationary** processes that often show up in communications (in connection with modulation and demodulation operations). Cyclostationarity (just like stationarity) comes in two flavors. A process is:

strictly cyclostationary with *period* T if and only if

$$\forall n,\ \forall\,(t_1, t_2, \cdots, t_n)$$

$$f_{X(t_1),X(t_2),\cdots,X(t_n)}\,(x_1, x_2, \cdots, x_n) = f_{X(t_1+T),X(t_2+T),\cdots,X(t_n+T)}\,(x_1, x_2, \cdots, x_n)$$

This condition is much less restrictive than the one for strict-sense stationarity: we do not require the condition to hold for any shift Δ, but only for a certain value, which is precisely the period T.

wide-sense cyclostationary with period T if and only if

$$\begin{aligned} \mu_X(t) &= \mathbb{E}\left[X(t)\right] = \mu_X(t+T) \\ R_X(t_1, t_2) &= R_X(t_1+T, t_2+T) \end{aligned}$$

The condition for the autocorrelation can be rewritten using the alternative notation

$$R_X(t+\tau, t) = R_X(t+\tau+T, t+T)$$

Again, we only impose the condition on the mean and autocorrelation function.

(Wide-sense) cyclostationary process

Both conditions above are about *periodicity*: a process is cyclostationary if

- the mean is periodic, and
- the autocorrelation function is periodic **with respect to time,** t (and not τ).

Caveat

Notice that a cyclostationary process is **not** stationary: its autocorrelation is a function of *two* variables.

WSS...and also cyclostationary?

If a process is WSS, then it's also cyclostationary, i.e.,

$$\text{WSS} \Rightarrow \text{cyclostationary}$$

(the converse is, of course, not true). The proof is straightforward:

- if the mean is constant (WSS), then it is also periodic with any period you like (cyclostationary), and
- if the autocorrelation only depends on τ (WSS), then it is periodic with **respect to** t, since it doesn't even depend on t (cyclostationary).

This means that wide-sense stationarity is a stronger condition than cyclostationarity (and comes with nicer results and properties). Hence, in practice, after proving that some process is WSS, you don't even need to check whether it is also cyclostationary (it's given, but the implications have less of an impact).

Quick quiz

Consider a process, $X(t)$, whose autocorrelation function is $R_X(t_1, t_2) = e^{-|t_1 - t_2|}$. Is it autocorrelation-cyclostationary?[1]

Quick quiz

Consider a process $X(t)$ whose autocorrelation function is $R_X(t + \tau, t) = e^{-2|\tau|}$. $X(t)$ is clearly autocorrelation-stationary since $R_X(t + \tau, t)$ doesn't depend on t. Is it also autocorrelation-*cyclo*stationary? [2]

Identifying the autocorrelation function

The autocorrelation of *any* random process is defined as

$$R_X(t_1, t_2) = \mathbb{E}\left[X(t_1)X(t_2)^*\right].$$

If the process is WSS, then the autocorrelation only depends on the time difference and we write it as a function of a single variable, $R_X(\tau)$, with $\tau = t_1 - t_2$. Going from

[1] The question amounts to decide whether $R_X(t+\tau, t)$ is periodic with respect to t. $R_X(t+\tau, t)$ is constant with respect to t, and a constant function is periodic with any period we like. Hence, $X(t)$ is indeed autocorrelation-cyclostationary. Also notice that wide-sense stationarity is a stronger condition than cyclostationarity (and one yielding nicer results), and hence we have WSS $\Rightarrow$ cyclostationary.

[2] One can rewrite the autocorrelation in the more familiar τ-notation to get $R_X(t + \tau, t) = e^{-|\tau|}$ with $\tau = t_1 - t_2$ and $t = t_2$. This way, it's clear that the process is autocorrelation-cyclostationary: it's periodic with respect to t (it doesn't even depend on t, i.e., it's constant with respect to t).

left to right in the above equation is definition. It is equally easy to go the other way around (from right to left) to identify the autocorrelation function whenever we see the expectation of a random process at a certain time instant multiplied by itself evaluated at some other (or the same) time instant. For instance:

$$\mathbb{E}\left[X(t+2)X(t+3)^*\right] = R_X(\underbrace{t+2}_{t_1}, \underbrace{t+3}_{t_2}) \overset{\substack{X(t) \text{ is} \\ \text{WSS}}}{=} R_X(\underbrace{t+2}_{t_1} - (\underbrace{t+3}_{t_2})) = R_X(-1)$$

$$\mathbb{E}\left[Y(t+\tau-1)Y(t-2)^*\right] = R_Y(t+\tau-1, t-2) \overset{\substack{Y(t) \text{ is} \\ \text{WSS}}}{=} R_Y(\tau+1).$$

1.2 Power and energy of a process

The definitions of energy and power of a stochastic process are analogous to the ones we have for deterministic signals but with the expectation operator.

Energy ...intuitively we are just putting a expectation around the definition of energy of a signal

$$E_X = \mathbb{E}\left[\int_{-\infty}^{\infty} |X(t)|^2 \, dt\right] = \mathbb{E}\left[\int_{-\infty}^{\infty} X(t)X^*(t)dt\right]$$

(expectation of the integral is the integral of the expectation...we can swap the integral and expectation operators)

$$= \int_{-\infty}^{\infty} \underbrace{\mathbb{E}\left[X(t)X^*(t)\right]}_{R_X(t,t)} dt = \int_{-\infty}^{\infty} R_X(t,t)dt \overset{\text{WSS}}{=} \int_{-\infty}^{\infty} R_X(0)dt$$

Notice that the last part of the equation is only true for *WSS processes*.

Power

$$P_X = \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |X(t)|^2 \, dt\right]$$

(the expectation of the limit is the limit of the expectation...afterwards, we use the same steps as before)

$$= \lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\int_{-\frac{T}{2}}^{\frac{T}{2}} |X(t)|^2 \, dt\right] = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} R_X(t,t)dt \overset{\text{WSS}}{=} \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} R_X(0)dt$$

$$= \lim_{T\to\infty} R_X(0) \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} dt = \lim_{T\to\infty} R_X(0) \frac{1}{\cancel{T}} \cancel{T} = R_X(0)$$

When dealing with deterministic signals we have

energy signals: their energy is finite and their power zero

power signals: their power is finite and their energy infinity

We have the same classification here:

- a stochastic process is **energy-type** or **finite-energy** if $0 < E_X < \infty$
- a stochastic process is **power-type** or **finite-power** if $0 < P_X < \infty$

This applies to any stochastic process, not just WSS processes.

1.2.1 WSS processes

What about WSS processes? Are they finite-energy or finite-power? We can use the formulas we got before for that particular case...and we see they both depend on the **autocorrelation function at** 0,

$$E_X = \int_{-\infty}^{\infty} R_X(0)dt$$
$$P_X = R_X(0).$$

We have two scenarios:

- If $R_X(\tau) \neq \delta(\tau) \Rightarrow R_X(0) = \text{constant}$[3]

$$\left.\begin{array}{l} E_X = \int_{-\infty}^{\infty} R_X(0)dt = \infty \\ P_x = R_X(0) = \text{constant} \end{array}\right\} \text{finite-power random process}$$

- If $R_X(\tau) = \delta(\tau)$ (what is the meaning of this? the correlation between any two *different* instants is 0!!)[4]

$$\left.\begin{array}{l} E_X = \int_{-\infty}^{\infty} R_X(0)dt = \int_{-\infty}^{\infty} \infty dt = \infty \\ P_x = R_X(0) = \infty \end{array}\right\} ?$$

  It is **not** a finite-power process...nor a finite-energy process. We don't have a name for this.

  Notice that, in any case, the energy of a WSS process is **always** infinite.

In summary, WSS processes are mostly finite-power random processes...except in the tricky case in which $R_X = \delta(\tau)$, which yields a process with infinite energy and power (not a finite-energy process nor a finite-power one)

[3] Different from 0 because if $R_X(\tau) = \mathbb{E}\left[X^2(t)\right] = 0$, then

$$\sigma^2_{X^2(t)} = \mathbb{E}\left[X^2(t)\right] - \mathbb{E}\left[X(t)\right]^2 = -\mathbb{E}\left[X(t)\right]^2,$$

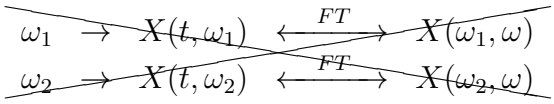
and the only possibility is that variance (and the expectation) be zero.

[4] A δ function might look like a weird function to have for autocorrelation, but we will encounter this later.

1.3 Spectral Characterization of Random Processes

A frequency-domain analysis of a stochastic process is very convenient, as we will see later[5]. How do we tackle this?

Recall that in a stochastic process we have a time signal for every outcome of the experiment. Then, the first thing that comes to mind is computing the Fourier transform of every deterministic signal encompassed by the stochastic process:



Here ω with a subindex refers to an outcome of the random experiment whereas without it refers to the independent variable for frequency (in radians per second). So, I see my random process as a bunch of signals and I compute the Fourier transform of each one...done!! In the end, we would get a new process that associates a spectrum to every outcome of the random experiment (from end to end we are mapping every outcome of the random experiment onto a frequency-domain signal), which is nice, and we could even compute the mean of all those FTs (if we want a single value for every frequency). However, this approach is not without problems...

Problem

It might be the case that the Fourier transform for a particular ω_i doesn't exist!

In order to overcome this difficulty, we define

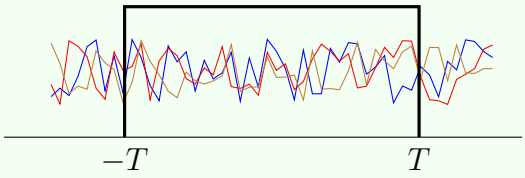
Definition 1.3.1: Power Spectral Density (PSD)

$$S_X(j\omega) = \lim_{T\to\infty} \frac{\mathbb{E}\left[|X_T(j\omega)|^2\right]}{2T}$$

where

$$X_T(j\omega) \xleftrightarrow{FT} X_T(t) = \begin{cases} X(t), & |t| < T \\ 0, & |t| \geq T \end{cases}$$

is the Fourier transform of the *truncated* process, $X_T(t)$ (every value of the signal after time $+T$ or before time $-T$ *is set* to 0).



That is, the PSD is the (time) *average* (given by the $1/2T$ factor) expectation of the squared modulus of the Fourier transform of the truncated process when the value at which we truncate approaches infinity. Why do we use this truncated process? Because, from the convergence properties of the FT, any finite signal (time-limited) has a Fourier

[5]For starters, we want to know how much bandwidth our stochastic process is going to take.

transform. Hence, the FT in the numerator *always* exists. Moreover, when T goes to ∞ there is no truncation and we recover the original process. As the name suggests, the PSD **measures the power of the process at every frequency**.

Notice that computing the PSD from the definition is not straightforward.

Caveat

The PSD is always positive (we are computing the expectation of the *squared modulus*).

1.4 Properties of the power spectral density

We have general properties, which hold for *any* random process, properties that only hold for WSS processes, and properties that only hold for cyclostationary processes.

general properties

$$P_X = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)d\omega$$
$$P_X^{\omega_1,\omega_2} = \frac{2}{2\pi}\int_{\omega_1}^{\omega_2} S_X(j\omega)d\omega$$

The first formula yields the power of any process given its PSD. The second one is the power in a particular frequency range[6].

if the process is WSS

$$S_X(j\omega) = \mathrm{FT}\left[R_X(\tau)\right] = \int_{-\infty}^{\infty} R_X(u)e^{-j\omega u}du \tag{1.1}$$

or, equivalently,

$$R_X(\tau) = \mathrm{FT}^{-1}\left[S_X(j\omega)\right] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)e^{j\omega\tau}d\omega \tag{1.2}$$

Notice that, for a WSS process, the autocorrelation function is a function of a single variable (that we usually denote as τ), and hence we have a regular (single variable) integral.

Notice that if we plug in 0 in (1.2), we get

$$R_X(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)d\omega = P_X,$$

which provides us with an easy way to compute the power of *WSS process* when the autocorrelation function is available.

[6]This is definition but, intuitively, we are considering this band of frequencies both in the positive and negative sides of the spectrum.

if the process is cyclostationary

$$S_X(j\omega) = \text{FT}\left[\tilde{R}_X(\tau)\right] \tag{1.3}$$

where

$$\tilde{R}_X(\tau) = \frac{1}{T_0}\int_0^{T_0} R_X(t+\tau, t)dt$$
$$T_0 \equiv \text{period of the cyclostationary process,}$$

is the average value of the autocorrelation function over a single period, T_0.

Notice that we are integrating with respect to t (not τ!!), and hence the result doesn't depend on t. Therefore, again we compute the FT of a function of a single variable, which is τ.

We also have a formula for power when the process is cyclostationary:

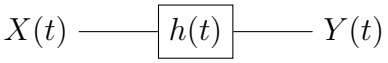
$$P_X = \tilde{R}_X(0).$$

Earlier we saw that it is not that easy to compute the PSD using the definition... However, if the process is stationary or cyclostationary, we can use these formulas connecting the PSD and the autocorrelation function[7]. Equations (1.1) and (1.3) are corollaries *of* the Wiener-Khinchin theorem (you will hear more about it in future courses).

? Quick quiz
What is the power of a WSS process with autocorrelation function $R_X(\tau) = 2\cos(2\tau)$ [8]

1.5 Response of LTI systems to a stochastic process

We want to know what happens when a stochastic process goes through a linear time-invariant (LTI) system:



If you don't know (or are uncertain about) what's going in, then you don't know what's going out (even though $h(t)$ is assumed to be known). In other words, if the input is a stochastic process, then so is the output.

Time signals that are produced by (or make up) the random process get modified and so do their statistical properties. In principle, we are interested in knowing what happens with

- the joint pdf of the random variables encompassed by the output process, $Y(t)$,

[7] If the process is not WSS or cyclostationary we have no easy way of computing the PSD.

[8] We just need to evaluate the autocorrelation function at 0: $P_X = 2\cos(2 \cdot 0) = 2$.

- properties such as stationarity, and
- the PSD.

Unless the input process is Gaussian, the expression for the joint pdf of variables from the output process is, in general, very complicated. However, we can derive things like the mean, the autocorrelation function, and the PSD of the output process given those of the input one.

We start with the definition of convolution that connects the input and output of an LTI system. The stochastic process at the output is given by

$$\begin{aligned} Y(t) = X(t) * h(t) &= \int_{-\infty}^{\infty} X(u)h(t-u)du \\ &= h(t) * X(t) = \int_{-\infty}^{\infty} h(u)X(t-u)du \end{aligned}$$

mean

$$\mu_Y(t) = \mathbb{E}\left[Y(t)\right] = \mathbb{E}\left[\int_{-\infty}^{\infty} X(u)h(t-u)du\right]$$

expectation and integral can be swapped; $h(t)$ is a deterministic signal, and hence can be pulled out of the expectation

$$\begin{aligned} &= \int_{-\infty}^{\infty} \mathbb{E}\left[X(u)\right] h(t-u)du = \int_{-\infty}^{\infty} \mu_X(u)h(t-u)du \\ &= \mu_X(t) * h(t) \overset{\text{WSS}}{=} \mu_X \int_{-\infty}^{\infty} h(u)du = \mu_X H(0), \end{aligned}$$

where $H(j\omega)$ is the frequency response associated with $h(t)$.

autocorrelation function

$$R_Y(t_1,t_2) = \mathbb{E}\left[Y(t_1)Y^*(t_2)\right] = \mathbb{E}\left[\int_{-\infty}^{\infty} X(t_1-l_1)h(l_1)dl_1 \int_{-\infty}^{\infty} X^*(t_2-l_2)h^*(l_2)dl_2\right]$$

we can collect the two integrals at the beginning, and reorder factors

$$= \mathbb{E}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} X(t_1-l_1)X^*(t_2-l_2)h(l_1)h^*(l_2)dl_1dl_2\right]$$

we swap integrals and expectation; $h(t)$ is a known deterministic signal

$$\begin{aligned} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underbrace{\mathbb{E}\left[X(t_1-l_1)X^*(t_2-l_2)\right]}_{R_X(t_1-l_1,t_2-l_2)} h(l_1)h^*(l_2)dl_1dl_2 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_X(t_1-l_1,t_2-l_2)h(l_1)h^*(l_2)dl_1dl_2 \overset{\text{WSS}}{=} \end{aligned}$$

so far true for any random process, we now assume WSS: the autocorrelation function only depends on the time difference

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_X(t_1 - l_1 - t_2 + l_2) h(l_1) h^*(l_2) dl_1 dl_2$$

we can see this expression as nested integrals, and $h^*(l_2)$, e.g., acts as a constant in one of them; we also reorder the argument of R_X

$$= \int_{-\infty}^{\infty} h^*(l_2) \left[\int_{-\infty}^{\infty} R_X(t_1 - t_2 + l_2 - l_1) h(l_1) dl_1 \right] dl_2$$

by giving $t_1 - t_2 + l_2$ a name, e.g., u, it is easy to see that what we have here is $\int_{-\infty}^{\infty} R_X(u - l_1) h(l_1) dl_1 = R_X(u) * h(u)$

$$= \int_{-\infty}^{\infty} h^*(l_2) \left[R_X(t_1 - t_2 + l_2) * h(t_1 - t_2 + l_2) \right] dl_2 \overset{f(t)=R_X(t)*h(t)}{=}$$

what is between brackets is actually a signal evaluated at $t_1 - t_2 + l_2$ (see the Convolution box below); if we give this signal a name, e.g., f, what we have is

$$= \int_{-\infty}^{\infty} h^*(l_2) f(\underbrace{t_1 - t_2}_{\tau} + l_2) dl_2 = \int_{-\infty}^{\infty} h^*(l_2) f(\tau + l_2) dl_2$$

we can time-reverse the signal inside the integral with respect to the integration variable (see the box below)

$$= \int_{-\infty}^{\infty} h^*(-l_2) f(\tau - l_2) dl_2$$

this is just another convolution: the integral (between $-\infty$ and ∞) of the product of two signals, one of them evaluated at the (dummy) integration variable, and the other at [the value of interest (here τ) minus the integration variable] (the *first* signal is here h time-reversed)

$$= h^*(-\tau) * f(\tau) = h^*(-\tau) * R_X(\tau) * h(\tau) = R_X(\tau) * h(\tau) * h^*(-\tau)$$

Notice that this result is only for WSS processes.

We have just derived the mean and autocorrelation function of the output process, and we have that **if the input process is WSS, so is the output process**.

Convolution

When we write, for instance,

$$R_x(t_1 - t_2 + l_2) * h(t_1 - t_2 + l_2)$$

or

$$R_x(5) * h(5)$$

we mean, respectively, $R_X(t) * h(t)\Big|_{t=t_1-t_2+l_2}$ and $R_X(t) * h(t)\Big|_{t=5}$. The convolution between two numbers, say $R_x(5)$ and $h(5)$, does not make any sense.

The value 5 in the expression $R_x(5) * h(5)$ is the time instant at which we evaluate the convolution (the "constant" inside the corresponding integral).

Time-reversing the function within an integral

Clearly,

$$\int_{-\infty}^{\infty} h(t)dt = \int_{-\infty}^{\infty} h(-t)dt$$

since the *area* under a signal doesn't change if the latter is time-reversed.

cross-correlation function

$$\begin{aligned} R_{YX}(t_1, t_2) &= \mathbb{E}\left[Y(t_1)X^*(t_2)\right] = \mathbb{E}\left[\int_{-\infty}^{\infty} X(t_1 - l_1)h(l_1)dl_1 X^*(t_2)\right] \\ &= \int_{-\infty}^{\infty} \mathbb{E}\left[X(t_1 - l_1)X^*(t_2)\right] h(l_1)dl_1 = \int_{-\infty}^{\infty} R_X(t_1 - l_1, t_2)h(l_1)dl_1 \\ &\overset{\text{WSS}}{=} \int_{-\infty}^{\infty} R_X(t_1 - l_1 - t_2)h(l_1)dl_1 = R_X(t_1 - t_2) * h(t_1 - t_2) = R_X(\tau) * h(\tau). \end{aligned}$$

As long as the input process is WSS, the cross-correlation between the input and output processes only depends on the time difference. Hence, by definition, they are **jointly stationary** (individually stationary, with a cross-correlation that only depends on the time difference).

Power spectral density For the case in which the **input process is WSS**, so is the output process, and we can obtain its PSD by computing the FT of the autocorrelation function,

$$\begin{aligned} S_Y(j\omega) &\overset{\text{WSS}}{=} \text{FT}\left[R_Y(\tau)\right] = \text{FT}\left[R_X(\tau) * h(\tau) * h^*(-\tau)\right] \\ &= S_X(j\omega)H(j\omega)H^*(j\omega) = S_X(j\omega)\left|H(j\omega)\right|^2 \end{aligned}$$

Quick quiz

Consider two zero-mean random processes $X(t)$ and $Y(t)$ with autocorrelation functions, $R_X(t_1, t_2) = e^{-|t_1 t_2|}$ and $R_Y(t_1, t_2) = e^{-|t_1 - t_2|}$, respectively. Are $X(t)$ and $Y(t)$ jointly stationary? [9]

In summary

$$X(t) \text{ WSS} \Rightarrow \begin{cases} Y(t) \text{ WSS (with known mean and autocorrelation function)} \\ X(t) \text{ and } Y(t) \text{ jointly stationary} \end{cases}$$

[9] $X(t)$ is not WSS, and hence it cannot be jointly stationary with *any* process.

1.6 Gaussian process

Recall that a Gaussian process is simply a random process in which the joint pdf of any set of random variables from the process (one per time instant) is Gaussian.

Definition 1.6.1: Gaussian process

$X(t)$ is a Gaussian process if, $\forall n, \forall (t_1, t_2, \cdots, t_n)$, random variables $\{X(t_i)\}_{i=1}^n$ have a joint Gaussian distribution.

This definition implies, among other things, that any individual random variable (associated with a certain time instant) is Gaussian (for $n = 1$).

They are an important kind of stochastic process in communications for two reasons:

thermal noise, which is present in any electronic device. It is the most relevant noise in any communications system, and is **Gaussian**. There are two questions here:

- why is thermal noise present in any electronic device? Because it is due to the random movement of electrons caused by the temperature. You need a temperature of 0 degrees Kelvin (that is, -273 degrees Celsius!!) for the electrons to stop moving. Hence, in practice you can never get rid of thermal noise.
- why is thermal noise Gaussian? We said it is due to the random movement of electrons. An electric current is just a huge number of electrons put together. If we assume the behavior of each electron is independent then we have a sum of a huge number of independent random variables. According to the **Central Limit Theorem** that approaches a Gaussian distribution.

sources of information Gaussian processes serve as a model for some sources of information.

Hence, Gaussian processes are useful for the analysis of thermal noise, on one hand, and for characterizing some sources of information on the other hand. Next we enumerate some interesting properties they exhibit.

1.6.1 Properties

- The mean and autocorrelation function amount to a *complete statistical description* of the process which, in turn, entails knowing the joint probability density function of any set of variables $X(t_1), X(t_n), \cdots, X(t_n)$. For jointly Gaussian random variables, the latter is given by

$$f_{X(t_1),X(t_2),\cdots,X(t_n)}(x_1, x_2, \cdots, x_n) = \frac{1}{\sqrt{(2\pi)^n \det(\mathbf{\Sigma_X})}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_\mathbf{X})^T \mathbf{\Sigma}_\mathbf{X}^{-1}(\mathbf{x}-\boldsymbol{\mu}_\mathbf{X})}$$

where "det" stands for determinant (of the argument),

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \equiv \text{vector comprising the values of the } n \text{ random variables,}$$

$$\boldsymbol{\mu_X} = \begin{bmatrix} \mu_{X(t_1)} \\ \mu_{X(t_2)} \\ \vdots \\ \mu_{X(t_n)} \end{bmatrix} \equiv \text{means vector, and}$$

$$\boldsymbol{\Sigma_X} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{bmatrix} \equiv \text{covariance matrix}$$

with[10]

$$C_{ij} = \text{Cov}\left(X(t_i), X(t_j)\right) = R_X(t_i, t_j) - \mu_X(t_i)\mu_X(t_j).$$

Therefore, the mean and autocorrelation functions allow to compute both the means vector and the covariance matrix, which completely determine the joint pdf.

- strict-sense stationarity $\iff$ wide-sense stationarity (they are equivalent)
- $X(t)$ Gaussian —— LTI —— $Y(t)$ Gausian

  If the input process to an LTI system is Gaussian, then so is the output process.
- $\text{Cov}\left(X(t_i), X(t_j)\right) = 0 \Leftrightarrow X(t_i)$ and $X(t_j)$ are independent

? Quick quiz

If a Gaussian process has zero mean (meaning, every individual random variable has zero mean), what is the connection between the autocorrelation and the covariance? [11]

1.7 White process

It is a particular kind of **wide-sense stationary** process.

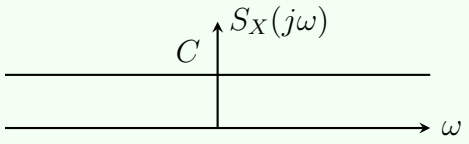
[10] Notice that the autocorrelation function is a function of two variables...a Gaussian process need not be stationary!!

[11] They are the same. More specifically, according to the above definition, the covariance between time instants t_i, and t_j is given by the autocorrelation evaluated at those particular time instants.

Definition 1.7.1: White process

A WSS process, $X(t)$, is white if its PSD is constant, i.e.,

$$S_X(j\omega) = C$$



It has the same power at every frequency. The name stems from physics: white light encompasses all the colors, and every color is associated with a different frequency.

1.7.1 Properties

Some consequences of the definition are:

- $R_X(\tau) = \mathrm{FT}^{-1}[C] = C\delta(\tau)$ (notice that we are using the fact that the process is WSS)

  We have seen before a WSS process whose autocorrelation function is a delta: it was the particular case (exception) in which a process was not finite-energy nor finite-power.

- The power of a white process is given by

$$P_X = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} C d\omega = \infty$$

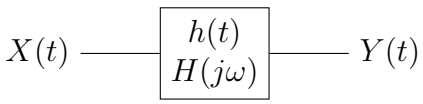
  and we saw earlier that WSS processes have infinite energy $E_X = \int_{-\infty}^{\infty} R_X(0)dt = \infty$.

1.7.2 Filtering of a white process

Let us consider a white process, $X(t)$, with PSD

$$S_X(j\omega) = C \Rightarrow R_X(\tau) = C\delta(\tau).$$

We want to know what happens to it after going through an LTI system,



Since $X(t)$ is white, it is also WSS, and therefore so is the output process $Y(t)$. Then, **we can apply all the properties we derived earlier**.

PSD at the output

$$S_Y(j\omega) \overset{\text{WSS}}{=} S_X(j\omega)\,|H(j\omega)|^2 = C\,|H(j\omega)|^2$$

Autocorrelation function

$$R_Y(\tau) \overset{\text{WSS}}{=} R_X(\tau) * h(\tau) * h^*(-\tau) = R_X(\tau) * \underbrace{r_h(\tau)}_{\substack{\text{self-similarity} \\ \text{for } h(t)}} = C r_h(\tau)$$

with

$$r_h(\tau) = h(\tau) * h^*(-\tau) = \int_{-\infty}^{\infty} h(t) h^*(t-\tau) dt \equiv \text{ self-similarity function.}$$

We already knew the process was going to be autocorrelation-stationary: if the input process is WSS, so is the output process.

Power

$$P_Y \overset{\text{Y(t) is WSS}}{=} R_Y(0) = C r_h(0)$$

Evaluating the self-similarity function at 0 we have

$$r_h(0) = \int_{-\infty}^{\infty} |h(t)|^2 \, dt = \varepsilon \left\{ h(t) \right\}.$$

Thus,

$$P_Y = C \varepsilon \left\{ h(t) \right\}.$$

1.8 Thermal noise

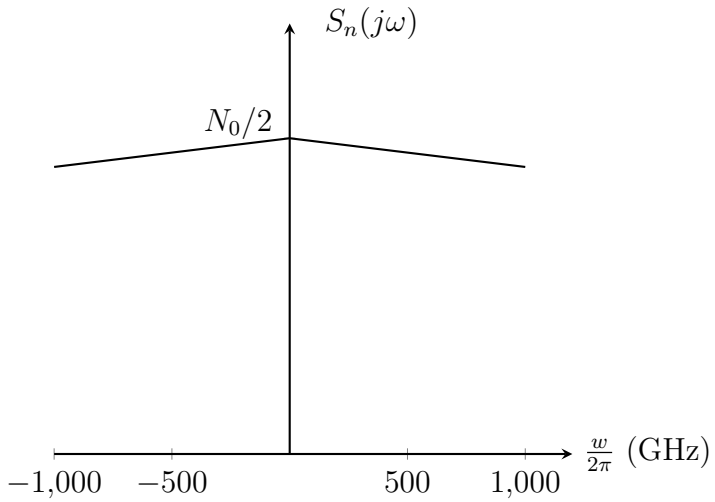
Earlier, we came to the conclusion that, according to the Central Limit Theorem, thermal noise must be Gaussian. Besides, quantum mechanics yields an analytical expression for its PSD:

$$S_n(j\omega) = \frac{h\omega}{4\pi \left(e^{\frac{h\omega}{2\pi kT}} - 1 \right)}$$

where

- h is Planck's constant $= 6.6 \cdot 10^{-34}$ joules/second
- k is Boltzmann's constant $= 1.38 \cdot 10^{-23}$ joules/degree Kelvin
- T is the temperature in degrees Kelvin, and
- ω is the frequency in radians per second.

If we plot the above formula we get

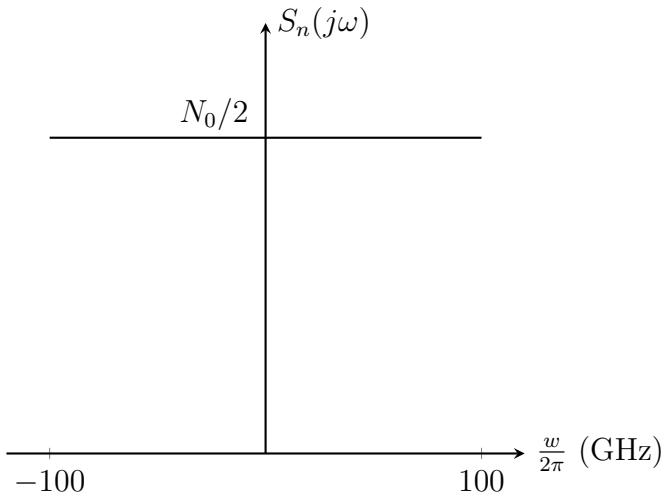


with

$$N_0 \triangleq kT$$

(we are giving kT a name[12]).

However, if we zoom in, between −100 and 100 GHz the PSD is approximately flat,



Hence, for all practical purposes[13] the noise process is **white** (which also entails it is **WSS**)!!

Putting together everything we have seen about thermal noise we arrive at the following...

1.8.1 Thermal noise model

For us, thermal noise is a random process, denoted by $n(t)$, that is

- Gaussian (from Central Limit Theorem)

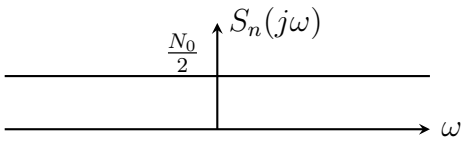
[12]This is something to keep in mind because N_0 will come up all the time: N_0 is just a constant that depends on the temperature.

[13]Electronic devices operate in a range of frequencies in which the PSD of thermal noise is fairly constant, and those are precisely the frequencies we care about.

- a white process (from quantum mechanics)
- WSS (since the process is white)

Additionally, we know that $n(t)$ is

- zero mean, $\mu_n(t) = 0$,
- with PSD $S_n(j\omega) = \frac{N_0}{2}$,



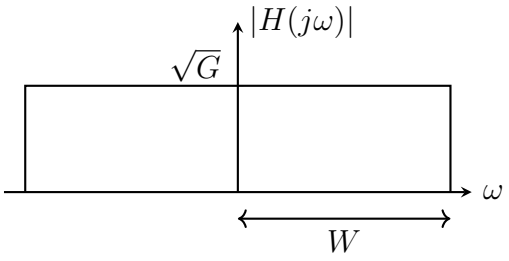
- and autocorrelation function $R_n(\tau) = \frac{N_0}{2}\delta(\tau)$ (since it's WSS).

Very often we will talk about **A**dditive **W**hite **G**aussian **N**oise or **AWGN**[14]. This is our *model*. Reality is what we saw earlier. They don't exactly match but it is still fine.

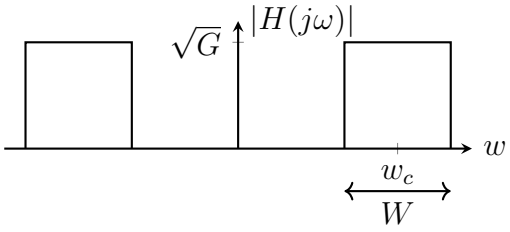
1.8.2 Power of thermal noise at the output of *ideal* filters with gain

We now consider an **ideal** filter (either baseband or passband) with bandwidth B Hz (or $W = 2\pi B$ rad/second) and gain $\sqrt{G}$.

Bandwidth of a signal/channel
The bandwidth is the width of the positive non-null (non-zero) frequencies.



...if the channel is baseband



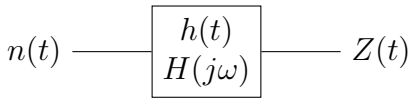
...if the channel is passband
(w_c is the center frequency)

We say that the filter $h(t)$ has

- *voltage gain* $\sqrt{G}$,
- and *power gain* G.

If the noise, $n(t)$, goes through a filter with impulse response $h(t)$ and frequency response $H(j\omega)$,

[14] About the *additive* part, remember that, in our model for a communications channel, noise is something that gets *added* up at the end, after any distortion that might be introduced by an LTI system representing the channel.



at the output we get

$$S_Z(j\omega) = \frac{N_0}{2} |H(j\omega)|^2 ,$$

where we have used that thermal noise is a WSS process with PSD $N_0/2$.

Then, the power of thermal noise at the output of the filter is[15]

$$\begin{aligned} P_Z &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S_Z(j\omega) d\omega = \frac{1}{2\pi} \frac{N_0}{2} \int_{-\infty}^{\infty} |H(j\omega)|^2 \, d\omega = \frac{1}{2\pi} \frac{N_0}{2} \int_{-W}^{W} G d\omega \\ &= \frac{1}{2\pi} \frac{N_0}{\cancel{2}} G \cancel{2} W = \frac{1}{2\pi} N_0 G W = \frac{1}{\cancel{2\pi}} N_0 G \cancel{2\pi} B = N_0 B G. \end{aligned}$$

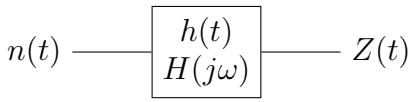
What happens if the filter has infinite bandwidth, i.e., if its impulse response is $h(t) = k\delta(t)$, with k being some constant? In such case, the power is infinity, and that's given: you are simply amplifying (multiplying) by k a white process, and we know that any white process has infinite power (see the properties of a white process). This becomes a problem when you have a signal tainted by thermal noise, as we will see later on when talking about signal-to-noise ratio

Power of thermal noise at the output of *ideal* filters with no gain

Sometimes people talk about *ideal filters* (just like that, without mentioning the gain) or *ideal filters* with **no** gain. That ultimately means the latter is $G = 1$, and we can still apply the previous result.

1.8.3 Equivalent noise bandwidth

We keep studying what happens to thermal noise when it goes through an LTI system,



but now it can by *any* system.

Why is this so important? Again, because thermal noise is everywhere and we care how it is transformed when going through an LTI system. For instance, imagine we have an audio signal, maybe assembled in a computer. For sure it is to some extent tainted by thermal noise (you cannot avoid that when capturing/storing the signal). If we are going to feed this signal into some device (e.g., speakers) we are concerned with how bad the noise is at the output of the latter.

In general, the PSD of thermal noise at the output of an LTI system with impulse response $h(t)$ and frequency response $H(j\omega)$ is given by,

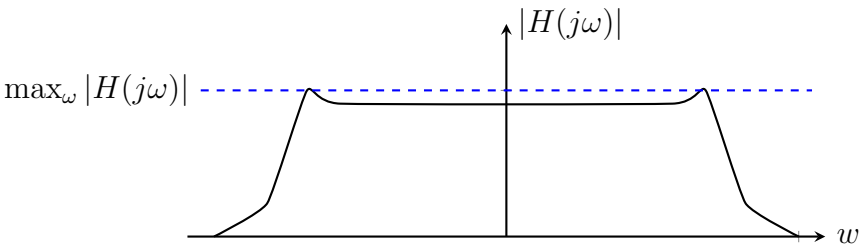
$$S_Z(j\omega) = \frac{N_0}{2} |H(j\omega)|^2 ,$$

[15]For the sake of simplicity, we are focusing on the baseband channel, but the same result is readily obtained for the passband case.

where we have used that thermal noise is WSS (according to our model). The power of the output process is then

$$P_Z = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_Z(j\omega)d\omega = \frac{N_0}{2}\underbrace{\frac{1}{2\pi}\int_{-\infty}^{\infty}|H(j\omega)|^2\,d\omega}_{\substack{\mathcal{E}\{h(t)\}\\ \text{(Parseval)}}}$$

Often it is hard to compute this integral (or, equivalently, the energy of $h(t)$). So, if you buy a device and the manufacturer tells you "this is the frequency response"



that is not so useful. That's why we define the...

Definition 1.8.1: Equivalent noise bandwidth, B_{eq}, of a filter $h(t)$

The bandwidth of an ideal (*brickwall*) filter whose gain matches the *maximum* gain of the given filter, and which yields the same thermal noise power at the output. Mathematically, if B_{eq} is the equivalent noise bandwidth of $h(t)$, then

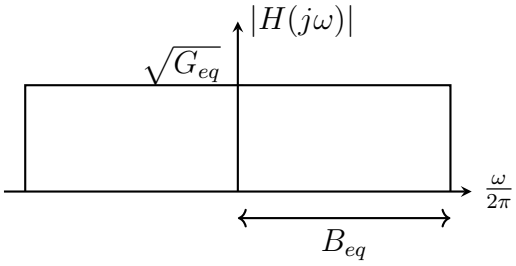
$$P_Z = N_0 B_{eq} G_{eq}$$

with

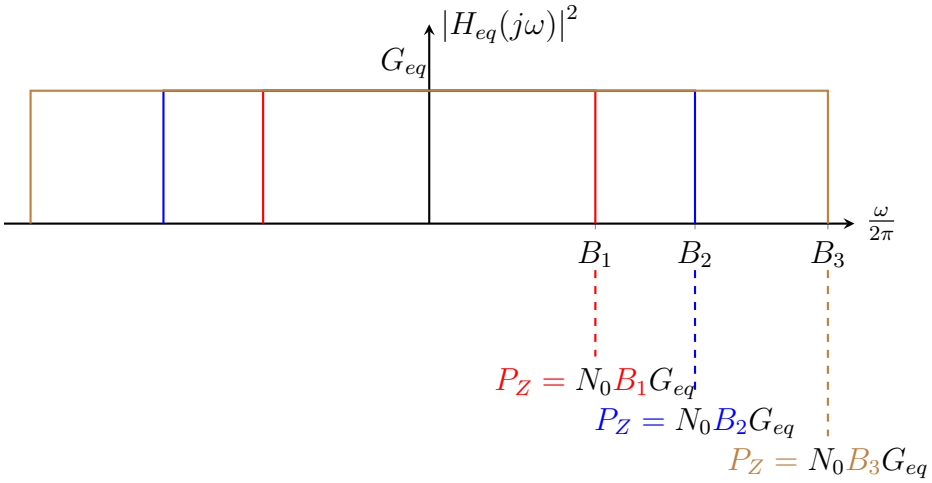
$$\begin{aligned} P_Z &\equiv \text{power of thermal noise at the output of the filter} \\ B_{eq} &\equiv \text{equivalent noise bandwidth} \\ G_{eq} &= \max_\omega |H(j\omega)|^2 \equiv \text{equivalent power gain} \end{aligned}$$

Notice the above formula is simply the power of thermal noise at the output of an ideal filter with bandwidth B_{eq} and gain G_{eq}.
The equivalent noise bandwidth of a device is usually provided by the manufacturer.

What is the meaning of this? The power of thermal noise at the output of the original filter (with impulse response $h(t)$) is the same as that at the output of an ideal filter with (power) gain G_{eq} and bandwidth B_{eq},



This ideal filter with power gain G_{eq} and bandwidth B_{eq} could be used as a *surrogate* for the filter of interest (at least, regarding thermal noise[16]). How do we get the surrogate filter to give the same power as the original filter? We are fixing its shape (we are saying "it's an ideal filter") and height ("it must be as high as the original filter"), and hence we can only play with its bandwidth



We must choose the value for the bandwidth, B_{eq}, such that we get the same power as before.

The point here is that, if we know the equivalent noise bandwidth of a generic (non-ideal!!) filter, we can easily compute the power thermal noise at the output using the formula for an ideal filter with gain.

? Quick quiz

What is the equivalent noise bandwidth of an ideal filter with bandwidth 10 Hz? [17]

[16] Right now, we only care about thermal noise!!

[17] It's $B_{eq} = 10$ Hz (the filter is already ideal).

Computation

We must choose B_{eq} so that this equation is satisfied

$$\begin{aligned} P_Z &= N_0 B_{eq} G_{eq} \\ &\Updownarrow \\ \frac{\cancel{N_0}}{2} \underbrace{\frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2 \, d\omega}_{\varepsilon\{h(t)\}} &= \cancel{N_0} B_{eq} G_{eq} \\ &\Updownarrow \\ B_{eq} &= \frac{\varepsilon\{h(t)\}}{2G_{eq}} \end{aligned}$$

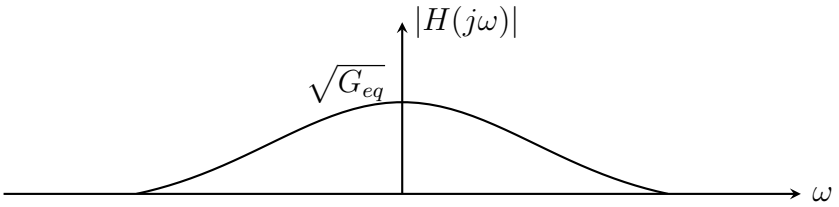
Notice that, according to Parseval's theorem, we can compute the energy of a signal either in the time-domain or the frequency-domain,

$$\varepsilon\{h(t)\} = \int_{-\infty}^{\infty} |h(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2 \, d\omega.$$

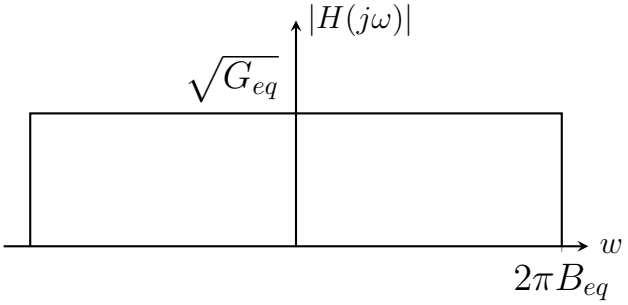
In general, it is not so easy to compute the energy of any arbitrary $h(t)$, but the manufacturer carries out this integral for you. That's the whole point.

Interpretation

If we are given a filter whose frequency response is



and whose equivalent noise bandwidth is B_{eq}, **for the purpose of thermal noise**, and only for that, we can think of the filter as something like this



and the latter allows to easily compute the output power.

Example: compute the equivalent noise bandwidth of a filter with a given frequency response

Compute the equivalent noise bandwidth of a filter whose frequency response is given by

$$|H(j\omega)| = \begin{cases} \sqrt{1+\frac{\omega}{W}}, & -W \leq \omega < 0 \\ \sqrt{1-\frac{\omega}{W}}, & 0 \leq \omega \leq W \\ 0, & \text{otherwise,} \end{cases}$$

where W is the bandwidth in radians per second.

We have a formula for the equivalent noise bandwidth, which is

$$B_{eq} = \frac{\varepsilon\{h(t)\}}{2G_{eq}} = \frac{\frac{1}{2\pi}\int_{-\infty}^{\infty}|H(j\omega)|^2\,d\omega}{2G_{eq}}.$$

Notice that we can compute the energy of $h(t)$ in the time-domain or in the frequency-domain, the latter being apparently easier here.

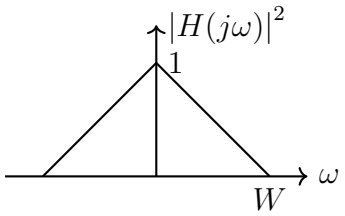
The squared modulus of $H(j\omega)$ is

$$|H(j\omega)|^2 = \begin{cases} 1+\frac{\omega}{W}, & -W \leq \omega < 0 \\ 1-\frac{\omega}{W}, & 0 \leq \omega \leq W \\ 0, & \text{otherwise,} \end{cases}$$

We can analyze this function piece-wise:

- on the negative side, $1+\frac{\omega}{W}$ is a straight line with
  - slope $\frac{1}{W}$,
  - intercept (the point at which it crosses the vertical axis) 1, and
  - evaluating the equation at both W and $-W$ we get 0
- on the positive side, we have another straight line with opposite slope, and crossing the y-axis at the same point.

Hence, the squared modulus looks like this



and we have

$$G_{eq} = \max_{\omega}|H(j\omega)|^2 = 1.$$

We still need to compute the energy of $h(t)$

$$\varepsilon\{h(t)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty}|H(j\omega)|^2\,d\omega \overset{\text{symmetry}}{=} 2\frac{1}{2\pi}\int_0^W\left(1-\frac{\omega}{W}\right)d\omega = \frac{1}{\pi}\int_0^W 1d\omega - \frac{1}{\pi}\int_0^W\frac{\omega}{W}d\omega$$
$$= \frac{\omega}{\pi}\Big|_0^W - \frac{1}{\pi W}\frac{\omega^2}{2}\Big|_0^W = \frac{W}{\pi} - \frac{1}{\pi\cancel{W}}\frac{\cancel{W}^2}{2} = \frac{W}{\pi} - \frac{W}{2\pi} = \frac{W}{\pi}\left(1-\frac{1}{2}\right) = \frac{W}{2\pi} = B,$$

which is the bandwidth in Hertz. That was the hard way...the easy way is: the integral we have just computed is the area of this triangle whose base is $2W$ and height is 1,

$$A = \frac{2W\cdot 1}{2} = W,$$

and we would still need to divide by 2π,

$$\varepsilon\{h(t)\} = \frac{A}{2\pi} = \frac{W}{2\pi} = B.$$

Integral of a symmetric function

If a function is symmetric, the area to the left of 0 is equal to the area to the right of 0, and hence the total area is twice the area on one side.

Putting it all together, we have

$$B_{eq} = \frac{\varepsilon\{h(t)\}}{2G_{eq}} = \frac{B}{2\times 1} = \frac{B}{2}.$$

Example: RC filter

In previous courses you have seen RC circuits: it is a circuit with a resistor and a capacitor. We are going to compute the equivalent noise bandwidth of an RC circuit acting as low-pass filter[18], whose frequency response is

$$H(j\omega) = \frac{1}{1+j\omega\tau}$$

with

$$\tau = RC \equiv \text{time constant}.$$

Again, we start with the formula for B_{eq},

$$B_{eq} = \frac{\varepsilon\{h(t)\}}{2G_{eq}} = \frac{\frac{1}{2\pi}\int_{-\infty}^{\infty}|H(j\omega)|^2\,d\omega}{2G_{eq}},$$

and we compute the modulus of $H(j\omega)$, which is

$$|H(j\omega)| = \frac{1}{\sqrt{1^2+\omega^2\tau^2}},$$

[18]This implies measuring the voltage across the capacitor.

where we have used that the modulus of the quotient of two complex numbers is quotient of the individual modulus.

The square of the modulus is

$$|H(j\omega)|^2 = \frac{1}{1^2 + \omega^2\tau^2},$$

from which we infer[19]

$$G_{eq} = \max_{\omega} |H(j\omega)|^2 = 1 \text{ (for } \omega = 0).$$

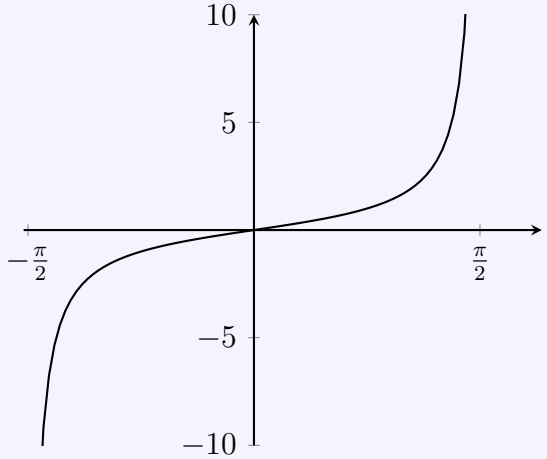
Next, we compute the energy of $h(t)$ as

$$\varepsilon\{h(t)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} |H(j\omega)|^2\, d\omega \overset{\text{symmetry}}{=} \cancel{2}\frac{1}{\cancel{2}\pi}\int_0^{\infty} |H(j\omega)|^2\, d\omega = \frac{1}{\pi}\int_0^{\infty} \frac{1}{1^2 + \omega^2\tau^2} d\omega$$

change of variable: $u = \omega\tau \Rightarrow \frac{du}{dw} = \tau \Rightarrow d\omega = \frac{du}{\tau}$, $\omega = 0 \to u = 0$, $\omega = \infty \to u = \infty$

$$= \frac{1}{\pi}\int_0^{\infty} \frac{1}{1+u^2}\frac{du}{\tau} = \frac{1}{\pi\tau}\int_0^{\infty} \frac{1}{1+u^2} du = \frac{1}{\pi\tau}\arctan(u)\Big|_0^{\infty} = \frac{1}{\cancel{\pi}\tau}\left(\frac{\cancel{\pi}}{2} - 0\right) = \frac{1}{2\tau}$$

Tangent function



It is 0 at 0 (hence, $\arctan(0) = 0$), and blows up at $\frac{\pi}{2}$ (hence, $\arctan(\infty)$ is $\frac{\pi}{2}$).

