

# Project: Performance Analysis of a realistic Cloud Radio Access Network (C-RAN)

Cédric Ware, Philippe Ciblat, Jean-Christophe Cousin, Chadi Jabbour,  
Germain Pham, Ghaya Rekaya-Ben Othman, Elie Awwad

2020-02-27 version v1.1-34-gd909654

## Contents

<b>1</b>	<b>General overview</b>	<b>3</b>
1.1	Learning objectives	3
1.2	System model	3
1.3	Wireless channel modeling	4
1.4	Wireless point-to-point communications	4
1.5	Optical point-to-point communications	11
1.6	Network communications	13
1.7	Octave program	14
<b>2</b>	<b>Deliverables</b>	<b>15</b>
D1	Simulation of system with ideal hardware	15
D2	Radio-frequency system and hardware modeling	16
D3	Optical point-to-point system performance	17
D4	Network simulation and global performance	18
<b>3</b>	<b>Practical information</b>	<b>19</b>
3.1	Planned schedule	19
3.2	Grading	19
<b>4</b>	<b>Appendix: guidelines and good practices</b>	<b>21</b>
4.1	Good practices: reports	21
4.2	Good practices: programming	24
<b>5</b>	<b>Appendix: BCH codes</b>	<b>26</b>

---

5.1 Encoder for BCH codes . . . . .	26
5.2 Decoder for BCH codes . . . . .	26

# 1 General overview

You will study information transmission between connected devices and a remote server through a Cloud Radio Access Network (C-RAN), which combines radio for access and optical fiber afterwards. The main framework will be the development of an Octave<sup>1</sup> simulator that models communication from the information bits to the physical signal; you will develop it, progressively deepening the model's complexity until you reach a realistic level, at which point you will then have to satisfy performance requirements that will be detailed in the deliverables below (chapter 2 on page 15). Furthermore, parts of the transmission chain will be studied on actual hardware, which will let you estimate calibration parameters for the simulator.

## 1.1 Learning objectives

The objective of the project is *not* the simulator in itself, but that you acquire the following knowledge and skills:

- Design an end-to-end communication system against performance requirements (such as error rates) taking into account realistic models of physical signal transmission (in electrical, radio and optical forms).
- Explain the mathematical techniques and models used for data communication in general (modulation, coding) and as a physical signal over specific devices (radio channel, optical fiber...)
- Implement a computer simulator of transmission chains.
- Work in small teams on a software project following best practices (individual deliverables and deadlines, code maintainability, clear presentation of results...)

## 1.2 System model

We will consider a Cloud Radio Access Network (C-RAN) composed by a set of nodes that may transmit information wirelessly to a basestation. This basestation is connected via an optical fiber to a remote server which corresponds to the final destination as described in Fig. 1.1.

In the project, we will consider that:

- the nodes have imperfect hardware devices;
- the wireless channels are modeled according to realistic configurations;
- the basestation may have dirty hardware devices and may or may not decode the information before forwarding it into the optical fiber;

---

<sup>1</sup>Free software equivalent to Matlab®.

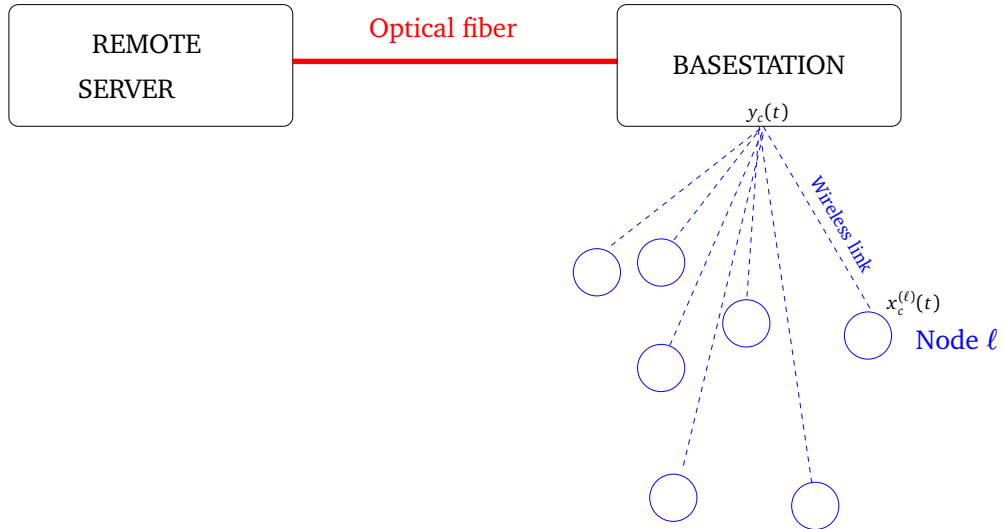


Figure 1.1: C-RAN system model

- the optical transmission over fiber may use either an intensity-modulated or a coherent system;
- the final server decodes the information.