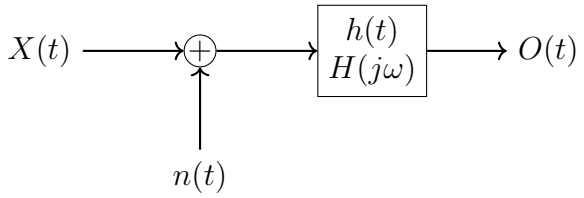
Putting it all together,

$$B_{eq} = \frac{\varepsilon\{h(t)\}}{2G_{eq}} = \frac{1/2\tau}{2 \times 1} = \frac{1}{4\tau} = \frac{1}{4RC}.$$

[19] As w gets away from zero, the denominator (always positive) increases. Hence, the maximum is attained at 0.

1.9 Signal-to-noise ratio (SNR) at the output of a filter

We have a stochastic process that encompasses a signal of interest plus some *thermal* noise (both the signal of interest and the noise are modeled as WSS stochastic processes), and we make it go through an LTI system



We want to know the signal-to-noise ratio (SNR) at the output, that is, we want to compare the power of the signal of interest and that of the noise at the output of the filter.

We start by computing the signal at the output of the filter by using the properties of the convolution,

$$O(t) = [X(t) + n(t)] * h(t) = \underbrace{X(t) * h(t)}_{Y(t)} + \underbrace{n(t) * h(t)}_{Z(t)} = Y(t) + Z(t).$$

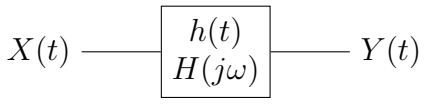
- $Y(t)$ is the process of interest after filtering
- $Z(t)$ is filtered noise

The SNR at the output is given by

$$\frac{S}{N} = \frac{P_Y}{P_Z} \quad \text{or, in dBs,} \quad \frac{S}{N}(\text{dBs}) = 10 \log_{10} \frac{P_Y}{P_Z}.$$

We compute P_Y and P_Z:

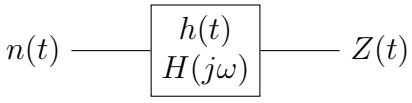
***Filtered* signal power** is the power at the output of this LTI system



and hence we have

$$P_Y = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_Y(j\omega) d\omega \overset{\text{WSS}}{=} \frac{1}{2\pi} \int_{-\infty}^{\infty} S_X(j\omega) \left|H(j\omega)\right|^2 d\omega$$

***Filtered* noise power** is the power at the output of this LTI system



and hence we have

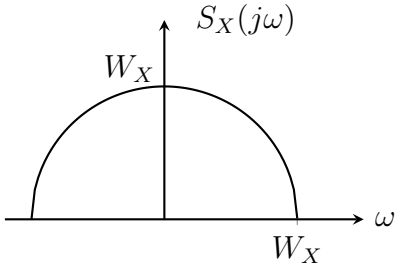
$$P_Z = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_Z(j\omega)d\omega = \frac{1}{2\pi}\frac{N_0}{2}\int_{-\infty}^{\infty} |H(j\omega)|^2\, d\omega$$

We know the result of this for a couple of particular cases

- ideal filters with gain
$$P_Z = N_0 B G$$
- filters with equivalent noise bandwidth B_{eq} and gain G_{eq}
$$P_Z = N_0 B_{eq} G_{eq}$$

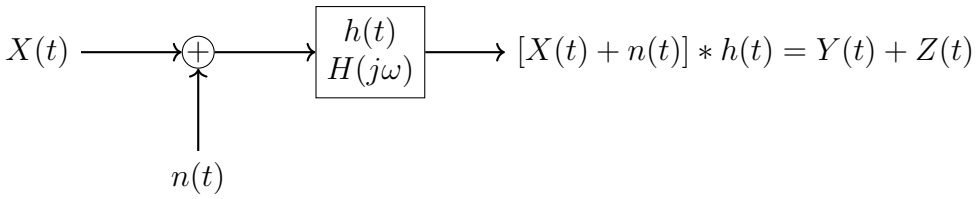
1.9.1 Example

Let us consider a WSS stochastic process, $X(t)$, with PSD

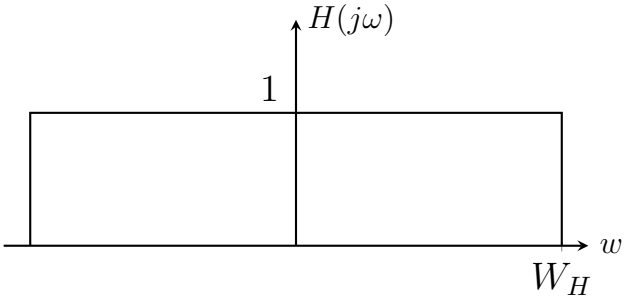


and thermal noise, $n(t)$ with $S_n(j\omega) = \frac{N_0}{2}$. The sum of processes $X(t)$ and $n(t)$ goes through an ideal filter with bandwidth $W_H > W_X$ (hence, the filter lets the signal go through). What is the S/N *at the output*?

We have this system



and the frequency response of the (ideal) filter is



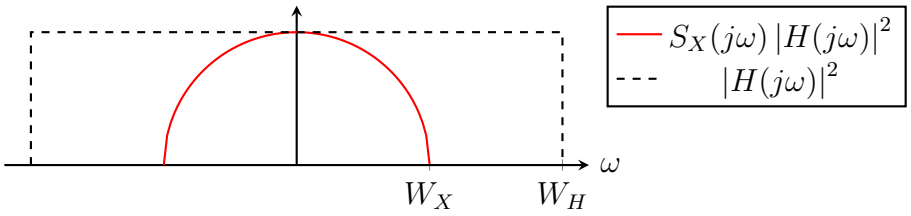
By definition,

$$\frac{S}{N}(\text{dBs}) = 10\log_{10}\frac{P_Y}{P_Z}.$$

We compute the power of the signal (at the output):

$$P_Y = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_Y(j\omega)d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)\left|H(j\omega)\right|^2 d\omega$$

How does it look the signal within the integral?



So, $S_X(j\omega)\left|H(j\omega)\right|^2$ is exactly $S_X(j\omega)$. Then,

$$P_Y = \frac{1}{2\pi}\int_{-W_X}^{W_X} S_X(j\omega)d\omega \overset{\text{area semicircle}}{=} \frac{1}{2\cancel{\pi}}\frac{\cancel{\pi}W_X^2}{2} = \frac{W_X^2}{4}.$$

Next, we compute the power of the noise at the output,

$$P_Z = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_Z(j\omega)d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{N_0}{2}\left|H(j\omega)\right|^2 d\omega = \frac{1}{2\pi}\frac{N_0}{2}\int_{-W_H}^{W_H} 1d\omega = \frac{1}{2\pi}\frac{N_0}{\cancel{2}}\cancel{2}W_H$$
$$= \frac{N_0}{2\pi}W_H,$$

i.e., the bandwidth of the filter in Hertz times N_0 (we already knew this).

Putting it all together,

$$\frac{S}{N}(\text{dBs}) = 10\log_{10}\frac{P_Y}{P_Z} = 10\log_{10}\frac{\frac{W_X^2}{4}}{\frac{N_0}{2\pi}W_H} = 10\log_{10}\frac{W_X^2\pi}{2N_0W_H}$$

What happens if the ideal filter has an infinite bandwidth (we are in this scenario if we don't use any filter at all)? We have

$$\frac{S}{N}(\text{dBs}) = 10\log_{10}\frac{W_X^2\pi}{2N_0W_H} = 10\log_{10}\frac{W_X^2\pi}{\infty} = -\infty,$$

and that's why in any communications receiver we must use bandlimited filter (to put a bound on the noise).

Chapter 2

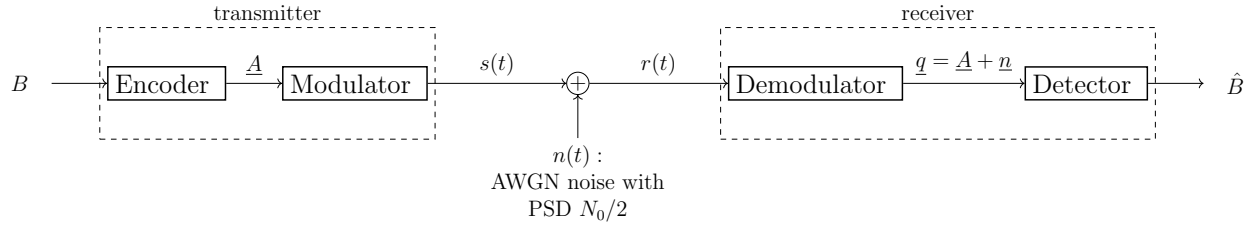
Modulation and detection

In the introduction of the course we talked about the difference between analog and digital communications systems: in an analog communications system any transmitted signal is possible, whereas in a digital one the signal to be transmitted (*symbol*) must be selected from a finite set of candidates (the *alphabet*). In the latter case we are bound to notice when bad things happen during transmission. This is the main advantage, but there are some extra ones such as versatility, easy (channel) coding or encryption.

In this module we focus on modulation and detection in **digital** communications systems. At the end of the course we will talk briefly about analog communications system, but the gist of the course is about digital ones.

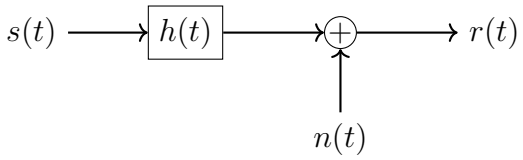
2.1 Model of a digital communications system

We'll be working with this general scheme for a digital communications system.



Notice that, for the time being, we are dismissing source coding, encryption and channel coding[1]. We are focusing on the basics. What we are omitting can be studied separately. Another important thing here is that the channel is simply adding noise to the signal transmitted (we don't have any distortion). This model of the channel is known as **Gaussian channel**. If we had a *real* channel, then it would go between the transmitter and the receiver, right after the modulator:

[1]The *encoder* in the diagram has nothing to do with source or channel coding.



For the sake of simplicity, we assume that we transmit a single information symbol, B (in practice, we will be transmitting a sequence), which is the input to this model. B can take one of M possible values, $b_1, b_2, \cdots, b_M$, thar are collected in a set called the **alphabet**. Hence, we have that is

$$B \in \underbrace{\{b_1, b_2, \cdots, b_M\}}_{\text{alphabet}}.$$

In general (though not always) we have that all the symbols are equally likely, i.e.,

$$P(b_i) = \frac{1}{M}, i = 1, \cdots, M.$$

Notice that (capital) B is a random variable whereas $b_1, b_2, \cdots b_M$ are the possible values it can take. For example, B might be the answer to a question and we could have $b_1 =$ "yes" and $b_2 =$ "no".

At the receiver the goal is to recover the symbol B, but it might be the case that we get a different one (still belonging to the alphabet). That's why we write $\hat{B}$ instead of B (the hat in $\hat{B}$ indicates the latter is an estimate of the *actual* B, which might be different).

2.1.1 Transmitter

The symbol to be transmitted is fed into the transmitter, which is in charge of mapping this symbol into an analog signal $s(t)$ that is fit to travel through the channel[2]. This transformation is carried out in two separate steps: encoding and modulation.

Encoder

It maps a symbol B into another symbol $\underline{A}$ using the correspondence

$$b_i \longrightarrow \underline{a}_i$$

Symbol $\underline{A}$ may be a real number, a complex number (ultimately, a 2D vector) or, more generally, an N-dimensional vector,

$$\underline{a}_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{iN} \end{bmatrix}.$$

[2]We cannot put a number into a transmission medium (e.g., the air)!! The system is only digital in the sense that we must pick the signal to be transmitted from a finite set, which is the alphabet. What we ultimately put in the channel is a continuous-time signal, a time-varying electromagnetic field.

The first subindex in a_{ij} indicates which symbol we are referring to (the i-th symbol), whereas the second one indicates the component, j, within the corresponding vector. The line under $\underline{A}$ is intended to stress the fact that $\underline{A}$ is a vector.

The alphabet of $\underline{A}$, that is, the set of possible values, is known as the **constellation**,

$$\underline{A} \in \underbrace{\{\underline{a}_1, \underline{a}_2, \cdots, \underline{a}_M\}}_{\text{constellation}}.$$

Notice that just like (capital) B is a random variable and b_i a realization thereof, $\underline{A}$ is a random *vector* and $\underline{a}_i$ a particular realization.

? Quick quiz

The constellation of a digital communications system is

$$\underline{a}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \underline{a}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

What are the values of M and N? [3]

Modulator

On the other hand (notice we are still on the transmitter side), the modulator performs a mapping from symbol $\underline{A}$ into an **analog** continuous-time signal, $s(t)$. If the element from the constellation transmitted was $\underline{a}_i$, then we have

$$\underline{a}_i \longrightarrow s_i(t).$$

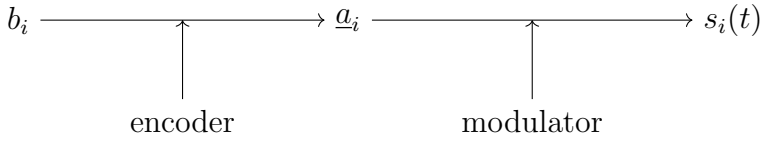
Hence, every element from the constellation has its own associated signal, and it holds that

$$s(t) \in \{s_1(t), s_2(t), \cdots, s_M(t)\}.$$

In general, $s_i(t)$ is

- a finite-energy signal,
- null outside the interval $0 \leq t \leq T$ (with T being the period).

In summary, the operation of the transmitter is



[3] We have $M = 2$ elements in the alphabet/constellation, each one of dimension $N = 3$.

This means that by knowing one of elements in the diagram above you know all of them. Indeed, there is a one to one mapping between any two of them.

Now the question that comes up is: why don't we go directly from b_i to $s_i(t)$? In other words, why don't we do $b_i \longrightarrow s_i(t)$? We don't because the distinction between encoder and modulator is useful. The reason is that the performance of the system (i.e, the probability of error) *only* depends on the constellation, which is given by the encoder, whereas the purpose of the modulator is simply to choose the signals $s_1(t), s_2(t), \cdots, s_M(t)$ that are better fitted to the nature of the channel (baseband/passband). Hence, we could use a different modulator (if the channel changes) with the same encoder, or we could plug a different encoder (to improve the performance) while keeping the same modulator.

2.1.2 Receiver

The receiver must *estimate* symbol B from the received signal, $r(t)$. Again, notice that the output of the receiver is an estimate, $\hat{B}$, of B (maybe they are equal, maybe not). Ideally $\hat{B} = B$, but it might well be that they are different due to disturbances and/or distortions during transmission. If $\hat{B}$ and B are different we say that an error occurred. Figure 2.1 illustrates what happens in your standard digital communications system. The

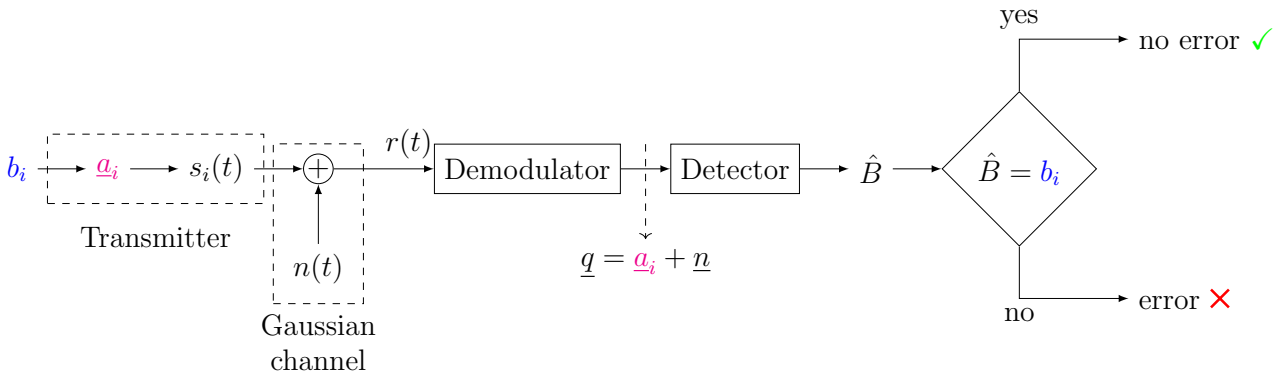


Figure 2.1: Making decisions at the receiver.

output of the demodulator, $\underline{q}$, is simply a *noisy* estimate of $\underline{a}_i$ ($\underline{n}$ is a discrete noise term stemming from continuous-time thermal noise $n(t)$).

We rely on this to measure the quality of a system. In particular (we have already talked about this) our metric is going to be the probability of error, which is defined as

$$P_e = P(\hat{B} \neq B),$$

i.e., the probability of $\hat{B}$ being different from B. The receiver is designed to minimize this probability of error, and it also operates in two steps.

Demodulator

The demodulator produces the *vector* $\underline{q}$ from the received signal, $r(t)$,

$$r(t) \rightarrow \underline{q} = \underline{A} + \underline{n},$$

and, ideally, $\underline{q}$ should resemble $\underline{A}$ (the element from the constellation that was transmitted) as much as possible[4]. Notice that $\underline{q}$ is actually the transmitted element $\underline{A}$ plus some noise vector $\underline{n}$ (this is shown in the diagram at the beginning of this section).

Detector

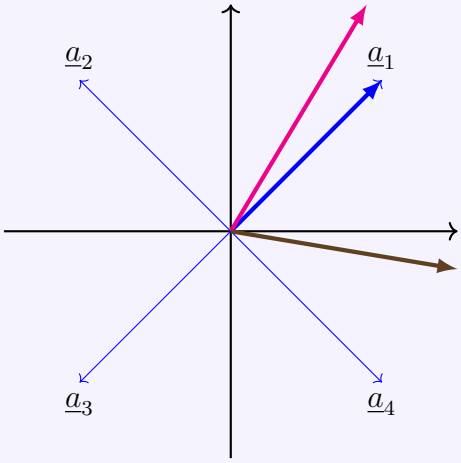
The last block in the diagram is the detector. It assigns a symbol $\hat{B}$ to vector $\underline{q}$ (recall that $\underline{q}$ need not be an element from the constellation). We should carry out this so that the error probability is minimized. Later on, we will see this *usually* entails deciding the symbol associated with the element in the constellation whose Euclidean distance to $\underline{q}$ is smaller.

Detection

- $N = 2 \Rightarrow$ the elements in the constellation, $\underline{a}_i$, are 2D-vectors
- $M = 4 \Rightarrow$ there are four possible symbols that can be transmitted (each one being associated with a 2D-vector, $b_i \rightarrow \underline{a}_i$).

Hence we have a constellation with four elements, $\underline{a}_1, \underline{a}_2, \underline{a}_3, \underline{a}_4$. Imagine we transmit b_1, which is associated with element $\underline{a}_1$

$$\underline{q} = \underline{a}_1 + \underline{n}$$
$$d(\underline{q}, \underline{a}_1) < d(\underline{q}, \underline{a}_i), i \neq 1 \Rightarrow \hat{B} = b_1 \rightarrow \textbf{good}$$



$$\underline{q}' = \underline{a}_1 + \underline{n}'$$
$$d(\underline{q}', \underline{a}_4) < d(\underline{q}', \underline{a}_i), i \neq 4 \Rightarrow \hat{B} = b_4 \rightarrow \textbf{error!!}$$

When we receive $\underline{q}$, we find that the distance from $\underline{q}$ to $\underline{a}_1$ is smaller than the distance from $\underline{q}$ to any other element in the constellation, $\underline{a}_i$. Hence, we decide $\underline{a}_1$...and that's the right decision. On the other hand, if we receive $\underline{q}'$ there is an error because $\underline{q}'$ is closer to $\underline{a}_4$. How did this happen? Because the noise $\underline{n}$ can shift the transmitted vector, $\underline{a}_1$ here, to any position in the plane.

$$d(\underline{a}_i, \underline{a}_j) \equiv \text{Euclidean distance between } \underline{a}_i \text{ and } \underline{a}_j$$

[4]This in turn means that $\underline{q}$ is **not**, in general, an element of the constellation.

2.2 Criteria for the design of the system

The goal is to design a system whose error probability is as small as possible. The main setback is, of course, the channel, that will cause disturbances and/or distortions in the transmitted signal. We will see the role played by each block above in achieving this goal.

2.2.1 Encoder

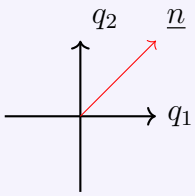
It is designed to minimize the probability of error, and we will soon see that this depends exclusively on the distance between the elements (vectors) in the constellation: the larger the distance between elements in the constellation, the lower the error probability,

$$\uparrow \text{distance between vectors} \Longrightarrow \downarrow P_e.$$

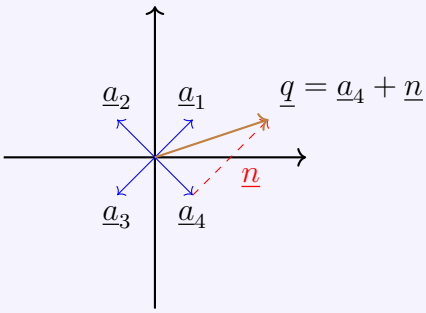
One way of increasing the distance between elements is enlarging the vectors, i.e., increasing their modules.

Impact of noise on the performance from the point of view of the constellation

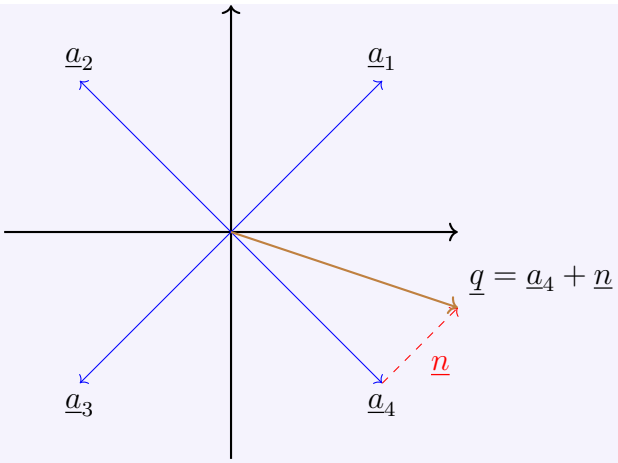
Let us assume the noise vector, $\underline{n}$, which is added to the transmitted symbols, is[a]



The constellation is the same as before, and $\underline{a}_4$ is transmitted



We decide in favor of the closest vector in the constellation, which is $\underline{a}_1$, and there is an error. However, if the vectors representing the elements of the constellation were longer, we would have



and there is no error (the transmitted vector stays in the appropriate quadrant even after adding the noise vector).

[a]Notice that in these kind of pictures, only the head of the arrow is relevant.

We will see later on that increasing the length of a vector in the constellation is tantamount to increasing the energy of the corresponding signal.

The bottom line here is: the further apart the elements of the constellation are from each other the better, and this is controlled by the encoder.

2.2.2 Modulator

Here, the rule is choosing the waveforms/signals that better fit the nature of the channel, i.e.,

- baseband channel → baseband signals, e.g., a rectangular signal
- passband channel → passband signals, e.g., sines and cosines (their frequency components are far away from zero)

2.2.3 Demodulator

The demodulator outputs a vector

$$\underline{q} = \underline{a}_i + \underline{n},$$

which is a *noisy* estimate of the element transmitted, $\underline{a}_i$. Hence it must be designed so that the noise vector $\underline{n}$ within $\underline{q}$ is as small as possible. Ideally, we would like

$$\underline{n} \to 0.$$

In other words, we want to get of rid of as much noise as possible

2.2.4 Detector

We'd like a rule to decide B from $\underline{q}$ that yields the least number of errors. We have seen before that this rule is *usually* proximity, i.e., we compute the distance between $\underline{q}$ and every possible $\underline{a}_i$, and choose that $\underline{a}_i$ which is at minimum distance.

$$P_e \to 0 \Leftrightarrow \#\text{number of erroneous decisions} \to 0$$

In summary

- encoder → minimize the error probability (only depends on the constellation!!)
- modulator → pick up signals fitted to the nature of the channel (baseband or passband)
- demodulator → make $\underline{n}$ within $\underline{q} = \underline{a}_i + \underline{n}$ as small as possible
- detector → decision rule that minimizes the number of errors[a] (we already know most of the time it is going to be proximity)

[a] There is always noise left after *demodulation*.

Quick quiz

If we are given a digital communications system and asked to decrease its error probability, which component or components should we tweak or fine tune? [5]

2.3 Hilbert space for finite-energy signals

When designing the modulator, we assign a waveform to every possible element in the constellation, $\underline{a}_i$ with $i = 1, \cdots, M$. Indirectly, this means assigning a signal $s_i(t)$ to every information symbol b_i. Remember the mapping

$$b_i \longrightarrow \underline{a}_i \longrightarrow s_i(t).$$

Intuitively, in order to minimize the error probability, the receiver should be able to easily tell apart two signals corresponding to different symbols. Then, for every $i \neq j$, signals $s_i(t)$ and $s_j(t)$ should be as different as possible...but it's not easy to tell how different two signals are.

Comparing signals

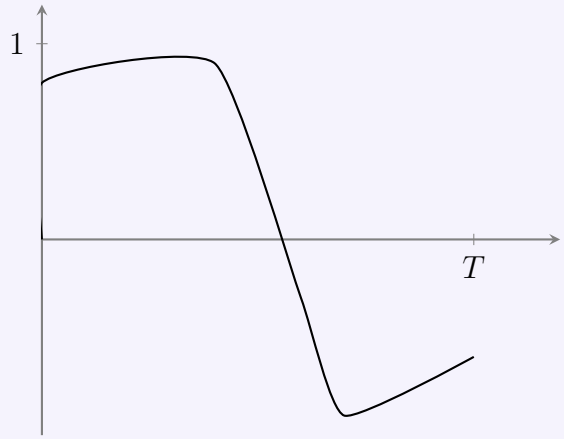
Which signals are the most different from each other? Those on the left or those on the right?

[5] The encoder: it's the the single component on which performance depends.

On the other hand, in the receiver we observe a signal $r(t)$, and we want to know if $r(t)$ resembles more closely $s_i(t)$ or $s_j(t)$. Sometimes it is not obvious...

Received signal not resembling anything possible

What if we transmit a positive rectangular signal and we receive this?



We need a formal/mathematical way of answering this questions. In order to tackle this problem we are going to use a **geometric representation of signals**: if every signal is a vector in a *vector space* then it is very easy to determine the distance between two signals by computing the distance between the corresponding vectors. Also, this geometric representation allows us to use many mathematical tools we are familiar with.

First of all, we need to check if a signal can be considered a vector in a vector space. We now review the definition of vector space.

Definition 2.3.1: Vector space

A **vector space** (also known as *linear* space) V is a set of elements, named *vectors* with the following properties:

1. There exists an operation called *addition*, represented by the symbol $+$ such that $x + y \in V$ for every $x, y \in V$, i.e.,

$$x, y \in V \Longrightarrow x + y \in V$$

   This operation must satisfy the following **properties**:

   a) Commutativity: $\forall x, y \in V, x + y = y + x$

   b) Associativity: $\forall x, y, z \in V, x + (y + z) = (x + y) + z$

   c) Identity element: $\exists\ 0 \in V \mid \forall x \in V, x + 0 = 0 + x = x$

   d) Inverse element: $\forall x \in V\ \exists(-x) \mid x + (-x) = 0$

2. There exists an operation called *scalar multiplication* that takes a scalar $\alpha \in F$ (with F being a *field*) and a vector $x \in V$ to yield another vector, i.e,

$$\forall x \in V,\ \forall \alpha \in F \Longrightarrow \alpha x \in V$$

   This operation must satisfy the following **properties**:

   a) Associativity: $\forall \alpha, \beta \in F, \forall x \in V, \alpha(\beta x) = (\alpha\beta)x$

   b) Identity element: $\exists\ 1 \in F \mid \forall x \in V, 1x = x$

   c) Distributivity of scalar multiplication with respect to vector addition:

$$\forall \alpha \in F, \forall x, y \in V, \alpha(x + y) = \alpha x + \alpha y$$

   d) Distributivity of scalar multiplication with respect to field addition:

$$\forall \alpha, \beta \in F, \forall x \in V, (\alpha + \beta)x = \alpha x + \beta x$$

We now wonder whether the set of complex continuous-time signals is a vector space (*our* V in the above notation). We can **define the operations** of *addition* and *scalar multiplication* **as we like**. If we define them[6] so that:

- addition $\equiv$ pointwise addition of signals (usual addition of signals)
- scalar multiplication $\equiv$ conventional product of a complex[7] scalar times a signal

then, it is straightforward to prove that this *addition* operation satisfies the required properties, and the same for the *scalar multiplication* operation.

[6] just like in the "Circuits & Systems" course...

[7] The set of complex numbers is a field.

⭐ Signals as vectors in a vector space

The set of complex continuous-time signals under these operations of *addition* and *scalar multiplication* has a structure of vector space.

This vector space algebra in itself is not very useful for us. Our final goal is to measure the distance or similarity between signals. We are trying to decide which set of signals $\{s_1(t), s_2(t), \cdots, s_M(t)\}$ should be used by the modulator (those that are as different as possible). So, we need something else. Hence, rather than talking about vector spaces, we'll be talking about Hilbert spaces. A **Hilbert space** is a vector space in which an **inner product** is defined and an...

Definition 2.3.2: Inner product

...in a vector space V over the field F is a mapping between the set of pairs of vectors and the field F,

$$f : (V, V) \to F,$$

that is, it assigns a scalar to every pair of vectors. The inner product between vectors $\underline{x}$ and $\underline{y}$ is denoted as $\langle \underline{x}, \underline{y} \rangle$ and must verify the following properties

1. $\langle \underline{x}, \underline{y} \rangle = \langle \underline{y}, \underline{x} \rangle^*$
2. $\langle \underbrace{\alpha \underline{x} + \beta \underline{y}}_{\substack{\text{also a vector} \\ \text{by definition}}}, \underline{z} \rangle = \alpha \langle \underline{x}, \underline{z} \rangle + \beta \langle \underline{y}, \underline{z} \rangle$
3. $\langle \underline{x}, \underline{x} \rangle \geq 0$
4. $\langle \underline{x}, \underline{x} \rangle = 0 \iff \underline{x} = 0$

An inner product *induces* a **norm**[8] (whenever we have an inner product we get a norm "for free") which is defined as

$$||\underline{x}|| = \sqrt{\underbrace{\langle \underline{x}, \underline{x} \rangle}_{\geq 0}}$$

and, in turn, the norm gives rise to the definition of **distance between vectors** we were looking for,

$$d(\underline{x}, \underline{y}) = \left|\left| \underline{x} - \underline{y} \right|\right|.$$

Using the norm, we can also compute the angle between two vectors $\underline{x}$ and $\underline{y}$ as

$$\theta = \arccos \frac{\mathrm{Re}\left\{ \langle \underline{x}, \underline{y} \rangle \right\}}{||\underline{x}|| \left|\left| \underline{y} \right|\right|}.$$

When $\langle \underline{x}, \underline{y} \rangle = 0$, the vectors form an angle $\theta = \pi/2$, and we say they are *orthogonal*.

Notice that the norm and the distance stem from (are determined by) the inner product, but any function which verifies the above conditions is an inner product. There

[8] A norm is just a function satisfying some properties.

are many ways in which one can define the inner product and, as expected, different definitions yield different Hilbert spaces which, in turn, have different metrics for the distance. We are going to define the inner product in the vector space of continuous-time signals (the *vectors* in our Hilbert space are complex continuous-time signals), and *our* definition is

$$\langle \underline{x}, \underline{y} \rangle = \int_{-\infty}^{\infty} x(t) y^*(t) dt.$$

With this definition, the resulting norm and distance between vectors are

$$||\underline{x}|| = \sqrt{\langle \underline{x}, \underline{x} \rangle} = \sqrt{\int_{-\infty}^{\infty} x(t) x^*(t) dt} = \sqrt{\int_{-\infty}^{\infty} |x(t)|^2 \, dt} = \sqrt{\varepsilon \{x(t)\}} \tag{2.1}$$

$$\begin{aligned} d(\underline{x}, \underline{y}) &= ||\underline{x} - \underline{y}|| = \sqrt{\langle \underline{x} - \underline{y}, \underline{x} - \underline{y} \rangle} = \sqrt{\int_{-\infty}^{\infty} (x(t) - y(t)) \, (x(t) - y(t))^* \, dt} \\ &= \sqrt{\int_{-\infty}^{\infty} x(t) x^*(t) dt - \int_{-\infty}^{\infty} x(t) y^*(t) dt - \int_{-\infty}^{\infty} y(t) x^*(t) dt + \int_{-\infty}^{\infty} y(t) y^*(t) dt} \\ &= \sqrt{\varepsilon \{x(t)\} + \varepsilon \{y(t)\} - \underbrace{\int_{-\infty}^{\infty} x(t) y^*(t) dt}_{\text{correlation}} - \underbrace{\int_{-\infty}^{\infty} y(t) x^*(t) dt}_{\text{correlation}}}. \end{aligned} \tag{2.2}$$

Intuitively, the above "correlation"s are measures of similarity, and we have

$$\uparrow \text{similarity} \Rightarrow \uparrow \text{correlation} \Rightarrow \downarrow \text{distance}.$$

The above formula for computing the distance between signals is not so convenient (it involves the integral of a product). However, from algebra we know that any element in a vector space can be represented as a set of coordinates (what we usually call a vector!!) with respect to some *basis*. That is going to make things much easier.

2.3.1 Orthonormal bases

First of all, we recall what a basis is.

Definition 2.3.3: Basis in a vector space

A **basis in a Hilbert space** H over a field F is a subset of elements[a] $\{\underline{b}_1, \underline{b}_2, \cdots\} \subset H$ that determine a set of *unique* coefficients, $\{c_1, c_2, \cdots\} \subset F$, for every vector in the space, x, such that the latter can be expressed as

$$\underline{x} = \sum_j c_j(\underline{x})\underline{b}_j, \tag{2.3}$$

where

$c_j(\underline{x}) \in F \equiv$ coordinate of vector $\underline{x}$ with respect to the j-th element in the basis

Notice that

- the coefficients, c_j, are dependent on the vector they aim to represent, and
- $\underline{x}$ is an element in the Hilbert space...and so are the $\{\underline{b}_1, \underline{b}_2, \cdots\}$.

[a]Notice that, for us, these are continuous-time signals!!

For us,

- H is the vector space of continuous-time signals, and
- the field F is the set of complex numbers, $\mathbb{C}$.

The above definition means that *any* element in the vector space can be represented as a linear combination of the elements in the basis, and this linear combination is given by a **unique**[9] set of coordinates (we have a coefficient associated with every element in the basis).

Using the set of coordinates for a signal instead of the signal itself is interesting for both

- representing the signals encompassed by the modulator in a more compact manner, and
- easily comparing two signals (we just need to compute the distance between their corresponding vectors).

There are a couple of properties that we would like for our basis:

$$\left.\begin{array}{l}\langle \underline{b}_i, \underline{b}_j \rangle = 0,\ \forall i \neq j \Rightarrow \text{basis is orthogonal} \\ \langle \underline{b}_j, \underline{b}_j \rangle = 1\end{array}\right\} \Rightarrow \text{basis is } \textbf{orthonormal}$$

[9]This is important: with respect to some basis, the coefficients are unique. In other words, we cannot find another set of coordinates that reconstruct the signal.

Energy of a signal

Notice that, in our Hilbert space (of signals), the inner product between a signal and itself, e.g., $\langle \underline{b}_j, \underline{b}_j \rangle$, is precisely its energy (see Equation (2.1)).

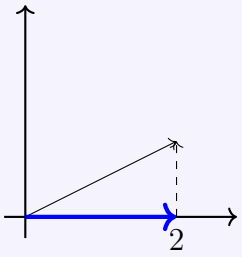
These properties are desirable because, **if the basis is orthonormal (and *only in this case*)**, then

$\langle \underline{x}, \underline{b}_j \rangle = c_j(\underline{x}) \equiv$ coordinate of vector $\underline{x}$ with respect to the j-th element in the basis.

Geometrically this inner product can be interpreted as **the projection of vector $\underline{x}$ on the j-th element in the basis**[10].

Example

In 2D-space, $\langle (2,1), (1,0) \rangle = 2$ is the coordinate of vector $(2,1)$ with respect to element $(1,0)$ in the basis (along the x-axis).



In our case, the elements in the vector space are continuous-time signals, and hence $\langle \sin(t), \log(t) \rangle$ is the projection of $\sin(t)$ over $\log(t)$.

It is straightforward to mathematically prove this statement. Assume $\{\underline{b}_1, \underline{b}_2, \cdots, \underline{b}_N\} \subset H$ is an *orthonormal* basis for Hilbert space H and $x \in H$. Then

$$\langle \underline{x}, \underline{b}_j \rangle =$$

since $\underline{x}$ is an element in the vector space, it can be represented as a linear combination of the elements of *any* basis

$$= \langle \sum_{l=1}^{N} c_l(\underline{x}) \underline{b}_l, \underline{b}_j \rangle$$

using the properties of the inner product

$$= \sum_{l=1}^{N} c_l(\underline{x}) \langle \underline{b}_l, \underline{b}_j \rangle$$

[10] Again, only if the basis is orthonormal does the projection match the coordinate.

$$\langle \underline{b}_l, \underline{b}_j \rangle = \begin{cases} 0, & l \neq j \\ 1, & l = j \end{cases}$$

i.e., since the base is orthonormal:

- the inner product between two different elements is 0
- the inner product of an element with itself is 1

$$= c_j(\underline{x})$$

In our case, we are going to represent every signal[11] in the modulator, $s_i(t)$, as

$$s_i(t) = \sum_{j=1}^{N} \underbrace{c_j(s_i(t))}_{a_{ij}} \phi_j(t) = \sum_{j=1}^{N} a_{ij}\phi_j(t)$$

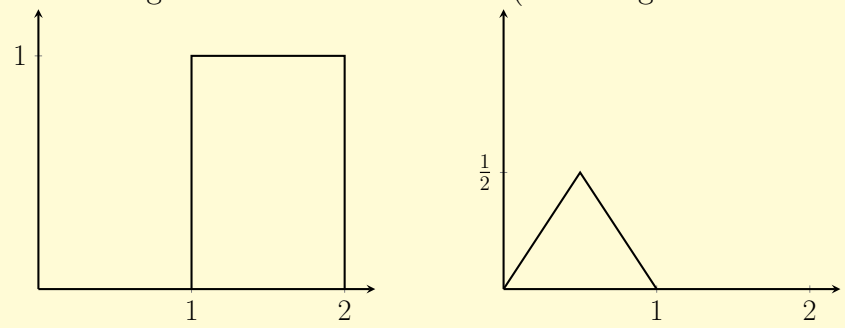
where a_{ij}, $j = 1, \cdots, N$, are the coordinates[12] of the signal $s_i(t)$ with respect to the basis $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$ (of continuous-time signals). We can collect all the coordinates for signal $s_i(t)$ in a vector to get

$$\underline{a}_i = \begin{bmatrix} a_{i,1} \\ \vdots \\ a_{i,N} \end{bmatrix} \overset{\text{with the previous notation}}{=} \begin{bmatrix} c_1(s_i(t)) \\ \vdots \\ c_N(s_i(t)) \end{bmatrix}.$$

Hence, given a basis $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$, $a_{i,j}$ is the coordinate of the i-th signal, $s_i(t)$, with respect to the j-th element in the basis, $\phi_j(t)$.

? Quick quiz

Are the signals below orthonormal (according to our definition of inner product)?



Justify your answer. [13]

[11] Our vector space is one of continuous-time signals: the $\underline{x}$ and $\underline{b}_j$'s in Equation (2.3) are signals.

[12] We said earlier that the coordinates depend on the specific element you want to represent...that's why we need the i in the subindex.

If for every signal, $s_i(t)$, we have a vector, $\underline{a}_i$, encompassing its corresponding coordinates with respect to some *orthonormal* basis, then computing the energy of a signal or the distance between any two given signals is much easier.

Energy

$$\varepsilon_{s_i} = \int_{-\infty}^{\infty} |s_i(t)|^2 \, dt = \int_{-\infty}^{\infty} s_i(t) s_i^*(t) dt = \int_{-\infty}^{\infty} \sum_{l=1}^{N} a_{il} \phi_l(t) \sum_{m=1}^{N} a_{im}^* \phi_m^*(t) dt$$

the integral of the summation is the summation of the integral; the coordinates do not depend on time and can be pulled out of the integral

$$= \sum_{l=1}^{N} \sum_{m=1}^{N} a_{il} a_{im}^* \int_{-\infty}^{\infty} \phi_l(t) \phi_m^*(t) dt = \sum_{l=1}^{N} \sum_{m=1}^{N} a_{il} a_{im}^* \langle \phi_l(t), \phi_m(t) \rangle$$

the basis is orthonormal, which means

$$\langle \phi_l(t), \phi_m(t) \rangle = \begin{cases} 0, & l \neq m \\ 1, & l = m \end{cases}$$

$$= \sum_{l=1}^{N} |a_{il}|^2 = |\underline{a}_i|^2 ,$$

i.e., we can compute the energy of a signal as the *squared*[14] modulus of the corresponding vector in the constellation.

[13] They are not. They are orthogonal, because the inner product is zero (signals don't overlap), but the second signal does not have unit energy.

[14] Energy is a "square magnitude"!!

Distance

$$
\begin{aligned}
d(s_i(t), s_j(t)) &= \|s_i(t) - s_j(t)\| = \sqrt{\langle s_i(t) - s_j(t), s_i(t) - s_j(t)\rangle} \\
&= \sqrt{\int_{-\infty}^{\infty} (s_i(t) - s_j(t)) \left(s_i(t) - s_j(t)\right)^*} = \sqrt{\int_{-\infty}^{\infty} |s_i(t) - s_j(t)|^2 \, dt} \\
&\overset{(2.2)}{=} \sqrt{\varepsilon_{s_i} + \varepsilon_{s_j} - \int_{-\infty}^{\infty} s_i(t) s_j^*(t) dt - \int_{-\infty}^{\infty} s_j(t) s_i^*(t) dt} \\
&\overset{\text{real signals}}{=} \sqrt{\varepsilon_{s_i} + \varepsilon_{s_j} - 2\int_{-\infty}^{\infty} s_i(t) s_j(t) dt} \\
&= \sqrt{\sum_{m=1}^{N} a_{im}^2 + \sum_{m=1}^{N} a_{jm}^2 - 2\int_{-\infty}^{\infty} \sum_{m=1}^{N} a_{im}\phi_m(t) \sum_{n=1}^{N} a_{jn}\phi_n(t) dt} \\
&= \sqrt{\sum_{m=1}^{N} a_{im}^2 + \sum_{m=1}^{N} a_{jm}^2 - 2\sum_{m=1}^{N}\sum_{n=1}^{N} a_{im}a_{jn}\int_{-\infty}^{\infty} \phi_m(t)\phi_n(t) dt} \\
&= \sqrt{\sum_{m=1}^{N} a_{im}^2 + \sum_{m=1}^{N} a_{jm}^2 - 2\sum_{m=1}^{N} a_{im}a_{jm}} = \sqrt{\sum_{m=1}^{N} a_{im}^2 + a_{jm}^2 - 2a_{im}a_{jm}} \\
&= \sqrt{\sum_{m=1}^{N} (a_{im} - a_{jm})^2} = d(\underline{a}_i, \underline{a}_j),
\end{aligned}
$$

where the latter distance refers to (regular) Euclidean distance between vectors[15]. Hence, we can compute the distance between two signals, $s_i(t)$ and $s_j(t)$ by simply computing the Euclidean distance between their corresponding vectors[16].

In summary, we can compute the

- energy of a signal
- distance between any two given signals,

without computing neither **integrals** nor products of signals!![17] This is the main benefit of a geometric representation of signals.

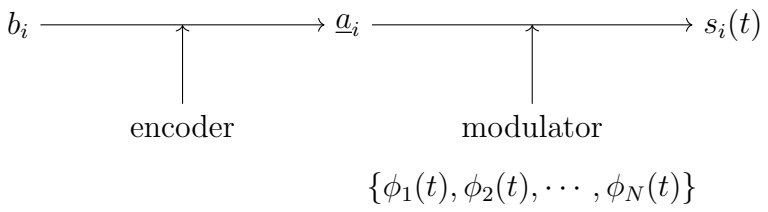
2.3.2 Encoder-modulator connection

Remember the scheme

[15] $d(\cdot, \cdot)$ is an *argument-wise* notation: if the arguments are continuous-time signals it refers to the Hilbert space definition of distance, whereas if the arguments are vectors then it refers to Euclidean distance.

[16] We assumed real signals, but it is straightforward to extend the proof for complex signals.

[17] Notice that the above formulas hold regardless of the particular orthonormal basis considered.

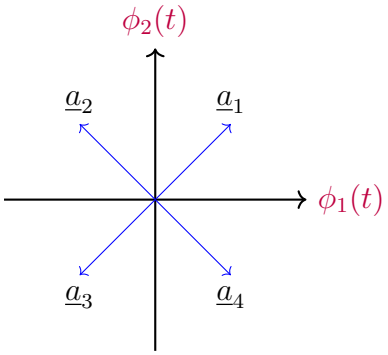


For us, the orthonormal basis is a property[18] of the modulator, and the output of the encoder, $\underline{a}_i$, for information symbol b_i is simply a vector encompassing the coordinates of $s_i(t)$ with respect to the basis of the modulator, $\phi_i(t), i = 1, \cdots, N$. In other words, the modulator has an orthonormal basis attached to it (embedded in it, so to say), and every time it receives a vector of coordinates (from the encoder), it uses this basis and the equation

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\phi_j(t),$$

to build the corresponding signal. Again, notice that the basis $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$ is something related to/associated with the modulator

We are now in a position to properly label the axes in the plot of a constellation: each axis is associated with a different element from the basis so that a_{i1} is associated with $\phi_1(t)$, a_{i2} with $\phi_2(t)$, and so forth and so on. For instance, if we label the axes of the plot for the constellation used in the examples of Section 2.2.1 we get



2.4 Gram-Schmidt process

In the previous section we have seen that a geometric representation of signals (relying on an orthonormal basis and the corresponding coordinates of every signal with respect to it) provides an easy way of telling how different or similar two signals are, which was the main problem posed at the beginning of that section. However, if we have a set of signals, such as those employed by the modulator, i.e., $\{s_1(t), s_2(t), \cdots, s_M(t)\}$, how can we find an orthonormal basis spanning the same subspace as those signals?[19] Using the Gram-Schmidt process!!

[18] ...sort of like in Object-Oriented programming (if you are a computer science-oriented person)

[19] Our basis should not yield signals that could not be obtained as linear combinations of $\{s_1(t), s_2(t), \cdots, s_M(t)\}$. In other words, we are looking for a basis with the minimum number of elements.

Our scenario is this: we have a set of M signals $\{s_1(t), s_2(t), \cdots, s_M(t)\}$ that are used to transmit information across the channel, and we want to obtain an **orthonormal** basis, $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$ with $N \leq M$, such that

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\phi_j(t),$$

where

$$a_{ij} \equiv \text{coordinate of } s_i(t) \text{ with respect to } \phi_j(t),$$

i.e., the signals in the orthonormal basis allow us to represent the signals in the modulator as vectors of coefficients,

$$\underline{a}_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{iN} \end{bmatrix},$$

in the sense that a vector of coefficients perfectly determines the corresponding signal. In a_{ij}, subindex i refers to a vector in the constellation and j to a coordinate within that vector.

The **Gram-Schmidt** process goes sequentially over *every one* of the input signals, and can be summarized as follows:

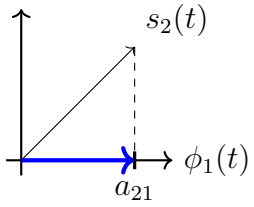
$\boldsymbol{s}_1(t)$ The first element in the basis is always the first signal in the set...but normalized so that its energy is 1 (we aim at an orthonormal basis), i.e.,

$$\phi_1(t) = \frac{s_1(t)}{\sqrt{\varepsilon_{s_1}}}$$

where ε_{s_i} stands for the energy of $s_i(t)$, which is a continuous-time signal.

So far, with our current basis, we can only represent signal $s_1(t)$.

$\boldsymbol{s}_2(t)$ We now move on and process the second signal, $s_2(t)$. We also want to represent $s_2(t)$ using the elements in the orthonormal basis. In order to check whether that's *already* possible or not, we project (that means computing an inner product) $s_2(t)$ over $\phi_1(t)$ and we have something like this



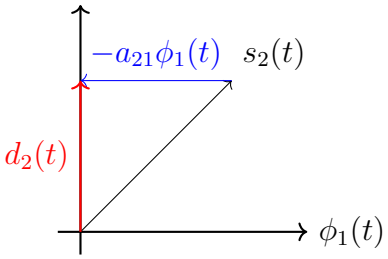
with

$$a_{21} = \langle s_2(t), \phi_1(t) \rangle.$$

In the above example $s_2(t)$ has a component along *the* $\phi_1(t)$ *axis*, but there is some part of $s_2(t)$ which cannot be represented using only $\phi_1(t)$ ($s_2(t)$ has a component along $\phi_1(t)$...but there is more). That part is

$$s_2(t) - a_{21}\phi_1(t) = d_2(t), \tag{2.4}$$

and we are giving that a name, which is $d_2(t)$ (as in *difference*). It is easy to see that $d_2(t)$ is orthogonal to $\phi_1(t)$ (which makes sense, since we have subtracted from $s_2(t)$ its projection over $\phi_1(t)$[20])



We can also check this mathematically: if $d_2(t)$ is orthogonal to $\phi_1(t)$, then their inner product should be zero,

$$\begin{aligned}\langle \phi_1(t), d_2(t)\rangle &= \langle \phi_1(t), s_2(t) - a_{21}\phi_1(t)\rangle \overset{\text{properties } \langle\cdot\rangle}{=} \langle \phi_1(t), s_2(t)\rangle - a_{21}\cancelto{1}{\langle \phi_1(t), \phi_1(t)\rangle} \\ &= a_{21} - a_{21} = 0\end{aligned}$$

What happens if $d_2(t) = 0$?

$$d_2(t) = 0 = s_2(t) - a_{21}\phi_1(t) \Rightarrow s_2(t) = a_{21}\phi_1(t),$$

i.e., $s_2(t)$ is $\phi_1(t)$ multiplied by some constant. In other words, $s_2(t)$ can be represented using the elements that are already in the basis (so far only $\phi_1(t)$), and we don't need to add a new signal to the basis in order to represent $s_2(t)$. If you check the picture, $d_2(t) = 0$ means $s_2(t)$ is along the $\phi_1(t)$ axis.

The other possibility is $d_2(t) \neq 0$. In such a case, we have that $d_2(t)$ is **orthogonal to** $\phi_1(t)$ and, at the sight of equation (2.4), it allows representing $s_2(t)$,

$$s_2(t) = a_{21}\phi_1(t) + d_2(t),$$

with $d_2(t) \neq 0$. Hence, we have a signal, $d_2(t)$, that is orthogonal to $\phi_1(t)$ and, along with the latter, will allow us to represent $s_2(t)$. We should add it to the basis, but we cannot add it as it is because the basis is *orthonormal*, that is, every element must have unit energy. It's not a problem: our second element in the basis is $d_2(t)$ normalized so that it has unit energy,

$$\phi_2(t) = \frac{d_2(t)}{\sqrt{\mathcal{E}_{d_2}}}$$

[20]Notice that, mathematically, orthogonal means 0 projection!!

(in order to normalize a signal you need to divide by the *square root* of its energy).

In summary, what we did here is computing the coordinates of $s_2(t)$ with respect to the elements that already are in the basis, we subtracted their contribution from $s_2(t)$, and we normalized the resulting signal before adding it to the basis.

$s_3(t)$ We continue *processing* $s_3(t)$...and we follow the same steps

- we compute the coordinates of $s_3(t)$ with respect to the elements that are already in the basis and, assuming there is now two elements in the latter[21], $\phi_1(t)$ and $\phi_2(t)$, we have

$$a_{31} = \langle s_3(t), \phi_1(t) \rangle$$
$$a_{32} = \langle s_3(t), \phi_2(t) \rangle$$

- we find out the part of $s_3(t)$ that we cannot represent with whatever we have right now in the basis (that's what we call $d_3(t)$),

$$d_3(t) = s_3(t) - a_{31}\phi_1(t) - a_{32}\phi_2(t)$$

  ($d_3(t)$ is orthogonal to every signal already in the basis)

- if $d_3(t) \neq 0$, we normalize it and add it to the basis,

$$\phi_3(t) = \frac{d_3(t)}{\sqrt{\varepsilon_{d_3}}}$$

In general, for $s_l(t)$, when there are $k-1$ elements in the basis , we process the l-th signal[22], with $l - 1 \geq k - 1$, as follows

- we compute the coordinates with respect to all the elements that are already in the basis

$$\begin{aligned} a_{l1} &= \langle s_l(t), \phi_1(t) \rangle \\ a_{l2} &= \langle s_l(t), \phi_2(t) \rangle \\ \vdots &\qquad \vdots \\ a_{lk-1} &= \langle s_l(t), \phi_{k-1}(t) \rangle \end{aligned}$$

  ($k-1$ is the number of elements currently in the basis).

- we compute the *difference signal*, $d_l(t)$, as

$$d_l(t) = s_l(t) - \sum_{j=1}^{k-1} a_{lj}\phi_j(t)$$

- two possibilities:

[21] i.e., assuming $d_2(t) \neq 0$ in the previous step

[22] ...meaning we have already processed $l-1$

– If $d_l(t) \neq 0$, we normalize it and add it to the basis,

$$\phi_k(t) = \frac{d_l(t)}{\sqrt{\varepsilon_{d_l}}}, \quad (d_l(t) \neq 0) \ .$$

– If $d_l(t) = 0$, we just go for the next signal.

Notice that $k \leq l$ because we don't add a new element to the basis every time we *process* a signal. At the end of the procedure we have a basis

$$\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t), \}$$

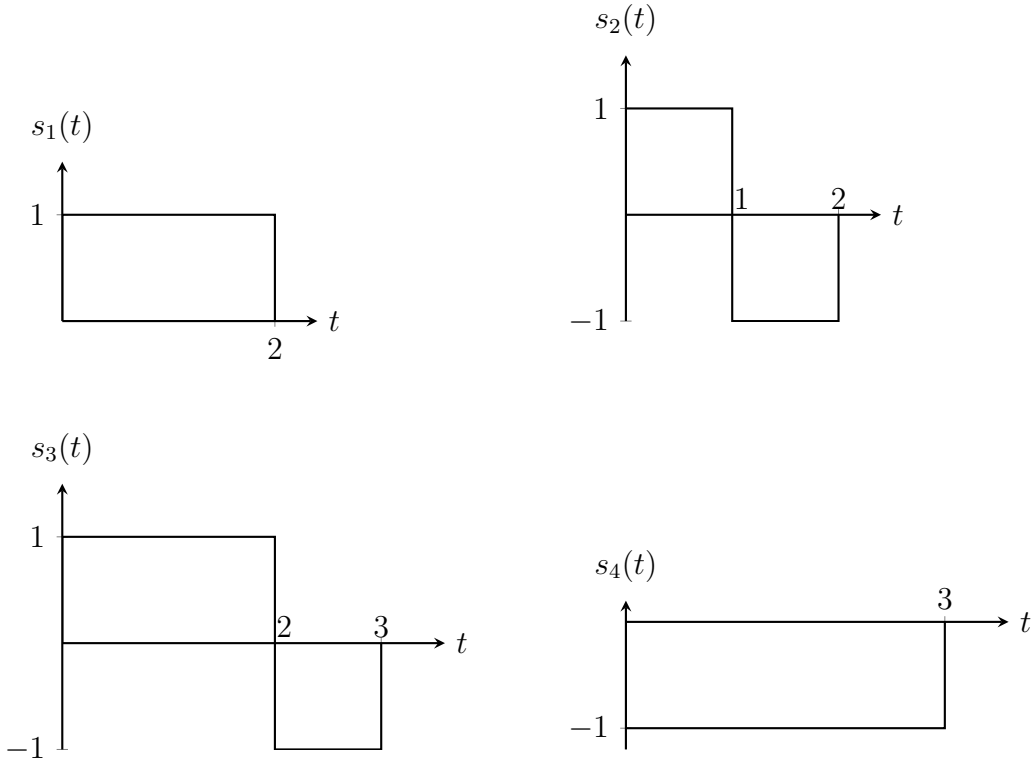
with

$$\overbrace{N}^{\substack{\text{\# elements} \\ \text{in the basis}}} \leq \overbrace{M}^{\substack{\text{\# signals we} \\ \text{want to represent}}} ,$$

where N is the dimension of the elements in our constellation.

2.4.1 Example: Gram-Schmidt process

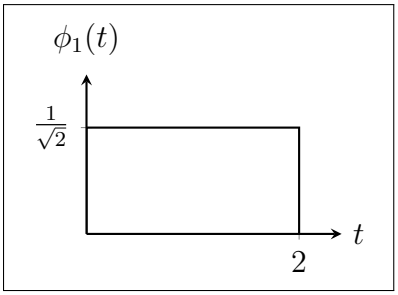
Let us use Gram-Schmidt to find an orthonormal basis for the 4 signals below.



We just apply the above steps:

$\boldsymbol{s}_1(t)$ The first signal goes directly into the basis after normalization, i.e.,

$$\phi_1(t) = \frac{s_1(t)}{\sqrt{\varepsilon_{s_1}}} = \frac{s_1(t)}{\sqrt{2}}$$



with

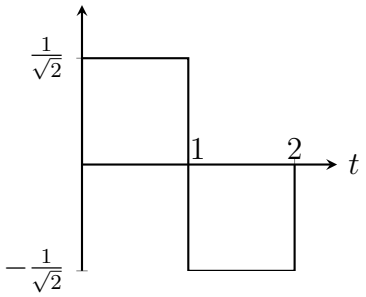
$$\varepsilon_{s_1} = \int_{-\infty}^{\infty} |s_1(t)|^2 \, dt = \int_0^2 1^2 dt = 2$$

$\boldsymbol{s_2(t)}$ We go for the next (second) signal in the given set. We need to compute its coordinates with respect to all the elements that are already in the basis. So far, we only have $\phi_1(t)$, and the coordinate of $\boldsymbol{s}_2(t)$ with respect to $\phi_1(t)$ is given by[23]

$$a_{21} = \langle s_2(t), \phi_1(t) \rangle = \int_{-\infty}^{\infty} s_2(t)\phi_1(t)dt.$$

An easy (and less error-prone) approach to compute the above integral is think of it as the area under the signal given by the product $s_2(t)\phi_1(t)$. The latter can be computed piecewise by evaluating the product in time intervals in which both signals are constant (and hence the product is straightforward). Here, both signals are constant during intervals $[0, 1]$ and $[1, 2]$ (although with different value in the latter)[24]. The product of the two signals is then

$s_2(t)\phi_1(t)$



$$s_2(t)\phi_1(t) = \begin{cases} \overbrace{1}^{s_2(t)} \cdot \overbrace{\frac{1}{\sqrt{2}}}^{\phi_1(t)}, & 0 \leq t < 1 \\ \underbrace{-1}_{s_2(t)} \cdot \underbrace{\frac{1}{\sqrt{2}}}_{\phi_1(t)}, & 1 \leq t < 2 \end{cases}$$

and hence

$$a_{21} = \langle s_2(t), \phi_1(t) \rangle = \int_0^1 \frac{1}{\sqrt{2}} dt + \int_1^2 \frac{-1}{\sqrt{2}} dt = \int_0^1 \frac{1}{\sqrt{2}} dt - \int_0^1 \frac{1}{\sqrt{2}} dt = \boxed{0}.$$

Next, **from** $s_2(t)$, we must build a signal that is orthogonal to every one of the elements in the (current) basis. It is the *difference* signal, $d_2(t)$, that we defined as

$$d_2(t) = s_2(t) - \cancelto{0}{a_{21}\phi_1(t)} = s_2(t).$$

This means $s_2(t)$ $(= d_2(t))$ is already orthogonal to $\phi_1(t)$...but that is something we already knew because the inner product between them, i.e., $\langle s_2(t), \phi_1(t) \rangle = a_{21}$,

[23] Notice that, since we are dealing with real signals, we are omitting the conjugate superindex everywhere.

[24] Choosing the larger interval $[0, 2]$ would not be so convenient because $s_2(t)$ varies within that interval.

is zero (the projection of $s_2(t)$ over $\phi_1(t)$ is null or, equivalently, the angle between them is $\pi/2$).

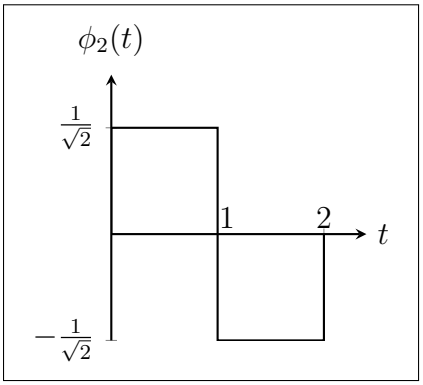
The last step is normalizing $d_2(t)$ before adding it to the basis,

$$\phi_2(t) = \frac{d_2(t)}{\sqrt{\varepsilon_{d_2}}} = \frac{s_2(t)}{\sqrt{\varepsilon_{s_2}}} = \frac{s_2(t)}{\sqrt{2}},$$

where ε_{s_2} was computed as

$$\varepsilon_{s_2} = \int_{-\infty}^{\infty} |s_2(t)|^2 \, dt = \int_0^1 1^2 dt + \int_1^2 (-1)^2 dt = 2.$$

$$\phi_2(t) = \frac{s_2(t)}{\sqrt{2}}$$

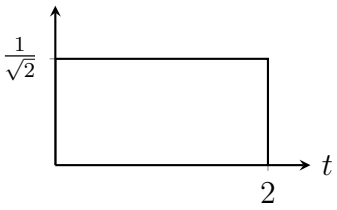


$\boldsymbol{s}_3(t)$ We go for the next (third) signal in the given set. We start by computing its coordinates with respect to all the elements (so far) in the basis.

$$a_{31} = \langle s_3(t), \phi_1(t) \rangle = \int_{-\infty}^{\infty} s_3(t)\phi_1(t)dt.$$

The signal to be integrated is

$s_3(t)\phi_1(t)$



$$s_3(t)\phi_1(t) = \begin{cases} 1 \cdot \frac{1}{\sqrt{2}}, & 0 \le t < 2 \\ -1 \cdot 0, & 2 \le t < 3 \end{cases} = \begin{cases} \frac{1}{\sqrt{2}}, & 0 \le t < 2 \\ 0, & 2 \le t < 3 \end{cases},$$

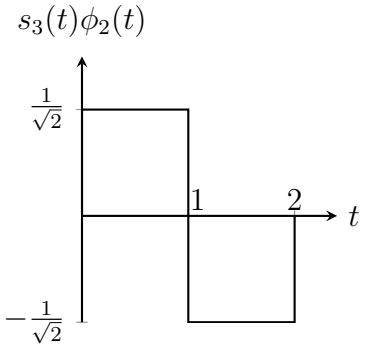
and hence

$$a_{31} = \langle s_3(t), \phi_1(t) \rangle = \int_{-\infty}^{\infty} s_3(t)\phi_1(t)dt = \int_0^2 \frac{1}{\sqrt{2}} dt = \frac{2}{\sqrt{2}} = \boxed{\sqrt{2}}.$$

On the other hand,

$$a_{32} = \langle s_3(t), \phi_2(t) \rangle = \int_{-\infty}^{\infty} s_3(t)\phi_2(t)dt,$$

with the signal in the integrand being

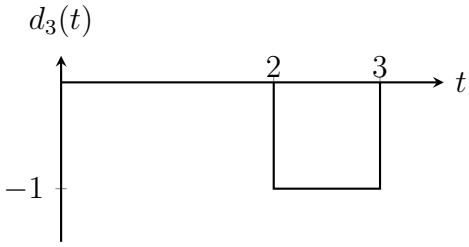


$$s_3(t)\phi_2(t) = \begin{cases} 1 \cdot \frac{1}{\sqrt{2}}, & 0 \le t < 1 \\ 1 \cdot -\frac{1}{\sqrt{2}}, & 1 \le t < 2 \\ -1 \cdot 0, & 2 \le t < 3 \end{cases} = \begin{cases} \frac{1}{\sqrt{2}}, & 0 \le t < 1 \\ -\frac{1}{\sqrt{2}}, & 1 \le t < 2 \\ 0, & 2 \le t < 3 \end{cases}.$$

Then, we have

$$a_{32} = \langle s_3(t), \phi_2(t) \rangle = \int_{-\infty}^{\infty} s_3(t)\phi_2(t)dt = \int_0^1 \frac{1}{\sqrt{2}}dt + \int_1^2 \frac{-1}{\sqrt{2}}dt = \boxed{0}.$$

From the coordinates we compute the *difference* signal $d_3(t)$, which is orthogonal to all the signals already in the basis,



$$\begin{aligned} d_3(t) &= s_3(t) - \sum_{j=1}^{2} a_{3j}\phi_j(t) \\ &= s_3(t) - \sqrt{2}\phi_1(t) - 0 \cdot \phi_2(t) \\ &= s_3(t) - \sqrt{2}\phi_1(t). \end{aligned}$$

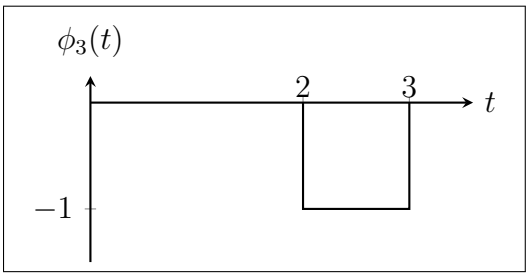
This signal goes into the basis after normalization,

$$\phi_3(t) = \frac{d_3(t)}{\sqrt{\varepsilon_{d_3}}} = d_3(t),$$

where we have used that

$$\varepsilon_{d_3} = \int_{-\infty}^{\infty} |d_3(t)|^2 \, dt = \int_2^3 |-1|^2 \, dt = 1.$$

$$\phi_3(t) = d_3(t) = s_3(t) - \sqrt{2}\phi_1(t)$$

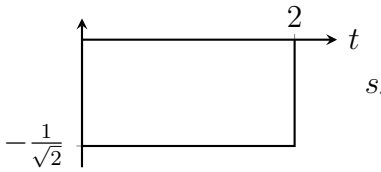


$\boldsymbol{s}_4(t)$ We go for the last (fourth) signal in the set. Again, we first compute its coordinates with respect to all the elements in the basis. The first coordinate is

$$a_{41} = \langle s_4(t), \phi_1(t) \rangle = \int_{-\infty}^{\infty} s_4(t)\phi_1(t)dt.$$

Now, the signal $s_4(t)\phi_1(t)$ is

$s_4(t)\phi_1(t)$



$$s_4(t)\phi_1(t) = \begin{cases} -1 \cdot \frac{1}{\sqrt{2}}, & 0 \le t < 2 \\ -1 \cdot 0, & 2 \le t < 3 \end{cases} = \begin{cases} -\frac{1}{\sqrt{2}}, & 0 \le t < 2 \\ 0, & 2 \le t < 3 \end{cases},$$

and hence we have

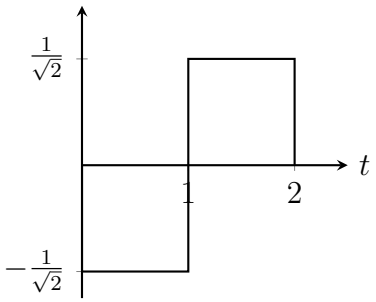
$$a_{41} = \langle s_4(t), \phi_1(t) \rangle = \int_{-\infty}^{\infty} s_4(t)\phi_1(t)dt = \int_0^2 -\frac{1}{\sqrt{2}}dt = -\frac{2}{\sqrt{2}} = \boxed{-\sqrt{2}}.$$

The second coordinate is

$$a_{42} = \langle s_4(t), \phi_2(t) \rangle = \int_{-\infty}^{\infty} s_4(t)\phi_2(t)dt,$$

and we need to integrate

$s_4(t)\phi_2(t)$



$$s_4(t)\phi_2(t) = \begin{cases} -1 \cdot \frac{1}{\sqrt{2}}, & 0 \le t < 1 \\ -1 \cdot \frac{-1}{\sqrt{2}}, & 1 \le t < 2 \\ -1 \cdot 0, & 2 \le t < 3 \end{cases} = \begin{cases} -\frac{1}{\sqrt{2}}, & 0 \le t < 1 \\ \frac{1}{\sqrt{2}}, & 1 \le t < 2 \\ 0, & 2 \le t < 3 \end{cases}.$$

Thus,

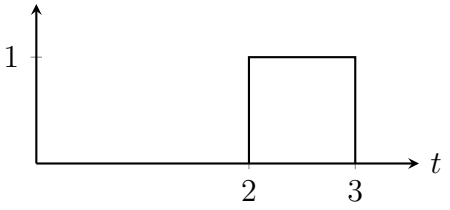
$$a_{42} = \langle s_4(t), \phi_2(t) \rangle = \int_{-\infty}^{\infty} s_4(t)\phi_2(t)dt = \int_0^1 -\frac{1}{\sqrt{2}}dt + \int_0^1 \frac{1}{\sqrt{2}}dt = \boxed{0}.$$

Last, the coordinate of $s_4(t)$ with respect to $\phi_3(t)$ is

$$a_{43} = \langle s_4(t), \phi_3(t) \rangle = \int_{-\infty}^{\infty} s_4(t)\phi_3(t)dt,$$

with

$s_4(t)\phi_3(t)$



$$s_4(t)\phi_3(t) = \begin{cases} -1 \cdot 0, & 0 \le t < 2 \\ -1 \cdot -1, & 2 \le t < 3 \end{cases} = \begin{cases} 0, & 0 \le t < 2 \\ 1, & 2 \le t < 3 \end{cases}.$$

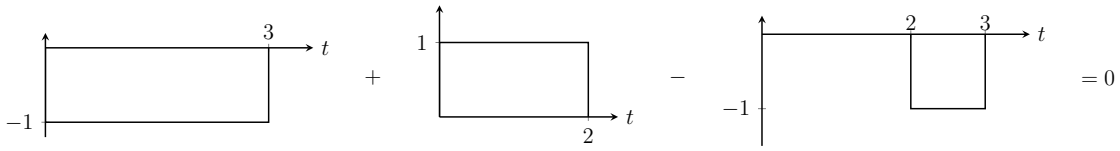
Then,

$$a_{43} = \langle s_4(t), \phi_3(t) \rangle = \int_{-\infty}^{\infty} s_4(t)\phi_3(t)dt = \int_2^3 1dt = \boxed{1}.$$

The *difference* signal is then

$$d_4(t) = s_4(t) - \sum_{j=1}^{3} a_{4j}\phi_j(t) = s_4(t) - a_{41}\phi_1(t) - \cancelto{0}{a_{42}\phi_2(t)} - a_{43}\phi_3(t)$$
$$= s_4(t) - (-\sqrt{2})\phi_1(t) - \phi_3(t) = s_4(t) + \sqrt{2}\phi_1(t) - \phi_3(t) = 0.$$

Graphically, what we have is



Signal $d_4(t)$ being equal to 0 means that $s_4(t)$ can be obtained as a linear combination of the elements that are already in the basis. Hence, we don't need to add any new element to the latter in order to represent $s_4(t)$.

Since we don't have any more signals, the procedure concludes here, and the resulting basis is

$$\{\phi_1(t), \phi_2(t), \phi_3(t)\}.$$

When to stop

We stop here because we don't have any more signals to *process* (**not** because the last *difference* signal was 0). If we had been given yet another signal, $s_5(t)$, our job would not be done.

Order of the signals

If we *process* the signals in a different order, say, e.g., $s_4(t)$, $s_2(t)$, $s_1(t)$ and $s_3(t)$, we will get a different (equally valid) basis. Notice that, in such a case, signal $s_4(t)$ will go into the basis after normalization (while it was irrelevant in the above example).

2.4.2 Computing the coordinates of the signals with respect to the orthonormal basis

Given

- a collection of signals, and
- and an orthonormal basis for them,

in order to get a geometric representation of signals, we still need to find the coordinates of each signal with respect to the basis. We will see this **through an example** by computing the coordinates for the previous scenario.

The coordinates for a certain signal are the coefficients in the linear combination of elements in the basis that allow to recover the signal. In other words, the coordinates of $s_i(t)$ are the a_{ij}'s that satisfy

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\phi_j(t). \tag{2.5}$$

There are two possibilities for obtaining the coordinates.

Using the inner product

We already know one way for computing the coordinates: the inner product (projection),

$$a_{ij} = \langle s_i(t), \phi_j(t) \rangle.$$

Hence, for this particular case we have

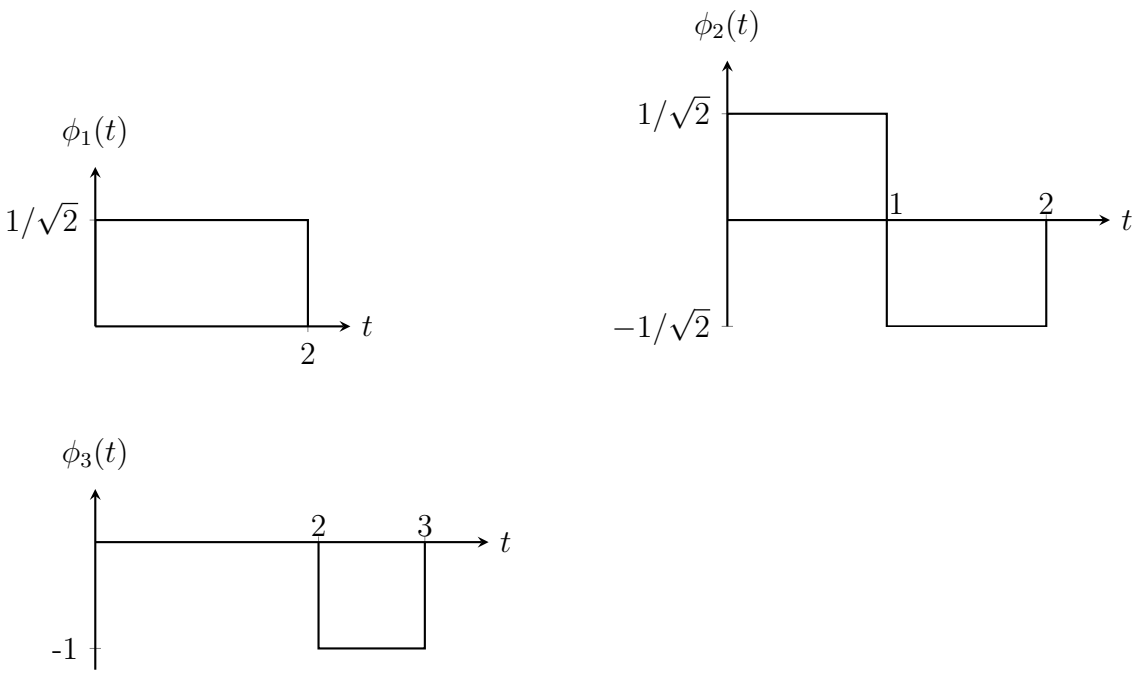
$$\begin{aligned}
\underline{a}_1 &\rightarrow \overbrace{\langle s_1(t), \phi_1(t)\rangle}^{a_{11}}, \overbrace{\langle s_1(t), \phi_2(t)\rangle}^{a_{12}}, \overbrace{\langle s_1(t), \phi_3(t)\rangle}^{a_{13}} \\
\underline{a}_2 &\rightarrow \langle \mathbf{s_2(t)}, \mathbf{\phi_1(t)}\rangle, \langle s_2(t), \phi_2(t)\rangle, \langle s_2(t), \phi_3(t)\rangle \\
\underline{a}_3 &\rightarrow \langle \mathbf{s_3(t)}, \mathbf{\phi_1(t)}\rangle, \langle \mathbf{s_3(t)}, \mathbf{\phi_2(t)}\rangle, \langle s_3(t), \phi_3(t)\rangle \\
\underline{a}_4 &\rightarrow \langle \mathbf{s_4(t)}, \mathbf{\phi_1(t)}\rangle, \langle \mathbf{s_4(t)}, \mathbf{\phi_2(t)}\rangle, \langle \mathbf{s_4(t)}, \mathbf{\phi_3(t)}\rangle.
\end{aligned}$$

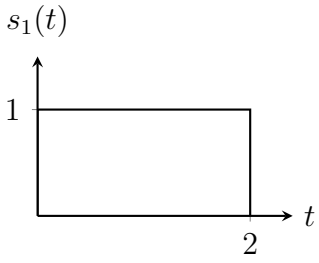
If we got the basis using Gram-Schmidt, we have **already computed** some of these.

By inspection

Equation (2.5) here becomes

$$s_i(t) = a_{i1}\phi_1(t) + a_{i2}\phi_2(t) + a_{i3}\phi_3(t)$$



- How do we build 

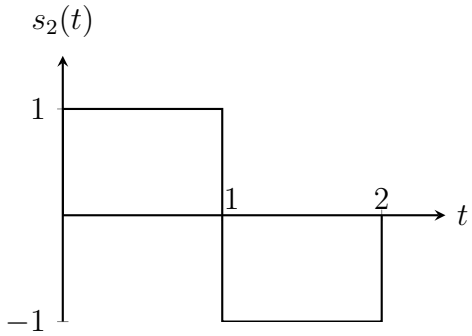
using the signals in the basis?

What should multiply the first element in the basis (i.e., what is the value of a_{11})? and the second (a_{12})? and the third (a_{13})?

We notice that this signal looks just like $\phi_1(t)$ but with a different amplitude. Hence, we just need to *scale* $\phi_1(t)$ multiplying it by $\sqrt{2}$ and we are done (we don't even need to use $\phi_2(t)$ or $\phi_3(t)$). Therefore, the coordinates for $s_1(t)$ would be

$$\underline{a}_1 = \begin{bmatrix} \sqrt{2} & 0 & 0 \end{bmatrix}$$

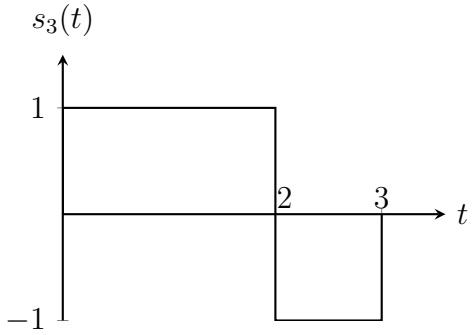
and there is no other possibility because the coefficients are unique!!

- How do we build 

using the signals in the basis?

In this case, $s_2(t)$ resembles $\phi_2(t)$. Moreover, $s_2(t)$ is $\phi_2(t)$ multiplied by $\sqrt{2}$, and hence

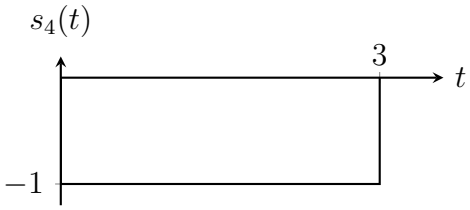
$$\underline{a}_2 = \begin{bmatrix} 0 & \sqrt{2} & 0 \end{bmatrix}.$$

- How do we build 

using the signals in the basis?

The first thing to notice here is that we are finally needing $\phi_3(t)$ because that's the only element in the basis that is non-zero between 2 and 3. On the other hand, $\phi_1(t)$ is only defined between 0 and 2, which suits us well here. The combination we need is

$$\underline{a}_3 = \begin{bmatrix} \sqrt{2} & 0 & 1 \end{bmatrix}.$$

- How do we build 

using the signals in the basis?

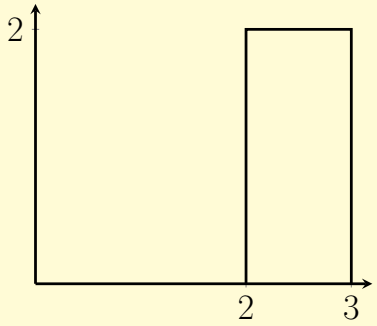
  $s_4(t)$ is easy to obtain by combining $\phi_1(t)$ and $\phi_3(t)$:

$$\underline{a}_4 = \begin{bmatrix} -\sqrt{2} & 0 & 1 \end{bmatrix}$$

  (notice the $-$ sign in the first coordinate: we are turning $\phi_1(t)$ upside down).

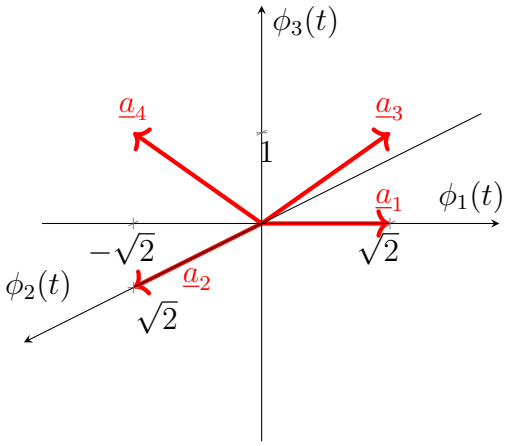
? Quick quiz

What are the coordinates of the signal



with respect to the above basis? [25]

Our signals now correspond with vectors and hence we can plot them in a **geometric representation**



Geometric representation of signals with respect to the orthonormal basis

A geometric representation is often very useful. Recall that the motivation for representing the signals in our modulator as elements in a Hilbert space was being able to compute *easily* how close or different two signals are...now we can!! The distance between two signals is simply the distance between the corresponding vectors.

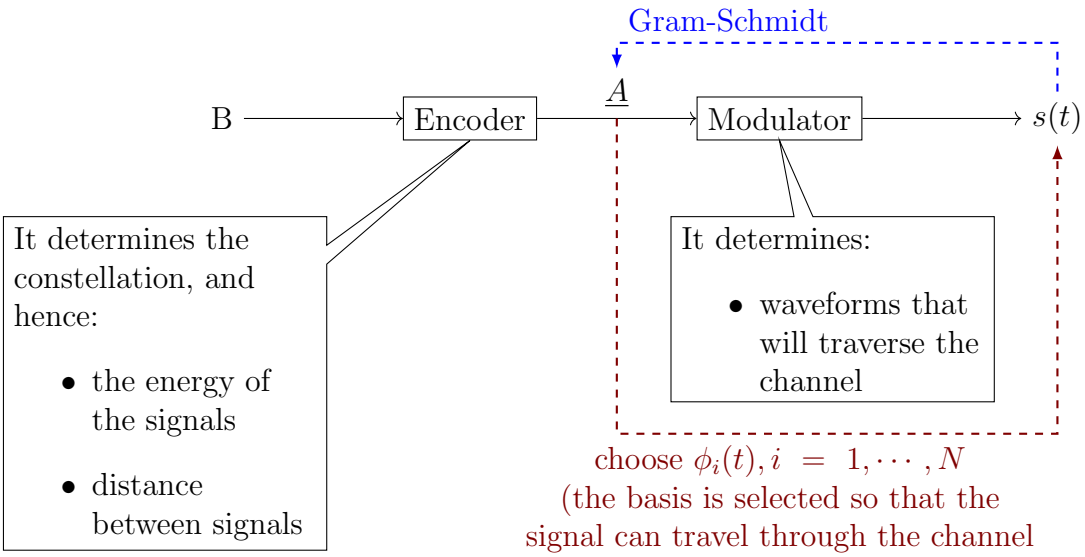
[25] They are $\begin{bmatrix} 0 & 0 & -2 \end{bmatrix}$

2.5 The transmitter

When designing the transmitter there are two possible starting points:

if we know/are given the signals that are going to be used for every information symbol, we can apply the Gram-Schmidt process to obtain an orthonormal basis, and then compute the coordinates of each signal with respect to that basis in order to get the constellation.

if we already have the constellation, in order to get signals in our modulator we just need to choose a basis, one that fits the characteristics of the channel.



The bottom line is:

- The *Encoder* determines the constellation, $\{\underline{a}_1, \underline{a}_2, \cdots, \underline{a}_M\}$, and we know:
  - The energy of the i-th element is
    $$\varepsilon_{s_i} = |\underline{a}_i|^2 .$$
    and
  - The distance between elements i and j is
    $$d(s_i(t), s_j(t)) = \sqrt{\sum_{m=1}^{N} |a_{im} - a_{jm}|^2}.$$

  Hence, the encoder determines the energy of the signals and the distance between them because, once we choose our constellation, they are already fixed **regardless of the orthonormal basis** given by the modulator (the energy/distance of the signals does **not** depend on the choice of the orthonormal basis: any one will yield the same energy/distance).

- The *Modulator* is only concerned with the nature of the channel (baseband vs. passband).

? Quick quiz

What is the connection between the orthonormal basis of the modulator and the energy of the signals? [26]

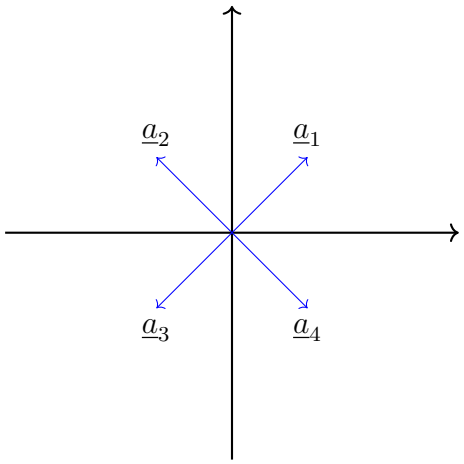
2.5.1 Energy of a constellation

Remember that earlier we said that the...

⭐ Peformance is completely determined by the constellation

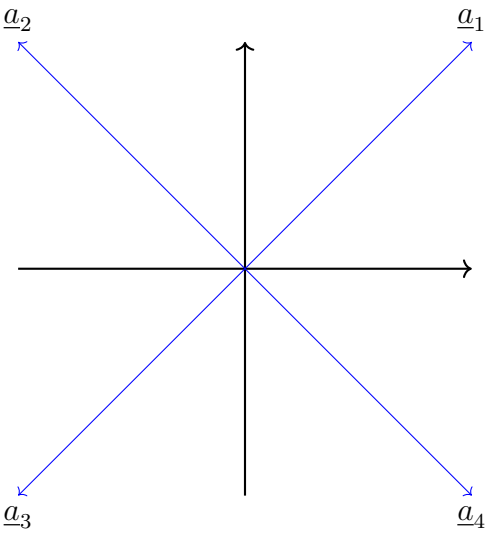
$$\uparrow \text{distance between vectors} \Longrightarrow \downarrow P_e$$

Hence, we can always improve the performance in our system (i.e., decrease the P_e) by increasing the distance between the elements in the constellation... and this can be easily accomplished by increasing the modulus of all the vectors (i.e., the energy of the signals because the squared modulus of a vector is the energy of the corresponding signal). For example, if we have this constellation



we can increase the distance between the elements in the constellation just by elongating the vectors

[26] None, the energy only depends on the constellation (i.e. on the encoder).



which is tantamount to spending more energy on each signal.

Therefore there is a clear trade-off between performance and energy: if you use more energy, you get a better performance. Right now we have a well-defined metric for the performance of a digital communications system: the probability of error, P_e (again, completely determined by the constellation). However, we don't have a metric for the energy of the system. We know how to compute the energy of any element in the constellation, but how do we compute the **energy of the constellation** *as a whole* (or, equivalently, of the set of signals $s_1(t), s_2(t), \cdots, s_M(t)$)? We use the **mean energy**, which is defined as

$$E_s = \sum_{i=1}^{M} P(s_i)\varepsilon_{s_i},$$

where

$$P(s_i) \equiv \text{the probability of transmitting } s_i(t),$$

and ε_{s_i} the energy of the latter. Intuitively, this is the expectation of the energy with respect to the random variable *signal transmitted*, s_i[27].

This parameter is going to very useful to decide between two different constellations that yield the same performance.

Mean energy is a *weighted* average

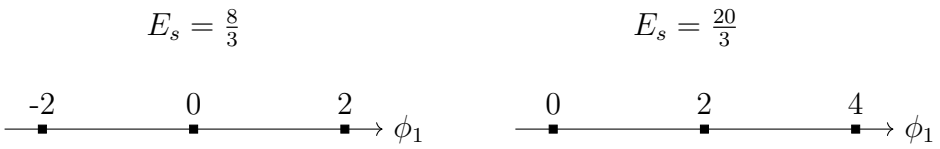
The mean energy is **not** the arithmetic mean of the energies of the different symbols, but a weighted average: every symbol is multiplied by the probability of it being transmitted. What happens most of the time (though not always) is that symbols are equally likely, in which case the weighted average becomes the arithmetic mean

$$E_s = \sum_{i=1}^{M} P(s_i)\varepsilon_{s_i} = \sum_{i=1}^{M} \frac{1}{M}\varepsilon_{s_i} = \frac{1}{M}\sum_{i=1}^{M} \varepsilon_{s_i}.$$

[27] s_i is a random variable and the energy is a function that depends on that random variable.

Centered constellation

Any constellation should be centered around 0. Otherwise, we are doing something wrong, since shifting the constellation apart from 0 increases the mean energy without having an impact on the performance!! Consider the constellations (of equally likely symbols)

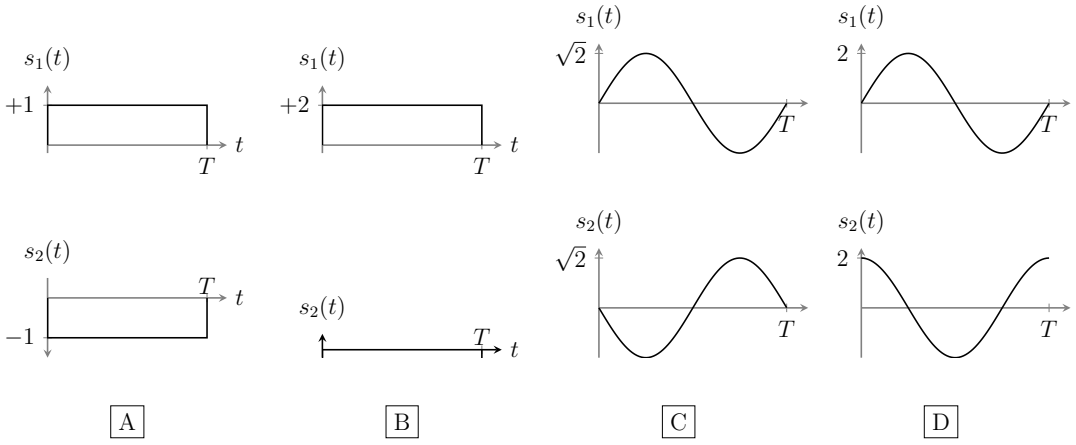


They exhibit the same performance since the distance between the elements of the constellation is the same. However, the mean energy on the right is higher. Hence, **the constellations should always be centered around** 0.

How do we center a constellation? We simply compute the mean and subtract it from every vector[28]. For instance, centering the constellation on the right entails subtracting from every element in the (1D) constellation (0, 2, and 4) the mean of all of them ($\frac{0+2+4}{3} = 2$).

2.5.2 Example: choosing the best set of signals for transmission

We are given four sets ($\boxed{A}$, $\boxed{B}$, $\boxed{C}$ and $\boxed{D}$) of two **equally likely** signals each, and we must choose the one giving the best performance using the least amount of energy.



In order to tackle this problem we will compute and plot the constellation for every set of signals. This entails first obtaining, in every case, an orthonormal basis.

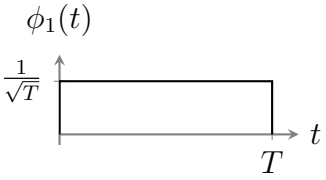
A The basis is obvious: one signal is the other multiplied by a constant, which means that a basis with just one signal in the modulator will allow us to build the two signals. The basis is given by $s_1(t)$...**after normalization**. In order to normalize the signal we compute its energy

$$\varepsilon_{s_1} = \int_{-\infty}^{\infty} |s_1(t)|^2 \, dt = T \times 1 = T$$

[28]This is always the case irregardless of the dimension, N, of the constellation.

Hence, the (only) element in the basis is

$$\phi_1(t) = \frac{s_1(t)}{\sqrt{\varepsilon_{s_1}}} = \frac{s_1(t)}{\sqrt{T}}.$$



What are the coordinates of signals $s_1(t)$ and $s_2(t)$ with respect to this basis?

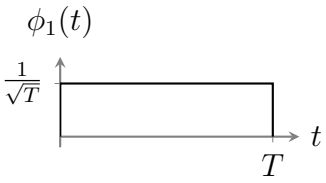
$$\underline{a}_1 = \sqrt{T}$$
$$\underline{a}_2 = -\sqrt{T}$$

(we plot the constellation just like before)

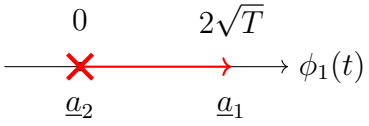
Once we have the constellation, it is very easy to compute the energy of every signal, and the distances:

- $\varepsilon_{s_1} = |\underline{a}_1|^2 = T = \varepsilon_{s_2} = E_s$ (mean energy)
- $d(s_1(t), s_2(t)) = d(\underline{a}_1, \underline{a}_2) = \sqrt{(a_1 - a_2)^2} = \sqrt{(\sqrt{T} - (-\sqrt{T}))^2} = \sqrt{(2\sqrt{T})^2} = 2\sqrt{T}$

B What is the basis now? The same as before!!



$$\underline{a}_1 = 2\sqrt{T}$$
$$\underline{a}_2 = 0$$



The energy of the signals is then computed using the square modulus of the vectors

$$\left.\begin{aligned}\varepsilon_{s_1} &= 4T \\ \varepsilon_{s_2} &= 0\end{aligned}\right\} \Rightarrow E_s = \frac{1}{2}(4T + 0) = 2T,$$

where we used that the symbols are equally likely.

Notice that the mean energy is twice that required by the previous set of signals. On the other hand, the distance

$$d(s_1(t), s_2(t)) = d(\underline{a}_1, \underline{a}_2) = 2\sqrt{T}$$

is the same as in $\boxed{A}$[29].

Therefore, if we had to choose between sets $\boxed{A}$ and $\boxed{B}$, we would pick set $\boxed{A}$ over set $\boxed{B}$ because they both have the same distance between signals, that is, the same peformance, but set $\boxed{A}$ takes up half the energy of set $\boxed{B}$.

C The two signals are opposite of each other, and hence a single element in the basis is again enough to represent both of them. That element is selected to be $s_1(t)$ after

[29] The problem with this constellation is, of course, that it is **not** centered.

normalization. Thus, we need to compute the energy of $s_1(t)$[30],

$$\varepsilon_{s_1} = \int_0^T \left|\sqrt{2}\sin\left(\frac{2\pi}{T}t\right)\right|^2 dt = \int_0^T 2\sin^2\left(\frac{2\pi}{T}t\right) dt$$

a trigonometric identity comes handy here, $1-\cos(2\alpha) = 2\sin^2(\alpha)$

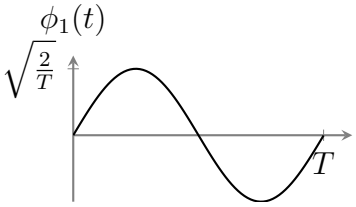
$$= \int_0^T 1dt - \cancelto{0}{\int_0^T \cos\left(2\cdot\frac{2\pi}{T}t\right) dt} = T,$$

where we have used that the period of the cosine is $T/2$ since

$$2\cdot\frac{2\pi}{T} = \frac{2\pi}{\frac{T}{2}},$$

(we multiply the frequency by 2) and hence we are integrating over an integer number of periods.

Therefore, we have



$$\underline{a}_1 = \sqrt{T}$$
$$\underline{a}_2 = -\sqrt{T}$$

We have exactly the same constellation what we had for set $\boxed{A}$. There is no need to compute again the (mean) energy of the constellation nor the distance between signals because they only depend on the constellation, and hence they are the same,

$$E_s = T$$
$$d(s_1(t), s_2(t)) = 2\sqrt{T}$$

D In this case the basis is not obvious, and therefore we should apply the Gram-Schmidt process:

- $\phi_1(t) = \frac{s_1(t)}{\sqrt{\varepsilon_{s_1}}} = \frac{2\sin\left(\frac{2\pi}{T}t\right)}{\sqrt{2T}} = \sqrt{\frac{2}{T}}\sin\left(\frac{2\pi}{T}t\right)$

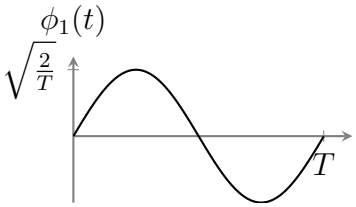
$$\varepsilon_{s_1} = \int_0^T 4\sin^2\left(\frac{2\pi}{T}t\right) dt = 2\underbrace{\int_0^T 2\sin^2\left(\frac{2\pi}{T}t\right) dt}_{\varepsilon_{s_1}\text{ in }\boxed{C}} = 2T,$$

since the signal $s_1(t)$ is the equal to the homonymous in the previous set, $\boxed{C}$, but multiplied by $\sqrt{2}$ and

$$\varepsilon_{ks} = \int_{-\infty}^{\infty} |ks(t)|^2 dt = k^2 \int_{-\infty}^{\infty} |s(t)|^2 dt = k^2\varepsilon_s,$$

with k being an arbitrary constant.

[30]For that, we need the analytical expression of the sine in the picture, which amounts to accounting for its period, T, and amplitude $\sqrt{2}$.



- $\phi_2(t) = \frac{d_2(t)}{\sqrt{\varepsilon_{d_2}}}$ where

$$d_2(t) = s_2(t) - a_{21}\phi_1(t)$$

with

$$a_{21} = \langle s_2(t), \phi_1(t)\rangle = \int_0^T 2\cos\left(\frac{2\pi}{T}t\right)\sqrt{\frac{2}{T}}\sin\left(\frac{2\pi}{T}t\right)$$

using the trigonometric indentity $\sin\alpha\cos\beta = \frac{\sin(\alpha+\beta)+\sin(\alpha-\beta)}{2}$

$$= \int_0^T \cancel{2}\sqrt{\frac{2}{T}}\frac{\sin\left(\frac{2\cdot 2\pi}{T}t\right) - \cancelto{0}{\sin(0)}}{\cancel{2}}dt = \sqrt{\frac{2}{T}}\int_0^T \sin\left(\frac{2\cdot 2\pi}{T}t\right)dt = 0.$$

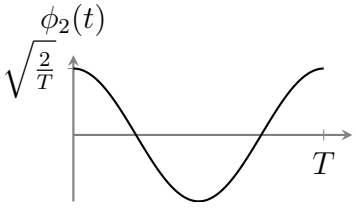
where the latter equality is due to the fact that we are integrating a sine over an integer number of periods.
This means sine and cosine are orthogonal!! Then $d_2(t) = s_2(t)$ and

$$\phi_2(t) = \frac{s_2(t)}{\sqrt{\varepsilon_{s_2}}} = \frac{2\cos\left(\frac{2\pi}{T}t\right)}{\sqrt{2T}} = \sqrt{\frac{2}{T}}\cos\left(\frac{2\pi}{T}t\right)$$

The energy of $s_2(t)$ is given by

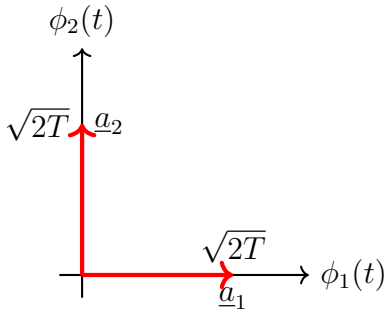
$$\varepsilon_{s_2} = \int_0^T |s_2(t)|^2\,dt = \int_0^T 4\cos^2\left(\frac{2\pi}{T}t\right)dt \overset{\text{exercise}}{=} 2T.$$

The second element in the orthonormal basis is



and the constellation

$$\underline{a}_1 = \left(\sqrt{2T}, 0\right)$$
$$\underline{a}_2 = \left(0, \sqrt{2T}\right)$$



Notice this is **not** a centered constellation. If you want to center it, you need

to subtract the mean,

$$\bar{\underline{a}} = \frac{\underline{a}_1 + \underline{a}_2}{2} = \left(\frac{\sqrt{2T}}{2}, \frac{\sqrt{2T}}{2} \right),$$

from every element.