The following sections will detail the modeling of these system parts and the C-RAN system as a whole. In terms of notations, we have

- $x_c^{(\ell)}(t)$  the carrier transmitted signal at node  $\ell$ ,  $x^{(\ell)}(t)$  the baseband corresponding signal.
- $y_c(t)$  the carrier received signal at the basestation,  $y(t)$  the baseband corresponding signal.

### 1.3 Wireless channel modeling

In this section, we introduce the model for one wireless link between a node and the basestation. We thus remove the node index  $\ell$ .

We consider in baseband a multipath channel with  $M$  paths, each of whose complex-valued attenuation is  $A_k$  and the delay is  $\tau_k$ .

Therefore we have

$$y(t) = \sum_{k=1}^M A_k x(t - \tau_k).$$

The channel described by  $\{A_k, \tau_k\}_{k=1, \dots, M}$  may be time-varying frame by frame where the frame size is  $N$  symbols.

Moreover, we assume throughout the project that the channels  $\{A_k, \tau_k\}_{k=1, \dots, M}$  are known at the basestation side and hence no estimation is required.

### 1.4 Wireless point-to-point communications

We model here the transmit and receive nodes. We consider single-carrier transmission. We first consider an ideal case where hardware devices are perfect and hence not considered.

### 1.4.1 Ideal simulations

We consider the following model given by Fig. 1.2.

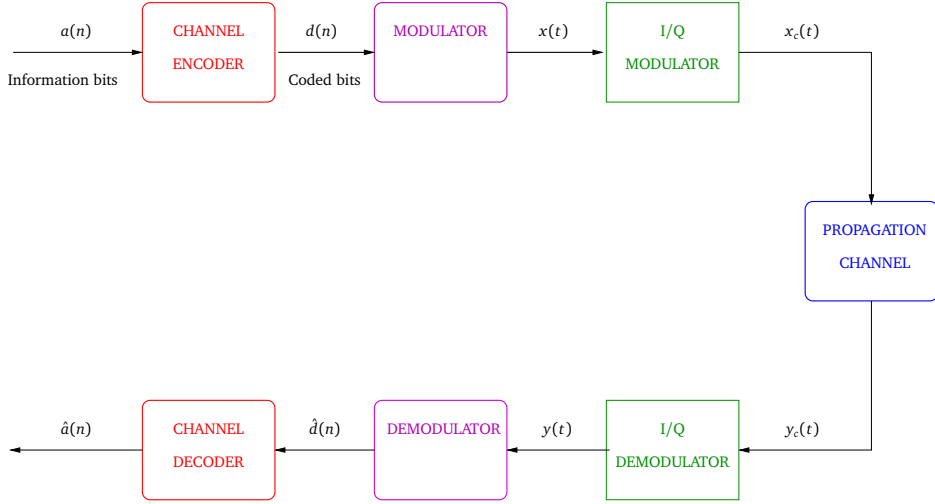


Figure 1.2: Ideal Transmitter and Receiver

At the transmitter side, we consider that

- the channel coding is a linear block coding described just below whose input is the information bits slotted  $a(n)$  of size  $K_c$  and the output is a binary sequence  $d(n)$  of size  $N_c$ . This binary sequence corresponds to the data to be sent. The considered codes are binary BCH codes of length  $N_c = 31$ . We will implement two codes:
  - the first BCH code corrects 1 transmission error ( $t = 1$ ),
  - and the second code corrects 2 transmission errors ( $t = 2$ ).

The standard form will be implemented to alleviate the decoding. We note that  $GF(32) = GF(2)[x]/(1 + x^2 + x^5)$ . The BCH codes have 2 different rates, which will allow to change the spectral efficiency of the transmission scheme.

- the modulation is linear which means that

$$x(t) = \sum_n s(n)g(t - nT_s)$$

where

- $s(n)$  are the symbols obtained via the binary sequence  $d(n)$ . The symbols may belong to any constellation (PAM, PSK, QAM).
- $T_s$  is the symbol period
- $g(t)$  is the shaping filter which satisfies the square-root Nyquist condition.