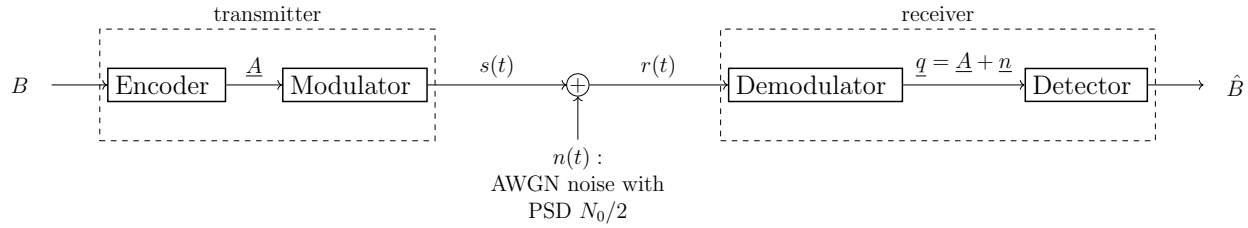
The energy (already computed), and distance are

$$\varepsilon_{s_1} = \varepsilon_{s_2} = E_s = 2T$$
$$d(s_1(t), s_2(t)) = |\underline{a}_1 - \underline{a}_2| = \sqrt{(\sqrt{2T} - 0)^2 + (0 - \sqrt{2T})^2} = \sqrt{4T} = 2\sqrt{T}.$$

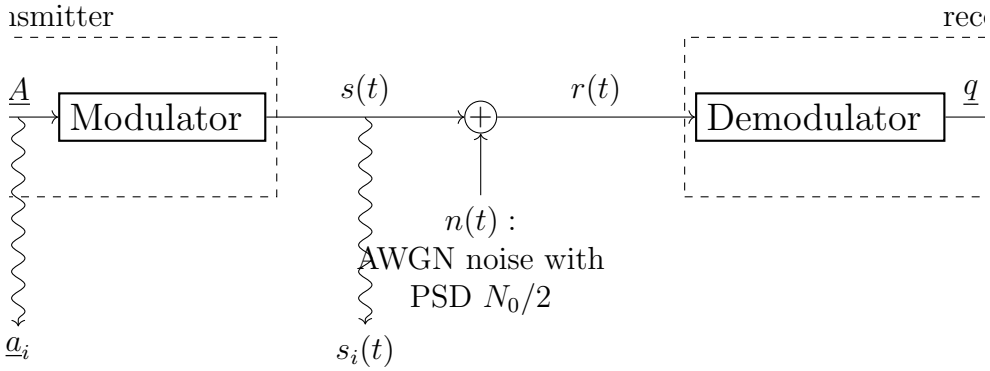
The distance between elements of the constellation is the same in every set (same performance), but the energy consumption changes. We would also have to take into account the nature of the channel: if the channel is baseband we cannot use sines or cosines.

2.6 Demodulator

So far, we have been talking about the transmitter. The demodulator is at the other end of the channel. In other words, it is already part of the receiver.



Right now, we forget about the encoder and detector, and focus only on the modulation/demodulation operations



We can annotate the picture to better indicate what's going on at every stage:

- $\underline{A}$ will be one of the elements in the constellation, say,

$$\underline{a}_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{iN} \end{bmatrix}$$

- which the modulator will turn into a signal, $s_i(t)$.

  We know that the components of vector $\underline{a}_i$ are actually the coordinates of signal $s_i(t)$ with respect to the orthonormal basis in the modulator, so that $s_i(t)$ can be expressed as a linear combination of the elements in the orthonormal basis,

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\phi_j(t)$$

- Thermal noise, $n(t)$, is added to the transmitted signal, $s_i(t)$, when the latter goes through the AWGN channel, and that results in the

- received signal, $r(t)$, which makes up the input to the demodulator.

Random variable v. realization

Notice that

- $\underline{A}$ is a random vector whereas $\underline{a}_i$ is a particular realization (value), and
- $s(t)$ is a random process whereas $s_i(t)$ is a deterministic signal.

The transmitter knows which $\underline{a}_i$ or, equivalently, which $s_i(t)$ was transmitted, but **from the standpoint of the receiver** we have

$$\underset{\text{r.v.}}{B} \quad \rightarrow \quad \underset{\text{random vector}}{\underline{A}} \quad \rightarrow \quad \underset{\text{random process}}{s(t)}$$

We don't use any subindex when we refer to things that are random. On the contrary, when we add a subindex we refer to a **realization** of the corresponding random variable/vector/process.

Looking at the picture, the modulator turns an element from the constellation, $\underline{a}_i$, which is a vector encompassing a set of coordinates, into a signal $s_i(t)$. **The demodulator** does the opposite: it **turns** a signal, **the received signal**, $r(t)$, into a set of coordinates that are collected in **vector** $\underline{q}$. Notice the received signal, $r(t)$, is contaminated with noise, and hence is **not** any of the signals $s_i(t)$.

So, how does the demodulator obtain coordinates from the received signal, $r(t)$? It computes its projection on the Hilbert space spanned by the orthonormal basis $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$. Therefore, vector $\underline{q}$ is the projection of the received signal $r(t)$ on the orthonormal basis $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$, with the j-th component in the vector, q_j,

being the projection on the axis $\phi_j(t)$. And how do we compute projections? By means of the inner product. Let us investigate the structure of an individual component, q_j. We have

$$q_j = \langle r(t), \phi_j(t) \rangle = \int_0^T r(t)\phi_j^*(t)dt$$

we keep the uncertainty about the signal transmitted: we use the stochastic process $s(t)$ rather than a realization thereof (the receiver doesn't know which $s_i(t)$ was transmitted)

$$= \int_0^T (s(t) + n(t))\phi_j^*(t)dt = \int_0^T s(t)\phi_j^*(t)dt + \underbrace{\int_0^T n(t)\phi_j^*(t)dt}_{\text{r.v. } n_j}$$

$$= \int_0^T s(t)\phi_j^*(t)dt + n_j$$

Signal $s(t)$ is one of the elements in the set $\{s_1(t), s_2(t), \cdots, s_M(t)\}$, but we don't know which one. That's why $s(t)$ is a random process. But, what are the coordinates of signal $s(t)$ with respect to the elements in the orthonormal basis? They are the components of vector $\underline{A}$, which can be written as

$$\underline{A} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix}.$$

Of course, $\underline{A}$ is a random vector that takes values in the set $\{\underline{a}_1, \underline{a}_2, \cdots, \underline{a}_M\}$ (recall the picture, it must be one of the elements in the constellation). Every element in a random vector is a random variable, and hence every A_j is a random variable.

From the coordinates of the corresponding element in the constellation and the orthonormal basis we can express $s(t)$ as

$$s(t) = \sum_{l=1}^{N} A_l \phi_l(t),$$

and plugging this into q_j we have

$$q_j = \int_0^T s(t)\phi_j^*(t)dt + n_j = \int_0^T \sum_{l=1}^{N} A_l \phi_l(t)\phi_j^*(t)dt + n_j = \sum_{l=1}^{N} A_l \underbrace{\int_0^T \phi_l(t)\phi_j^*(t)dt}_{\langle \phi_l(t), \phi_j(t) \rangle} + n_j$$

$$= \sum_{l=1}^{N} A_l \langle \phi_l(t), \phi_j(t) \rangle + n_j = A_j + n_j,$$

where, again, we have used

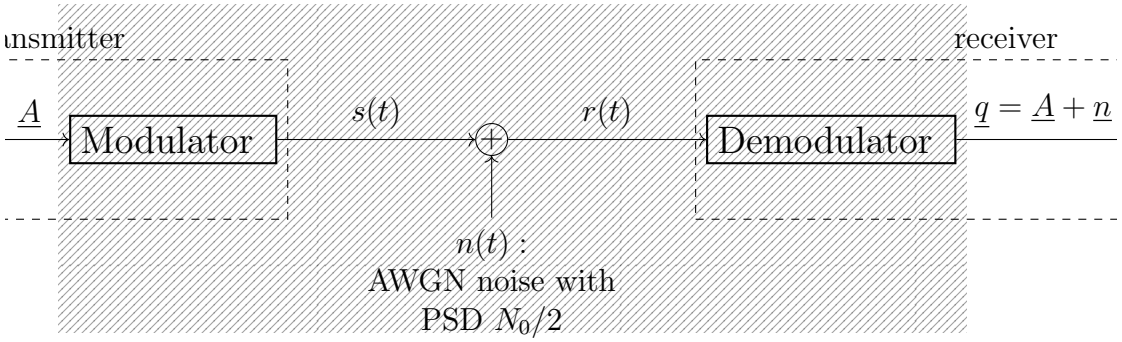
$$\langle \phi_l(t), \phi_m(t) \rangle = \begin{cases} 0, & l \neq m \\ 1, & l = m \end{cases}.$$

We can do the same for every q_j, with $j = 1, \cdots, N$, to get

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} = \begin{bmatrix} A_1 + n_1 \\ A_2 + n_2 \\ \vdots \\ A_N + n_N \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{bmatrix} = \underline{A} + \underline{n}.$$

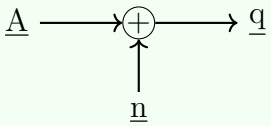
That is, $\underline{q}$ is the element of the constellation transmitted, $\underline{A}$, plus a vector of noise, $\underline{n}$. In other words, $\underline{q}$ **is a *noisy* estimate of** $\underline{A}$. Ideally, in order to minimize the number of errors, the distance between $\underline{q}$ and $\underline{A}$ should be as small as possible or, equivalently, the noise vector should be close to zero.

In summary,



Sometimes, we can *forget* about the modulation/demodulation operations and work with a model that is a simplification of the real thing

Definition 2.6.1: Equivalent discrete-time channel



Equivalent discrete-time channel

We transmit a vector, $\underline{A}$, which gets mixed with some noise, $\underline{n}$, to give vector $\underline{q}$, which is what, ultimately, the detector gets to observe. The nice thing about this model is that it allows us to work in a discrete-time world.

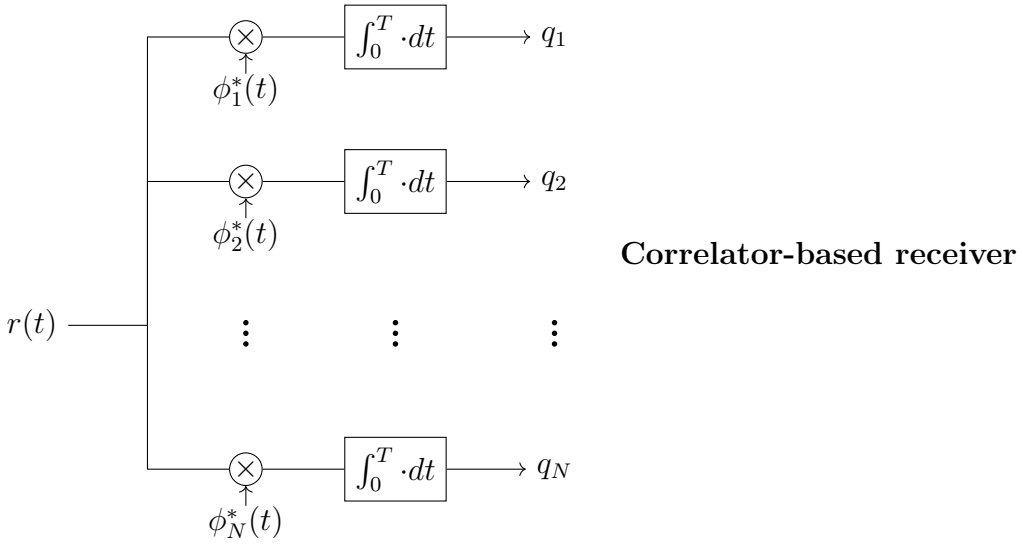
? Quick quiz

What component or components in our digital communications system are affected if we change the orthonormal basis used to build the signals transmitted? [31]

[31] We are affecting both the modulator and the demodulator. The basis is used by the former to assemble the signals transmitted, and by the latter to project the received signal into the Hilbert space of signals encompassing the constellation.

2.6.1 Block diagram of the demodulator

We now know how to compute every coordinate in vector $\underline{q}$ at the output of the demodulator. This is illustrated by the block diagram[32]



This is known as the **correlator-based receiver**. In order to understand the reason behind the name, let us review the concept of

Definition 2.6.2: Cross-correlation between two signals...

...$f(t)$ and $g(t)$ is

$$R_{fg}(\tau) = \int_{-\infty}^{\infty} f(t)g^*(t+\tau)dt$$

Notice that the expression on the right-hand side kind of resembles the inner product. Let us now bring back the expression for computing the j-th component in vector $\underline{q}$:

$$q_j = \langle r(t), \phi_j(t) \rangle = \int_{-\infty}^{\infty} r(t)\phi_j^*(t)dt = \int_{-\infty}^{\infty} r(t)\phi_j^*(t+0)dt = R_{r\phi_j}(0),$$

i.e., q_j is the cross-correlation between $r(t)$ and $\phi_j^*(t)$ evaluated at $\tau = 0$. Therefore, the projection of signal $r(t)$ on the basis $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$ can be interpreted as a cross-correlation, evaluated at $\tau = 0$, between the received signal and each element in the basis. Notice that

- we have as many branches as coordinates, and
- we need the elements in the orthonormal basis, $\phi_i(t), i = 1, \cdots, N$.

[32]Here we are assuming that the signals of the modulator, $s_1(t), s_2(t), \cdots, s_M(t)$, are non-zero only in the interval $[0, T]$.

2.7 Statistical characterization of vector q

Earlier, we talked about the **equivalent discrete-time channel**

$$\underline{q} = \underline{A} + \underline{n}$$

which is a simplified model for the transmission in a digital communications system. According to this model, the observations vector, $\underline{q}$, is the sum of two random vectors, and hence it is itself a random vector. As such, it has a probability density function

$$f_{\underline{q}} = f_{q_1, q_2, \cdots, q_N}$$

(the pdf of a random vector is simply the joint pdf of its components).

Let us study the marginal pdf of any arbitrary random variable within vector $\underline{q}$,

$$q_j = A_j + n_j,$$

which is the sum of two random variables:

A_j, a **discrete** random variable taking values in the set $\{a_{1j}, a_{2j}, \cdots, a_{Mj}\}$, and

n_j, a **continuous** random variable that is obtained by integrating the Gaussian process $n(t)$ (defined in Section 2.6),

$$n_j = \int_0^T n(t)\phi_j^*(t)dt.$$

2.7.1 Noise, n_j

We first focus on the noise term, n_j. Since it's Gaussian, it is perfectly characterized by its mean and variance.

Probability density function of a Gaussian random variable

$$X \sim \mathcal{N}\left(\mu, \sigma^2\right) \Rightarrow f_X(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Let us compute those. The mean is

$$\mathbb{E}\left[n_j\right] = \mathbb{E}\left[\int_0^T n(t)\phi_j^*(t)dt\right]$$

expectation of the integral is the integral of the expectation

$$= \int_0^T \mathbb{E}\left[n(t)\phi_j^*(t)\right] dt$$

$\phi_j(t)$ is known, and hence deterministic

$$= \int_0^T \mathbb{E}\left[n(t)\right] \phi_j^*(t)dt$$

thermal noise has zero mean

$$= \int_0^T 0 \ \phi_j^*(t)dt = 0$$

Now we compute the covariance between any two noise components, n_j and n_k as

$$\sigma^2_{n_j n_k} = \mathbb{E}\left[(n_j - 0)(n_k - 0)^*\right] = \mathbb{E}\left[n_j n_k^*\right].$$

Notice that when $j = k$, the above expression amounts to the (individual) variance of the j-th noise component. That is,

$$\mathbb{E}\left[n_j n_k^*\right] = \begin{cases} \text{covariance between variables } n_j \text{ and } n_k, & j \neq k \\ \text{variance of variable } n_j, & j = k \end{cases},$$

and hence when computing the above covariance for arbitrary j and k, we get *for free* the variance of any noise variable.

$$\mathbb{E}\left[n_j n_k^*\right] = \mathbb{E}\left[\int_0^T n(t)\phi_j^*(t)dt \int_0^T n^*(u)\phi_k(u)du\right]$$

only $n(t)$ and $n(u)$ are random

$$= \int_0^T \int_0^T \mathbb{E}\left[n(t)n^*(u)\right] \phi_j^*(t)\phi_k(u)dtdu$$

what we have here is the autocorrelation function evaluated at $t-u$, $\mathbb{E}\left[n(t)n^*(u)\right] = R_n(t-u)$; on the other hand, we know that $n(t)$ is white[33](Additive **White** Gaussian Noise) and, moreover, we know its autocorrelation function $R_n(\tau) = \frac{N_0}{2}\delta(\tau)$, hence $\mathbb{E}\left[n(t)n^*(u)\right] = \frac{N_0}{2}\delta(t-u)$

$$= \frac{N_0}{2} \int_0^T \int_0^T \delta(t-u)\phi_j^*(t)\phi_k(u)dtdu$$

$\phi_k(u)$ does not depend on t, and hence can be pulled out of the *inner* integral

$$= \frac{N_0}{2} \int_0^T \phi_k(u) \left[\int_0^T \delta(t-u)\phi_j^*(t)dt\right] du$$

$\delta(t-u)\phi_j^*(t)$ is a delta at time u with *amplitude* $\phi_j^*(u)$, hence when you integrate it you get the *amplitude* $\phi_j^*(u)$ (notice that u is between 0 and T, and thus a delta at u is *captured* by an integral from 0 to T)

$$= \frac{N_0}{2} \int_0^T \phi_k(u)\phi_j^*(u)du = \frac{N_0}{2}\langle \phi_k(u), \phi_j(u)\rangle = \begin{cases} \frac{N_0}{2}, & j = k \\ 0, & \text{otherwise} \end{cases}.$$

In summary, what we have is

[33]White process implies WSS.

- $\sigma^2_{n_j} = \mathbb{E}\left[n_j^2\right] = \frac{N_0}{2}$, i.e., the variance of any noise variable is $\frac{N_0}{2}$,
- $\left.\begin{array}{r} \sigma^2_{n_j n_k} = 0, j \neq k \;\Rightarrow\; \text{the noise variables are } \textbf{uncorrelated} \\ \text{the noise variables are } \textbf{Gaussian} \end{array}\right\} \Rightarrow$ the noise variables, n_j with $j = 1 \cdots N$, are **independent**!!

This means, we know the...

⭐...distribution of the noise vector

$$\underline{n} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{N_0}{2} & 0 & \cdots & 0 \\ 0 & \frac{N_0}{2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{N_0}{2} \end{bmatrix} \right) = \mathcal{N}\left(\underline{0}, \frac{N_0}{2}\underline{\underline{I}}\right),$$

and hence we have an expression for its pdf[34].

$$f_{\underline{n}}(\underline{n}) = f_{n_1,n_2,\cdots,n_N}(n_1, n_2, \cdots, n_N) = \prod_{j=1}^{N} f_{n_j}(n_j) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi \frac{N_0}{2}}} e^{-\frac{(n_j-0)^2}{2\frac{N_0}{2}}}.$$

2.7.2 Observation q_j

Now we know the distribution of the noise, and taking advantage of it we aim at characterizing the pdf of the observation $q_j = A_j + n_j$. This is hard to do since we don't know neither A_j nor n_j (both are random variables, the former discrete and the latter continuous). However, it would be a completely different story if we knew A_j. Let us focus on computing

$$f_{q_j|A_j=a_{ij}} \equiv \begin{array}{l} \text{the pdf of } j\text{-th observation, } q_j, \text{ conditional on the } j\text{-th component of} \\ \text{the transmitted vector being that corresponding to the } i\text{-th element} \\ \text{in the constellation, } \underline{a}_i, \end{array}$$

which will be enough for our purposes. Notice that assuming $A_j = a_{ij}$ is tantamount to assuming $\underline{a}_i$ was the element from the constellation transmitted.

How can we compute the above pdf? What is the distribution of $q_j = A_j + n_j$ when $A_j = a_{ij}$? In such case we have

$$q_j = \underbrace{a_{ij}}_{\substack{\text{known} \\ \text{constant...we} \\ \text{know the} \\ \text{constellation!!}}} + \underbrace{n_j}_{\substack{\text{Gaussian r.v.} \\ \text{with zero mean} \\ \text{and variance} \\ \sigma_n^2 = \frac{N_0}{2}}}.$$

Now, n_j is a Gaussian random variable with zero mean and variance $N_0/2$, and we are simply adding a *known* constant to it. That amounts to modifying the mean of the r.v.

[34]Notice that here we are slightly abusing notation in denoting the same way a random variable and its realization. It's very common.

Therefore $q_j|A_j = a_{ij}$ is a random variable with the same distribution as n_j but mean $0 + a_{ij}$, i.e.,

$$q_j|A_j = a_{ij} \sim \mathcal{N}\left(a_{ij}, \frac{N_0}{2}\right) \Rightarrow f_{q_j|A_j=a_{ij}} = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{\left(q_j - a_{ij}\right)^2}{N_0}}.$$

So far, we have characterized the pdf of a single component, q_j, in the vector of observations, but we still need to characterize the conditional pdf of the whole vector,

$$f_{\underline{q}|\underline{A}}(\underline{q} \mid \underline{a}_i),$$

i.e., the pdf of vector $\underline{q}$ conditional on the element of the constellation transmitted being $\underline{a}_i$. We can reason in an analogous way: if $\underline{n}$ is a Gaussian random vector distributed according to $\underline{n} \sim \mathcal{N}\left(\underline{0}, \frac{N_0}{2}\underline{\underline{I}}\right)$, then adding to it a known (deterministic) vector $\underline{a}_i$ to get

$$\underline{q} = \underline{a}_i + \underline{n}$$

just *shifts* the distribution[35], i.e., changes its mean from 0 to $0 + \underline{a}_i = \underline{a}_i$. Then, we know the...

⭐...*conditional* distribution of the observations vector

$$\underline{q} \mid \underline{A} = \underline{a}_i \sim \mathcal{N}\left(\underline{a}_i, \frac{N_0}{2}\underline{\underline{I}}_N\right).$$

Notice that the individual random variables inside vector $\underline{q}$ are independent because they are uncorrelated (every covariance is 0) and Gaussian. We can exploit this[36] to write the mathematical expression for the pdf of $\underline{q}$,

$$\begin{aligned} f_{\underline{q}|\underline{A}}\left(\underline{q} \mid \underline{a}_i\right) &= f_{q_1,q_2,\cdots,q_N|A_1,A_2,\cdots,A_N}\left(q_1, q_2, \cdots, q_N \mid a_{i1}, a_{i2}, \cdots, a_{iN}\right) \\ &\overset{\text{indep.}}{=} \prod_{j=1}^{N} f_{q_j|A_j}\left(q_j \mid a_{ij}\right) = \prod_{j=1}^{N} \frac{1}{\left(\pi N_0\right)^{1/2}} e^{-\frac{\left(q_j - a_{ij}\right)^2}{N_0}} \\ &= \frac{1}{\left(\pi N_0\right)^{N/2}} e^{-\frac{\sum_{j=1}^{N}\left(q_j - a_{ij}\right)^2}{N_0}} \end{aligned}$$

what we have in the numerator of the exponential is the squared Euclidean distance between $\underline{q}$ and $\underline{a}_i$ squared

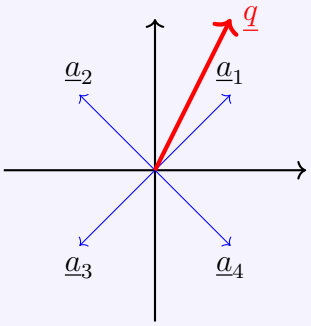
$$= \frac{1}{\left(\pi N_0\right)^{N/2}} e^{-\frac{d^2(\underline{q}, \underline{a}_i)}{N_0}} \tag{2.6}$$

At the sight of this result, the smaller the distance between vectors $\underline{q}$ and $\underline{a}_i$ (the more similar they are), the higher the value of this pdf, which intuitively makes sense: if we condition on a symbol $\underline{a}_i$ which is close to $\underline{q}$, then the density is large.

[35] True for Gaussian random variables/vectors, but not for *every* distribution.

[36] The joint pdf of a collection of random variables is the product of the marginal pdf's.

Elements closer to $\underline{q}$ yield a larger value for the pdf

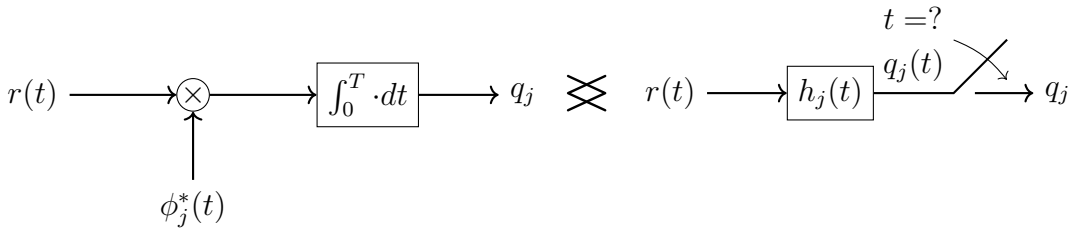


$$d(\underline{a}_1, \underline{q}) < d(\underline{a}_2, \underline{q}) \Rightarrow f_{\underline{q}|\underline{A}}\left(\underline{q} \mid \underline{A} = \underline{a}_1\right) > f_{\underline{q}|\underline{A}}\left(\underline{q} \mid \underline{A} = \underline{a}_2\right)$$

2.8 Matched filter-based demodulator (matched receiver)

Earlier, we saw how to implement the demodulator by means of correlators, and now we are going to see an equivalent implementation using filters.

Specifically, we want q_j to be a sample at the output of an LTI system (notice that at the output of the filter we get a signal rather than a number!!), i.e., the "old" construction using a correlator is replaced with a filter and a sampler,



$$q_j = \int_{-\infty}^{\infty} r(u)\phi_j^*(u)du$$

$$q_j = q_j(t)\Big|_{t=?} = r(t) * h_j(t)\Big|_{t=?} = \int_{-\infty}^{\infty} r(u)h_j(t-u)du\Big|_{t=?}$$

In order for the two expressions to yield the same result we need to choose

- $h_j(t)$, and
- the sampling time.

If we sample at time $t = 0$, then the expression using a linear filter becomes very close to that using a correlator,

$$q_j = q_j(t)\Big|_{t=0} = \int_{-\infty}^{\infty} r(u)h_j(-u)du$$

and we just need to choose $h_j(t)$...how do we do it? It is straightforward: we choose $h_j(-u) = \phi_j^*(u) \Rightarrow h_j(u) = \phi_j^*(-u)$

$$= \int_{-\infty}^{\infty} r(u)\phi_j^*(u)du.$$

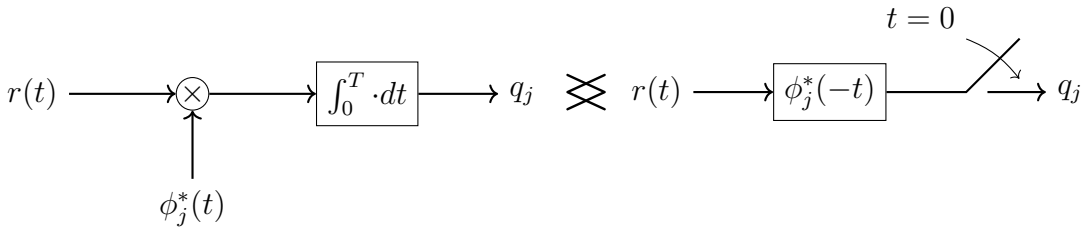
We say that $h_j(t)$ is a filter *matched* to the signal $\phi_j(t)$ and, in general,

Definition 2.8.1: Matched filter

A filter with impulse response $h(t) = x^*(-t)$ is matched to the signal $x(t)$.

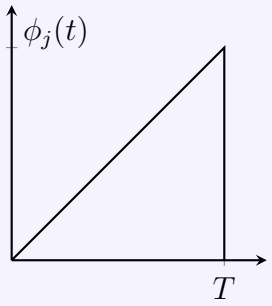
A matched filter can be though of as a *signal detector* (for the corresponding signal it is matched to).

In summary, what we have is

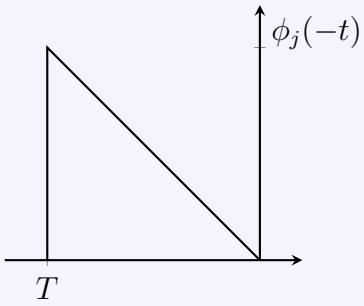


There is a catch...

One "problem" with the implementation of this filter is that if $\phi_j(t)$ is a signal between 0 and T (this is the case for us), e.g.,



then

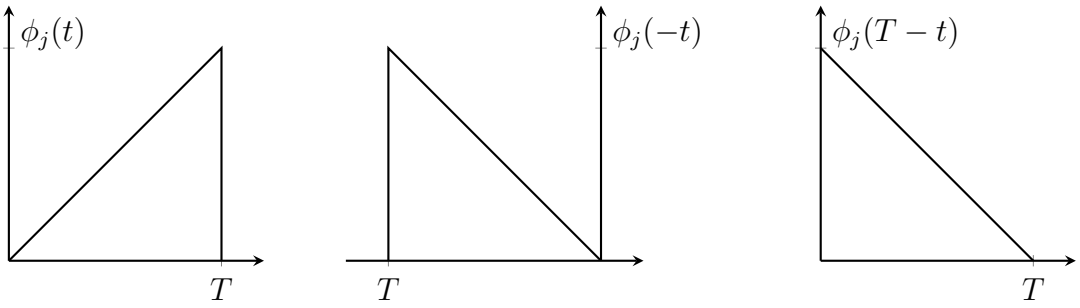


is not a **causal** filter[a] (it is anti-causal), which implies that the output of the filter at a

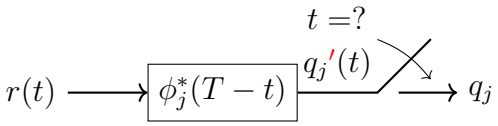
certain time instant depends on future inputs. In other words, in order to compute the output of the filter at time t, you need the inputs from time t up to $t+T$.

[a]For an LTI system to be causal the impulse response must be zero on the negative part of the time axis.

How can we fix this? We can simply add some delay to the filter in order to make it a causal filter[37].



Then, what we have now is



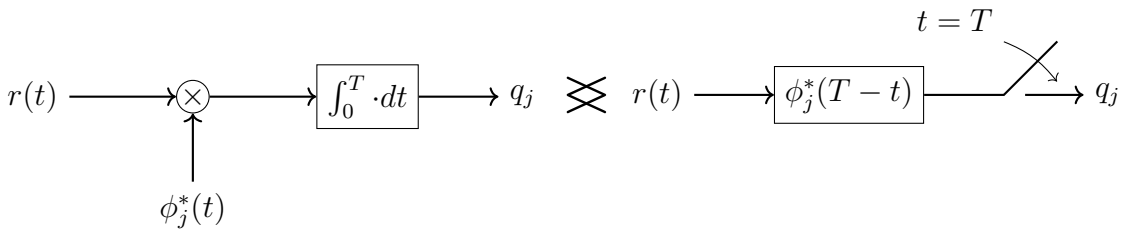
and the output of the filter is not the same as before[38],

$$q_j'(t) = r(t) * h_j'(t) = \int_{-\infty}^{\infty} r(u)\phi_j^*(T-(t-u))du = \int_{-\infty}^{\infty} r(u)\phi_j^*(T-t+u)du,$$

and if we want to get the same result as before we need to evaluate the convolution at time T,

$$q_j'(T) = \int_{-\infty}^{\infty} r(u)\phi_j^*(\cancel{T}-\cancel{T}+u)du = \int_{-\infty}^{\infty} r(u)\phi_j^*(u)du,$$

which is exactly what we were looking for: the result given by the correlator. We had this coming: what happens to the output of an LTI system when we delay the impulse response? The output gets equally delayed[39]. So, if earlier we were sampling at 0, now we should be sampling at T (because that's the delay we applied to the filter), and we have that these two implementations are equivalent



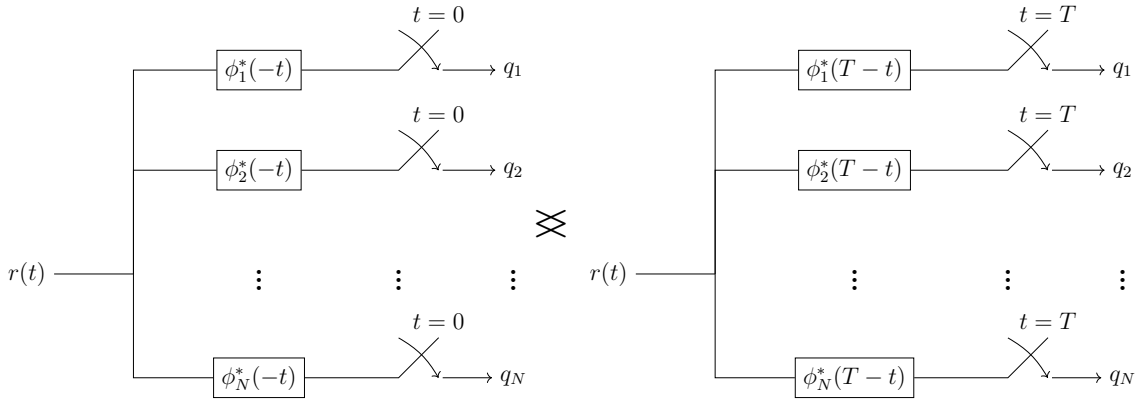
[37]We obtain a causal filter from an anti-causal filter by *shifting* the filter to the right.

[38]That's why we write the *prime*.

[39]In an LTI system, a shift at the input, or impulse response, yields the same shift at the output.

Notice that, in this last implementation, in order to get the output of the filter at time t, we need to wait until time $t+T$ (it's not a problem: T is usually very short).

In summary, we have two additional implementations for the demodulator



Matched filters-based receiver ***Causal*** **matched filters-based receiver**

Both are equivalent to each other, and also to the correlators-based one we saw earlier.

An important property of the matched filter: the matched filter maximizes the SNR. This means the SNR at every q_j is maximum (notice that $q_j = A_j + n_j$ has a signal component, A_j, and a noise component, n_j).

Recall that

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\phi_j(t) = a_{i1}\phi_1(t) + a_{i2}\phi_2(t) + \cdots a_{iN}\phi_N(t).$$

The first branch of the demodulator aims at detecting the coefficient that multiplies $\phi_1(t)$, the second branch that which multiplies $\phi_2(t)$, and so forth and so on. In the next section we prove that if you want to detect a certain signal (whatever that is) *contaminated with noise*, a matched filter will yield maximum SNR.

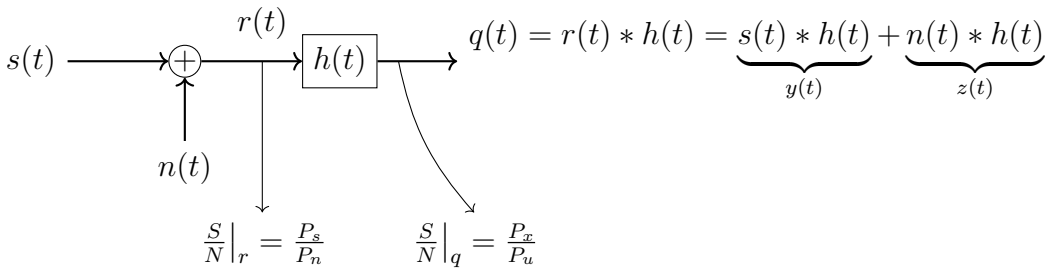
2.8.1 Properites of the matched filter: Maximum Signal-to-noise ratio

Implementing the demodulator using matched filters instead of correlators doesn't provide any benefit in terms of operation (they both are equivalent and do the same thing!!). However, thinking about the demodulator in terms of filtering allows to easily derive some properties...and the most important property is that our modulator maximizes the signal-to-noise ratio (SNR)[40]. Recall that

$$\uparrow SNR \Rightarrow \uparrow \text{performance of the system.}$$

We have a scheme like this

[40]Remember: it's the ratio between the power of the signal and that of the noise. It's a metric of the quality of a system, and we'd like the SNR to be as high as possible.



in which the signal of interest, $s(t)$, is known.

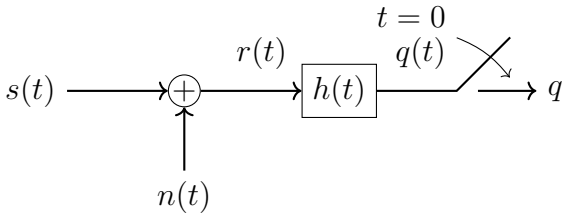
At the output of the filter, the signal of interest is $x(t)$ (the filtered version of the signal of interest at the input).

Ideally,

$$\left.\frac{S}{N}\right|_q \geq \left.\frac{S}{N}\right|_r,$$

that is, the SNR at the output of the filter is higher than the one we had at the input $\Rightarrow$ the filter alleviates the noise, it doesn't add more. Now, we'll prove that if $h(t)$ is adapted to $s(t)$ (the signal of interest!!), then the SNR of q is maximum.

We start from the scheme we saw before, but we sample the output at $t = 0$ (the same way we did in the implementation of the demodulator). We only care about the SNR at the sampling instant.



The output of the filter is

$$q(t) = s(t) * h(t) + n(t) * h(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau + \int_{-\infty}^{\infty} n(\tau)h(t-\tau)d\tau,$$

and sampling at time $t = 0$ we get

$$q(0) = \underbrace{\int_{-\infty}^{\infty} s(\tau)h(-\tau)d\tau}_{y} + \underbrace{\int_{-\infty}^{\infty} n(\tau)h(-\tau)d\tau}_{z} = y + z.$$

Of course, $q(0)$ is a number, and so are y and z.

Now, the SNR of $q = q(0)$ is

$$\left.\frac{S}{N}\right|_q = \frac{\mathbb{E}\left[y^2\right]}{\mathbb{E}\left[z^2\right]} \overset{y \text{ is deterministic}}{=} \frac{y^2}{\mathbb{E}\left[z^2\right]}.$$

We compute separately numerator and denominator...but before that we need to review the...

Cauchy–Schwarz inequality

The Cauchy–Schwarz inequality states that for all vectors $x, y \in V$, where V is a vector space with an inner product defined, it holds that

$$|\langle x, y\rangle|^2 \leq \langle x, x\rangle \cdot \langle y, y\rangle$$

with equality when x and y are linearly dependent, that is, $x = K \cdot y$.

With our definition of inner product, this is

$$\left|\int_{-\infty}^{\infty} x(t)y(t)dt\right|^2 \leq \int_{-\infty}^{\infty} x^2(t)dt \int_{-\infty}^{\infty} y^2(t)dt$$

Let's go back to computing the SNR:

power of the noise

$$\begin{aligned}\mathbb{E}\left[z^2\right] &= \mathbb{E}\left[\int_{-\infty}^{\infty} n(\tau)h(-\tau)d\tau \int_{-\infty}^{\infty} n(u)h(-u)du\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbb{E}\left[n(\tau)n(u)\right] h(-\tau)h(-u)d\tau du \\ &= \frac{N_0}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(\tau - u)h(-\tau)h(-u)d\tau du = \frac{N_0}{2}\int_{-\infty}^{\infty} h(-u)\left[\int_{-\infty}^{\infty} \delta(\tau - u)h(-\tau)d\tau\right] du \\ &= \frac{N_0}{2}\int_{-\infty}^{\infty} h(-u)h(-u)du = \frac{N_0}{2}\int_{-\infty}^{\infty} h^2(-u)du\end{aligned}$$

power of the signal

$$y^2 = \left(\int_{-\infty}^{\infty} s(\tau)h(-\tau)d\tau\right)^2 \overset{\text{Cauchy-Schwartz}}{\leq} \int_{-\infty}^{\infty} s^2(\tau)d\tau \int_{-\infty}^{\infty} h^2(-\tau)d\tau = \varepsilon_s \int_{-\infty}^{\infty} h^2(-\tau)d\tau. \tag{2.7}$$

Then, the SNR of q is

$$\left.\frac{S}{N}\right|_q = \frac{y^2}{\mathbb{E}\left[z^2\right]} = \frac{\left(\int_{-\infty}^{\infty} s(\tau)h(-\tau)d\tau\right)^2}{\frac{N_0}{2}\int_{-\infty}^{\infty} h^2(-u)du} \leq \frac{\varepsilon_s \cancel{\int_{-\infty}^{\infty} h^2(-\tau)d\tau}}{\frac{N_0}{2}\cancel{\int_{-\infty}^{\infty} h^2(-\tau)d\tau}} = \frac{2\varepsilon_s}{N_0},$$

that is, the SNR at the output is always less than or equal to $\frac{2\varepsilon_s}{N_0}$. But, according to the Cauchy-Schwartz theorem, we know that we get equality in (2.7) when one of the signals is a scaled version of the other, i.e., here when

$$h(-t) = Ks(t).$$

If $h(t)$ is a filter matched to $s(t)$, then

$$h(t) = s(-t) \Rightarrow h(-t) = 1 \cdot s(t),$$

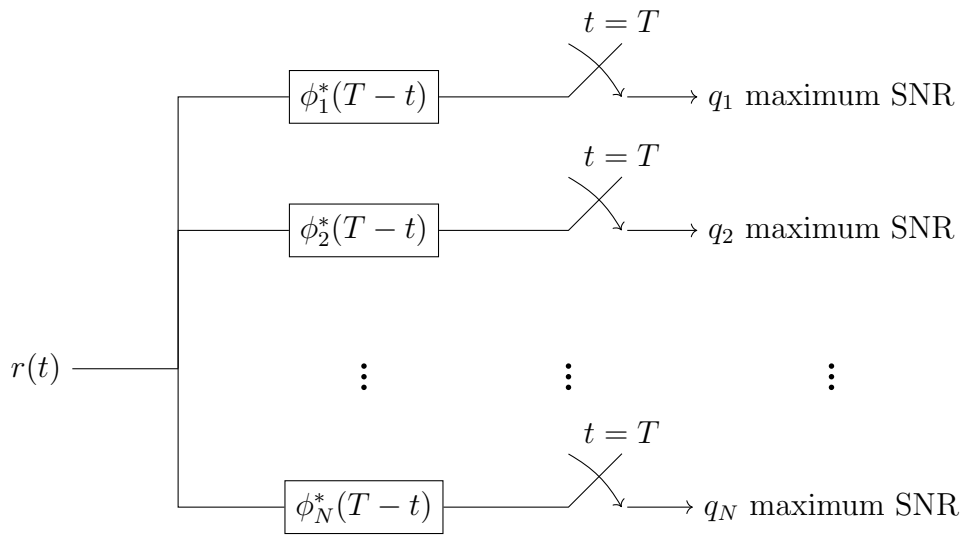
and we have equality in (2.7), that is,

$$y^2 = \varepsilon_s \int_{-\infty}^{\infty} h^2(-\tau)d\tau,$$

which, in turn, means that the SNR at the output is maximum,

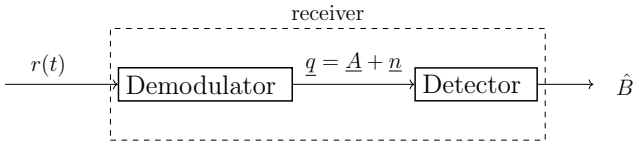
$$\left.\frac{S}{N}\right|_q = \frac{2\varepsilon_s}{N_0}.$$

Recall our filters-based implementation of a demodulator



and hence we are detecting every component (coordinate) in our signal with maximum SNR.

2.9 Detector

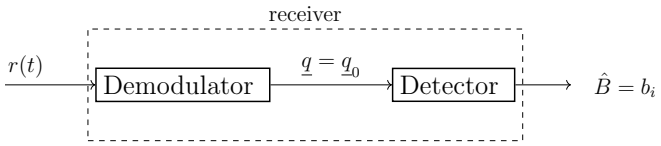


We have studied the first component of the receiver, and now we are going to discuss the other one, which is also the last component in our digital communications system: the detector. Its goal is to make a decision about which symbol was transmitted based on the observations vector $\underline{q}$.

At the beginning of this module we saw that the design criterion/rule for the detector is to minimize the probability of error. Hence, we are going to analyze the connection between the decisions we make and the probability of error.

Let us assume the demodulator computes (yields at its output) a vector $\underline{q} = \underline{q}_0$ ($\underline{q}_0$ is a realization of random vector $\underline{q}$) to which the detector assigns the symbol $\hat{B} = b_i$, i.e,

$$\underline{q} = \underline{q}_0$$
$$\hat{B} = b_i.$$



Then, our goal is to compute the probability of this being an erroneous decision. We will denote it as $P_e(\hat{B} = b_i, \underline{q} = \underline{q}_0)$, i.e., this is the probability of an erroneous decision when the vector of observations is $\underline{q}_0$ and we decide b_i,

$$P_e(\hat{B} = b_i, \underline{q} = \underline{q}_0) \triangleq P(B \neq b_i \mid \underline{q} = \underline{q}_0)$$

this is 1 minus the probability of the complementary event

$$= 1 - P(B = b_i \mid \underline{q} = \underline{q}_0)$$

we are going to write this probability in a more compact manner

$$= 1 - \underbrace{p_{B|\underline{q}}(b_i \mid \underline{q}_0)}_{\substack{\text{probability of} \\ \text{being right}}} \tag{2.8}$$

This is the probability of error when we decide the symbol b_i **and** the observations vector is $\underline{q}_0$...but what we are really interested in is the probability of error when we decide b_i regardless of the observed vector (the *mean* probability of error). We can obtain it by averaging the latter with respect to $\underline{q}_0$. It's given by the same expression we had before but now without conditioning on a particular observation,

$$P_e(\hat{B} = b_i) = P(B \neq b_i)$$

using the law of total probability[41]

$$= \int_{\mathbb{R}^N} P(B \neq b_i \mid \underline{q}_0) f_{\underline{q}}(\underline{q}_0) d\underline{q}_0 \overset{\text{using (2.8)}}{=} \int_{\mathbb{R}^N} \left(1 - p_{B|\underline{q}}(b_i \mid \underline{q}_0)\right) f_{\underline{q}}(\underline{q}_0) d\underline{q}_0$$
$$= \int_{\mathbb{R}^N} f_{\underline{q}}(\underline{q}_0) d\underline{q}_0 - \int_{\mathbb{R}^N} p_{B|\underline{q}}(b_i \mid \underline{q}_0) f_{\underline{q}}(\underline{q}_0) d\underline{q}_0 = 1 - \int_{\mathbb{R}^N} p_{B|\underline{q}}(b_i \mid \underline{q}_0) f_{\underline{q}}(\underline{q}_0) d\underline{q}_0$$

Some remarks here:

- In order to minimize this probability we need to maximize the resulting integral by choosing b_i for every value of $\underline{q}_0$[42].
- The function inside the integral is always positive

  $$\underbrace{p_{B|\underline{q}}(b_i \mid \underline{q}_0)}_{\geq 0} \underbrace{f_{\underline{q}}(\underline{q}_0)}_{\geq 0} \geq 0,$$

  and hence the integral gets maximized if we maximize the value of this function (with respect to b_i) for every possible $\underline{q}_0$[43]. In other words, the maximum is achieved if every addend (in the integral's infinite summation) is maximum.

[41] $P(A) = \int_{-\infty}^{\infty} P(A \mid X = x) f_X(x) dx$.

[42] The detector receives $\underline{q}_0$ as input and its only purpose/task is choosing the output, b_i, for that input.

[43] For every $\underline{q}_0$ we choose an appropriate b_i.

- the factor $f_{\underline{q}}(\underline{q}_0)$ inside the integral does not depend on the decision we make, that is, the detector cannot affect this term, so we can forget about it when maximizing, and we only need to care about maximizing

$$p_{B|\underline{q}}(b_i|\underline{q}_0),$$

which is the probability of the symbol transmitted being equal to b_i when we have observed vector $\underline{q}_0$.

In summary,

$$\min_{b_i} \underbrace{P_e(\hat{B}=b_i)}_{\substack{\text{the probability of error} \\ \text{when we decide } \hat{B}=b_i}} \Leftrightarrow \max_{b_i} p_{B|\underline{q}}(b_i|\underline{q}_0) \ \forall \underline{q}_0 \equiv \textbf{Maximum A Posteriori (MAP) rule}$$

We will see in a minute the reason behind the naming. For now, just notice that both minimization and maximization are with respect to b_i. The above result goes to say that if you want to minimize the *mean* error probability, then *for every observations vector*, $\underline{q}_0$, you need to choose b_i so that it maximizes the probability of the symbol transmitted, B, at the sight of those observations.

Some notation we will be using from now on:

- $p_{B|\underline{q}}(b_j|\underline{q}_0), j = 1, \cdots, M$ are called ***posterior* probabilities**: they are the probabilities of the symbols once we know the observations vector $\underline{q}_0$ (the observations provide information about the symbol transmitted),

and as opposed to them we have

- $p_B(b_j), j = 1, \cdots, M$ are the ***prior* probabilities** of the symbols: they are the probabilities of the symbols **before** *seeing* the observations. Most of the time, the symbols will be equally likely.

What is the meaning of this **MAP rule** in practice? Given a vector of observations, $\underline{q}_0$, the detector must compute the (finite!!) set of probabilities $\left\{p_{B|\underline{q}}(b_j|\underline{q}_0), j = 1, \cdots, M\right\}$ and choose b_i such that

$$p_{B|\underline{q}}(b_i|\underline{q}_0) > p_{B|\underline{q}}(b_j|\underline{q}_0) \ \forall j \neq i.$$

If

$$p_{B|\underline{q}}(b_i|\underline{q}_0) = p_{B|\underline{q}}(b_k|\underline{q}_0) > p_{B|\underline{q}}(b_j|\underline{q}_0) \ \forall i \neq j \neq k,$$

i.e., there is a tie between symbols i and k, the detector decides b_i or b_k arbitrarily and that does not affect the probability of error.

This completely determines the operation of the detector, and now our task is to find a closed-form expression for computing the *posterior* probabilities $p_{B|\underline{q}}(b_i|\underline{q}_0)$ (we don't know how to compute this directly). We apply Bayes theorem to get[44]

$$p_{B|\underline{q}}(b_i \mid \underline{q}_0) = \frac{f_{\underline{q}|B}(\underline{q}_0 \mid b_i) p_B(b_i)}{f_{\underline{q}}(\underline{q}_0)}$$

[44]Recall that we use f to denote pdf's (rather than pmf's).

since $B = b_i \Leftrightarrow \underline{A} = \underline{a}_i$ (these two conditions are the same because we have a one-to-one mapping between information symbols and elements in the constellation), we have $f_{\underline{q}|B}(\underline{q}_0 \mid b_i) = f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_i)$

$$= \frac{f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_i) p_B(b_i)}{f_{\underline{q}}(\underline{q}_0)}.$$

Then,

$$\begin{gathered} p_{B|\underline{q}}(b_i|\underline{q}_0) > p_{B|\underline{q}}(b_j|\underline{q}_0),\ \forall j \neq i \\ \Downarrow \\ \frac{f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_i) p_B(b_i)}{f_{\underline{q}}(\underline{q}_0)} > \frac{f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_j) p_B(b_j)}{f_{\underline{q}}(\underline{q}_0)},\ \forall j \neq i, \end{gathered}$$

but $f_{\underline{q}}(\underline{q}_0)$ is positive and independent of the symbol transmitted (it's a constant in the comparisons!!). Hence, we can simplify the above inequality and the MAP rule amounts to...

MAP rule

Choose b_i such that

$$f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_i) p_B(b_i) > f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_j) p_B(b_j),\ \forall j \neq i$$

(notice that the expressions being compared are likelihoods times *prior* probabilities).

In summary, given $\underline{q} = \underline{q}_0$, the MAP rule proceeds by

1. computing
$$\begin{gathered} f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_1) p_B(b_1) \\ f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_2) p_B(b_2) \\ \vdots \\ f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_M) p_B(b_M) \end{gathered}$$
2. choosing the maximum, and deciding its corresponding b_i,

that is, we compute the above product for every element in the constellation and choose the one that yields the maximum.

Quick quiz

What is the maximum value $f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_i)$ can take? [45]

[45] Just like any probability density function, $f_{\underline{q}|\underline{A}}$ is unbounded above (though always greater than or equal to 0).

Likelihood
The pdf $f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \underline{a}_i)$ is know as the *likelihood* (here, of $\underline{a}_i$). A likelihood is always the probability (or probability density) of some (known) observation(s) conditional on some unknown parameter(s) we are interested in.

2.9.1 Example I

- $M = 2$, $N = 1$
- noise is AWGN
- equally likely symbols

($\underline{q}$ and $\underline{A}$ are scalars)

If the output of the demodulator is $\underline{q} = \underline{q}_0$, we have to compute

$$f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid -\sqrt{T})p(b_1), \quad \text{and} \quad f_{\underline{q}|\underline{A}}(\underline{q}_0 \mid \sqrt{T})p(b_2). \tag{2.9}$$

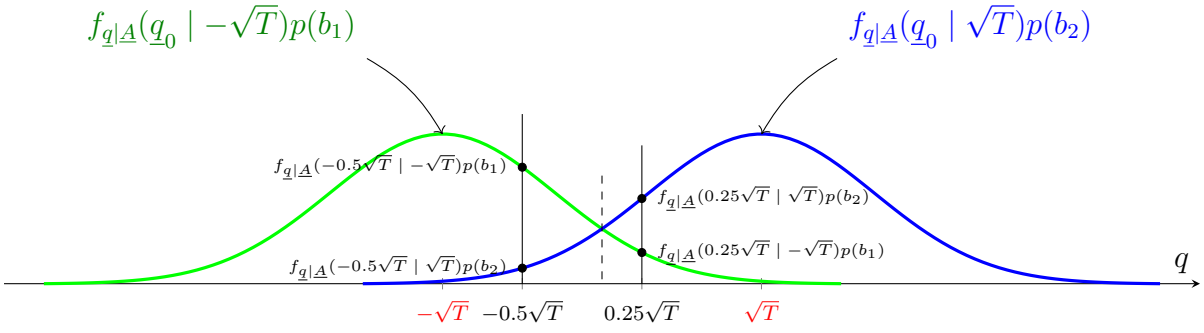
Since the noise is Gaussian, we know the distribution of the observations vector conditional on the symbol transmitted[46]

$$\underline{q} \mid \underline{A} = \underline{a}_i \sim \mathcal{N}\left(\underline{a}_i, \frac{N_0}{2}\underline{\underline{I}}_N\right),$$

and hence

$$q \mid A = -\sqrt{T} \sim \mathcal{N}\left(-\sqrt{T}, \frac{N_0}{2}\right)$$
$$q \mid A = \sqrt{T} \sim \mathcal{N}\left(\sqrt{T}, \frac{N_0}{2}\right).$$

Since the above random variables are Gaussian (with known mean and variance), we know their pdf's (the f's), and the expressions to be compared when applying the MAP rule in (2.9) are simply functions of q (above evaluated at $q = q_0$) given by the product of a Gaussian pdf (each one with its own mean) and a constant equal to 1/2 (the prior probability of each symbol).



[46] If the noise is not Gaussian, then we have to figure things out again just like we did in Section 2.7.

We can apply the MAP rule for several values of q_0:

- if $q_0 = 0.25\sqrt{T}$

$$f_{q|A}(0.25\sqrt{T} \mid \sqrt{T})p(b_2) > f_{q|A}(0.25\sqrt{T} \mid -\sqrt{T})p(b_1)$$

  and we decide $\hat{B} = b_2$

- if $q_0 = -0.5\sqrt{T}$

$$f_{q|A}(-0.5\sqrt{T} \mid -\sqrt{T})p(b_1) > f_{q|A}(-0.5\sqrt{T} \mid \sqrt{T})p(b_2)$$

  and we decide $\hat{B} = b_1$.

If we are given a particular q_0, there is no need to plot these functions to make a decision, but if we do, we can immediately see the decision to be made for every value of q_0. For which values of q_0 should we decide b_1? and b_2? Intuitively, we can see that

- we decide b_1 if $q_0 \leq 0 \Leftrightarrow q_0 \in I_1 = (-\infty, 0]$
- we decide b_2 if $q_0 \geq 0 \Leftrightarrow q_0 \in I_2 = [0, \infty)$.

$q_t = 0$ is known as the *decision threshold*, and we say that

$$I_1 \equiv \text{decision region for } b_1$$
$$I_2 \equiv \text{decision region for } b_2.$$

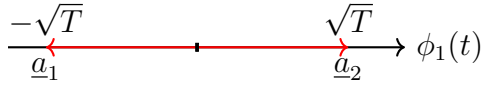
So now you know what to decide for every value of q_0. Notice that a detection error happens when the observation q_0 falls outside the decision region of the symbol actually transmitted. If q_0 falls within the region I_i then, *according to the MAP rule* (which is optimal), we should decide b_i. This means, that when we transmit b_i, there is no error as long as q_0 falls within the decision region I_i.

Why is the decision threshold at $q_t = 0$? Because it's the value at which the two curves cross/intersect, i.e., the value of q_0 satisfying

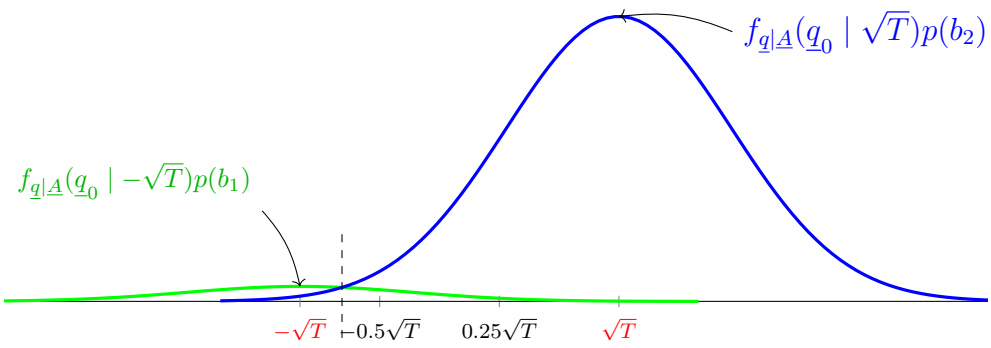
$$f_{q|A}(q_t \mid \underline{A} = -\sqrt{T})p(b_1) = f_{q|A}(q_t \mid \underline{A} = \sqrt{T})p(b_2).$$

2.9.2 Example II

- $M = 2$, $N = 1$
- noise is AWGN
- $p(b_2) > p(b_1)$



(same constellation as before)



The decision threshold has shifted. Let us see what happens for the same (example) values of q_0 we investigated before:

- if $q_0 = 0.25\sqrt{T}$

$$f_{q|A}(0.25\sqrt{T} \mid \sqrt{T})p(b_2) > f_{q|A}(0.25\sqrt{T} \mid -\sqrt{T})p(b_1)$$

  and we decide $\hat{B} = b_2$

- if $q_0 = -0.5\sqrt{T}$ (right **after** the decision threshold)

$$f_{q|A}(-0.5\sqrt{T} \mid \sqrt{T})p(b_2) > f_{q|A}(-0.5\sqrt{T} \mid -\sqrt{T})p(b_1)$$

  and we decide $\hat{B} = b_2$.

Since the probability of transmitting b_2 is larger than that of transmitting b_1, it's only natural that we decide b_2 more often than we decide b_1. Hence, the probability of the symbols matters when making decisions

2.9.3 Maximum Likelihood (ML) rule

If the symbols are equally likely (the usual scenario), the MAP rule for decision can be simplified. Given $\underline{q} = \underline{q}_0$, we need to solve the optimization problem

$$\max \left\{ \begin{array}{c} f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_1)p(b_1) \\ f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_2)p(b_2) \\ \vdots \\ f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_M)p(b_M) \end{array} \right\} \overset{p(b_1)=p(b_2)=\cdots p(b_M)}{=} \max \left\{ \begin{array}{c} f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_1) \\ f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_2) \\ \vdots \\ f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_M) \end{array} \right\}$$

and the result is the...

Maximum Likelihood (ML) rule

Choose b_i such that

$$f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_i) > f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_j) \; \forall j \neq i.$$

This is true as long as the symbols are equally likely (very common!!).

2.9.4 Proximity rule

Additionally, **if the noise is Gaussian** (and only in that case!!), we have

$$f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_i) > f_{\underline{q}|\underline{A}}(\underline{q}_0|\underline{a}_j) \; \forall j \neq i$$

$$\Downarrow$$

$$\frac{1}{(\pi N_0)^{N/2}} e^{-\frac{d^2(\underline{q}_0, \underline{a}_i)}{N_0}} > \frac{1}{(\pi N_0)^{N/2}} e^{-\frac{d^2(\underline{q}_0, \underline{a}_j)}{N_0}} \; \forall j \neq i$$

we take logs at both sides of the inequality

$$\begin{gathered}\Downarrow \\ -d^2(\underline{q}_0, \underline{a}_i) > -d^2(\underline{q}_0, \underline{a}_j)\ \forall j \neq i \\ \Downarrow \\ d^2(\underline{q}_0, \underline{a}_i) < d^2(\underline{q}_0, \underline{a}_j)\ \forall j \neq i \\ \Downarrow \\ d(\underline{q}_0, \underline{a}_i) < d(\underline{q}_0, \underline{a}_j)\ \forall j \neq i\end{gathered}$$

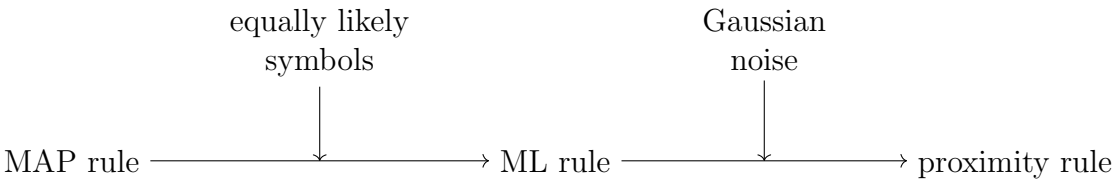
So, when the **noise is Gaussian**[47], the ML rule amounts to computing the Euclidean distance (similarity) between the observations vector $\underline{q}$ and every possible element in the constellation $\underline{a}_i, i = 1, \cdots, M$, and deciding the one that is closest (most similar!!). This yields the...

Proximity rule
Choose b_i such that

$$d(\underline{q}_0, \underline{a}_i) < d(\underline{q}_0, \underline{a}_j)\ \forall j \neq i.$$

2.9.5 The appropriate rule

The take-home message here is: the MAP rule amounts to the ML rule when the symbols are equally likely, and the latter amounts to a proximity rule when the noise is Gaussian



If the first condition (equally likely symbols) is not met, we **cannot** go further along the line.

Quick quiz
Consider a binary digital communication system with $p(b_1) = 0.6$ and $p(b_2) = 0.4$ operating in a Gaussian channel. What is the appropriate decision rule? [48]

[47] We obtained the above expression for $f_{\underline{q}|A}$ in Section 2.7 assuming Gaussian (thermal) noise.

[48] It's the MAP rule since the symbols not being equally likely won't allow us to *move forward* from the MAP rule to the ML rule.

2.10 Decision regions

The decision regions are regions defined in the space of the observations vector, $\underline{q}$ (for us, $\underline{q} \in \mathbb{R}^N$, and hence the decision regions are in $\mathbb{R}^N$),

$$I_1, I_2, \cdots, I_M \subset \mathbb{R}^N,$$

that allow to decide the symbol transmitted, $\hat{B}$, directly from vector $\underline{q}$ without computing probabilities. Specifically, I_i is the region in the space $\mathbb{R}^N$ for which we should decide b_i according to the MAP rule. So, the whole point of decision regions is to avoid the computation of probabilities every time a new observation arrives. We have as many decision regions as symbols and they are *defined* so that:

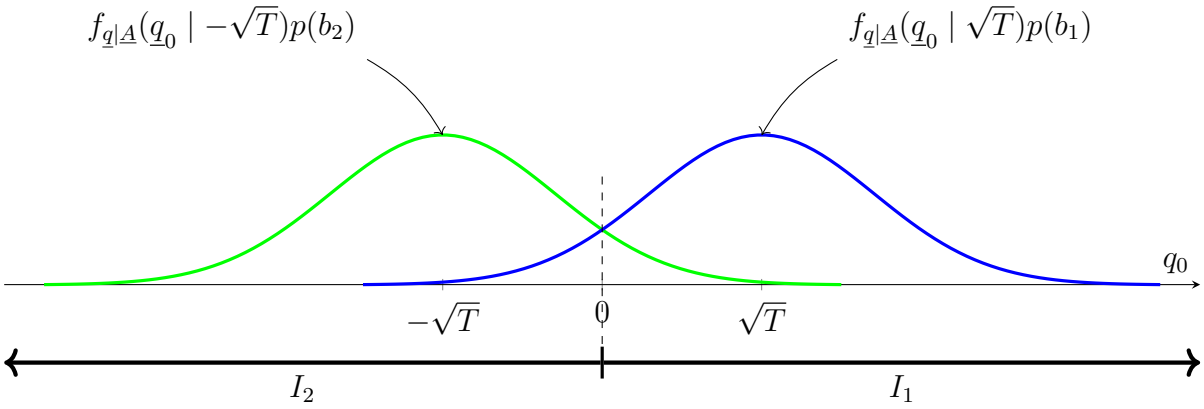
- they are disjoint, $I_i \cap I_j = \emptyset, i \neq j$
- they cover the entire space in which $\underline{q}$ is defined, $\cup_i I_i = \mathbb{R}^N$ (we need to know what to do for every observations vector $\underline{q}$)
- they must respect the MAP rule, i.e., we *define* I_i such that

$$\underline{q} \in I_i \Leftrightarrow \underbrace{f_{\underline{q}|\underline{A}}(\underline{q}|\underline{a}_i)p(b_i) > f_{\underline{q}|\underline{A}}(\underline{q}|\underline{a}_j)p(b_j)\forall j \neq i}_{\text{MAP rule}} \Leftrightarrow \hat{B} = b_i$$

In summary, **they are defined so that the optimal decision is b_i if $\underline{q}$ falls within the region I_i**. Then, when you receive some observation, you find out which decision region it belongs to, and then you decide for the corresponding symbol.

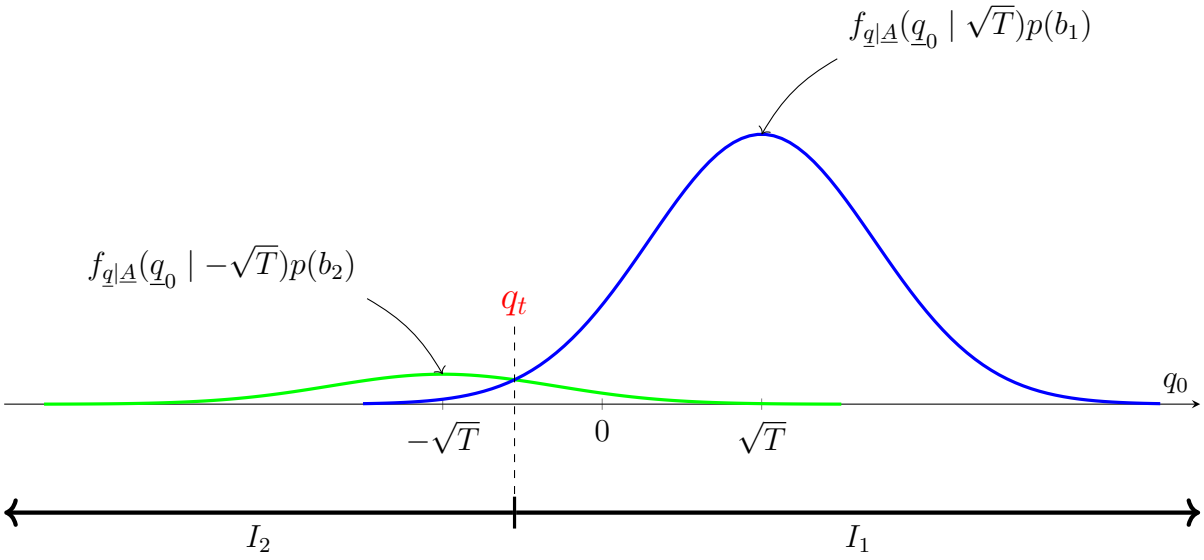
Let us compute the decision regions in a few different situations.

2.10.1 Example: $M = 2$, $N = 1$, $p(b_1) = p(b_2)$, Gaussian noise



The rule is proximity. In order to apply it we can draw a line between the two elements and, after that, the perpendicular bisector will tell us which points in the space are closer to one element than to the other.

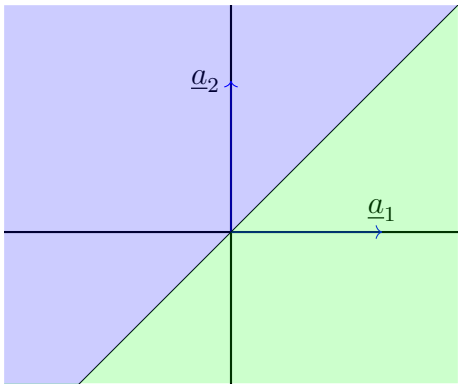
2.10.2 Example: $M = 2$, $N = 1$, $p(b_1) > p(b_2)$, Gaussian noise



When the probabilities of the symbols are different, the decision threshold is **not** in the middle (the rule is **not** proximity but MAP). Later on, we will see an example of how to compute the threshold when the symbols are not equally likely (although you already know the equation to be solved).

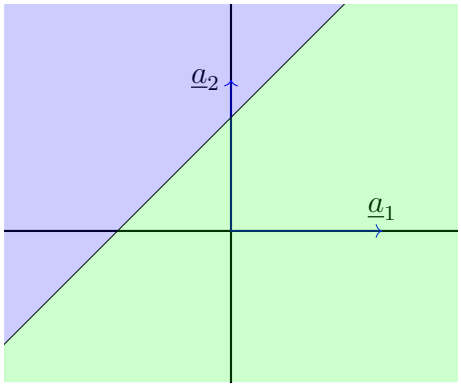
2.10.3 Example: $M = 2$, $N = 2$, $p(b_1) = p(b_2)$, Gaussian noise

The rule is proximity. Notice that, when choosing the appropriate rule, the number of elements in the constellation, M, and the dimension of the latter, N, are irrelevant.



2.10.4 Example: $M = 2$, $N = 2$, $p(b_1) > p(b_2)$, Gaussian noise

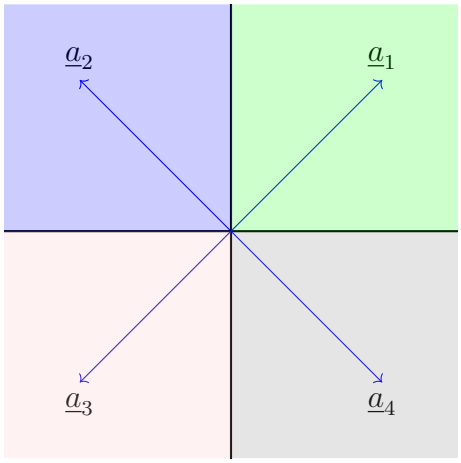
The rule is MAP because symbols are **not** equally likely.



Since $\underline{a}_1$ has been more likely transmitted, its decision region is bigger[49].

2.10.5 Example: $M = 4$, $N = 2$, $p(b_i) = 1/4$, Gaussian noise

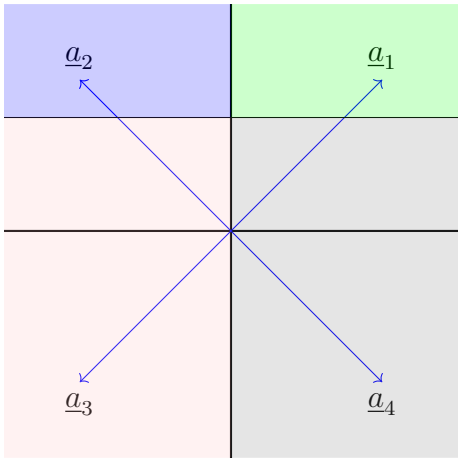
The rule is proximity.



2.10.6 Example: $M = 4$, $N = 2$, $p(b_1) = p(b_2) < p(b_3) = p(b_4)$, Gaussian noise

The rule is MAP because symbols are **not** equally likely.

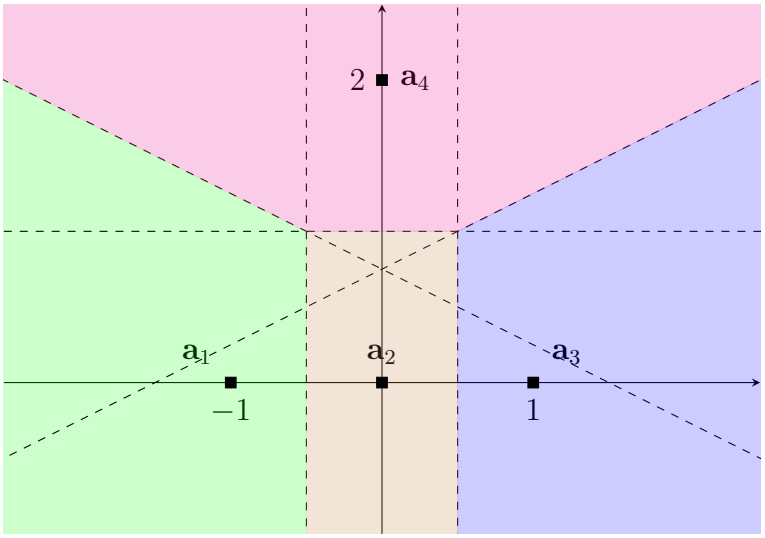
[49]Rigorously speaking, both regions encompass an infinite area. However, the area under the (unconditional) pdf of the observations is larger for $\underline{a}_1$'s decision region (meaning we will be deciding $\underline{a}_1$ more often).



Since symbols b_3 and b_4 are more likely than b_1 and b_2, more often than not we will decide they were transmitted, and hence their decision regions should be larger.

2.10.7 Example: generic 2D constellation with equally likely symbols and Gaussian noise

The rule is proximity.



We focus on computing the decision region for $\underline{a}_2$: we need to find the points in the space that are closer to this symbol than to any other symbol. The points in that region should be

- closer to $\underline{a}_2$ than to $\underline{a}_3$, **and at the same time**
- closer to $\underline{a}_2$ than to $\underline{a}_4$, **and at the same time**
- closer to $\underline{a}_2$ than to $\underline{a}_1$.

Hence we need to compute the intersection of three regions, and each one is obtained using a perpendicular bisector:

- We join $\underline{a}_2$ and $\underline{a}_3$, find the perpendicular bisector, and the points to the **left** of the line are closer to $\underline{a}_2$ than they are to $\underline{a}_3$.
- We join $\underline{a}_2$ and $\underline{a}_4$, find the perpendicular bisector, and the points **below** the line are closer to a_2 than they are to a_4.
- We join $\underline{a}_2$ and $\underline{a}_1$, find the perpendicular bisector, and the points to the **right** of the line are closer to $\underline{a}_2$ than to $\underline{a}_1$.

The intersection of those three regions is closer to $\underline{a}_2$ than it is to any other symbol: it comprises its decision region.

? Quick quiz

What is the correct decision to be made for an observation, $\underline{q}_0$, that exactly falls on the border of two regions (corresponding to two different elements of the constellation)? [50]

2.11 Computation of the probability of error

Let us see how to compute the probability of error attained by our detector. In order to do it we are going to use the law of total probability,

$$P_e = P_{e|b_1} p(b_1) + P_{e|b_2} p(b_2) + \cdots + P_{e|b_M} p(b_M),$$

and the conditional probabilities, $P_{e|b_i}, i = 1, \cdots, M$, that show up here are computed using the decision regions[51]. In particular, we know that, *if the symbol transmitted was* $B = b_i$:

- $\nexists$ error if $\underline{q} \in I_i$ (the decision region for the symbol actually transmitted)
- $\exists$ error if $\underline{q} \notin I_i$

Then,

$$P_{e|b_i} = P_{e|\underline{a}_i} = P\left(\underbrace{\underline{q}|\underline{A} = \underline{a}_i}_{\text{r.v.}} \notin I_i \right),$$

and we can compute the probability of $\underline{q}$ not belonging to the decision region I_i conditional on the fact that the symbol transmitted was $\underline{a}_i$ as long as we know the corresponding pdf, $\underline{q}|\underline{A} = \underline{a}_i$. That's always going to be the case and, e.g., for AWGN we have (see 2.7.2)

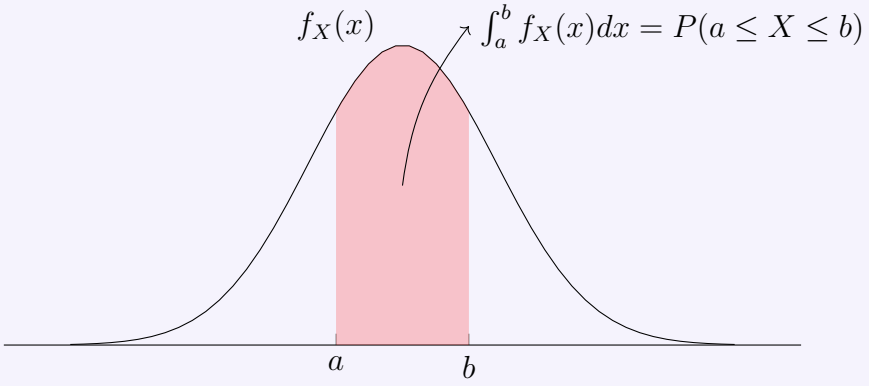
$$\underline{q}|\underline{A} = \underline{a}_i \sim \mathcal{N}\left(\underline{a}_i, \frac{N_0}{2} \underline{\underline{I}}_N \right).$$

[50] Along the border there is a tie and both symbols will yield the same probability of error. Hence, we can choose either of the two.

[51] Notice that $P_{e|b_i} = P_{e|\underline{a}_i}$.

Probability from pdf

If we know the pdf, $f_X(x)$, of a random variable X we can compute the probability of the random variable taking values in a given interval of interest, let's say (a, b)



$$P(a \leq X \leq b) = \int_a^b f_X(x)dx$$

It's the same reasoning here, but now the random variable is $\underline{q}|\underline{A} = \underline{a}_i$. Hence, putting it all together we have

$$P_{e|b_i} = P_{e|\underline{a}_i}$$

an error happens when the observation vector $\underline{q}$ doesn't fall within the correct decision region (that for b_i)

$$= P\left(\underbrace{\underline{q}|\underline{A} = \underline{a}_i}_{\text{r.v.}} \notin I_i\right) = \int_{q \notin I_i} f_{\underline{q}|\underline{A}}(\underline{q}|\underline{a}_i)dq,$$

and we are simply integrating a Gaussian random variable (now is when the Q function comes handy). Notice that for computing the above P_e we need the **decision regions**. Hence, finding the latter is always the **first step** for computing the probability of error.

Next, we see how to apply this in different scenarios.

2.11.1 Example: $M = 2, N = 1$, equally likely symbols, Gaussian noise

This is an important case because it's very common.

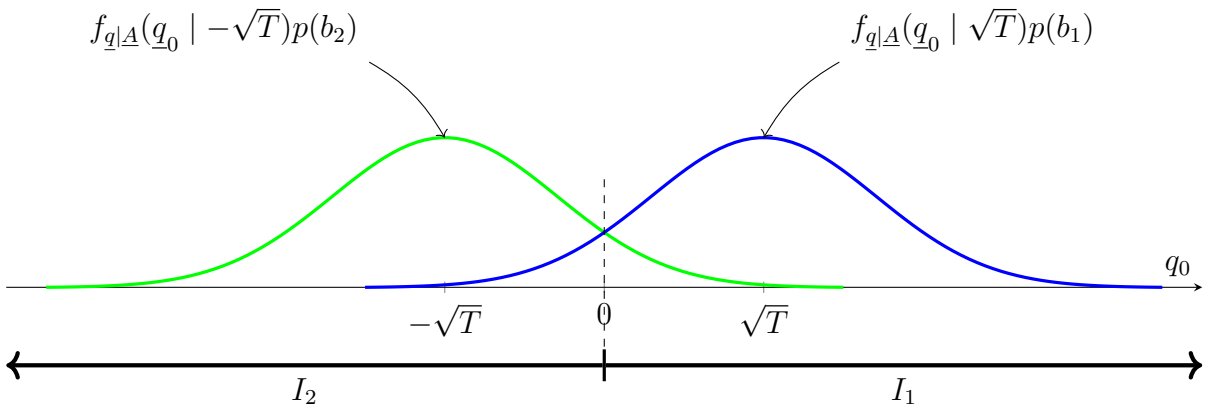
Let us assume that the PSD of thermal noise (i.e., the variance of the noise) is $\frac{N_0}{2}$.

The steps we follow are always the same:

Compute the decision regions. In principle we must apply the MAP rule, which states that we should decide b_1 if

$$f_{q|A}(q_t \mid \underline{A} = \sqrt{T})p(b_1) > f_{q|A}(q_t \mid \underline{A} = -\sqrt{T})p(b_2),$$

and b_2 otherwise. If we make a plot, we get something like this



and now our goal is to find the threshold, say q_t, at which both curves intersect. It can be obtained by solving

$$f_{q|A}(q_t \mid \underline{A} = \sqrt{T})p(b_1) = f_{q|A}(q_t \mid \underline{A} = -\sqrt{T})p(b_2).$$

However, since the symbols are equally likely and the noise is Gaussian, we set the decision regions using the proximity rule: we decide b_1 for the values of q (scalar) that are closer to a_1 than a_2. Hence, the decision threshold is $q_t = 0$, the middle point between the two elements.

If the symbols were not equally likely, we should apply the MAP rule and solve the above equation.

The probability of error is written in terms of the probability of error conditional on each symbol: we just apply the law of total probability

$$P_e = P_{e|\sqrt{T}}p(b_1) + P_{e|-\sqrt{T}}p(b_2) = P_{e|\sqrt{T}}\frac{1}{2} + P_{e|-\sqrt{T}}\frac{1}{2},$$

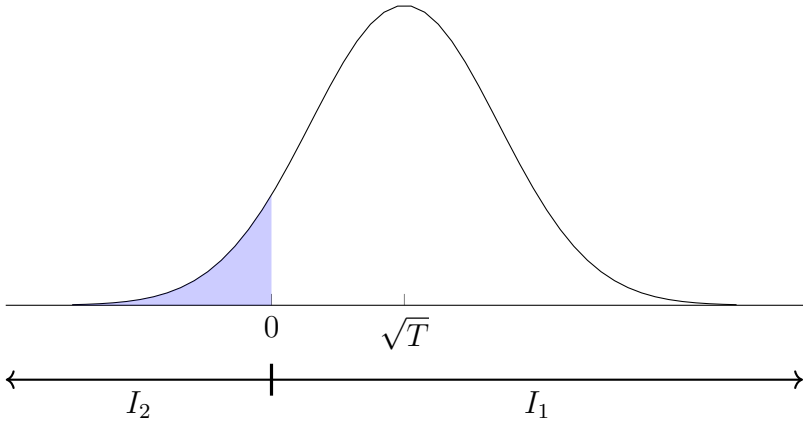
where we have used that the symbols are equally likely, and hence their probability is $\frac{1}{2}$.

Compute each conditional probability of error on its own.

- The conditional error probability when $\sqrt{T}$ was transmitted is given by

$$\begin{aligned} P_{e|\sqrt{T}} &= p\left(q \mid A = \sqrt{T} \notin [0, \infty)\right) = p\left(q \mid A = \sqrt{T} \in (-\infty, 0)\right) \\ &= \int_{-\infty}^{0} f_{q|A}(q \mid \sqrt{T})dq \end{aligned}$$

in plotting this pdf we observe that we are integrating the tail of a Gaussian random variable with know mean and variance: we can use the Q function!



$$= Q\left(\frac{\sqrt{T}}{\sqrt{N_0/2}}\right)$$

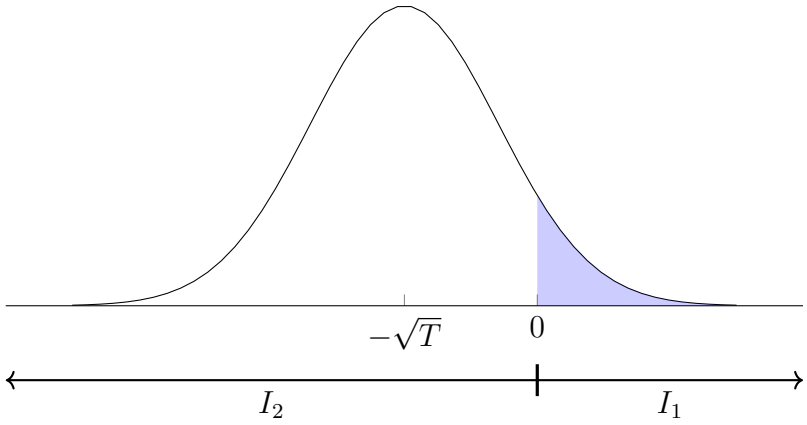
Notice that the numerator in the Q function is the distance between the mean and the threshold of integration, whereas the denominator is simply the standard deviation.

- We need to do the same thing for the probability of error when $-\sqrt{T}$ was transmitted. Again, there is an error when the observation, q here, falls outside the decision region for the transmitted symbol, and hence,

$$P_{e|-\sqrt{T}} = p\left(q \mid A = -\sqrt{T} \notin (-\infty, 0)\right)$$

if it doesn't belong to the negative part of the axis, then it must belong to the positive part

$$= p\left(q \mid A = -\sqrt{T} \in [0, \infty)\right) = \int_0^\infty \underbrace{f_{q|A}(q \mid -\sqrt{T})}_{\substack{\text{pdf of } q \text{ when} \\ A=-\sqrt{T}}} dq$$



$$= Q\left(\frac{\sqrt{T}}{\sqrt{N_0/2}}\right)$$

The area in both tails is the same. We could have argued that both error probabilities are equal due to the symmetry of the problem: the Gaussian pdf is symmetric and in the first case we computed the area of a tail that is $\sqrt{T}$ away to the **left** of the mean, whereas now we are computing the area of a tail that is $\sqrt{T}$ away to the **right** of the mean.

Pictures for computing P_e
Usually, you don't make a picture for the pdf of the observations conditional on every symbol, but "recycle" the one you used for establishing the decision regions.

The final overall probability of error is

$$P_e = \frac{1}{2}P_{e|\sqrt{T}} + \frac{1}{2}P_{e|-\sqrt{T}} = \frac{1}{2}\left(P_{e|\sqrt{T}} + P_{e|-\sqrt{T}}\right) = \frac{1}{2}2 \times P_{e|\sqrt{T}} = P_{e|\sqrt{T}} = Q\left(\frac{\sqrt{T}}{\sqrt{N_0/2}}\right)$$

For this constellation, we have already computed the energy and the distance between the elements in the constellation[52]

$$E_s = \frac{1}{2}T + \frac{1}{2}T = T$$
$$d(a_1, a_2) = 2\sqrt{T}$$

and we have

$$E_s = T \Rightarrow \sqrt{E_s} = \sqrt{T} \tag{2.1}$$
$$d(a_1, a_2) = 2\sqrt{T} \Rightarrow \sqrt{T} = \frac{d(a_1, a_2)}{2}. \tag{2.2}$$

Using this, we can rewrite the expression for the probability of error,

$$P_e = Q\left(\frac{\sqrt{T}}{\sqrt{N_0/2}}\right)$$

using equation (2.2) as

$$= Q\left(\frac{d(a_1, a_2)}{2\sqrt{N_0/2}}\right) \leftarrow \text{the most widely used}$$

and using equation (2.1) as

$$= Q\left(\sqrt{\frac{E_s}{\frac{N_0}{2}}}\right). \tag{2.3}$$

Using equation (2.2) we get an expression as a function of the distance between the elements in the constellation, whereas using equation (2.1) we get an expression as a function of the mean energy. The former is the more widely used.

Notice that

[52] This was the constellation of two different sets in Section 2.5.2 (we were choosing the best set of signals for transmission accounting for the energy and distance between signals).

- $\uparrow E_s \Rightarrow \downarrow P_e$
- $\uparrow SNR \left(= \frac{E_s}{\frac{N_0}{2}}\right) \Rightarrow \downarrow P_e$ (E_s is the power of the signal of interest[53] and $\frac{N_0}{2}$ is the power of the noise)

Validity of these formulas

These are expressions for the probability of error in any constellation as long as

- it's binary,
- the two symbols are equally likely, and
- the noise is Gaussian.

If these conditions are met, we can use any of these expressions regardless of the dimension, N, of the constellation[a].

[a] Equation (2.3) further assumes the constellation is centered (which *should* always be the case). Otherwise you don't have two elements with the exact same energy, and equal to E_s. Another take on this: when you shift/rotate the constellation the performance, i.e., the probability of error, doesn't change, but the mean energy does; hence, E_s is not really *tied* to the probability of error, and Equation (2.3) is only meant to be used when the constellation is centered (the usual scenario).

Quick quiz

Where does thermal noise (the model we discussed in 1.8) fit in the above equations? [54]

2.11.2 Example: $M = 3, N = 1$, equally likely symbols, Gaussian noise

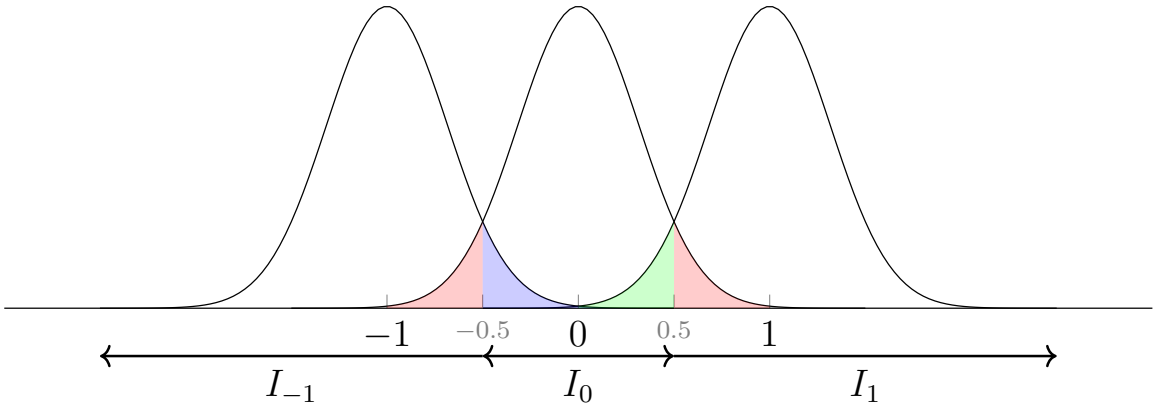
We have a 1D constellation with $M = 3$ symbols: -1, 0, and 1. Assuming equally likely symbols and Gaussian noise with variance $\frac{N_0}{2} = 1$, what is the probability of error?

We follow the same steps as before:

We compute the decision regions. Since the symbols are equally likely and the noise Gaussian, the rule is proximity. If these conditions didn't hold we should apply MAP or ML. In any case, it is useful to do a plot

[53] The power is the average of the energy.

[54] The term $\frac{N_0}{2}$ is the power of thermal noise.



We write the probabiliy of error in terms of the *conditional* probabilities of error.

$$P_e = P_{e|-1}p(-1) + P_{e|0}p(0) + P_{e|+1}p(+1) \overset{\substack{\text{eq. likely}\\ \text{symbols}}}{=} \frac{1}{3}\left(P_{e|-1} + P_{e|0} + P_{e|+1}\right)$$

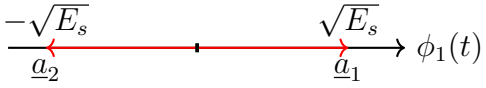
We compute the conditional error probabilities.

$$\begin{aligned}
P_{e|-1} &= p\left(q \mid A = -1 \notin I_{-1}\right) = p\left(q \mid A = -1 \in [0.5, \infty)\right) \\
&= \mathrm{Q}\left(\frac{|\text{threshold} - \text{mean}|}{\text{standard deviation}}\right) = \mathrm{Q}\left(\frac{|-0.5 - (-1)|}{1}\right) = \mathrm{Q}\left(0.5\right) \\
P_{e|0} &= \mathrm{Q}\left(\frac{|0 - (-0.5)|}{1}\right) + \mathrm{Q}\left(\frac{|0.5 - 0|}{1}\right) = 2\mathrm{Q}\left(0.5\right) \\
P_{e|1} &= \mathrm{Q}\left(\frac{|1 - 0.5|}{1}\right) = \mathrm{Q}\left(0.5\right)
\end{aligned}$$

Hence, the overall error probability is

$$P_e = \frac{1}{3}\left(P_{e|-1} + P_{e|0} + P_{e|+1}\right) = \frac{1}{3}\left(\mathrm{Q}\left(0.5\right) + 2\mathrm{Q}\left(0.5\right) + \mathrm{Q}\left(0.5\right)\right) = \frac{4}{3}\mathrm{Q}\left(0.5\right).$$

2.11.3 Example: $M = 2, N = 1$, NOT equally likely symbols, Gaussian noise

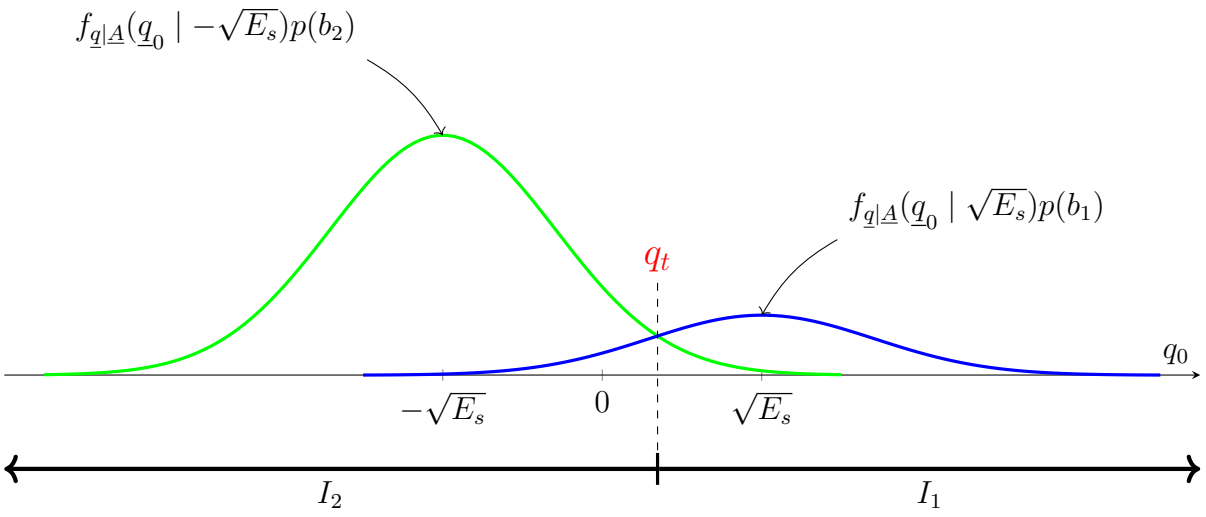


with

$$\begin{aligned}
p(b_1) &= p(\sqrt{E_s}) = 1 - p \\
p(b_2) &= p(-\sqrt{E_s}) = p
\end{aligned}$$

and $p > 0.5$ (the probability of b_2 is strictly larger than that of b_1). Assume the PSD of the noise is still $\frac{N_0}{2}$.

We compute the decision regions. Since the symbols are not equally likely we **cannot** apply the ML rule nor the proximity rule (it wouldn't be optimal), and we need to apply the MAP rule.



Therefore, it's not so clear anymore what are the decisions regions: we need to *mathematically* compute the decision threshold, q_t, the point at which both *functions* (they are not simply likelihoods) intersect/cross. For that purpose, we recall the mathematical expression we obtained in Equation (2.6) for the likelihood (notice that here the constellation is 1D, and hence $N = 1$).

$$
\begin{aligned}
f_{q|A}\left(q_t \mid -\sqrt{E_s}\right) p &= f_{q|A}\left(q_t \mid \sqrt{E_s}\right)(1-p) \\
&\Downarrow \\
\cancel{\frac{1}{(\pi N_0)^{\frac{1}{2}}}} e^{-\frac{(q_t+\sqrt{E_s})^2}{N_0}} p &= \cancel{\frac{1}{(\pi N_0)^{\frac{1}{2}}}} e^{-\frac{(q_t-\sqrt{E_s})^2}{N_0}}(1-p) \\
&\Downarrow \\
\frac{p}{1-p} &= e^{\frac{-(q_t-\sqrt{E_s})^2+(q_t+\sqrt{E_s})^2}{N_0}} \\
&\Downarrow \\
\log\frac{p}{1-p} &= \frac{-(q_t-\sqrt{E_s})^2+(q_t+\sqrt{E_s})^2}{N_0} \\
&\Downarrow \\
N_0 \log\frac{p}{1-p} &= -(\cancel{q_t^2}+\cancel{E_s}-2q_t\sqrt{E_s})+\cancel{q_t^2}+\cancel{E_s}+2q_t\sqrt{E_s} \\
&\Downarrow \\
N_0 \log\frac{p}{1-p} &= 4q_t\sqrt{E_s} \\
&\Downarrow \\
q_t &= \frac{N_0}{4\sqrt{E_s}} \log\frac{p}{1-p}
\end{aligned}
$$

We can add the threshold to the plot.

We write the probabiliy of error in terms of the conditional probabilities of error,

$$P_e = P_{e|-\sqrt{E_s}} p + P_{e|\sqrt{E_s}}(1-p).$$

We compute the conditional error probabilities in the usual way:

- $P_{e|-\sqrt{E_s}} = \mathrm{Q}\left(\frac{q_t+\sqrt{E_s}}{\sqrt{\frac{N_0}{2}}}\right)$
- $P_{e|+\sqrt{E_s}} = \mathrm{Q}\left(\frac{\sqrt{E_s}-q_t}{\sqrt{\frac{N_0}{2}}}\right)$

Hence,

$$P_e = \mathrm{Q}\left(\frac{q_t+\sqrt{E_s}}{\sqrt{\frac{N_0}{2}}}\right) p + \mathrm{Q}\left(\frac{\sqrt{E_s}-q_t}{\sqrt{\frac{N_0}{2}}}\right)(1-p).$$

⭐ Where the decision rule matters

Notice that the decision rule is only important to find the decision regions.

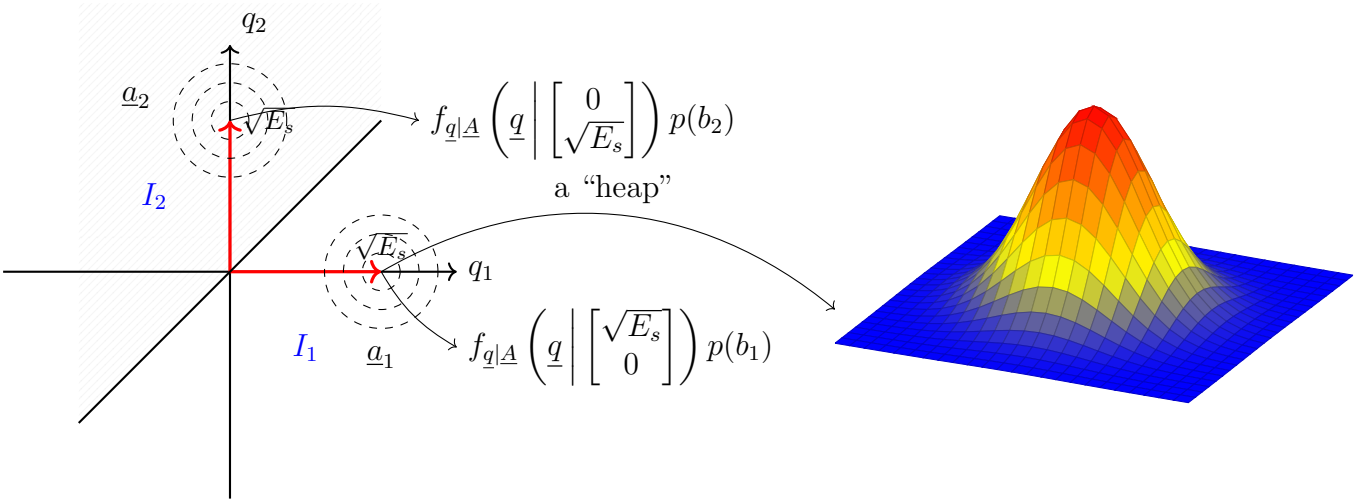
The constellation at the receiver

What matters when computing the probability of error is (prior probabilities aside) the distribution of the observations given each element in the constellation, i.e., $f_{\underline{q}|\underline{A}}(\cdot|\underline{a}_i)$, $i = 1, \cdots, M$. This implies that the important thing is not the original constellation we have at the transmitter, but the one we *observe* at the other end of the channel. Hence, for computing $P_{e|a_i}$, we are not concerned with the value of $\underline{a}_i$ (*at the transmitter*), but with the distribution of the observations (*at the receiver*) when $\underline{a}_i$ is transmitted. What happens is that in this course, most of the time, the latter distribution is centered at $\underline{a}_i$ (but it could be centered somewhere else).