At the receiver side,

- the received signal in baseband is

$$y(t) = \sum_n s(n)p(t - nT_s) + v(t)$$

where

- $p(t)$  is the channel encompassing the shaping filter and the propagation channel,

$$p(t) = \sum_{k=0}^M A_k g(t - \tau_k)$$

- $v(t)$  is the white Gaussian noise

- we apply the matched filter related to  $g(t)$  only (in order to simplify). Consequently, we have

$$\tilde{z}(t) = \sum_n s(n)\tilde{h}(t - nT_s) + \tilde{v}(t)$$

where

- $\tilde{h}(t) = \overline{g(-t)} \star p(t)$
- $\tilde{v}(t)$  the colored noise with  $\tilde{v}(t) = \overline{g(-t)} \star v(t)$ .

- we then sample at the symbol rate  $1/T_s$ , which leads to

$$z(n) = \sum_{\ell=-L}^L h(\ell)s(n - \ell) + w(n) \quad (1.1)$$

with  $z(n) = \tilde{z}(nT_s)$ ,  $h(n) = \tilde{h}(nT_s)$  and  $w(n) = \tilde{v}(nT_s)$  a white noise with zero-mean and variance  $N_0$  (per complex dimension). In Eq. (1.1), we consider that the global filter  $h(n)$  has non-zero values over  $2L + 1$  symbol times.

Therefore if perfect hardware devices are used, we can assume that Eq. (1.1) holds. As InterSymbol Interference occurs, equalizers have to be employed.

In order to analyze the digital communications transmitters and receivers, we rewrite Eq. (1.1) by using the following matrix framework

$$\mathbf{z} = \mathbf{H}\mathbf{s} + \mathbf{w} \quad (1.2)$$

where

- $\mathbf{z} = [z(N), \dots, z(1)]^T$  with  $N$  being the frame size. To avoid Inter-Frame Interference, each frame is pre-fixed and post-fixed by a null sequence whose length is tailored according to the filter memory length. The superscript  $(\bullet)^T$  stands for the transposition.
- $\mathbf{s} = [s(N), \dots, s(1)]^T$  the data frame.
- $\mathbf{H}$  is a  $N \times N$  Toeplitz matrix described as follows

$$\mathbf{H} = \begin{bmatrix} h(0) & h(1) & \dots & h(L) & 0 & \dots & 0 \\ h(-1) & h(0) & h(1) & \dots & h(L) & 0 & \dots \\ \dots & 0 & h(-L) & \dots & h(-1) & h(0) & h(1) \\ 0 & \dots & 0 & h(-L) & \dots & h(-1) & h(0) \end{bmatrix}.$$

- $\mathbf{w} = [w(N), \dots, w(1)]^T$  the white Gaussian noise

We now apply various detectors:

- the threshold detector,
- the Zero-Forcing (ZF) equalizer,
- and the Decision-Feedback Equalizer (DFE).

The work to be done associated with this section 1.4.1 is described in the deliverable [D1 on page 15](#). The numerical values of each of the above-introduced parameters are also given in deliverable [D1 on page 15](#).

## 1.4.2 Hardware impairments

The purpose of this section of the project is to explore key baseband and RF analog impairments in a transceiver and to demonstrate how to model their impact from a communication system design point of view. The ultimate purpose is to optimize a mixed-signal platform by taking into account the characteristics of the RF/analog front-end.

There are two main non-idealities in electronic systems that degrade their resolution and precision: noise and non linearity. It is worth noting that there are many other non-idealities that exist in practice such as process variations and linear distortions but in the scope of the project, we will limit our studies to noise and non linearity.

RF Bandwidth	30 MHz
Useful Bandwidth	20 MHz
Central Frequency	2400 MHz
SNR receiver output	10 dB
Pout TX	20 dBm
Minimum distance	1.4 m
Maximum distance	1400 m
Matching resistance	50 $\Omega$
Temperature	290 K
ACPR Tx output	45 dB
ADC full scale	$\pm 1$ V
DAC full scale	$\pm 1$ V

Table 1.1: *General Specifications of the system*

### Non-Idealities

**Noise:** Noise is a random or a pseudo random signal that could be the result of a physical phenomenon such as thermal agitation (thermal noise) or of a systematic operation that leads to a pseudo-random error such as quantization. Most noise sources are white, i.e., have a constant power spectral density over the bandwidth of interest; but some, such as *flicker noise*, are “colored”, i.e. they have a non-flat frequency response. Noise can affect communication systems anywhere but for analysis purposes, noise impact will modeled by adding a random perturbation to the input signal.