2.12 Computation of the probability of error when $N > 1$

So far we have seen examples of computing the probability of error in one-dimensional constellations ($N = 1$). Let us now see a couple of characteristic examples when $N = 2$.

2.12.1 Example: $M = 2, N = 2$, equally likely symbols, Gaussian noise with $S_n(j\omega) = \frac{N_0}{2}$



This is one of the constellations for which we computed the mean energy and distance in Section 2.5.2: the corresponding signals were a sine and a cosine (they are orthogonal).

We follow the usual steps:

We compute the decision regions. The optimal decision rule is *proximity* because the symbols are equally likely and the noise is Gaussian. Hence, in order to get the decision regions we just draw the perpendicular bisector of the line that joins both symbols: the points to the left of this line are closer to $\underline{a}_2$ than to they are to $\underline{a}_1$, and vice versa. In any case we can plot *likelihood* $\times$ *prior* for every symbol to see what's going on. They are two-dimensional Gaussians, and we are watching them from above in the companion (left-hand-side) picture above.

The probability of error is written in terms of the conditional probabilities of error.

$$P_e = P_{e|b_1} p(b_1) + P_{e|b_2} p(b_2)$$

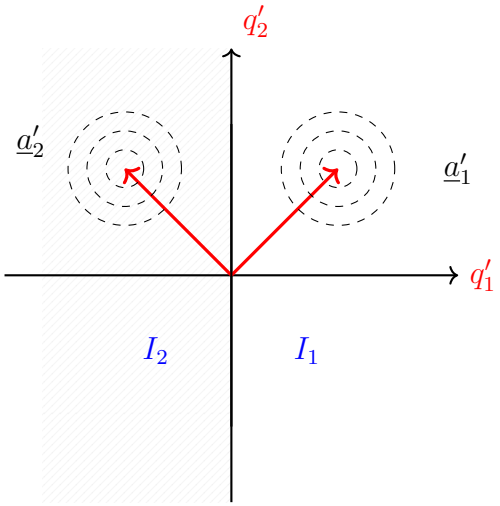
We compute the conditional error probabilities. First, we compute $P_{e|b_1}$, which is the probability of the observations vector $\underline{q}$ falling within the region ▨ when $\underline{A} = \left[\sqrt{E_s}, 0\right]$ was transmitted. We have

$$\underline{q}|\underline{A} = \left[\sqrt{E_s}, 0\right] \sim \mathcal{N}\left(\begin{bmatrix}\sqrt{E_s}\\ 0\end{bmatrix}, \frac{N_0}{2}\underline{\underline{I}}_2\right),$$

and, in order to compute the probability of error conditional on $\underline{A} = \left[\sqrt{E_s}, 0\right]$ we need to integrate the corresponding 2D pdf in the *incorrect* decision region, i.e., we need to compute the double integral

$$P_{e|\underline{a}_1} = \int_{\underline{q}\in I_2} f_{\underline{q}|\underline{A}}\left(\underline{q} \mid \underline{a}_1\right) d\underline{q} = \int_{-\infty}^{\infty}\int_{q_1}^{\infty} f_{q_1,q_2|A_{11},A_{12}}\left(q_1, q_2 \mid a_{11}, a_{12}\right) dq_2 dq_1,$$

This is complicated and the easy way to solve this problem is by rotating the axes (which amounts to a change of variable). If we do so, we get[55]



Notice that rotating/shifting the 2D Gaussians only affects their mean: ultimately, we are simply applying a known linear function to both of them.

With respect to the new axes, the integral becomes

$$P_{e|\underline{a}_1'} = \int_{-\infty}^{0}\int_{-\infty}^{\infty} f_{q_1',q_2'|A_1,A_2}\left(q_1', q_2' \mid a_{11}', a_{12}'\right) dq_2' dq_1',$$

and this is a double integral we can handle[56]. However, we don't have to: the decision threshold is now a vertical line, and hence to decide the region to which

[55] Notice that the axes are now q_1' and q_2'.

[56] We have *uncoupled* the integrals (the bounds are now fixed), and hence we can compute one after the other. The inner one amounts to *marginalization* (i.e., the law of total probability), and hence

$$\int_{-\infty}^{0}\int_{-\infty}^{\infty} f_{q_1',q_2'|A_1,A_2}\left(q_1', q_2' \mid a_{11}', a_{12}'\right) dq_2' dq_1' = \int_{-\infty}^{0} f_{q_1'|A_1}\left(q_1' \mid a_{11}'\right) dq_1'.$$

The result of the remaining (outer) integral can be obtained (as usual) by means of the Q function.

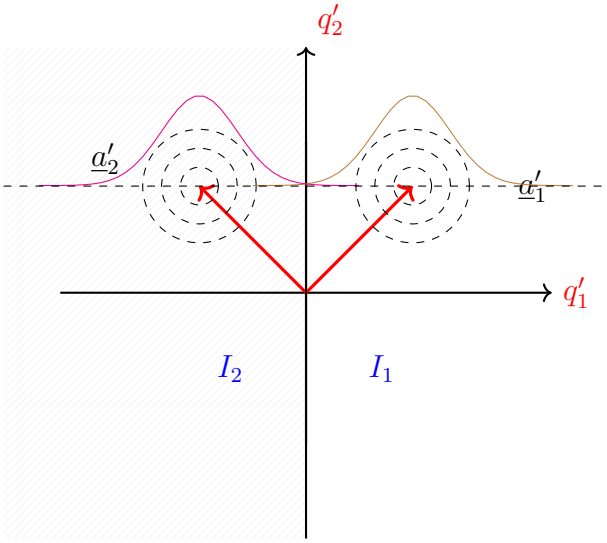
an observation $\underline{q}_0$ belongs, we only need to check the horizontal component. We know the (marginal) distribution, and hence the pdf, of any individual component in a random Gaussian vector. Indeed, given the joint pdf of $\underline{q}$ conditional on the element of the constellation transmitted,

$$\underline{q}'|\underline{A} = \underline{a}'_i \sim \mathcal{N}\left(\begin{bmatrix} a'_{i1} \\ a'_{i2} \end{bmatrix}, \begin{bmatrix} \frac{N_0}{2} & 0 \\ 0 & \frac{N_0}{2} \end{bmatrix}\right),$$

we know both marginals

$$q'_1|\underline{A} = \underline{a}'_i \sim \mathcal{N}\left(a'_{i1}, \frac{N_0}{2}\right)$$
$$q'_2|\underline{A} = \underline{a}'_i \sim \mathcal{N}\left(a'_{i2}, \frac{N_0}{2}\right),$$

i.e., every one of them is also Gaussian with mean and variance given by the corresponding elements in, respectively, the means vector and the covariance matrix[57]. The figure below extends the previous one by plotting the pdfs for the horizontal components of both 2D Gaussians.



The probability of error given $\underline{a}_1$ can then be computed as

$$P_{e|\underline{a}'_1} = \int_{-\infty}^{0} f_{q'_1|A_1}\left(q'_1 \mid a'_{11}\right) dq'_1$$

with $q'_1 \mid a'_{11} \sim \mathcal{N}\left(a'_{11}, N_0/2\right)$ (again, shifting the Gaussians only affects their means: $\underline{a}_1 \to \underline{a}'_1$ and $\underline{a}_2 \to \underline{a}'_2$), so that we are integrating a 1D Gaussian random variable whose pdf is known, and we can resort to the Q function

$$= Q\left(\frac{\sqrt{2E_s}}{2\sqrt{\frac{N_0}{2}}}\right).$$

[57] This is similar to what happens in a Gaussian process: you pick any time instants you like (a single one maybe) and the joint distribution of the corresponding random variables is still Gaussian.

Notice that both vectors, $\underline{a}_1$ and $\underline{a}_2$, after rotation, still have length $\sqrt{E_s}$. Then, Pythagoras theorem can be used in the above figure to compute the distance between them.

Due to the symmetry of the problem, we also have

$$P_{e|\underline{a}'_2} = Q\left(\frac{\sqrt{2E_s}}{2\sqrt{\frac{N_0}{2}}}\right),$$

and hence the final result is

$$P_e = Q\left(\frac{\sqrt{2E_s}}{2\sqrt{\frac{N_0}{2}}}\right)$$

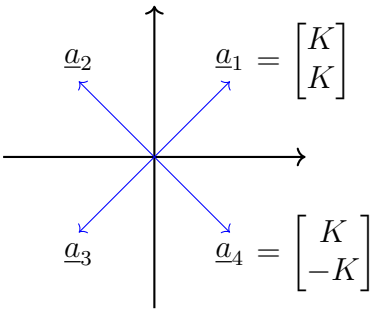
or, equivalently

$$= \mathrm{Q}\left(\frac{d(\underline{a}'_1, \underline{a}'_2)}{2\sqrt{\frac{N_0}{2}}}\right) = \mathrm{Q}\left(\frac{d(\underline{a}_1, \underline{a}_2)}{2\sqrt{\frac{N_0}{2}}}\right),$$

which is the result we got before for binary constellations with equally likely symbols and Gaussian noise.

From the beginning, we could have said: "well, this is a binary constellation, and we already know the probability of error in a binary constellation when the symbols are equally likely and the noise is Gaussian". However, this trick is useful in other situations!!

2.12.2 Example: 4-QAM, equally likely symbols, Gaussian noise

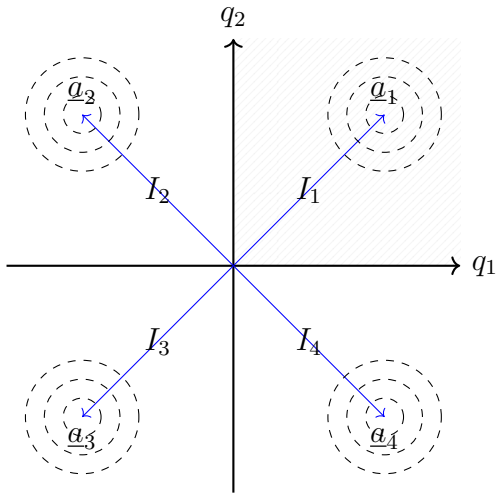


$$p(\underline{a}_1) = p(\underline{a}_2) = p(\underline{a}_3) = p(\underline{a}_4) = \frac{1}{4}$$

PSD of the noise is $S_n(j\omega) = \frac{N_0}{2}$

The usual steps:

W obtain the decision regions and, assuming Gaussian noise and equally likely symbols, the rule is *proximity*. This was one of the examples we used in 2.10.5 to illustrate the decision regions: the decision regions are the quadrants.



The (overall) probability of error is expressed in terms of the conditional error probabilities,

$$P_e = P_{e|\underline{a}_1} p(\underline{a}_1) + P_{e|\underline{a}_2} p(\underline{a}_2) + P_{e|\underline{a}_3} p(\underline{a}_3) + P_{e|\underline{a}_4} p(\underline{a}_4).$$

We compute the conditional probabilities as

$$P_{e|\underline{a}_1} = \underbrace{p(\underline{q} \mid \underline{A} = \underline{a}_1 \notin I_1)}_{\text{hard to compute}} = 1 - \underbrace{p(\underline{q} \mid \underline{A} = \underline{a}_1 \in I_1)}_{\substack{\text{it can be} \\ \text{computed}}}$$

How do we compute $p(\underline{q} \mid \underline{A} = \underline{a}_1 \in I_1)$? We know that

$$\underline{q}|\underline{A} = \underline{a}_1 \sim \mathcal{N}\left(\underline{a}_1, \frac{N_0}{2}\underline{I}_2\right) = \mathcal{N}\left(\begin{bmatrix} a_{i1} \\ a_{i2} \end{bmatrix}, \begin{bmatrix} \frac{N_0}{2} & 0 \\ 0 & \frac{N_0}{2} \end{bmatrix}\right),$$

which entails that the distributions for the *individual* components (the *marginal* distributions) are:

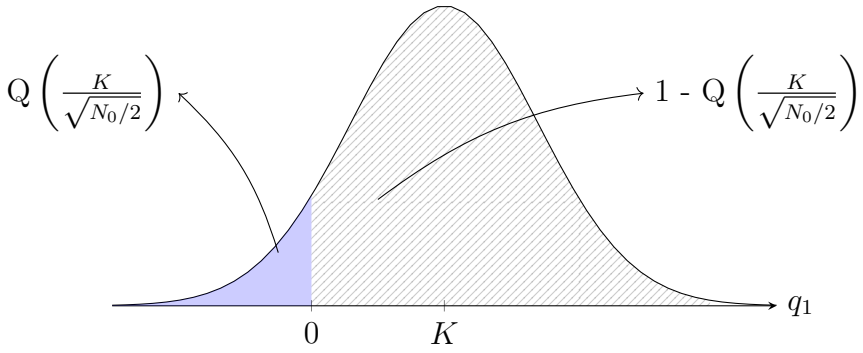
$$q_1|\underline{A} = \underline{a}_1 \sim \mathcal{N}\left(a_{11}, \frac{N_0}{2}\right) \overset{a_{11}=K}{=} \mathcal{N}\left(K, \frac{N_0}{2}\right)$$
$$q_2|\underline{A} = \underline{a}_1 \sim \mathcal{N}\left(a_{12}, \frac{N_0}{2}\right) \overset{a_{12}=K}{=} \mathcal{N}\left(K, \frac{N_0}{2}\right).$$

Moreover, these two random variables are independent (the covariance matrix is diagonal)![58] Hence, we can say

$$p(\underline{q} \in I_1) = p(q_1 > 0, q_2 > 0) = p(q_1 > 0)p(q_2 > 0),$$

and, for the first factor, we need to compute the probability of a Gaussian random variable with mean K and variance $N_0/2$ taking on values greater than 0. We know how to do this using the Q function. Notice that, once again, we end up working with one-dimensional Gaussians.

[58] In order to make the notation less cumbersome, from now on we are omitting the conditioning, $|\underline{a}_1$. It is understood (we focus on the probability of error conditional on $\underline{a}_1$).



Hence,

$$p(q_1 > 0) = 1 - \mathrm{Q}\left(\frac{K}{\sqrt{N_0/2}}\right)$$

...and we do the same for q_2 to get

$$p(q_2 > 0) = p(q_1 > 0) = 1 - \mathrm{Q}\left(\frac{K}{\sqrt{N_0/2}}\right).$$

Therefore,

$$p(\underline{q} \in I_1) = p(q_1 > 0)p(q_2 > 0) = \left(1 - \mathrm{Q}\left(\frac{K}{\sqrt{N_0/2}}\right)\right)^2,$$

and

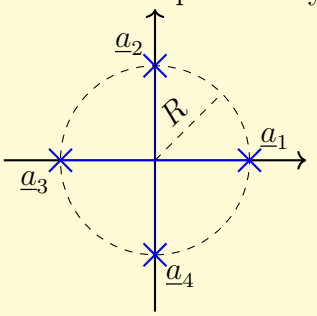
$$P_{e|\underline{a}_1} = 1 - \left(1 - \mathrm{Q}\left(\frac{K}{\sqrt{N_0/2}}\right)\right)^2.$$

We could do the same for the remaining symbols but, due to the symmetry of the problem, we are going to get the same result and, therefore,

$$P_e = P_{e|\underline{a}_1} = P_{e|\underline{a}_2} = P_{e|\underline{a}_3} = P_{e|\underline{a}_4} = 1 - \left(1 - \mathrm{Q}\left(\frac{K}{\sqrt{N_0/2}}\right)\right)^2.$$

? Quick quiz

What is the probability of error for the constellation below when $R = K\sqrt{2}$?



[59]

2.13 Bounds on the probability of error

In constellations with more than one dimension it is usually very difficult to compute the *exact*[60] probability of error (only in a few cases, like the ones we have just studied, this is feasible). In more complicated constellations, we resort to bounds: we compute an upper bound for the probability of error rather than the exact value (instead of saying, e.g., $P_e = 2.3 \cdot 10^{-6}$, we are going to say $P_e \leq 10^{-5}$).

2.13.1 The union bound

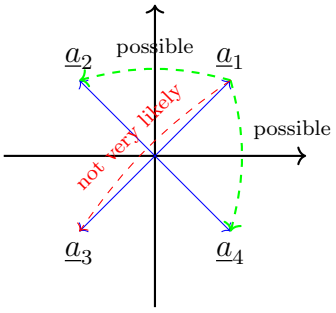
The first one is the union bound, which says that

$$P_e \leq \sum_{i=1}^{M} p(b_i) \sum_{\substack{j=1 \\ j \neq i}}^{M} \mathrm{Q}\left(\frac{d(\underline{a}_i, \underline{a}_j)}{2\sigma_n}\right)$$

where σ_n is the standard deviation of the noise (for us, it will be $\sqrt{N_0/2}$ most of the time).

2.13.2 The looser bound

Many $Q(\cdot)$ terms in the union bound will be very small (those in which the distance is large[61]), and hence their contribution to the probability of error negligible. The looser bound is a simplification of the union bound that assumes that only errors between adjacent symbols are possible, that is,



Here, when we transmit $\underline{a}_1$ we might decide $\underline{a}_2$ or $\underline{a}_4$ because they are the adjacent symbols, but we will hardly decide $\underline{a}_3$. In other words, most of the time we will only confuse a symbol with those that are the closest to it (there can be more than one). This means assuming a whole lot of errors are not possible.

For the looser bound we need a parameter which is...

[59] It is the same as before since this constellations is the previous one rotated 45°.

[60] Notice that, so far, we have been computing *exact* error probabilities.

[61] Remember that Q function is a (rapidly) decreasing function.

- $d_{min} = \min_{\substack{i,j=1,\cdots,M \\ j\neq i}} \left\{ d(\underline{a}_i, \underline{a}_j) \right\}$, that is, the minimum distance between any two elements in the constellation. So, in order to obtain this, we need to compute the (Euclidean) distance between any two elements in the constellation and, afterwards, choose the minimum.

Once we know d_{min}, and *assuming equally likely symbols*, the looser bound is given by

$$P_e \leq (M-1)\mathrm{Q}\left(\frac{d_{min}}{2\sigma_n}\right)$$

Union bound vs. Looser bound
The looser bound is much less accurate than the union bound.

2.14 *kiss number*-based approximation for the probability of error

If *symbols are equally likely*, the most common **approximation** (not a bound!!) of the probability of error is given by

$$P_e \approx \kappa \mathrm{Q}\left(\frac{d_{min}}{2\sigma_n}\right)$$

where

- σ_n is the standard deviation of the noise, and
- κ is the *kiss number*, which is defined as

$$\kappa = \max_{i=1}^{M} \{\#\text{ elements that are at distance } d_{min} \text{ from } \underline{a}_i\}.$$

Notice that we are computing the maximum of a *set*. In words, *kappa* is the maximum number of symbols that are at distance d_{min} from any given symbol. It's better seen through an example

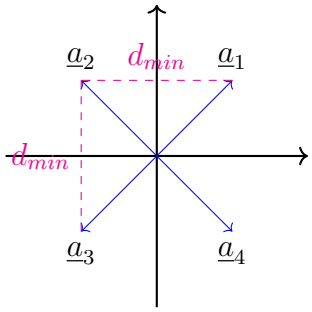
2.14.1 Example: computing κ in a constellation with 4 elements

For every element in the constellation we find out how many are at distance d_{min} from it. For instance, if

- $\underline{a}_1 \rightarrow$ has $\boxed{3}$ elements that are at d_{min} from it,
- $\underline{a}_2 \rightarrow$ has $\boxed{1}$ element that is at d_{min} from it,
- $\underline{a}_3 \rightarrow$ has $\boxed{2}$ elements that are at d_{min} from it, and
- $\underline{a}_4 \rightarrow$ has $\boxed{2}$ elements that are at d_{min} from it

then the *kiss number* is the maximum of 3, 1, 2, and 2, i.e., $\kappa = 3$.

2.14.2 Example: computing the *kiss number* in a 4-QAM



- $\underline{a}_1 \rightarrow$ has $\boxed{2}$ elements that are at d_{min} from it
- $\underline{a}_2 \rightarrow$ has $\boxed{2}$ elements that are at d_{min} from it
- $\underline{a}_3 \rightarrow$ has $\boxed{2}$ elements that are at d_{min} from it
- $\underline{a}_4 \rightarrow$ has $\boxed{2}$ elements that are at d_{min} from it

Hence, $\kappa = \max(\{2, 2, 2, 2\}) = 2$.

Validity of this approximation and the previous bounds

This approximation and the previous bounds are only valid when the **noise** is **Gaussian**[a]. Additionally, the looser bound and the *kiss number*-based approximation also assume equally likely symbols, though they could be generalized for elements with different probabilities.

[a] Q () is for integrating Gaussian pdfs!!

2.15 Connection between bits and symbols

If M is the number of elements in our constellation:

- The **number of bits per symbol** is

$$m = \log_2 M$$

- The (average) **bit energy** is defined as

$$E_b = \frac{E_s}{m}$$

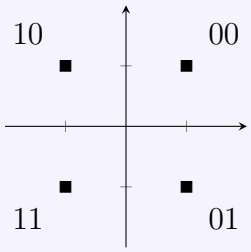
- We can also talk about bit error rate or BER, which is the probability of having an erroneous bit, and can be bounded

$$\frac{P_e}{m} \leq BER \leq P_e$$

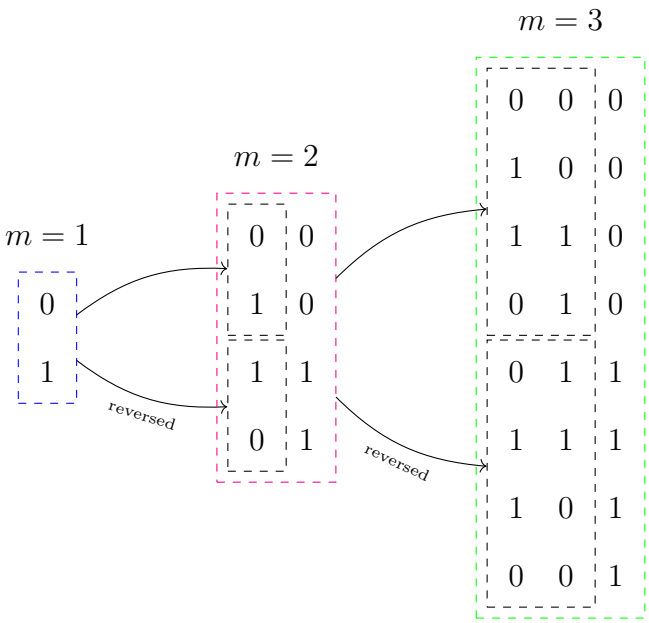
2.15.1 Gray mapping

When assigning sequences of bits to elements in the constellation, it is a good thing to assign sequences of bits that only differ in one bit to adjacent elements in the constellation. That way, a *symbol error* will give rise to a single *bit error*. Keep in mind that we are assuming that errors between adjacent elements in the constellation are far more likely that those between elements far away.

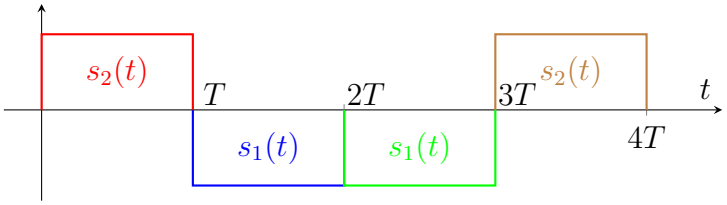
Example: $M = 4, N = 2$



There is a systematic way to build the list of sequences for a given number of bits per symbol, m, giving the list for $m - 1$. That along with the fact that Gray mapping for $m = 1$ is straightforward allows to build the list of sequences for any value of m:



2.16 Transmission of a sequence of symbols



When transmitting a sequence of symbols, we can talk about **symbol rate**, which is the number of symbols (from the constellation) transmitted per second. It is the inverse of the symbol period, which is the time taken to transmit a single symbol (the duration of the signals in our modulator),

$$R_s = \frac{1}{T} \frac{\text{symbols}}{\text{second}}.$$

The units, $\frac{\text{symbols}}{\text{second}}$, are sometimes referred to as *bauds*.

Likewise we can define a bit rate, which is the number of bits transmitted per second (think your internet connection!!). It is connected with the symbol rate through

$$R_b = m \cdot R_s.$$

Chapter 3

Fundamental limits in communications

3.1 Information

We need two things to measure the performance of a system using information:

- a (probabilistic) channel model, and
- a *quantitative* measure of information.

We have seen the first one, and now we focus on the second one: how do we measure information? Notice that we would like a quantitative measure, that is, a number!!

Intuitively, an unusual event, **when it happens**, bears a lot of information, whereas an event that occurs very often does not (we expect it to happen).

Example: to rain or not to rain
The event "it's now raining in the Sahara desert" bears a lot of information: it doesn't happen very often (it is highly unlikely). However, the event "it's now raining in the north of Spain" does not provide that much information. That is very common (highly likely!!).

Example: Playing roulette
Foretelling the number is very different than foretelling the color... If I tell you what *number* is gonna come up next, I'm giving you a lot of information (you have a lot of uncertainty about that). On the other hand, if I tell which *color* is gonna come next, that's much less information (odds are 50-50).

So, the more likely an event is, the less information it bears (you were expecting it to happen), i.e.,

$$\uparrow \text{probability} \Rightarrow \downarrow \text{information},$$

and, conversely, the more unlikely (surprising) an event is, the larger the amount of information it provides, i.e.,

$$\downarrow \text{probability} \Rightarrow \uparrow \text{information}.$$

We want a metric for information that is consistent with this, i.e., a metric that will associate a large amount of information to low-probability events, and a small quantity of information to high-probability events.

Definition 3.1.1: Information

Given a random experiment characterized by a *discrete* random variable X, such that,

$$X \in \{x_1, x_2, \cdots, x_M\}$$

with known $p(x_i), i = 1 \cdots, M$, the information associated with x_i (or *autoinformation* of x_i) is

$$I_{x_i} = \log_2 \frac{1}{p(x_i)} = \log_2 1 - \log_2 p(x_i) = -\log_2 p(x_i) \textbf{ bits}.$$

Notice the information, I_{x_i}, is a number.

Where does the above formula come from?

Why $\frac{1}{p(x_i)}$? We want the information to decrease as the probability increases.

Why logarithm? Because if two events are independent, then the information they provide jointly should be the sum of information provided by each one of them.

Let x and y denote two independent events (for example, different tosses of a coin, or realizations of independent random variables). The information provided by the joint event (x, y) is

$$I_{xy} = \log_2 \frac{1}{p(x,y)} \overset{\text{indep.}}{=} \log_2 \frac{1}{p(x)p(y)} = \log_2 \frac{1}{p(x)} + \log_2 \frac{1}{p(y)} = I_x + I_y$$

Why is 2 the base of the logarithm? It doesn't have to. Sometimes, natural logarithm is used and, in such case, the units are *nats*.

$$I_{x_i} = \log_2 \frac{1}{p(x_i)} \text{ bits}$$
$$I_{x_i} = \log \frac{1}{p(x_i)} \text{ nats}$$

Example: computing autoinformation

X is a r.v. whose sample space is given by $\Omega = \{A, B, C, D\}$, i.e.

$$X \in \{A, B, C, D\}.$$

The information of every possible value of the r.v. is computed as

$$\begin{array}{ll} I_A = \log_2 \frac{1}{p(A)}, & I_B = \log_2 \frac{1}{p(B)} \\ I_C = \log_2 \frac{1}{p(C)}, & I_D = \log_2 \frac{1}{p(D)} \end{array}.$$

If all the values are equally likely, i.e., $p(A) = p(B) = p(C) = p(D)$,

$$I_A = I_B = I_C = I_D = \log_2 4 = 2 \text{ bits}.$$

We are now in position to compute the information associated with any event, but we also care about the amount of information in a random variable or, equivalently, a source of information[1].

? Quick quiz

What do you need to compute the information provided by a certain event happening?
[2]

3.2 Entropy

In order to measure the information in a random variable, we would like to account for all the possible values it can take. Hence, it seems reasonable to define the information in a random variable as the expectation of the information over all the possible values it can take. This quantity for a random variable X is known as the **entropy** of X and it's denoted as $H(X)$.

Definition 3.2.1: Entropy

Let X be a random variable such that

$$X \in \{x_1, x_2, \cdots, x_M\},$$

then the entropy of X is defined as

$$H(X) = \sum_{i=1}^{M} \underbrace{p(x_i)}_{\text{prob.}} \underbrace{\log_2 \frac{1}{p(x_i)}}_{\text{info.}} = \sum_{i=1}^{M} -p(x_i) \log_2 p(x_i) = -\sum_{i=1}^{M} p(x_i) \log_2 p(x_i) \frac{\text{bits}}{\text{symbol}}$$

Notice that

- the units are bits per symbol[3],
- the entropy is a number, **not** a random variable,
- using **L'Hopital** it can be shown that

$$0 \log_2 0 = 0.$$

[1] Even though a source of information is a sequence of random variables, for us they all have the same distribution, and hence any one of them is enough to characterize the source.

[2] You "only" need the probability of that event.

[3] This is an average: you multiply the autoinformation of a value by its probability and you sum over all the possible values.

Example: computing the entropy

Consider a r.v. $X \in \{0, 1, 2, 3\}$ with

$$\begin{array}{ll} p(X=0) = 1/2, & p(X=2) = 1/4 \\ p(X=1) = 1/4, & p(X=3) = 0 \end{array} .$$

The entropy is given by

$$H(X) = -\sum_{i=1}^{M} p(x_i) \log_2 p(x_i) \overset{\text{M=4}}{=} -\Big(1/2 \log_2 1/2 + 1/4 \log_2 1/4 + 1/4 \log_2 1/4 + \cancelto{0}{0 \log_2 0}\Big)$$

$$= -\left(1/2(-1) + 1/4(-2) + 1/4(-2)\right) = 1/2 + 1/2 + 1/2 = 3/2 \, \frac{\text{bits}}{\text{symbol}}$$

A related example: different sampling space, same probabilities

Consider a r.v. $X \in \{a, b, c, d\}$ with

$$\begin{array}{ll} p(X=a) = 1/4, & p(X=c) = 1/2 \\ p(X=b) = 1/4, & p(X=d) = 0 \end{array} .$$

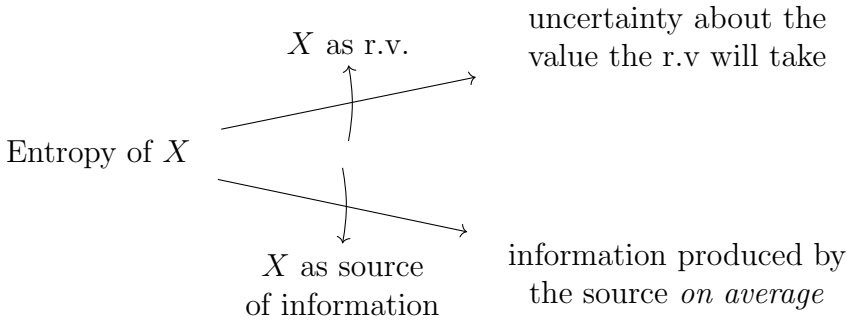
The entropy is given by

$$H(X) = -\Big(1/4 \log_2 1/4 + 1/4 \log_2 1/4 + 1/2 \log_2 1/2 + \cancelto{0}{0 \log_2 0}\Big) = 3/2 \, \frac{\text{bits}}{\text{symbol}}$$

The values the random variable takes on make no difference in the computation of the entropy (only their probabilities!!). So, two different random variables with the same distribution have the same entropy.

3.2.1 Interpretation of the entropy

There are two possible interpretations for the entropy. When talking about random variables, the entropy may be interpreted as the *uncertainty* about the value that the random variable is going to take. On the other hand, when we interpret X as a source of information, the entropy is the *information* the source produces on average:



Notice that we interpret the entropy as either information or uncertainty. Interpretation often comes handy when we need to compute the entropy of a certain variable.

Example: entropy of an uncertainty-less source

Let us consider a source of information (modeled as a random variable), X, taking values in the set $\{x_1, x_2, \cdots, x_M\}$, and we have $p(x_2) = 1$, and $p(x_i) = 0, \forall i \neq 2$, then

$$H(X) = -\left(0\log_2 0 + 1\log_2 1 + 0\log_2 0 + \cdots + 0\log_2 0\right) = 0\,\frac{\text{bits}}{\text{symbol}}.$$

This means the source does not produce any information...or if we think of X as a r.v., there is no uncertainty.

Quick quiz

What is the entropy of a *constant* (say, e.g., 5)? [4]

3.2.2 Properties of the entropy

If $X \in \{x_1, x_2, \cdots, x_M\}$ is a *discrete* random variable[5], we have

- $H(X) \geq 0$ since

$$0 \leq \boxed{p(x_i) \leq 1}$$
$$\Downarrow$$
$$\frac{1}{p(x_i)} \geq 1 \Leftrightarrow \log_2 \frac{1}{p(x_i)} \geq 0$$

  where it must be taken into account that $\log_2$ is a monotonically increasing function, and hence computing the inverse on both sides of the inequality doesn't change the *direction* thereof.

- $H(X) \leq \log_2 M$ and, moreover,

$$H(X) = \log_2 M \Leftrightarrow p(x_i) = \frac{1}{M},$$

  the entropy is maximum when the symbols are equally likely.

[4] You can think of a constant as a random variable taking a certain value with probability 1 and any other value with probability 0. This is exactly the above example. Hence, the entropy is zero.

[5] For the time being we are focusing on discrete random variables.

3.3 Binary entropy function

Let us consider the entropy of a binary random variable which can only take two values: a and b,

$$X \in \{a, b\}$$

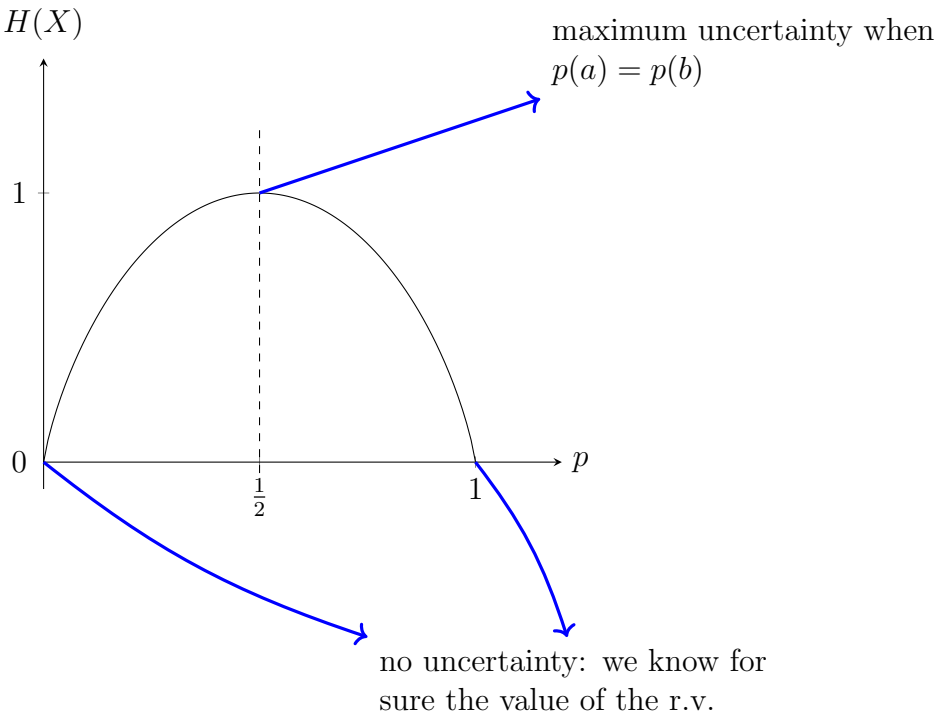
with probabilities

$$p(X = a) = p$$
$$p(X = b) = 1 - p.$$

The entropy is given by

$$\begin{aligned} H(X) &= -p(X = a) \log_2 p(X = a) - p(X = b) \log_2 p(X = b) \\ &= -p \log_2 p - (1 - p) \log_2 (1 - p). \end{aligned}$$

$H(X)$ is a function of just one variable, p, and we can easily plot it



Since the values of the random variable don't matter (only the probabilities), this is valid for *any* **binary** random variable. The entropy of a binary r.v. shows up often when computing the entropies of other (non-binary) random variables. That's why we have a notation for it

Definition 3.3.1: Binary entropy function

$$H_b(p) = -p \log_2 p - (1 - p) \log_2 (1 - p).$$

Hence, whenever we have the entropy of a binary random variable, we will resort to this notation. Notice that the argument of H_b is **not** the random variable, but rather the probability of one of the two symbols (any of them).

Binary entropy function is symmetric around $1/2$
An important property of the binary entropy function is that

$$H_b(p) = H_b(1-p),$$

i.e., the function is symmetric around $1/2$.

Also, from the above plot, we can see that

$$\max\{H_b(p)\} = 1$$

for $p = 1/2$. This is just a particular case of the last property in Section 3.2.2.

3.4 Joint entropy

The definition of entropy can be extended to more than one variable. Let us consider random variables X and Y such that

$$X \in \{x_1, x_2, \cdots, x_M\}, \text{with known } p(x_i), i = 1, \cdots, M$$
$$Y \in \{y_1, y_2, \cdots, y_L\}, \text{with known } p(y_j), j = 1, \cdots, L$$

i.e., they have different alphabets and probabilities. We define the joint entropy of X and Y as

Definition 3.4.1: Joint entropy

$$H(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(x_i,y_j)} = -\sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2 p(x_i,y_j)\ \frac{\text{bits}}{\text{symbol}}.$$

We are simply averaging the information of all the *joint* events. It measures the information associated with the random experiment characterized by those two random variables.

If variables X and Y are independent, then we have

$$p(x_i,y_j) = p(x_i)p(y_j),$$

and

$$
\begin{aligned}
H(X,Y) &= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(x_i,y_j)} = \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i)p(y_j)\log_2\frac{1}{p(x_i)p(y_j)} \\
&= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i)p(y_j)\left(\log_2\frac{1}{p(x_i)} + \log_2\frac{1}{p(y_j)}\right) \\
&= \sum_{i=1}^{M}\sum_{j=1}^{L}\left(p(x_i)p(y_j)\log_2\frac{1}{p(x_i)} + p(x_i)p(y_j)\log_2\frac{1}{p(y_j)}\right) \\
&= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i)p(y_j)\log_2\frac{1}{p(x_i)} + \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i)p(y_j)\log_2\frac{1}{p(y_j)} \\
&= \sum_{i=1}^{M} p(x_i)\log_2\frac{1}{p(x_i)}\cancel{\sum_{j=1}^{L} p(y_j)}^{1} + \cancel{\sum_{i=1}^{M} p(x_i)}^{1}\sum_{j=1}^{L} p(y_j)\log_2\frac{1}{p(y_j)} \\
&= \underbrace{\sum_{i=1}^{M} p(x_i)\log_2\frac{1}{p(x_i)}}_{H(X)} + \underbrace{\sum_{j=1}^{L} p(y_j)\log_2\frac{1}{p(y_j)}}_{H(Y)} = H(X)+H(Y)
\end{aligned}
$$

Furthermore, we know the above final result is the maximum value $H(X,Y)$ can take, i.e., the joint entropy of two random variables satisfies

$$H(X,Y) \leq H(X) + H(Y),$$

because if the variables are not independent, then they are related and there is some common information both provide[6]. Intuitively, there is some intersection between the information provided by one variable and that provided by the other, and hence you don't expect their joint entropy to be the sum. This is related to the concept of conditional entropy.

? Quick quiz

How much information do you get by tossing a fair coin twice? [7]

3.5 Conditional entropy

We consider the same setup as before, i.e.,

$$
\begin{aligned}
&X \in \{x_1, x_2, \cdots, x_M\}, \text{with known } p(x_i), i = 1, \cdots, M \\
&Y \in \{y_1, y_2, \cdots, y_L\}, \text{with known } p(y_j), j = 1, \cdots, L.
\end{aligned}
$$

[6] As an example, if you know the height of a person, you can approximately estimate its weight. Then the information provided by both variables shouldn't be the sum of the individual informations.

[7] Since the coin is fair, you have a uniform distribution, meaning the entropy/information you get in a toss is $\log_2 2 = 1$ bit. Now, since every toss is independent, the information in two tosses is the sum of the information in the individual tosses, i.e., two.

Definition 3.5.1: Conditional entropy

Let X and Y be two random variable characterizing a random experiment. The conditional entropy of X given Y is defined as

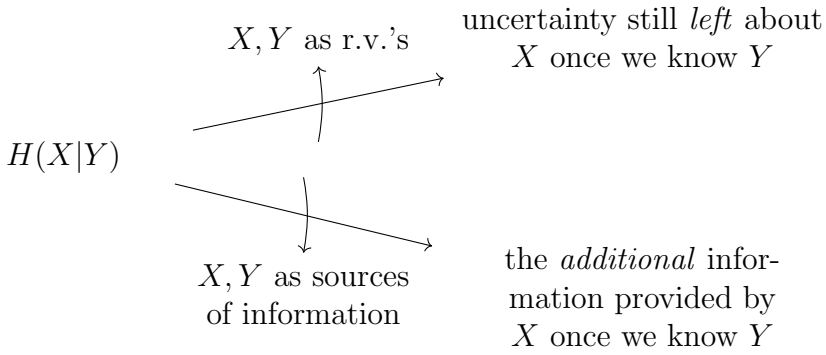
$$\begin{aligned} H(X|Y) &= \sum_{j=1}^{L} p(y_j) H(X|Y=y_j) \\ &= \sum_{j=1}^{L} p(y_j) \sum_{i=1}^{M} p(X=x_i|Y=y_j) \log_2 \frac{1}{p(X=x_i|Y=y_j)} \\ &= \sum_{j=1}^{L} \sum_{i=1}^{M} p(x_i, y_j) \log_2 \frac{1}{p(X=x_i|Y=y_j)} \frac{\text{bits}}{\text{symbol}}. \end{aligned}$$

Notice that $X|Y=y_j$ is just a regular random variable[a], and hence we can compute its entropy in the usual way: the weighted average of the information over all the possible values it can take.

[a] ...otherwise, we would be defining the conditional entropy using the conditional entropy!!

In other words, the conditional entropy is the expected entropy of a random variable (X above) conditional on the value of another random variable (Y above)[8].

Again, we have two interpretations



It will be very useful. For example, if X and Y are independent, then

$$H(X|Y) = H(X),$$

and this can be inferred using the above two interpretations:

- The uncertainty of X after knowing Y is the same we had before (the uncertainty we have about X is given by $H(X)$). Since the variables are independent we have not removed any uncertainty.
- If we know Y, what *additional* information does X provide? The same it provided before knowing Y, which is $H(X)$.

[8] The expectation is computed with respect to the latter.

The joint and conditional entropy are related[9]

$$
\begin{aligned}
H(X,Y) &= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(x_i,y_j)} = \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(y_j|x_i)p(x_i)} \\
&= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\left(\log_2\frac{1}{p(y_j|x_i)} + \log_2\frac{1}{p(x_i)}\right) \\
&= \underbrace{\sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(y_j|x_i)}}_{H(Y|X)} + \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(x_i)} \\
&= H(Y|X) + \sum_{i=1}^{M}\log_2\frac{1}{p(x_i)}\underbrace{\sum_{j=1}^{L} p(x_i,y_j)}_{p(x_i)} = H(Y|X) + \underbrace{\sum_{i=1}^{M}\log_2\frac{1}{p(x_i)}p(x_i)}_{H(X)} \\
&= H(Y|X) + H(X)
\end{aligned}
$$

Likewise we can show that

$$H(X,Y) = H(X|Y) + H(Y).$$

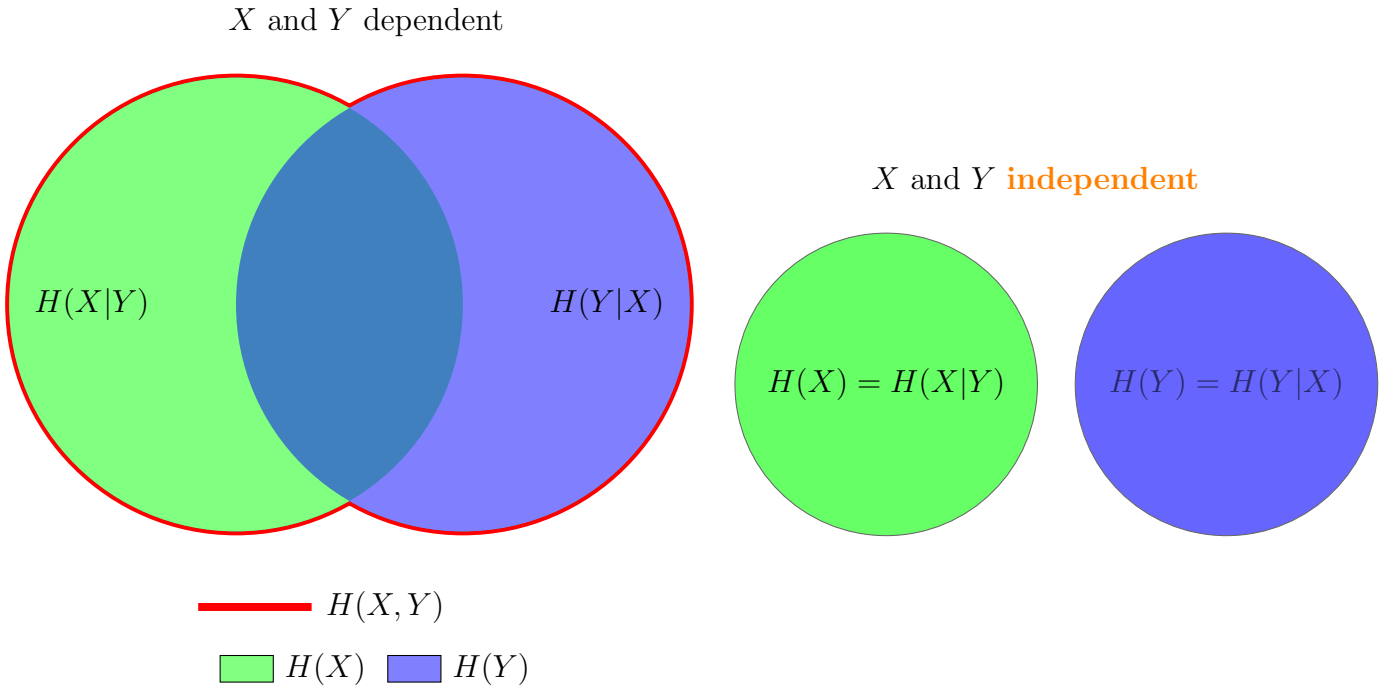
Joint entropy and conditional entropy

In summary, we have

$$H(X,Y) = H(X|Y) + H(Y) = H(Y|X) + H(X) \tag{3.1}$$

A nice way of looking at this connection is using Venn diagrams. We represent every entropy as a set, and the joint entropy as the union. Let us consider two different scenarios: either X and Y are dependent, or they are independent.

[9] We are not making any assumption here, and hence this equality **always** holds.



Joint entropy is the *union* of the sets

We must interpret $H(X,Y)$ as the union of the entropies of X and Y (despite the fact that in statistics $p(A,B)$ refers to the intersection of the probabilities of A and B.)

From Equation (3.1) relating conditional and joint entropy, we can infer what are the pieces to the left and right of the intersection:

if we remove $H(Y)$ from the union...

$$H(X,Y) = H(X|Y) + H(Y) \Rightarrow H(X,Y) - H(Y) = H(X|Y)$$

and, if we remove $H(X)$ from the union...

$$H(X,Y) = H(Y|X) + H(X) \Rightarrow H(X,Y) - H(X) = H(Y|X)$$

Notice that we still don't have a name for the intersection.

Quick quiz

What is the entropy of the random variable "room temperature in degrees Kelvin given the room temperature in degrees celsius"? [10]

[10] It's zero: one variable determines the other, i.e., once you know the temperature in degrees celsius/Kelvin, you don't have any uncertainty about the temperature in Kelvin/celsius. In a Venn diagram representation both *sets*, the one for "room temperature in degrees Kelvin" and that for "room temperature in degrees celsius" would exactly overlap.

3.5.1 Probabilistic channel model

We can apply what we have learned so far about information theory to our (probabilistic) channel model. Assume that, as usual, X is the input and Y the output,

- If X and Y are independent

$$H(X|Y) = H(X),$$

that is, knowing Y does not provide any new information about X, which means, the uncertainty we have about X is the same we had before knowing Y. In other words, the transmission was useless.

- If $X = Y$

$$H(X|Y) = H(Y|X) = 0$$

  - what is the uncertainty left about X after knowing Y?...none
  - what is the information provided by X if we know Y?...none

Furthermore, we have that,

$$H(X) = H(Y) = H(X, Y).$$

In this case, the channel is *perfect* (no distortion or disturbance).

3.6 Mutual information

It is a concept complementary to conditional entropy. Given the usual setup

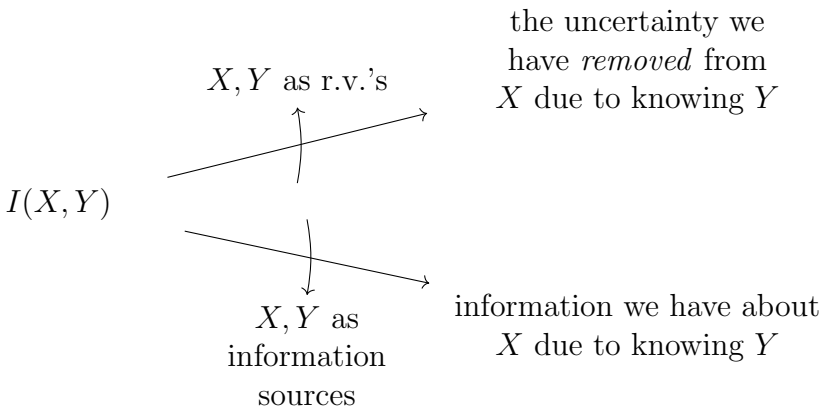
$$X \in \{x_1, x_2, \cdots, x_M\}, \text{with known } p(x_i), i = 1, \cdots, M$$
$$Y \in \{y_1, y_2, \cdots, y_L\}, \text{with known } p(y_j), j = 1, \cdots, L,$$

the formal definition is

Definition 3.6.1: Mutual information

$$I(X, Y) = \sum_{i=1}^{M} \sum_{j=1}^{L} p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(x_i)p(y_j)} \frac{\text{bits}}{\text{symbol}}$$

We can think about the mutual information in two different ways



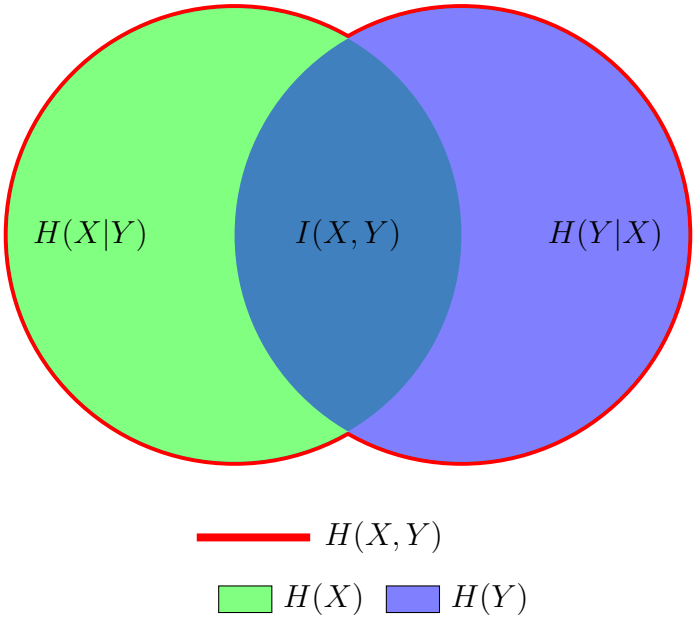
Mutual information and entropy are related

$$
\begin{aligned}
I(X,Y) &= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{p(x_i,y_j)}{p(x_i)p(y_j)} = \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{p(x_i|y_j)\cancel{p(y_j)}}{p(x_i)\cancel{p(y_j)}} \\
&= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\left(\log_2\frac{1}{p(x_i)} + \log_2 p(x_i|y_j)\right) \\
&= \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(x_i)} + \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2 p(x_i|y_j) \\
&\overset{\log_2 x = -\log_2\frac{1}{x}}{=} \sum_{i=1}^{M}\log_2\frac{1}{p(x_i)}\underbrace{\sum_{j=1}^{L} p(x_i,y_j)}_{p(x_i)} - \sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(x_i|y_j)} \\
&= \underbrace{\sum_{i=1}^{M} p(x_i)\log_2\frac{1}{p(x_i)}}_{H(X)} - \underbrace{\sum_{i=1}^{M}\sum_{j=1}^{L} p(x_i,y_j)\log_2\frac{1}{p(x_i|y_j)}}_{H(X|Y)} = H(X) - H(X|Y)
\end{aligned}
$$

Likewise, it can be shown that

$$I(X,Y) = H(Y) - H(Y|X). \tag{3.2}$$

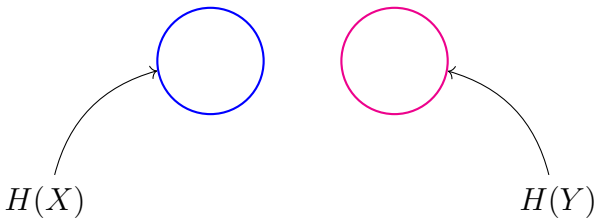
If we go back to the graphical representation based on Venn diagrams,



we can see that the mutual information is the **intersection** between the entropies of X and Y. If we keep in mind this diagram we can easily deduce any formula we need!!

Two special situations:

- If X and Y are independent

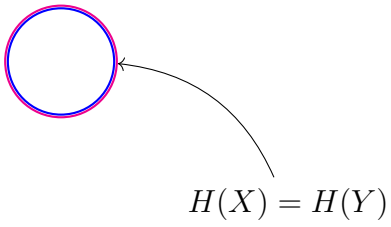


the mutual information is 0, and it can be easily proved mathematically:

$$I(X,Y) = H(X) - H(X|Y) \overset{H(X|Y)=H(X)}{=} H(X) - H(X) = 0,$$

in other words, Y does not provide any information about X.

- If $X = Y$



the mutual information is

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \overset{H(X|Y)=0}{=} H(X) - 0 \\ &= H(Y) - H(Y|X) = H(Y) \end{aligned}$$

? Quick quiz

What is the mutual information between two tosses of a dice? [11]

3.6.1 Properties

- $I(X,Y) = I(Y,X) \geq 0$

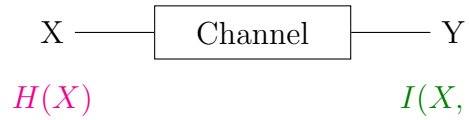
  This symmetry ($I(X,Y) = I(Y,X)$) of the mutual information goes to say that the information between X and Y flows both ways: if X provides a certain amount of information about Y, then that same amount of information is provided by Y about X.

- $I(X,Y) \leq H(X)$
- $I(X,Y) \leq H(Y)$

All these properties can be easily inferred from the Venn diagram.

3.6.2 Mutual information from the standpoint of a communications system

If X is the symbol transmitted and Y is what we get at the other end of the channel,



then $H(X)$ is the information we have at the input of the channel[12] and $I(X,Y)$ is the information at the output since it is "the information we have about X due to knowing Y". In other words, it is the amount of information that gets through the channel. A couple of remarks are in order here: when we say "information that gets through the channel" we mean

- *error-free information*, that is, if we say that the mutual information is, e.g., $2\frac{\text{bits}}{\text{symbol}}$ (recall mutual information is measured in bits per symbol), those 2 bits reach the receiver with vanishing probability of error[13], i.e., $P_e \to 0$,
- *per channel use*, that is, every time we use the channel, that's the amount of information received at the other end.

Hence, the mutual information is the amount of error-free information (measured in bits) that can get through per channel use.

Recall that in this module we are studying another way (other than the probability of error) of measuring the performance of a communications system...this is it!!

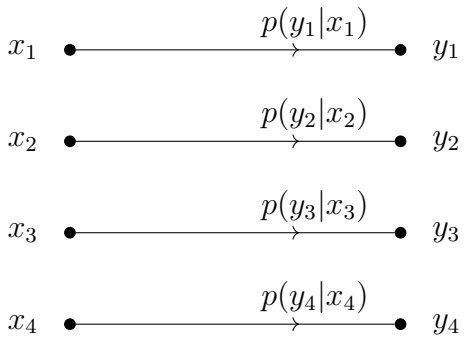
[11] (the tosses are independent).
It's zero: the first toss will not remove any uncertainty about what's gonna happen in the second toss

[12] This is one of the interpretations of the entropy.

[13] i.e., an error probability as small as we like, say, 10^{-2000}

3.6.3 Example I

Let us consider the DMC



Notice that the probability of transmitting a symbol and receiving a different one is 0, and hence

$$p(y_i|x_i) = 1$$
$$p(y_j|x_i) = 0, \ \forall i \neq j.$$

Caveat

When there is only one branch leaving a node, then its label (which is the probability conditional on that node), must be 1[a]. For that reason, it is usually omitted.

[a] Recall that the sum of the labels on the branches *leaving* a node is 1.

Additionally, we assume $p(x_1)$, $p(x_2)$, $p(x_3)$ and $p(x_4)$ are known.

What is happening here is: $X \in \{x_1, x_2, x_3, x_4\}$ is transmitted, $Y \in \{y_1, y_2, y_3, y_4\}$ is received, but there is one-to-one mapping between the x's and the y's, and hence every time you receive a y you know perfectly well which x was sent, and we have an error-free transmission. In other words, all the information put at the input reaches the output. Let us verify this using what we have learned so far.

The entropy of X is given by

$$H(X) = \sum_{i=1}^{4} p(x_i) \log_2 \frac{1}{p(x_i)}.$$

Since $p(y_i) = p(x_i)$ we have

$$H(Y) = \sum_{i=1}^{4} p(y_i) \log_2 \frac{1}{p(y_i)} = \sum_{i=1}^{4} p(x_i) \log_2 \frac{1}{p(x_i)} = H(X).$$

Remember that entropy only depends on the probabilities (the distribution), **not** on the values taken by the random variable.

Next, we are going to compute the mutual information using the definition

$$
\begin{aligned}
I(X,Y) &= \sum_{i=1}^{4}\sum_{j=1}^{4} p(x_i,y_j)\log_2\frac{p(x_i,y_j)}{p(x_i)p(y_j)} \\
&= \sum_{i=1}^{4}\sum_{j=1}^{4} p(y_j|x_i)p(x_i)\log_2\frac{p(y_j|x_i)\cancel{p(x_i)}}{\cancel{p(x_i)}p(y_j)} \\
&\overset{\substack{p(y_j|x_i)=0\\ \forall j\neq i}}{=} \sum_{i=1}^{4} p(y_i|x_i)p(x_i)\log_2\frac{p(y_i|x_i)}{p(y_i)} \\
&\overset{p(y_i|x_i)=1}{=} \sum_{i=1}^{4} p(x_i)\log_2\frac{1}{p(y_i)} \\
&\overset{p(y_i)=p(x_i)}{=} \sum_{i=1}^{4} p(x_i)\log_2\frac{1}{p(x_i)} = H(X) = H(Y)
\end{aligned}
$$

The information at the output of the channel is the same we had at the input!! We could have inferred this using the interpretation we have been talking about all along:

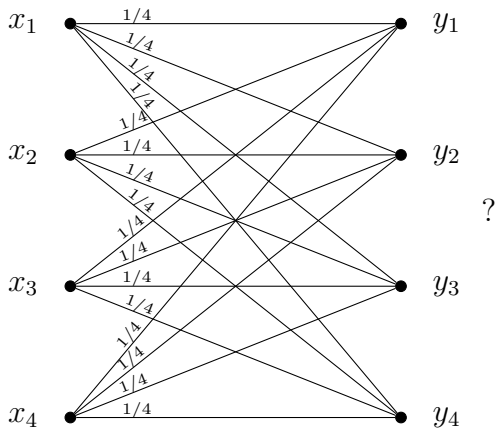
$$I(X,Y) = H(X) - H(X|Y),$$

and $H(X|Y)$ is 0 because there is no uncertainty about Y if we know X, so that

$$I(X,Y) = H(X) - \cancel{H(X|Y)}^{0} = H(X).$$

3.6.4 Example II

What happens if, instead of the previous DMC, we had



Notice that, in this case, $p(y_j|x_i) = \frac{1}{4}\ \forall i,j$, i.e., regardless of the symbol we transmit we receive any of the output symbols with equal probability. According to the definition,

the mutual information is given by

$$\begin{aligned} I(X,Y) &= \sum_{i=1}^{4}\sum_{j=1}^{4} p(x_i,y_j)\log_2\frac{p(x_i,y_j)}{p(x_i)p(y_j)} \\ &= \sum_{i=1}^{4}\sum_{j=1}^{4} p(y_j|x_i)p(x_i)\log_2\frac{p(y_j|x_i)\cancel{p(x_i)}}{\cancel{p(x_i)}p(y_j)} \\ &= \sum_{i=1}^{4}\sum_{j=1}^{4} \frac{1}{4}p(x_i)\log_2\frac{1/4}{p(y_j)}, \end{aligned} \tag{3.3}$$

and we need

$$p(y_j) = \sum_{i=1}^{4} p(y_j,x_i) = \sum_{i=1}^{4} p(y_j|x_i)p(x_i) = \frac{1}{4}\sum_{i=1}^{4} p(x_i) = \frac{1}{4},$$

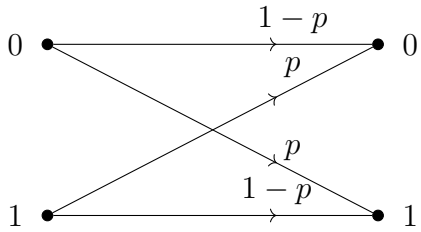
which does not depend on the input!! Using this in (3.3) we get

$$I(X,Y) = \sum_{i=1}^{4}\sum_{j=1}^{4} \frac{1}{4}p(x_i)\log_2\frac{1/4}{p(y_j)} = \sum_{i=1}^{4}\sum_{j=1}^{4} \frac{1}{4}p(x_i)\log_2\frac{\cancel{1/4}}{\cancel{1/4}} = 0,$$

i.e. **no** information gets through the channel: Y does not provide any information about X because whatever X is transmitted every Y is equally likely. Notice that this ultimately means that X and Y are **independent** (in which case the mutual information, given by the **intersection in the Venn diagram**, is 0), and we can actually check it

$$p(x_i|y_j) = \frac{p(y_j|x_i)p(x_i)}{p(y_j)} = \frac{1/4p(x_i)}{1/4} = p(x_i) \text{ (Bayes Theorem)}.$$

3.6.5 Example III: Binary Symmetric Channel



$$\begin{array}{rcc} p & \equiv & \text{probability of error} \\ p(X=0) & = & \alpha \\ p(X=1) & = & 1-\alpha \end{array} .$$

We want to compute the mutual information, which is given by

$$I(X,Y) = H(Y) - H(Y|X). \tag{3.4}$$

The formula involves computing $H(Y)$, and for that we need the probabilities of the output symbols. Using the *law of total probability* we have

$$\begin{aligned} p(Y=0) &= p(\overbrace{0}^{Y}|\overbrace{0}^{X})p(X=0) + p(0|1)p(X=1) = (1-p)\alpha + p(1-\alpha) \\ &= \alpha - p\alpha + p - p\alpha = \alpha + p - 2p\alpha \\ p(Y=1) &= p(1|0)p(X=0) + p(1|1)p(X=1) = p\alpha + (1-p)(1-\alpha) \\ &= p\alpha + 1 - \alpha - p + p\alpha = 1 + 2p\alpha - (\alpha + p), \end{aligned}$$

and we can easily check that $p(Y=0)+p(Y=1)=1$ ✓. Using these probabilities we obtain $H(Y)$ as

$$H(Y)=p(Y=0)\log_2\frac{1}{p(Y=0)}+p(Y=1)\log_2\frac{1}{p(Y=1)},$$

but since we are computing the entropy of a binary random variable, we can use the binary entropy function to express it, i.e.,

$$\begin{aligned}H(Y)&=H_b(p(Y=0))=\boxed{H_b(\alpha+p-2p\alpha)}\\&=H_b(p(Y=1))=H_b(1-p(Y=0))=H_b(1+2p\alpha-(\alpha+p)).\end{aligned}\tag{3.5}$$

Caveat

Notice that the argument of the binary entropy function is **not** the (binary) random variable, but rather its parameter, i.e., the probability that the r.v. takes on one of the values (either one of them).

Only $H(Y|X)$ is left to compute in order to calculate the mutual information. Using the definition, we have

$$\begin{aligned}H(Y|X)&=\sum_{i=1}^{M}p(x_i)H(Y|X=x_i)=p(X=0)H(Y|X=0)+p(X=1)H(Y|X=1)\\&=p(X=0)\left[\underbrace{p(0|0)\log_2\frac{1}{p(0|0)}+p(1|0)\log_2\frac{1}{p(1|0)}}_{H(Y|X=0)}\right]+\\&\quad+p(X=1)\left[\underbrace{p(0|1)\log_2\frac{1}{p(0|1)}+p(1|1)\log_2\frac{1}{p(1|1)}}_{H(Y|X=1)}\right].\end{aligned}$$

Both $Y|X=0$ and $Y|X=1$ are binary r.v.'s: we can express their entropies using the binary entropy function (actually, we are just applying the definition of entropy to each conditional random variable)[14]. Hence,

$$\begin{aligned}H(Y|X)&=\alpha H_b(p(0|0))+(1-\alpha)H_b(p(0|1))=\alpha H_b(1-p)+(1-\alpha)H_b(p)\\&\overset{H_b(p)=H_b(1-p)}{=}\alpha H_b(p)+(1-\alpha)H_b(p)=H_b(p)(\cancel{\alpha}+1-\cancel{\alpha})=\boxed{H_b(p)}.\end{aligned}\tag{3.6}$$

Using equations (3.5) and (3.6) in Equation (3.4) the resulting mutual information is

$$I(X,Y)=H_b(\alpha+p-2p\alpha)-H_b(p).$$

Let us see what happens to this expression for three particular cases:

[14]Every time we see an expression involving logarithms and only two probabilities that are complementary, we should ask ourselves whether it's possible to express that using the binary entropy function.

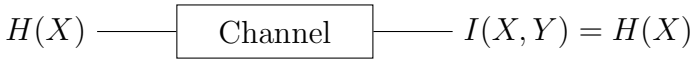
- $p = 0$, that is, the probability of error is 0: we never make a mistake

$$I(X,Y) = H_b(\alpha) - \cancelto{0}{H_b(0)} = H_b(\alpha)$$

X is a binary random variable and hence its entropy is the binary entropy function evaluated at the probability of one of the values, say 0, and hence $H(X) = H_b(\alpha)$

$$= H(X),$$

i.e., all the information goes through the channel[15] (it makes sense since there are no errors), and when depicted just like in Section 3.6.2, we have



with the information going in at the transmitter, $H(X)$ (as usual), reaching the other end of the channel.

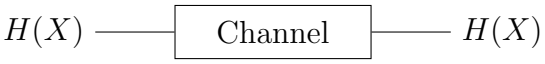
- $p = 1$, that is, we always mix up the two symbols

$$I(X,Y) = H_b(1+\alpha-2\alpha) - H_b(1) = H_b(1-\alpha) - \cancelto{0}{H_b(1)}$$

H_b is symmetric around 1/2 and hence $H_b(1-\alpha) = H_b(\alpha)$

$$= H_b(\alpha) = H(X),$$

i.e., again all the information goes through the channel,

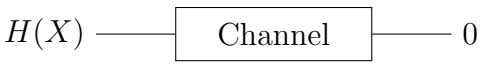


This makes sense since if we know the channel *always* mixes up the symbols, by deciding the symbol opposite to that which we have received we will always be right.

- $p = 0.5$, that is, we make mistakes half of the time

$$I(X,Y) = H_b(\cancel{\alpha} + 0.5 - \cancel{\alpha}) - H_b(0.5) = H_b(0.5) - H_b(0.5) = 0,$$

i.e., no information goes through the channel,

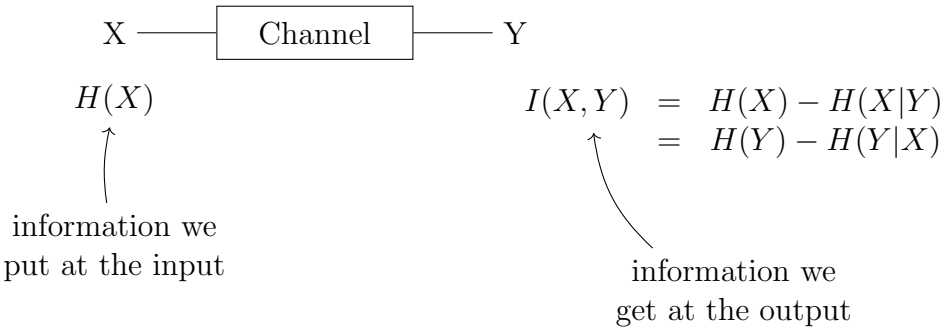


It is also reasonable: the channel *lies* half of the time and hence we know nothing of the symbol transmitted at the sight of the one received (whatever we transmit, the two output symbols are equally likely...a toss of a coin).

[15] Notice that $H(X)$ is the information at the input!!

3.7 Channel capacity

We have seen before that the information that gets through the channel is given by the **mutual information**,



For a specific channel, we are interested in the *maximum* amount of information that can get through it, that is, we want to know the maximum mutual information that can be attained...and this is known as the **channel capacity**,

$$C = \max I(X,Y) \frac{\text{bits}}{\text{symbol}} \text{ or } \frac{\text{bits}}{\text{channel use}},$$ [16]

where the **maximization is over the probability distribution of the input**. Remember from Section 3.6.2 that when we talk about mutual information we presume error-free information per channel use. Since capacity is simply the maximum of the mutual information, it is also endowed with these two properties. Hence, the channel capacity is the maximum amount of error-free information that can get through the channel per channel use.

Notice that the channel capacity depends on the mutual information and this, in turn, on the conditional probabilities that characterize the channel. Therefore, the capacity of the channel depends too on the conditional probabilities that characterize the channel.

3.7.1 Properties

$$X \in \{x_1, x_2, \cdots, x_M\}$$
$$Y \in \{y_1, y_2, \cdots, y_L\}$$

- $C \geq 0$, since $I(X,Y) \geq 0$
- $C \leq \log_2 M$ since

$$C = \max I(X,Y) = \max \{H(X) - H(X|Y)\}$$

$H(X|Y) \geq 0$ and hence the above expression is maximum when $H(X|Y) = 0$

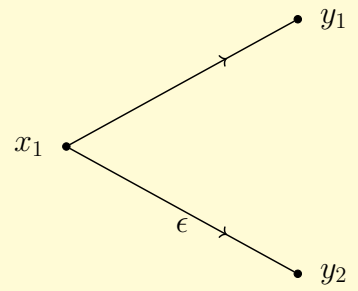
$$\leq \max \{H(X)\} = \log_2 M$$

- $C \leq \log_2 L$ (analogous proof)

[16] Every time we use the channel we transmit a symbol, and hence $\frac{\text{bits}}{\text{symbol}}$ is tantamount to $\frac{\text{bits}}{\text{channel use}}$.

? Quick quiz

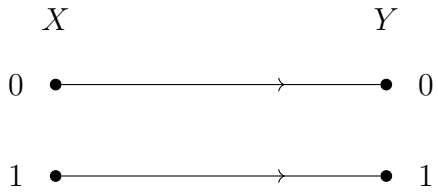
What is the capacity of the channel



? [17]

3.7.2 Example I

$$\begin{aligned} p(X=0) &= \alpha \\ p(X=1) &= 1-\alpha \end{aligned}$$



This example is very similar to that in Section 3.6.3. Again, it's clear that there will be no errors. What is the capacity? In order to obtain the capacity we need to compute the mutual information and maximize it,

$$I(X,Y) = H(Y) - H(Y|X) \overset{H(Y|X)=0}{=} H(Y)$$

What is the condition for the mutual information to be maximum? In other words, what is condition for $H(Y)$ to be maximum? Y is a discrete random variable, and hence, its entropy is maximum when the distribution is uniform[18], that is, when $\alpha = 1/2$[19]. And what is the value of that maximum? It's $\log_2 L = 1 \frac{\text{bits}}{\text{channel use}}$ (L being the number of values Y can take). Hence

$$C = 1 \frac{\text{bits}}{\text{channel use}},$$

and it is attained for $\alpha = 1/2$.

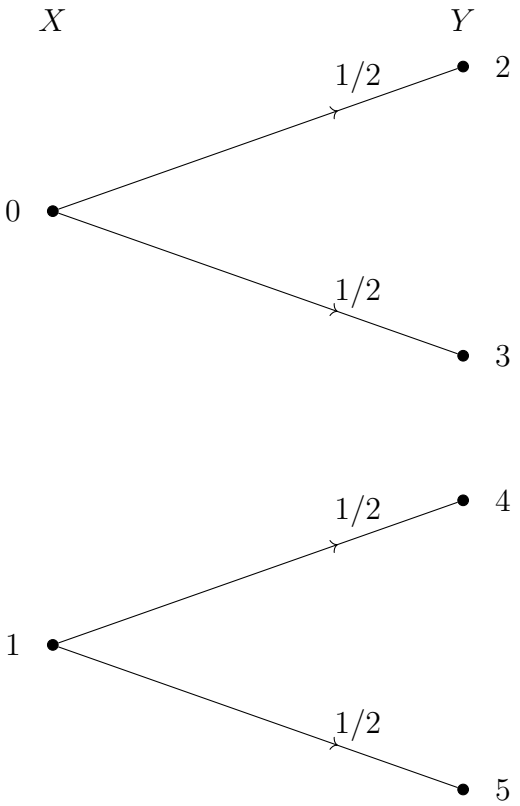
We could have reasoned: all the information goes through (there are no errors!!), hence the mutual information is equal to the information at the input, which is the entropy of X. Now, X is a discrete r.v., and thus the maximum of its entropy is $\log_2 2 = 1 \frac{\text{bits}}{\text{symbol}}$.

[17] It's zero: here $M = 1$ and we have, using the above properties that $C \leq \log_2 1 = 0$ (and $C \geq 0$).

[18] This is one of the properties for the entropy of a *discrete* random variable: it is maximum when the distribution of the random variable is uniform.

[19] Notice that since $X = Y$ we have $p(Y=0) = p(X=0) = \alpha$ and $p(Y=1) = p(X=1) = 1-\alpha$.

3.7.3 Example II



$$\begin{aligned} p(X=0) &= \alpha \\ p(X=1) &= 1-\alpha \end{aligned}$$

If at the sight of the output we always know the symbol transmitted, this means that all the information we put at the input reaches the output. Using the terminology from this module, the mutual information between X and Y (the information at the output) is equal to the entropy of X (the information at the input),

$$I(X,Y) = H(X).$$

Now, since the capacity is the maximum of the mutual information, we have

$$C = \max I(X,Y) = \max H(X),$$

and the maximum of the entropy is (the logarithm of the number of possible values) $\log_2 M = \log_2 2 = 1 \frac{\text{bits}}{\text{channel use}}$. Furthermore, we know that the maximum of the entropy is always attained for a uniform distribution, and hence we reach the capacity when $p(X=0) = p(X=1) = 1/2$. This is the confident-smart guy approach to solve this problem...but we can also show this mathematically by following the standard procedure that involves computing the mutual information and maximizing it afterwards. Using Equation (3.2), the mutual information is given by

$$I(X,Y) = H(Y) - H(Y|X).$$

We start by computing the entropy of Y, and for that we first need to compute the probabilities $p(2)$, $p(3)$, $p(4)$ and $p(5)$. Using the law of total probability we have

$$p(Y=2) = p(2,0) + \cancelto{0}{p(2,1)} = p(2|0)p(X=0) = \frac{1}{2}\alpha,$$

where we have used that input 1 and output 2 is something that (jointly) cannot happen. If we do the same for the remaining output symbols, we get

$$p(Y=3) = \frac{1}{2}\alpha$$
$$p(Y=4) = \frac{1}{2}(1-\alpha)$$
$$p(Y=5) = \frac{1}{2}(1-\alpha).$$

The entropy of Y is then given by

$$\begin{aligned}
H(Y) &= p(Y=2)\log_2\frac{1}{p(Y=2)} + p(3)\log_2\frac{1}{p(3)} + p(4)\log_2\frac{1}{p(4)} + p(5)\log_2\frac{1}{p(5)} \\
&= \frac{1}{2}\alpha\log_2\frac{2}{\alpha} + \frac{1}{2}\alpha\log_2\frac{2}{\alpha} + \frac{1}{2}(1-\alpha)\log_2\frac{2}{(1-\alpha)} + \frac{1}{2}(1-\alpha)\log_2\frac{2}{(1-\alpha)} \\
&= \alpha\log_2\frac{2}{\alpha} + (1-\alpha)\log_2\frac{2}{(1-\alpha)} \\
&= \alpha\cancelto{1}{\log_2 2} + \alpha\log_2\frac{1}{\alpha} + (1-\alpha)\cancelto{1}{\log_2 2} + (1-\alpha)\log_2\frac{1}{1-\alpha} \\
&= \cancel{\alpha} + \alpha\log_2\frac{1}{\alpha} + 1 - \cancel{\alpha} + (1-\alpha)\log_2\frac{1}{1-\alpha} = 1 + H_b(\alpha).
\end{aligned}$$

On the other hand, the conditional entropy of Y given X is

$$\begin{aligned}
H(Y|X) &= \sum_{i=1}^{M} p(x_i)H(Y|X=x_i) = p(X=0)H(Y|X=0) + p(X=1)H(Y|X=1) \\
&= \alpha\left(p(2|0)\log_2\frac{1}{p(2|0)} + p(3|0)\log_2\frac{1}{p(3|0)} + \cancelto{0}{p(4|0)\log_2\frac{1}{p(4|0)}} + \right. \\
&\quad \left. + \cancelto{0}{+p(5|0)\log_2\frac{1}{p(5|0)}}\right) + (1-\alpha)\left(\cancelto{0}{p(2|1)\log_2\frac{1}{p(2|1)}} + \right. \\
&\quad \left. + \cancelto{0}{p(3|1)\log_2\frac{1}{p(3|1)}} + p(4|1)\log_2\frac{1}{p(4|1)} + p(5|1)\log_2\frac{1}{p(5|1)}\right) \\
&= \alpha\cancelto{1}{\left(\frac{1}{2}\log_2 2 + \frac{1}{2}\log_2 2\right)} + (1-\alpha)\cancelto{1}{\left(\frac{1}{2}\log_2 2 + \frac{1}{2}\log_2 2\right)} = 1.
\end{aligned}$$

Putting it all together we have

$$I(X,Y) = H(Y) - H(Y|X) = 1 + H_b(\alpha) - 1 = H_b(\alpha).$$

Interpretation of the conditional entropy provides a shortcut to arrive at the same conclusion: since there is no uncertainty about the value of X if we know Y, then $H(X|Y) = 0$ and

$$I(X,Y) = H(X) - \cancelto{0}{H(X|Y)} = H(X) = H_b(\alpha),$$

where the last equality stems from the fact that X is a binary random variable (whose entropy is then given by the binary entropy function) with α being the probability of one of the values it can take.

Once we have the mutual information, the capacity is its maximum value, and the maximum of the binary entropy function is 1 when $\alpha = 1/2$. Hence

$$C = \max I(X,Y) = \max H_b(\alpha) = 1 \frac{\text{bits}}{\text{channel use}},$$

i.e., what we predicted at the beginning.

Remark

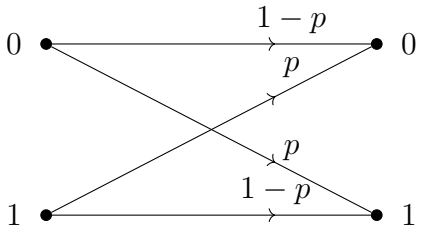
Notice that whenever there is no overlap between outputs corresponding to different inputs (i.e., different inputs do not yield the same output), the mutual information is equal to the entropy of the input, and hence the capacity (its maximum) is $\log_2 M$.

Maximizing a function

Computing the capacity always entails maximizing the mutual information. Now, we have a standard procedure for finding the maximum of a function: we take the derivative and set it equal to 0. Why are we not doing that here? Because it turns out that, in the above example, the mutual information is equal to a function (the binary entropy function), whose maximum we know beforehand. However, keep in mind that, in general, you need to differentiate the mutual information to find its maximum.

3.7.4 Capacity of a Binary Symmetric Channel (BSC)

$$\begin{array}{rcl} p & \equiv & \text{probability of error} \\ p(X=0) & = & \alpha \\ p(X=1) & = & 1-\alpha \end{array}$$



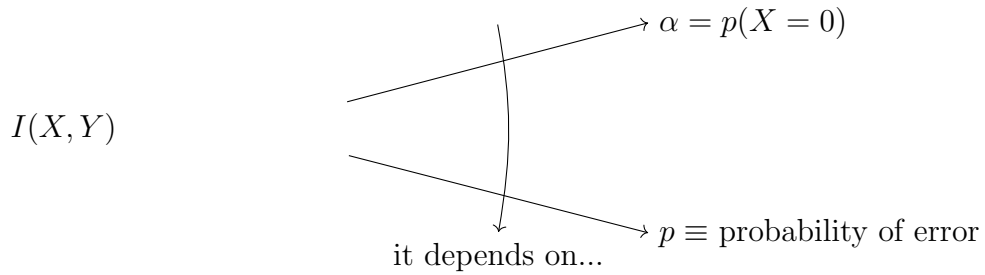
We know from Section 3.6.5 the mutual information for this channel:

$$I(X,Y) = H_b(\alpha + p - 2p\alpha) - H_b(p).$$

The capacity is the maximum of the mutual information, i.e.,

$$C = \max I(X,Y),$$

but how do we obtain this maximum? With respect to which parameter should we maximize $I(X,Y)$?



We have no control whatsoever over p (it is inherent to the channel!!) but we have over α (we choose the input and hence its distribution). Therefore, we maximize $I(X,Y)$ with respect to α[20],

$$C = \max_{\alpha} I(X,Y) = \max_{\alpha} \{H_b(\alpha + p - 2p\alpha) - H_b(p)\}$$

$H_b(p)$ doesn't depend on α

$$= \max_{\alpha} H_b(\alpha + p - 2p\alpha) - H_b(p)$$

the maximum of the binary entropy function is 1

$$= 1 - H_b(p),$$

where the last equality exploits the fact that the maximum of the binary entropy function is 1. Moreover, we know this maximum is attained when the argument of H_b is $1/2$, and hence we can find the particular value of α (which completely determines the input distribution) that yields the capacity of the channel,

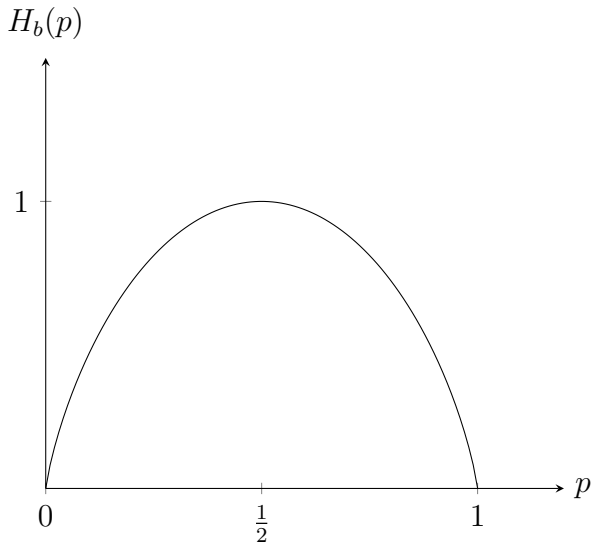
$$\alpha + p - 2p\alpha = \frac{1}{2} \Rightarrow \alpha(1-2p) + p = \frac{1}{2} \Rightarrow \alpha = \frac{\frac{1}{2} - p}{1-2p} = \frac{\frac{1-2p}{2}}{1-2p} = \frac{1}{2}.$$

Therefore, in summary,

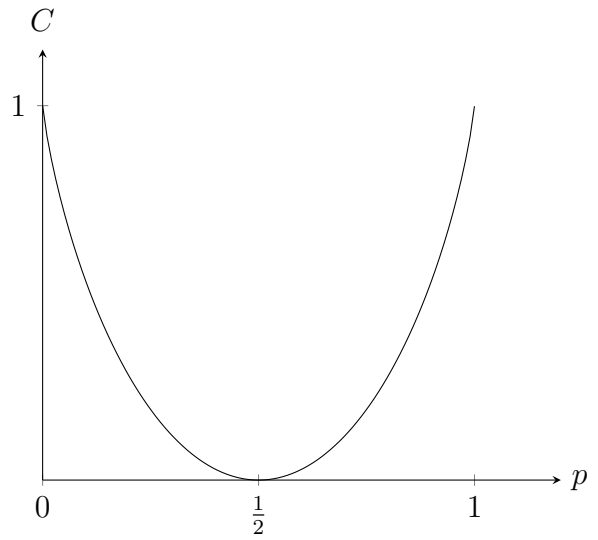
$$C = 1 - H_b(p),$$

for $\alpha = 1/2$, i.e., for a uniform input distribution. Notice the capacity of a BSC only depends on the probability of error, p. We know the function $H_b(p)$ (both its mathematical expression and shape)

[20]We *always* maximize with respect to the distribution of the input. See Section 3.7.



and hence the plot for the capacity of a BSC is



If we focus only on the region between 0 and 1/2 (the probability of error should always be less than 0.5; otherwise we flip every bit we receive), we see that the capacity goes down as the probability of error increases

$$\uparrow\ p \longrightarrow \downarrow\ C.$$

3.7.5 Computation of the capacity in the general case

In general,

$$C = \max_{p(x_i)} I(X,Y) = \max_{p(x_i)} \{H(Y) - H(Y|X)\}\ \frac{\text{bits}}{\text{channel use}},$$

and we need to find the input distribution $p(x_1), p(x_2), \cdots$ that maximizes the mutual information. How do we go about it? Both $H(Y)$ and $H(Y|X)$ are functions[21] of the

[21] *different* functions...

input probabilities...but also of the conditional probabilities that characterize the channel

$$H(Y) = f\left(p(x_1), p(x_2), \cdots, p(x_M), \overbrace{p(y_1|x_1), \cdots, p(y_L|x_M)}^{\text{these...we know}}\right)$$

$$H(Y|X) = g\left(p(x_1), p(x_2), \cdots, p(x_M), p(y_1|x_1), \cdots, p(y_L|x_M)\right),$$

and $p(x_1), p(x_2), \cdots, p(x_M)$ are variables with respect to which we have to maximize with the constraint

$$p(x_1) + p(x_2) + \cdots + p(x_M) = 1.$$

Hence, the computation of the capacity is not straightforward in general, though there are analytical expressions for certain channel models, e.g., symmetric channels.

Maximization

Maximization is carried with respect to the input distribution, i.e., the probabilities of the input symbols. This means that, whenever you are computing the capacity of some channel, you **cannot** assume any value for $p(x_1)$, $p(x_2)$.... Rather, the latter are *free* variables that you must select/*tune* in order to maximize the mutual information.

Quick quiz

Why is it, in general, hard to compute the capacity of a channel? [22]

3.8 Noisy-channel coding theorem

In the design of any communications system, a primary goal is to maximize the reliability of the system. In other words, we want the probability of error to be as small as possible. At first sight it might seem that the probability of error can never go down below a certain value...we would think it should depend on the amount of noise added by the channel. However, a fundamental result in information theory states that a **reliable transmission**[23] is always possible, regardless of the level of noise...as long as the **transmission rate is below** a certain threshold, which is exactly **the channel capacity**.

Definition 3.8.1: Noisy-channel coding theorem (Shannon, 1948)

It is possible to transmit with an arbitrarily low probability of error if the transmission rate is below the channel capacity.

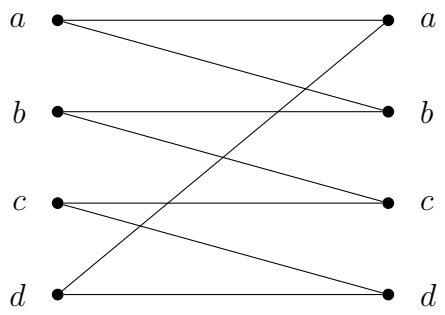
The conclusion to draw from this is that the **level of noise in a channel does not limit the reliability of the communication but rather its speed**. So, for

[22] Because we must solve an optimization problem involving a bunch of variables *subject to a constraint* (the probabilities of the inputs must add up to 1).

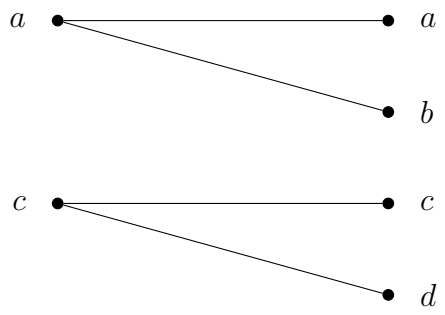
[23] When we talk about a reliable transmission, we just mean an error-free one (see Section 3.6.2), i.e., a transmission in which we can attain a probability of error as small as we like.

instance, can we transmit with probability of error 10^{-200}? Yes, we can, as long as the transmission rate is below the capacity. Hence, the latter ultimately indicates how fast you can transmit through the channel.

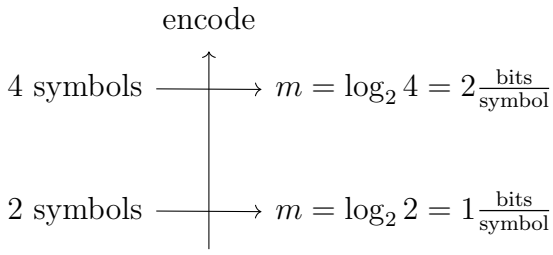
We are going to see an intuitive (informal) proof of the theorem. Let us consider a DMC with 4 inputs and 4 outputs



If we get an a at the receiver, we cannot tell whether the symbol transmitted was a or d, and the same applies for the remaining symbols. Hence, this is not an error-free transmission and, ultimately, the probability of error will depend on how often the channel "lies" to us. On the other hand, if only a or c can be transmitted

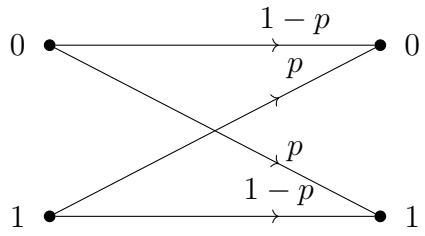


there is no ambiguity and the probability of error is 0 (see the example and final remark in Section 3.7.3). If we use the second DMC instead of the first one, we are decreasing the transmission rate

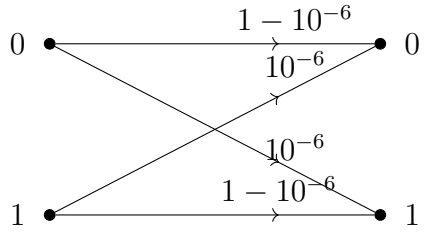


but this is the price we have to pay for a reliable transmission!! This is the fundamental idea behind the noisy-channel coding theorem: in the transmitter we must use symbols that are *far apart enough* so that after going through the channel they don't overlap...or, if this is not possible, so that the overlap does not occur very often.

If we try to apply this idea on a binary symmetric channel (BSC),



we immediately see that it's not possible. In general, this idea (of employing only a subset of the input symbols) is hardly applicable to any DMC. Then, a more realistic goal is to lower the probability of error so that "the overlap is very small", e.g.,



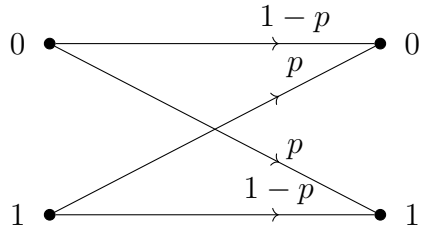
There is overlap, but it's very small, meaning that every time we receive, for instance, a 0, only once in a million the bit transmitted was 1.

We need some technique to lower the probability of error, and that technique is (channel) **coding**. It consists in adding redundancy to the information we want to transmit so that our chances to reliably recover it increase.

? Quick quiz

You are given a BSC with a probability of error equal to 0.49 (almost the flip of a coin). Is a reliable transmission possible? [24]

3.9 Channel coding by example



In this channel, by definition, when an error happens during transmission (either $0 \to 1$ or $1 \to 0$), the bit is *flipped*, i.e., we receive a bit which is not the one transmitted. We could build a more robust system:

[24] Sure, the noisy-channel coding theorem states that, by using channel coding (i.e., adding redundancy) you can have an error probability as small as you like, but transmission will be slow (specifically, for this BSC, of only $C = 1 - H_b(0.49) \approx 3 \times 10^{-4}$ bits per channel use).

- instead of transmitting 0, we transmit 000,

$$0 \longrightarrow 000, \tag{3.7}$$

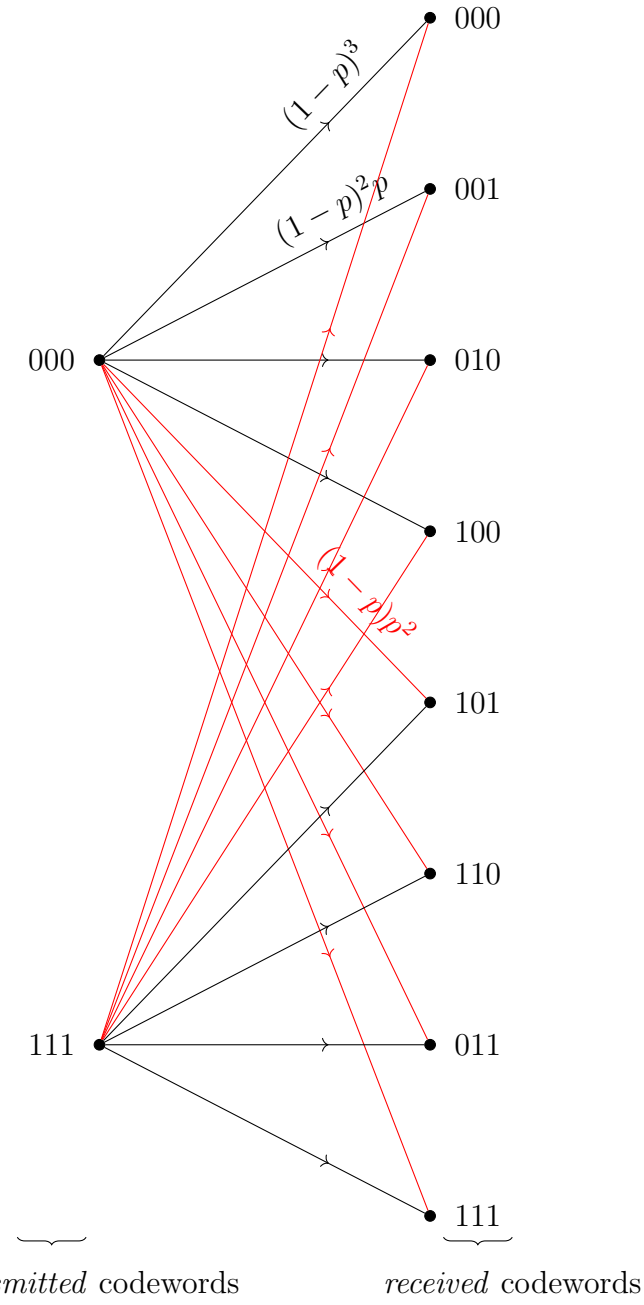
- and instead of transmitting 1, we transmit 111,

$$1 \longrightarrow 111. \tag{3.8}$$

Sequences of bits 000 and 111 are the so-called **codewords** (what is actually going to be transmitted), each one representing the output of the *coding scheme* (given by the mapping specified by equations (3.7) and (3.8)) for a piece of information (or *message*).

At the receiver, if we get something that is not a codeword, we try to *fix* it by *assuming* that the codeword actually transmitted is the one that entails flipping the lesser number of bits. If, for instance, we receive 110 then we take it that 111 was transmitted because going from 111 to 110 entails flipping a single bit, while in order to go from 000 to 110 we must flip two of them. At the sight of the *coding scheme* ($0 \leftrightarrow 000$ and $1 \leftrightarrow 111$), this amounts to deciding "the bit that was transmitted is the one that is repeated the most".

The resulting system is

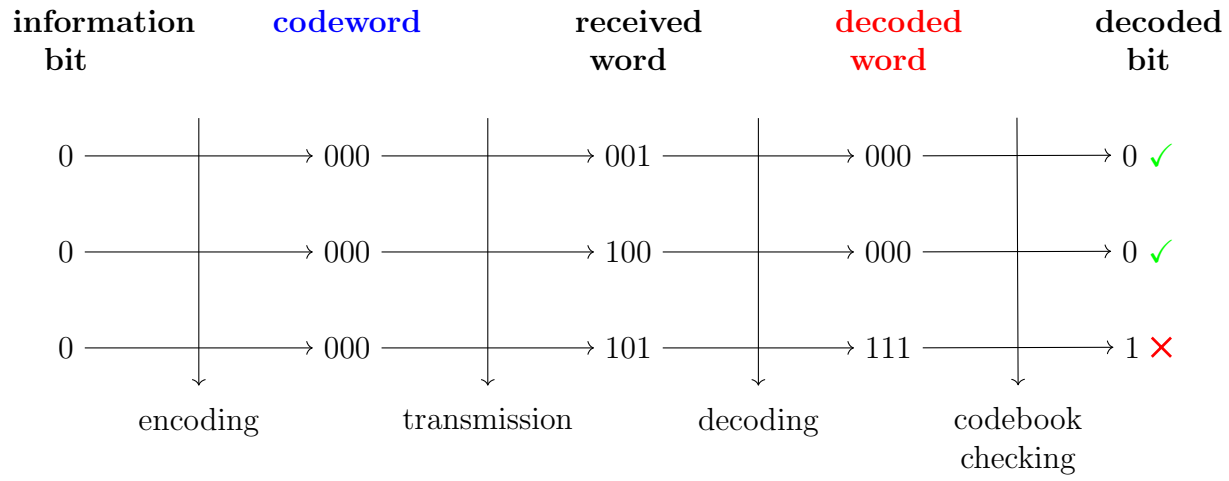


Notice that now every *channel use* corresponds to **three** channel uses in the original system[25]. Also, errors occur independently (the DMC is *memoryless*), and thus the probability of, e.g., receiving 001 when 000 was transmitted is the probability of no error, $(1-p)$, times the probability of no error, $(1-p)$, times the probability of error, (p), i.e., $(1-p)^2p$. Mathematically,

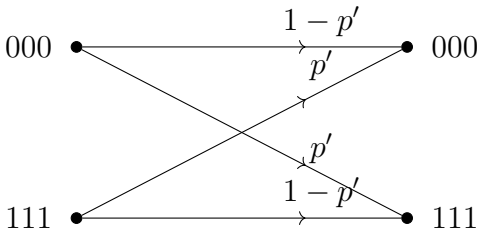
$$p(000 \to 001) = p(0 \to 0)p(0 \to 0)p(0 \to 1).$$

If the received word is *decoded* as the bit that is repeated the most, then the system is more robust because for an information bit to be received incorrectly two errors must happen. Below, some examples of end-to-end transmission of a single bit are presented.

[25]..meaning that we need to spend three times the energy we were spending before!!



If we think of the system as transmitting a certain codeword and decoding another (possibly different) one, the equivalent DMC is



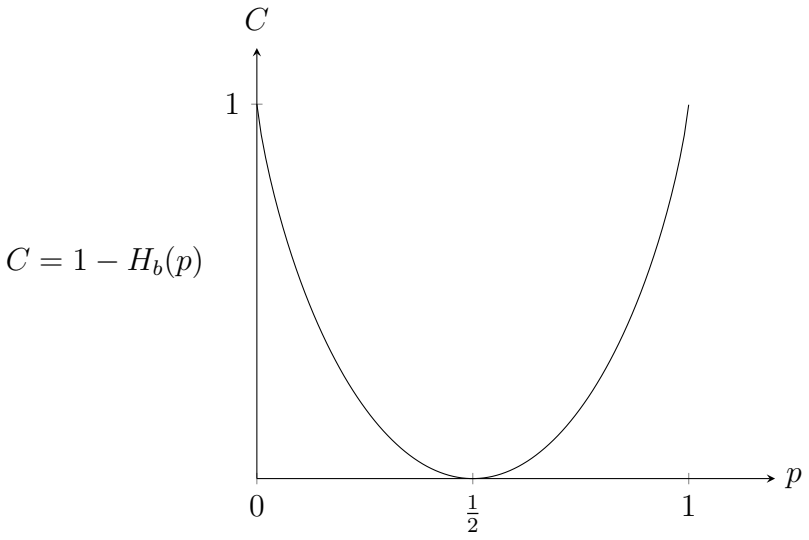
$$C' = 1 - H_b(p')$$

(still a BSC)

and it can be shown that

$$p' < p,$$

which means that using channel coding gives rise to an increase in the capacity because



However, the capacity is the maximum amount of information *per channel use*, and using this *composite* channel is equivalent to 3 channel uses of the original one. In both cases, we are transmitting the same amount of information, 1 bit, but now we need

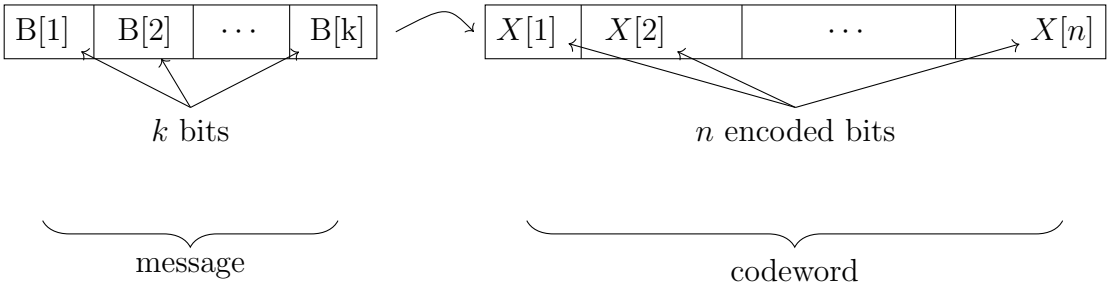
three (old) channel uses. Therefore, in order to decide if this coding scheme is worth it what we should be assessing is whether

$$\underbrace{\frac{C'}{3}}_{\substack{\text{information per use} \\ \text{of the } \textit{original} \text{ channel}}} > C.$$

The code we have just seen is the so-called *repetition code*, but there are many coding schemes. Here we only focus on **block codes** like the one in the example.

3.10 Block codes

The coding schemes we'll be talking about here are essentially mappings between sequences of *information* bits onto sequences of *encoded* bits. The input is a word (vector) of k bits and the output is another word, a *codeword*, of n bits.



with $n > k$ (since otherwise we would not be adding any redundancy).

The **code rate** is

$$R = \frac{k}{n},$$

and can be interpreted as the number of *information* bits per bit *transmitted*. Notice that (since $n > k$) R is always between 0 and 1, i.e., $0 < R < 1$.

? Quick quiz

If R is large, are we adding a lot or very little redundancy? [26]

We say that the code is (n, k), with n being the *block length*.

Example: BSC

For the block code

$$\begin{array}{rcl} 0 & \rightarrow & 000 \\ 1 & \rightarrow & 111 \end{array},$$

we have

$$\left.\begin{array}{l} k = 1 \\ n = 3 \end{array}\right\} \Rightarrow R = \frac{1}{3}.$$

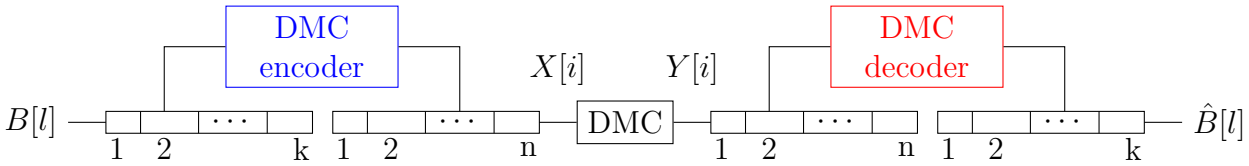
[26] If R is large that means k (number of bits in the message) is very close to n (the number of bits in the codeword), and hence we are adding very little redundancy.

The same code rate is achieved for other block lengths: the code

$$\begin{array}{rcl} 00 & \rightarrow & 000000 \\ 01 & \rightarrow & 111000 \\ 10 & \rightarrow & 011000 \\ 11 & \rightarrow & 101000 \end{array} ,$$

has $R = \frac{2}{6} = \frac{1}{3}$.

Where does channel coding fit in our communications system? We put the *encoder*[27] before the DMC and the *decoder* afterwards.



Roughly, what happens in a typical system is

1. we wait until we have k bits to be transmitted, that are then passed to the...
2. ...DMC encoder, which yields a codeword with n bits.
3. These are transmitted through the DMC[28], and the received n bits make up the input for...
4. ...the DMC decoder, which is meant to recover the k bits transmitted (as usual, the decoded bits might match or not those transmitted, i.e., again we have *estimates*).

Knowing these channel coding concepts, we can state the noisy-channel coding theorem in a more precise manner:

1. If the code rate, R, times the number of bits per symbol[29], m, is below the channel capacity, then for $\delta > 0$[30], there exists a code with block length n (for n large enough) whose probability of error is below δ, i.e.,

$$mR < C \Rightarrow \forall \delta > 0, \exists \text{ code yielding } P_e < \delta$$

2. If $mR > C$, the probability of error of *any* code has a non-zero lower bound, i.e.,

$$mR > C \Rightarrow P_e > \epsilon,$$

where $\epsilon > 0$ is a constant.

[27] Notice that this encoder has nothing to do with the encoder of the previous module!!

[28] This *discrete memory-less channel* encompasses the encoder, decoder, (Gaussian) channel, demodulator and detector blocks that we studied in the previous module.

[29] Every time we use the channel we transmit an element of the constellation that carries $m = \log_2 M$ bits (M being the number of elements in the constellation), of which only mR contain information.

[30] ...and we can choose δ as small as we like!!

3. There exist codes attaining the channel capacity, i.e., such that $mR = C$.

   Great!!...but how do we build them? The theorem doesn't provide a *recipe*. It only states *they exist*. In general:

   - R low: easy to find a code attaining the capacity.
   - R high: hard to find a code attaining the capacity.

   Notice that R is low when n is large, in which case the code is adding a lot of redundancy, $(n-k)$[31]. On the contrary, if $R = 1$, then we would not be adding any redundancy (**not** meaningful).

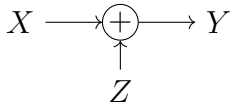
Example

Assume the capacity of a certain channel is $C = 2\frac{\text{bits}}{\text{channel use}}$, and we would like a system with a probability of error $P_e = 10^{-6}$. Is it possible to find a code allowing a transmission rate $mR = 0.1$? Yes, and it should be fairly easy to find such a code (transmission rate is far away from the capacity). Is it possible to find a code allowing a transmission rate $mR = 1.9$? Yes, but it's going to be hard to find such a code (we are very close to the capacity!!).

3.11 Capacity of the Gaussian channel

So far, we have been working with *discrete* memoryless channels (DMCs). Now we are going to extend the previous definitions for the Gaussian channel where the input and output are continuous (rather than discrete) random variables. The output of the channel is modeled as a random variable Y whose structure is given by[32]

$$Y = X + Z$$



with

- $X \equiv$ continuous r.v.
- $Z \equiv$ Gaussian r.v. with zero mean and variance σ_z^2.

Since both X and Z are continuous random variables, so is Y (their sum), and the latter is related with the input through the conditional pdf $f_{Y|X}(y|x)$ (Gaussian with mean X and variance σ_z^2).

The capacity is still the maximum of the mutual information between the input and output variables, but these are now continuous and we need to extend the concepts of entropy, joint entropy, conditional entropy and mutual information to the continuous case.

[31]...and hence our transmission is not very efficient: we are transmitting very few (mR with R small) *information* bits per channel use...we want R to be large

[32]Notice that this is the Gaussian channel considered in the previous module.

3.11.1 Differential entropy

The counterpart of entropy for continuous random variables is the

Definition 3.11.1: Differential entropy

$$h(X) = \int_{-\infty}^{\infty} f_X(x) \log_2 \frac{1}{f_X(x)} dx \text{ bits}$$

This metric does **not** have the intuitive meaning that conventional entropy has (either uncertainty or average amount of information produced)[33] because the probability of a continuous random variable taking on a specific value is null (and hence the corresponding *autoinformation* is ∞!!). Notice the units are not $\frac{\text{bits}}{\text{symbol}}$ anymore but simply bits.

Example: uniform r.v. in the interval $[0, b]$

$X \sim \mathcal{U}[0, b]$

$$h(X) = \int_0^b \frac{1}{b} \log_2 b \, dx = \log_2 b$$

Notice:

- $b < 1 \Rightarrow h(X) < 0$, which goes against the *non-negativity* property of the discrete entropy, and
- $b = 1 \Rightarrow h(X) = 0$, which, according to the interpretation we have for a discrete r.v., means that uncertainty is 0. However, X is not determinist and hence **there is** uncertainty.

Differential entropy for two continuous probability distributions of interest

It can be shown that

- If $X \sim \mathcal{N}(\mu, \sigma_X^2)$, then

  $$h(X) = \frac{1}{2} \log_2 \left(2\pi e \sigma_X^2\right) \text{ bits}$$

  ...**regardless of the mean!!** That is given: if you compute the entropy of a Gaussian random variable with zero mean, the entropy of another Gaussian with mean μ must be the same since the probabilities have not changed, only their *labels*. Recall that, at the beginning of this module, we said that the entropy (of a discrete r.v.) does not depend on the particular values the variable take but only on the probabilities.

- If $X \sim \mathcal{U}[a, b]$ then

  $$h(X) = \log_2(b - a)$$

[33] Still, we can think of differential entropy as a *proxy* for the uncertainty of a random variable.

Bounds on the differential entropy

- If X is a continuous *unbounded* (it takes on values from $-\infty$ to ∞, i.e., $X \in (-\infty, \infty)$) r.v. with variance σ_X^2, then the distribution that yields the maximum entropy is the Gaussian distribution, i.e.,

$$h(X) \text{ maximum} \Leftrightarrow X \sim \mathcal{N}\left(\cdot, \sigma_X^2\right),$$

  regardless of the mean. A corollary of this result is that, for any continuous *unbounded* r.v. X,

$$h(X) \leq \frac{1}{2} \log_2 \left(2\pi e \sigma_X^2\right) \text{ bits}$$

- If X is a continuous *bounded* r.v. between a and b (notice that a Gaussian r.v. is not bounded!!), then the distribution that yields the maximum entropy is the uniform

$$h(X) \text{ maximum} \Leftrightarrow X \sim \mathcal{U}\left[a, b\right],$$

  and hence for any continuous *bounded* r.v. X,

$$h(X) \leq \log_2(b-a)$$

3.11.2 Two continuous random variables

For continuous random variables, joint and conditional (differential) entropy as well as mutual information are analogously defined by, essentially, replacing summations with integrals, and pmf's with pdf's.

Definition 3.11.2: Joint differential entropy

$$h(X,Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x,y) \log_2 \frac{1}{f_{X,Y}(x,y)} dxdy$$

Definition 3.11.3: Conditional differential entropy

$$h(X|Y) = \int_{-\infty}^{\infty} f_Y(y) \int_{-\infty}^{\infty} f_{X|Y}(x|y) \log_2 \frac{1}{f_{X|Y}(x|y)} dxdy$$

or, equivalently,

$$h(X|Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x,y) \log_2 \frac{1}{f_{X|Y}(x|y)} dxdy$$

They are natural extensions to the definitions seen for discrete random variables, and the same identities hold, e.g.,

$$h(X,Y) = h(X|Y) + h(Y)$$

Definition 3.11.4: Mutual information

$$I(X,Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x,y) \log_2 \frac{f_{X,Y}(x,y)}{f_X(x) f_Y(y)} dxdy$$

In order to compute differential entropies we need to take into account

$$\boxed{0 \log_2 \frac{0}{0} = 0}.$$

Just like for discrete random variables, mutual information can be expressed using the entropy (*differential*, in this case),

$$I(X,Y) = h(Y) - h(Y|X) = h(X) - h(X|Y)$$

Unlike differential entropy, the mutual information between continuous random variables has the same meaning as in the discrete case: the information/knowledge that one variable provides about the other. Furthermore, the same properties hold

- $I(X,Y) \geq 0$ (non negative function)
- $I(X,Y) = 0 \Leftrightarrow X$ and Y independent
- $I(X,Y) = I(Y,X)$

After this huge parenthesis we go back to...

3.11.3 Computation of the capacity for the Gaussian Channel

The capacity is computed just like before. We start from

$$Y = X + Z$$

with

$$\left.\begin{array}{ll} X & \equiv \text{input variable} \\ Y & \equiv \text{output variable} \\ Z & \sim \mathcal{N}(0, \sigma_z^2) \end{array}\right\} \text{They are related through } f_{Y|X}(y|x)$$

and the capacity is given by

$$C = \max_{f_X(x)} I(X,Y)$$

where $f_X(x)$ is the pdf of the *input* random variable (and hence it determines its distribution) and

$$I(X,Y) = h(Y) - h(Y|X).$$

The entropy of Y, $h(Y)$, is not bounded[34], something that did not happen in the discrete case, for which the maximum was $\log_2 M$, with M being the number of inputs. This entails the maximum can be infinity!! We are going to cope with this difficulty by adding a constraint to the maximization problem: the power of X must be below certain threshold P,

$$\mathbb{E}\left[X^2\right] \leq P.$$

This is a natural constraint in a communications system: you cannot use all the power you want.

[34]...since we know that Y is a *continuous* random variable (because its definition involves Z) and hence the differential entropy can be as large as $h(Y) = 1/2 \log_2 2\pi e \sigma_Y^2$ with σ_Y^2 unbounded.

The resulting optimization problem is

$$C = \max_{f_X(x)/\mathbb{E}[X^2]\leq P} I(X,Y).$$

Going back to the expression for the mutual information, we have

$$I(X,Y) = h(Y) - h(Y|X),$$

and we know the distribution of $Y|X$,

$$Y|X \sim \mathcal{N}\left(X, \sigma_z^2\right)$$

because Y is simply X plus Gaussian noise with variance σ_z^2, and adding a constant to a Gaussian r.v. only shifts its mean. Therefore, we have

$$I(X,Y) = h(Y) - \frac{1}{2}\log_2 2\pi e\sigma_z^2.$$

We know nothing about the distribution of Y, but the latter is anyway a continuous (unbounded) random variable with variance, say σ_Y^2, and hence

$$h(Y) \leq \frac{1}{2}\log_2 2\pi e\sigma_Y^2$$

with equality holding when Y is Gaussian. Notice that the variance of this r.v. is, in general, different from that of the noise (i.e., $\sigma_Y^2 \neq \sigma_z^2$). Then,

$$\begin{aligned} I(X,Y) &\leq \frac{1}{2}\log_2 2\pi e\sigma_Y^2 - \frac{1}{2}\log_2 2\pi e\sigma_z^2 \\ &= \frac{1}{2}\log_2 \frac{\cancel{2\pi e}\sigma_Y^2}{\cancel{2\pi e}\sigma_z^2} = \frac{1}{2}\log_2 \frac{\sigma_Y^2}{\sigma_z^2}, \end{aligned}$$

and we are going to express the variance of Y in a different way[35]

$$\begin{aligned} \sigma_Y^2 &= \mathbb{E}\left[Y^2\right] - \mathbb{E}\left[Y\right]^2 = \mathbb{E}\left[(X+Z)^2\right] - \mathbb{E}\left[X+Z\right]^2 \\ &= \mathbb{E}\left[X^2+Z^2+2XZ\right] - \left(\mathbb{E}\left[X\right] + \cancelto{0}{\mathbb{E}\left[Z\right]}\right)^2 \\ &= \mathbb{E}\left[X^2\right] + \mathbb{E}\left[Z^2\right] + 2\mathbb{E}\left[XZ\right] - \mathbb{E}\left[X\right]^2 \\ &\overset{\substack{X,Z \text{ indep:} \\ \mathbb{E}[XZ]=\mathbb{E}[X]\mathbb{E}[Z]}}{=} \mathbb{E}\left[X^2\right] + \mathbb{E}\left[Z^2\right] + 2\mathbb{E}\left[X\right]\cancelto{0}{\mathbb{E}\left[Z\right]} - \mathbb{E}\left[X\right]^2 \\ &= \mathbb{E}\left[X^2\right] + \mathbb{E}\left[Z^2\right] - \mathbb{E}\left[X\right]^2. \end{aligned}$$

In order to maximize $I(X,Y)$ we need to maximize σ_Y^2, and for that we would like $\mathbb{E}\left[X\right] = 0$ (this might bring back memories of centered constellations). Hence,

$$\sigma_Y^2 \leq \mathbb{E}\left[X^2\right] + \mathbb{E}\left[Z^2\right] = \mathbb{E}\left[X^2\right] + \sigma_z^2,$$

[35]The notation $\mathbb{E}\left[\cdot\right]^2$ means the square of the expectation.

where we have used that $\mathbb{E}[Z^2]$ is the variance of the noise, σ_z^2, since the latter has zero mean.

The above expression is the maximum value for σ_Y^2, and going back to the expression for the mutual information we have

$$I(X,Y) \leq \frac{1}{2}\log_2 \frac{\sigma_Y^2}{\sigma_z^2} = \frac{1}{2}\log_2 \frac{\mathbb{E}[X^2]+\sigma_z^2}{\sigma_z^2} = \frac{1}{2}\log_2\left(1+\frac{\mathbb{E}[X^2]}{\sigma_z^2}\right),$$

and the capacity is given by

$$C = \max_{f_X(x)/\mathbb{E}[X^2]\leq P} I(X,Y) \leq \max_{f_X(x)/\mathbb{E}[X^2]\leq P} \frac{1}{2}\log_2\left(1+\frac{\mathbb{E}[X^2]}{\sigma_z^2}\right)$$

if we want to maximize the expression inside the logarithm, we should choose $\mathbb{E}[X^2]$ as large as we can, i.e., $\mathbb{E}[X^2] = P$

$$= \frac{1}{2}\log_2\left(1+\frac{P}{\sigma_z^2}\right)\ \frac{\text{bits}}{\text{symbol}}.$$

Sometimes you see a similar expression in which it is assumed

- the channel is baseband with bandwidth B, which means that the symbol rate should be[36]
$$R_s = 2B\,\frac{\text{symbols}}{\text{second}},$$
- at the receiver there is an ideal filter with the same bandwidth[37] as the channel, that is meant to bound the power of the noise[38], which is then (see Section 1.8.2)
$$\sigma_z^2 = N_0 B.$$

In such case, the capacity *in bits per second* is given by

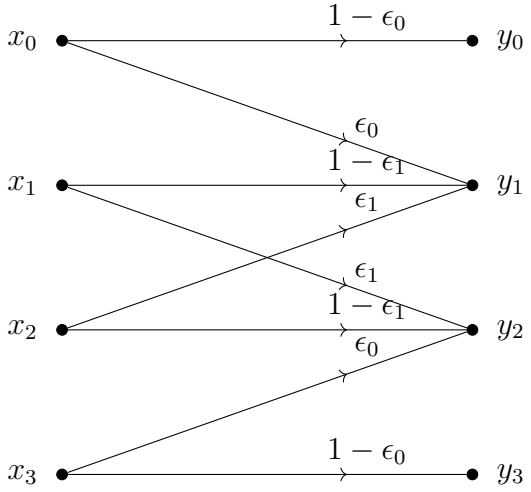
$$C = \frac{1}{2}\log_2\left(1+\frac{P}{N_0 B}\right)\frac{\text{bits}}{\cancel{\text{symbol}}}2B\frac{\cancel{\text{symbols}}}{\text{second}} = B\log_2\left(1+\frac{P}{N_0 B}\right)\frac{\text{bits}}{\text{second}}.$$

[36]...in order to alleviate nuisances such as inter-symbol interference (discussed in more advanced courses)

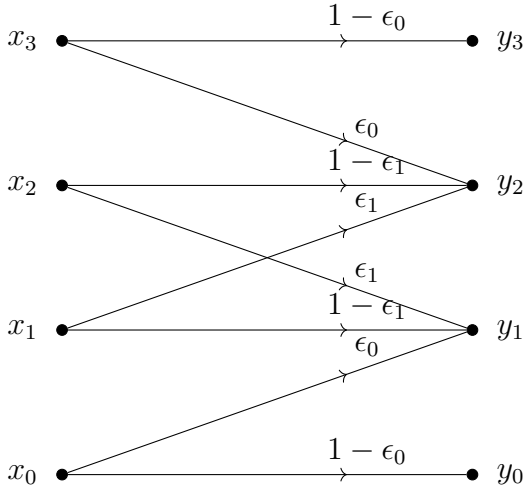
[37]We could use a filter with a larger bandwidth (never lower since that would effectively reduce the bandwidth of the signals that can be transmitted without distortion) but that wouldn't help the transmitted signal since the maximum bandwidth was already determined by the channel, and it would make noise (at the receiver) worse.

[38]Recall that there must be a filter at the receiver or, otherwise, the power of the thermal noise will be infinity!!

3.12 Appendix: pairwise symmetry in DMCs



There is symmetry in this DMC (although this is **not** a strongly symmetric channel). Specifically, x_0 and x_3 play the same role: they are connected (through the same conditional probabilities) to the output symbol across, and the neighboring inner one. Thus, when maximizing the mutual information to obtain the capacity, their probabilities should be the same...Why? Let us assume that we compute the capacity and it is attained for a distribution in which the first symbol is more likely than the last one, i.e., $p(x_0) > p(x_3)$. If we turn the channel upside down,



the connections and their corresponding labels are the same. Then, computing the capacity all over again *following the same procedure*[39], we would get that, again, it is attained for an input distribution in which the first symbol is more likely than the last symbol, i.e, for $p(x_3) > p(x_0)$...but the channel is the same!!. Since we arrived at a contradiction, it is not possible that $p(x_0) > p(x_3)$. But we can apply the same reasoning assuming $p(x_0) < p(x_3)$, and that would lead to another contradiction. Then, it must be the case that $p(x_0) = p(x_3)$.

[39]Forget where you are coming from: you are given this channel and you apply the standard procedure.

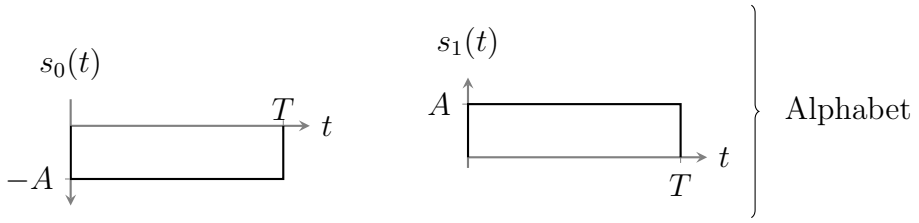
We could apply the same reasoning to justify that the capacity must be attained for $p(x_1) = p(x_2)$.

Chapter 4

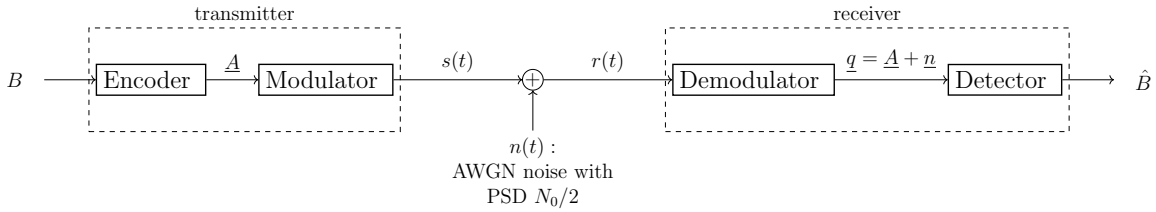
Analog modulation techniques

At the beginning of the course we saw there are two ways of representing information:

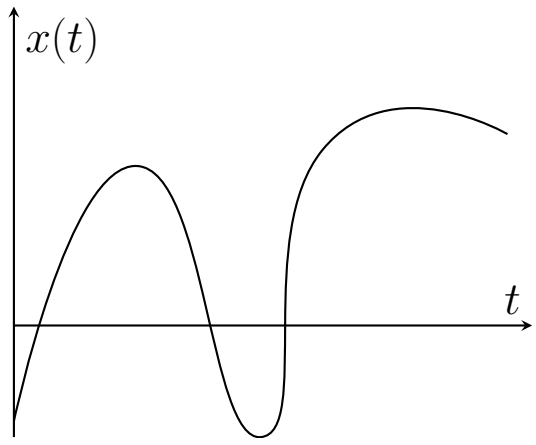
digital information is carried by *symbols* belonging to a finite alphabet, which are simply finite-duration signals.



We already know the model for this



analog information is carried by a continuous *waveform*



This could be either voice, video, audio...

Here we do **not** have the concept of symbol: the number of waveforms the transmitter can use is infinite and their duration is not limited.

In order to transmit information in analog format we have two possibilities

- discretize the signal and transmit it using a digital system
- transmit it directly using an analog system.

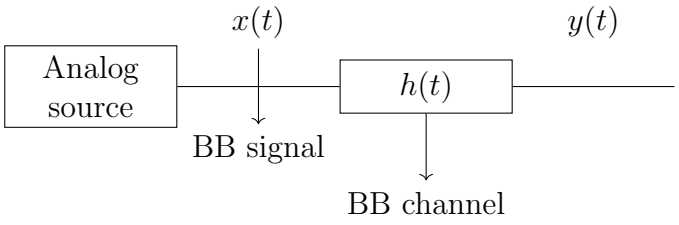
Nowadays, the usual approach is the first one. However, there still exist some analog communications systems, and that's why it's interesting to know how they work on a basic level.

Depending on the type of channel, transmission can be

- Baseband
- Passband

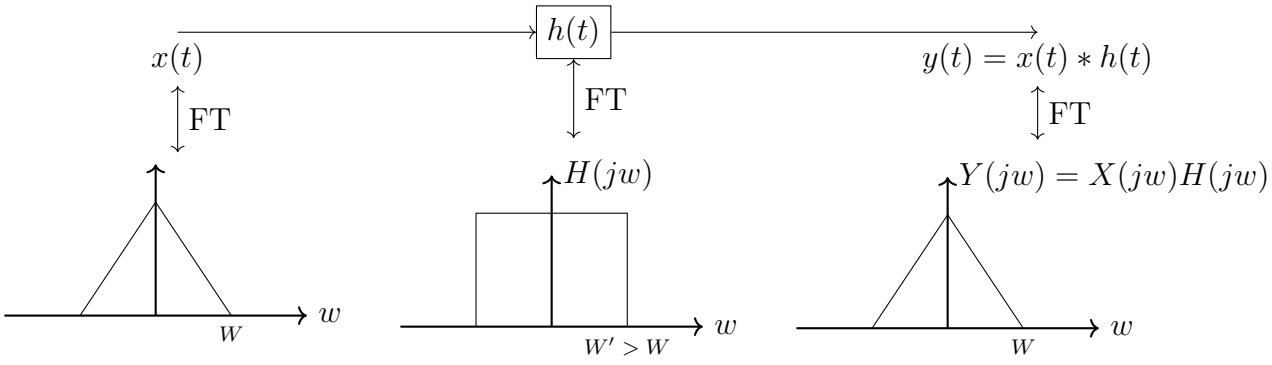
4.1 Baseband analog transmission

The information signal (a continuous waveform) is transmitted as is



Notice that we are denoting as $y(t)$ the signal at the output of the channel.

The channel acts as an LTI system



Some examples are:

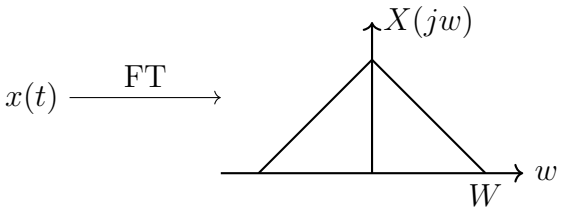
- telephone subscriber loop (not used that much nowadays...)

- public address systems
- closed-circuit TV

These system are in practice obsolete nowadays, and hence we are going to focus on...

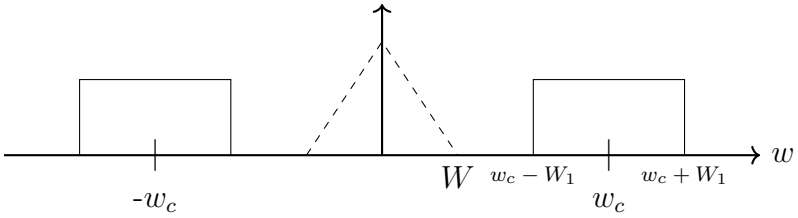
4.2 Passband analog transmission (modulation)

At the beginning of the course we talked very briefly about modulation and why it is necessary. Let us assume that we want to transmit the same baseband signal as before



- baseband signal
- bandwidth W

but now the channel is passband

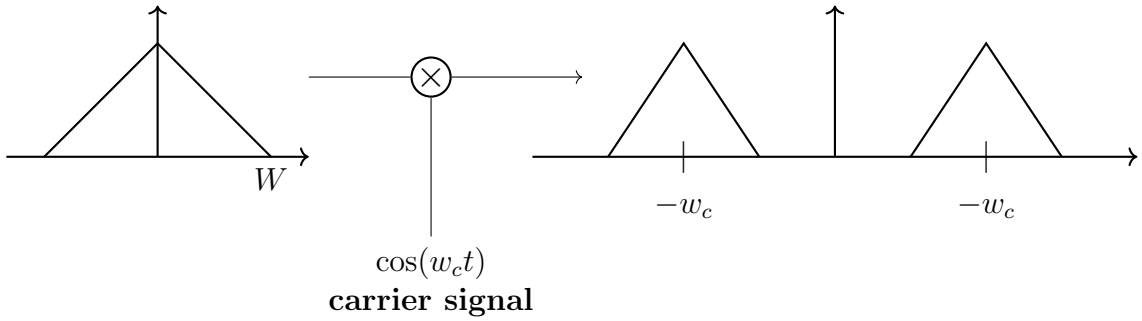


- **passband** channel
- bandwidth $2W_1$

with $w_c - W_1 > W$, that is, the spectrum of transmitted signal does not *fit* in the channel. Mathematically, the signal **cannot** go through the channel because

$$Y(jw) = X(jw)H(jw) = 0.$$

In order to transmit $x(t)$ through this channel, we need to *shift* (translate) the spectrum of $x(t)$, $X(jw)$, to the frequency band in which the spectrum of the channel is non-null...and that is achieved using a *carrier* signal



Mathematically,

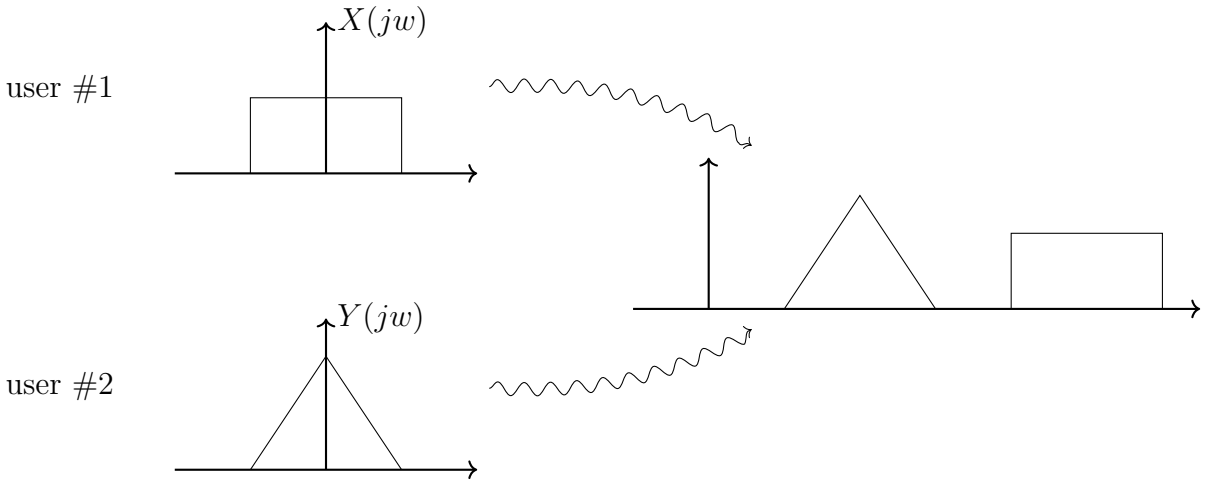
$$x(t)\cos(w_c t) \xleftrightarrow{DFT} \frac{1}{2}X(j(w - w_c)) + \frac{1}{2}X(j(w + w_c))$$

This operation is called **modulation**. Notice the three signals that come into play here

$$\underbrace{y(t)}_{\substack{\text{modulated}\\ \text{signal}}} = \underbrace{x(t)}_{\substack{\text{modulating}\\ \text{signal}}} \cdot \underbrace{\cos(\omega_c t)}_{\substack{\text{carrier}\\ \text{signal}}}$$

4.2.1 Goals of modulation

- To adapt the transmission to the characteristics of the channel (a passband channel will not allow baseband signals to pass through!!)
- To allow the transmission of multiple information signals



When you switch TV or radio stations, you are actually selecting a new band of frequencies.

- To spread the signals across a wider bandwidth. Sometimes, spreading the bandwidth of the transmitted signal provides some protection against noise or unauthorized users.

4.2.2 Types of modulation

For us, $x(t)$ is:

- a baseband signal, i.e., $X(jw) = 0, \forall\, |w| > W$
- a power signal, i.e., whose power is finite
- a realization of a band-limited WSS random process $X(t)$ with $S_X(jw) = 0, \forall\, |w| > W$

Transmission of signal $x(t)$ through the channel is achieved by embedding it in a carrier signal of the form

$$A_c \cos(\, w_c\, t + \phi_c)$$

amplitude — frequency — phase

and the signal $x(t)$, containing the information, can be *imposed* in

- the amplitude,
- the frequency, or
- the phase,

and we say that $x(t)$ modulates the amplitude, frequency or phase of the carrier.

In general, the modulated signal (that resulting from modulation, which is transmitted through the channel) has the form

$$\text{modulated signal} \equiv y(t) = r(t)\cos(w_c t + \varphi)$$

- if $x(t)$ is embedded in $r(t)$ ($\not\Rightarrow r(t) = x(t)$[1]) $\longrightarrow$ **linear or amplitude modulation**

  Then, we have a modulated signal of the form

  $$y(t) = r(t)\cos(w_c t + \varphi)$$

  with *constant* frequency and phase.

- if $x(t)$ is embedded in $\varphi(t) \longrightarrow$ **angular modulation**

  We have a modulated signal of the form

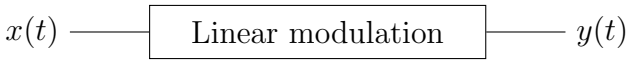
  $$y(t) = A\cos(w_c t + \varphi(t))$$

  with *constant* amplitude and frequency.

In both cases, the spectrum of the information signal is moved to another frequency band.

4.3 Linear or amplitude modulation

The information signal, $x(t)$ is imposed upon the amplitude of the carrier



$$\begin{aligned} y(t) &= r(t)\cos(w_c t + \varphi) = r(t)\left(\cos(w_c t)\cos\varphi - \sin(w_c t)\sin\varphi\right) \\ &= \underbrace{r(t)\cos\varphi}_{\substack{x_i(t) \\ \text{in-phase component}}}\cos(w_c t) - \underbrace{r(t)\sin\varphi}_{\substack{x_q(t) \\ \text{quadrature component}}}\sin(w_c t), \end{aligned}$$

where we have used that

$$\cos(A + B) = \cos A\cos B - \sin A\sin B.$$

This is the structure of the modulated signal in any linear modulation, and depending on how we choose the in-phase and quadrature components, we get a different kind of linear modulation.

We are going to rewrite the in-phase and quadrature components as a function of some parameters that will allow to decide the type of the linear modulation:

[1] ...since some transformation might be applied

- $x_i(t) = r(t)\cos\varphi = A_c + A_m x(t)$
- $x_q(t) = r(t)\sin\varphi = A_n \tilde{x}(t)$

where $\tilde{x}(t)$ is a certain transformation of $x(t)$.

Then, depending on the values of A_c, A_m and A_n, we will have different modulations:

- $A_c \neq 0$, $A_m \neq 0$, $A_n = 0 \rightarrow$ AM modulation (*conventional* amplitude modulation)
- $A_c = 0$, $A_m \neq 0$, $A_n = 0 \rightarrow$ DSB modulation (Double Side Band modulation)
- $A_c = 0$, $A_m \neq 0$, $A_n \neq 0 \rightarrow$ SSB modulation (Single Side Band modulation)

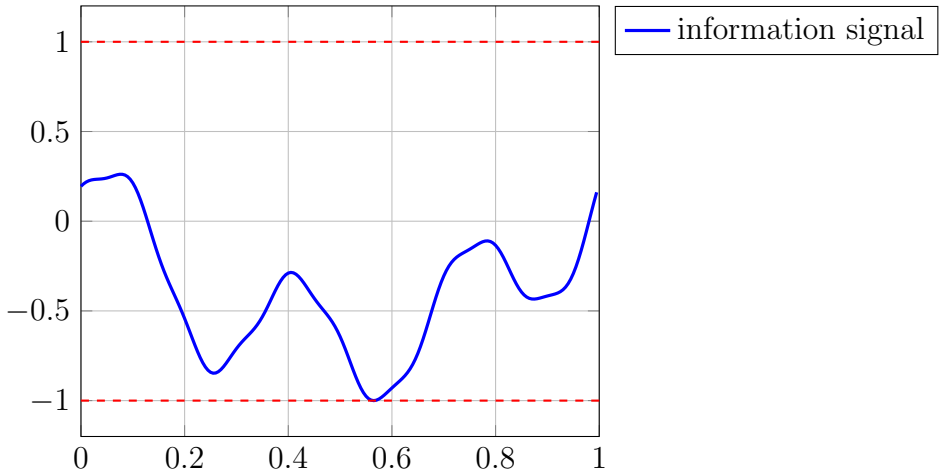
Notice that $A_n = 0$ means the quadrature component is null.

4.4 AM modulation

$$y(t) = (A_c + A_m x(t))\cos(w_c t) \tag{4.1}$$

Let us assume that $x(t)$ is normalized, that meaning

$$|x(t)| \leq 1$$



If it was not normalized, we could normalize it by doing

$$x_n(t) = \frac{x(t)}{\max|x(t)|},$$

but let us assume it already is.

Equation (4.1) can be rewritten as

$$y(t) = (A_c + A_m x(t))\cos(w_c t) \overset{A_m = A_m \frac{A_c}{A_c}}{=} A_c\left(1 + \frac{A_m}{A_c}x(t)\right)\cos(w_c t)$$
$$\overset{\frac{A_m}{A_c}=m}{=} A_c\,(1 + m x(t))\cos(w_c t)$$

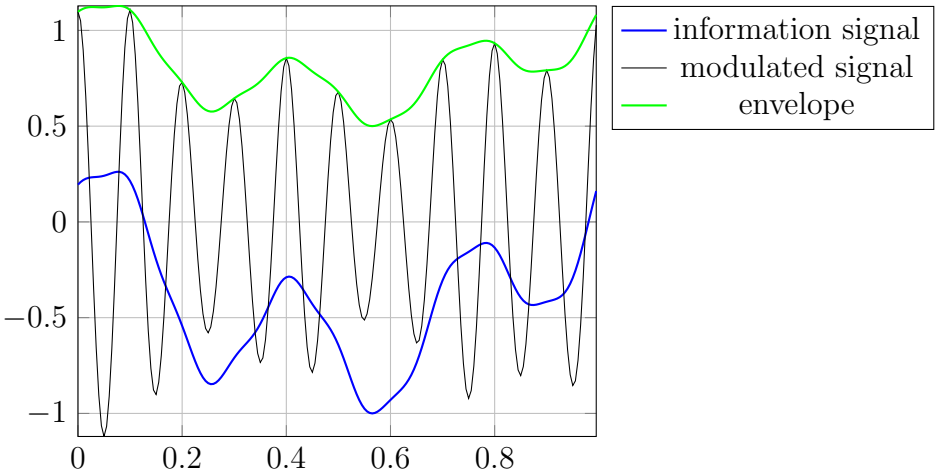
where

$$m = \frac{A_m}{A_c} \equiv \text{modulation index}$$

We would like $A_c\,(1 + mx(t))$ to be greater than or equal to zero, i.e.,

$$A_c\,(1 + mx(t)) \geq 0$$

Why? Because in such case the modulated signal is a cosine multiplied by a positive signal $\Rightarrow$ this positive signal is the **envelope** of the resulting signal and demodulation is very easy.



Notice that the cosine varies much more rapidly than $A_c\,(1 + mx(t))$, and somehow the positive peaks of the former are "sampling" the latter.

Envelope of a signal

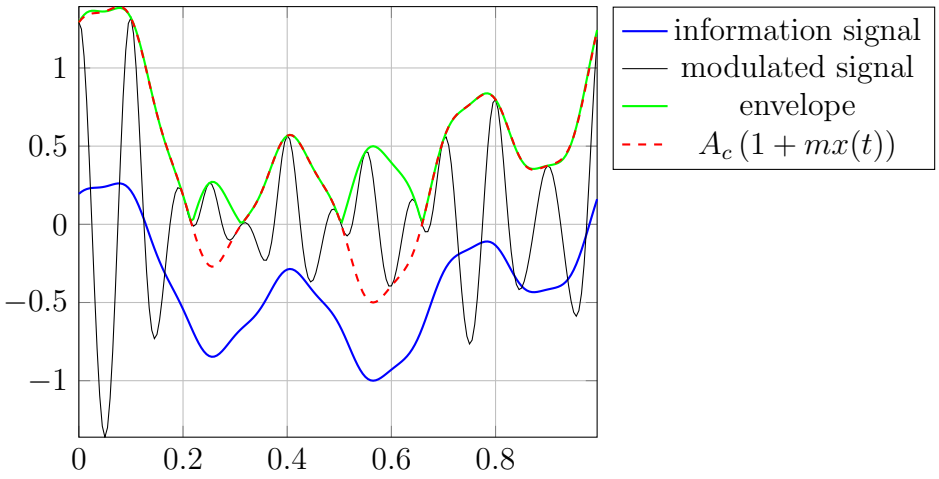
The **envelope** of an oscillating signal (for example, a cosine) is a *smooth* signal that outlines its extremes. We have upper and lower envelopes. Intuitively, by *smoothly* joining the peaks of the signal that are above 0 we get the upper envelope.

How can we make sure that $A_c\,(1 + mx(t)) \geq 0$? If the signal is normalized, i.e., $|x(t)| \leq 1$, then it is enough to choose $0 < m \leq 1$:

$$\left.\begin{array}{l} 0 < m \leq 1 \\ |x(t)| \leq 1 \end{array}\right\} \Rightarrow (1 + mx(t)) \geq 0 \Rightarrow \text{demodulation is very easy}$$

$$\begin{array}{ccccc} & \text{extract envelope} & & \text{solve for } x(t) & \\ & \uparrow & & \uparrow & \\ y(t) & \longleftarrow & A_c(1 + mx(t)) & \longrightarrow & x(t) \end{array}$$

What happens if $\boxed{m > 1}$? It might happen that $A_c\,(1 + mx(t)) < 0$ at some point, in which case



The envelope of the modulated signal is not $A_c\,(1+mx(t))$ anymore (some samples have changed sign and are now in the negative part). This phenomenon is called **overmodulation**,

$$m > 1 \Rightarrow \text{overmodulation},$$

and demodulation is not that easy anymore.

4.5 Angular modulations

The information is in the argument of the cosine signal, either in the phase or the frequency.

These are much more complex modulations than their linear counterparts. Sometimes, they can only be studied through approximations. Moreover, the required bandwidth is bigger. However, the advantage is that they are less affected by noise. They trade off bandwidth for immunity against noise.

The general expression for the modulated signal is now

$$y(t) = A\cos\left(w_c t + \varphi(t)\right) = A\cos\phi(t),$$

where

$$\phi(t) = w_c t + \varphi(t) \equiv \text{instantaneous phase (everything inside the cosine)}$$

From it, we have

$$\frac{d\phi(t)}{dt} = w_i(t)\frac{\text{rad}}{\text{second}} \equiv \text{instantaneous frequency in radians.},$$

which in hertz is

$$\frac{1}{2\pi}\frac{d\phi(t)}{dt} = f_i(t)\text{Hz} \equiv \text{instantaneous frequency.}$$

Doing some algebra, we get

$$\frac{1}{2\pi}\frac{d\phi(t)}{dt} = \frac{1}{2\pi}\frac{d\left(w_c t + \varphi(t)\right)}{dt} \overset{w_c=2\pi f_c}{=} f_c + \frac{1}{2\pi}\frac{d\varphi(t)}{dt}$$

If $x(t)$ is the information signal

- $\varphi(t) = \beta x(t) \longrightarrow$ Phase modulation (PM)

$$\beta \equiv \text{phase deviation constant}$$

- $f_i(t) = f_c + f_d x(t) \longrightarrow$ Frequency modulation (FM)

$$f_d \equiv \text{frequency deviation}$$

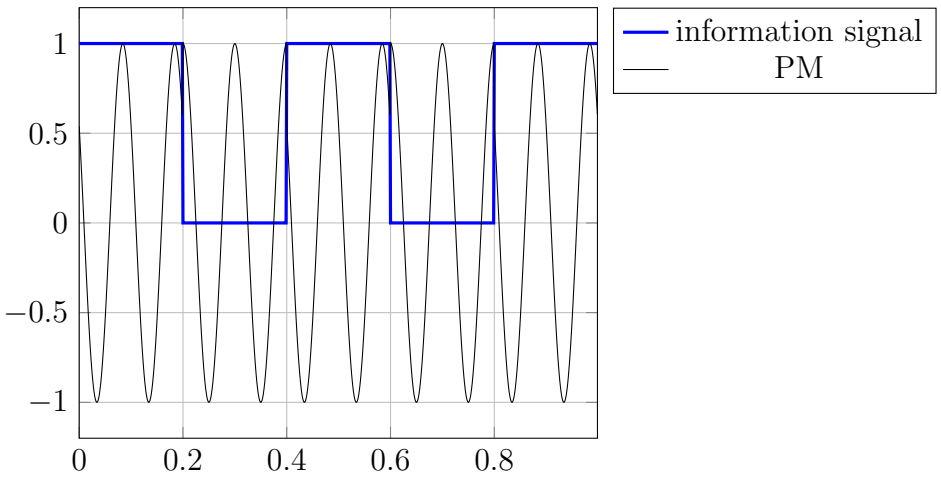
> **Caveat**
> The frequency being modulated is the *instantaneous* frequency, and **not** the carrier frequency.

It can be shown that in FM the above equation entails the phase varies according to

$$\varphi(t) = 2\pi f_d \int_{-\infty}^{t} x(u)du.$$

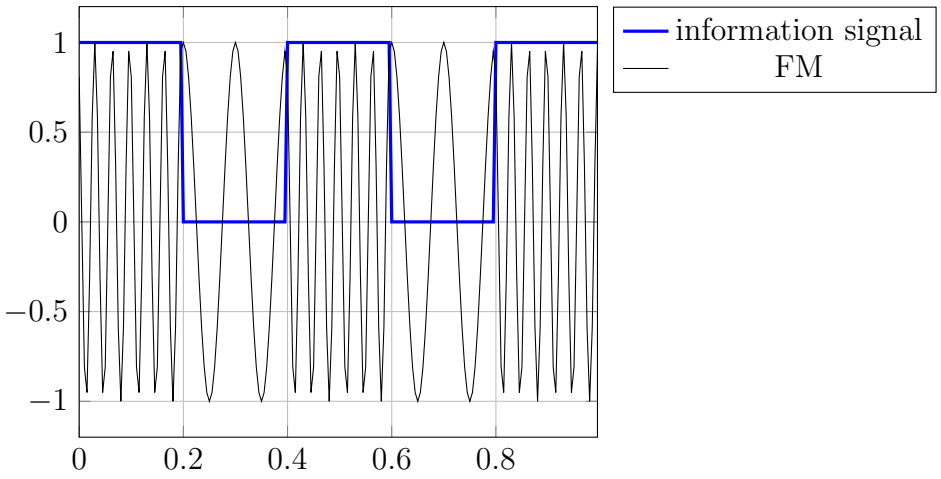
Hence,

- $y(t) = A\cos(w_c t + \beta x(t)) \longrightarrow$ PM



  Here, quick changes in the information signal, $x(t)$, indeed cause sudden changes in the modulated signal.

- $y(t) = A\cos\left(w_c t + 2\pi f_d \int_{-\infty}^{t} x(u)du\right) \longrightarrow$ FM



Here the integral will not allow the quick changes in $x(t)$ to cause sudden changes in the modulated signal

Index