**Non linearity:** In a linear system, the change of the output is strictly proportional to the change of the input. The huge majority of electronic systems are designed to be linear, however many imperfections make them in practice non-linear. The causes of non linearities are diverse, e.g., transistor's non-linear behavior, slew rate and some kind of Inter Symbol Interference (ISI). A large number of non-linear models exist in the literature, the most complete ones are called memory aware, i.e. the output does not only depend on the current value of the input but also on its past values. Among most employed memory aware models, we can cite the Volterra model and the memory polynomial model. In the scope of this project, for the sake of simplicity, we will limit our modeling to memoryless models such as the polynomial model described in Eq. (1.3):

$$y(t) = \sum_{k=1}^{K_{NL}} \alpha_k x(t)^k, \quad (1.3)$$

where  $K_{NL}$  is the non-linearity order,  $\alpha_k$  the  $k^{th}$ -order non-linearity coefficient,  $y(t)$  the output signal and  $x(t)$  the input signal. Note that this equation is given for real signals but similar expression will be used during the project for complex signals.

There are three main consequences of non-linearity: unwanted harmonics, inter-modulation products and gain compression. They were studied at length in the scope of course TELECOM201.

## Metrics

To measure the impact of noise and non-linearity on the performance of an electronic system, a large number of metrics could be used. A selection of the fundamental ones are introduced hereafter (few additional metrics will be introduced during labs):

**Signal to noise ratio (SNR)** It measures the ratio between signal power and noise power. It is usually expressed in dB:

$$SNR_{dB} = 10 \log_{10} \left( \frac{P_{Signal}}{P_{Noise}} \right). \quad (1.4)$$

**Signal to noise and distortion ratio (SNDR)** is similar to SNR but takes into account distortions also.

$$SNDR_{dB} = 10 \log_{10} \left( \frac{P_{Signal}}{P_{Noise} + P_{Distortions}} \right). \quad (1.5)$$

The  $SNDR$  can be easily computed from time-domain simulations or from measurements. Two approaches are available according to the knowledge of the input signal:

- Time domain: Any deterministic signal could be used as an input, the  $SNDR$  is then given by the following expression

$$SNDR = 10 \log_{10} \left( \frac{\sum_{n=1}^N x[n]^2}{\sum_{n=1}^N (x[n] - y[n])^2} \right),$$

where  $N$  is the number of samples of observation.



- Frequency domain: A sinewave should be used as an input signal, this allows to distinguish the useful signal from the noise and non linearity in the output spectrum. The  $SNDR$  is then given by the following expression

$$SNDR_{dB} = 10 \log_{10} \left( \frac{PSD(f_{in})}{\int_B PSD(f)df - PSD(f_{in})} \right),$$

where  $f_{in}$  is the frequency of the input sinewave,  $B$  the band of interest and  $PSD$  the power spectral density.

**1 dB Compression Point** Due to physics the output of an amplifier cannot exceed the level of the DC power that makes it work. So, when the input of an amplifier is too large, its output is clipped to a constant value. In this situation, the amplifier is said to be saturated. In semiconductor devices, the transition between the linear region and the saturated region is more or less smooth and the value of the gain is said to be *compressed* since its value decreases as the input amplitude increases.

The 1 dB Compression Point is obtained when the input power is high enough to have the actual gain of the amplifier 1 dB lower than the ideal gain in the linear conditions. In the following figure, this point is denoted as  $P_{1dB}$  which is actually the output value ; this output value corresponds to a specific input value (which is conveniently denoted  $IP_{1dB}$ ).

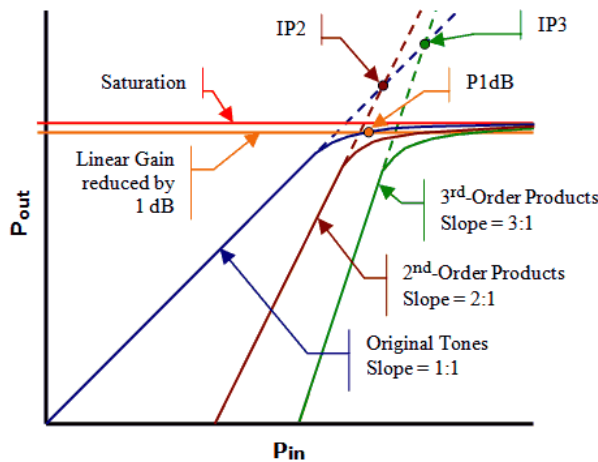


Figure 1.3: Illustration of the 1 dB Compression Point (and others non-linearity metrics)

**Adjacent Channel Power Ratio (ACPR):** When a transmitter is emitting on a given channel, power is also emitted or leaked on adjacent channels due to non-linearity and noise. The ACPR is a metric that measures this leakage. This measure is important to make sure that the disturbance to adjacent channels' users is lower than the value fixed in the standard. We can compute the ACPR from the power of the main useful channel and the power of the unwanted signal that is in the adjacent channel.

$$ACPR = 10 \log_{10} \left( \frac{P_{main\ channel}}{P_{adjacent\ channel}} \right) \quad (1.6)$$

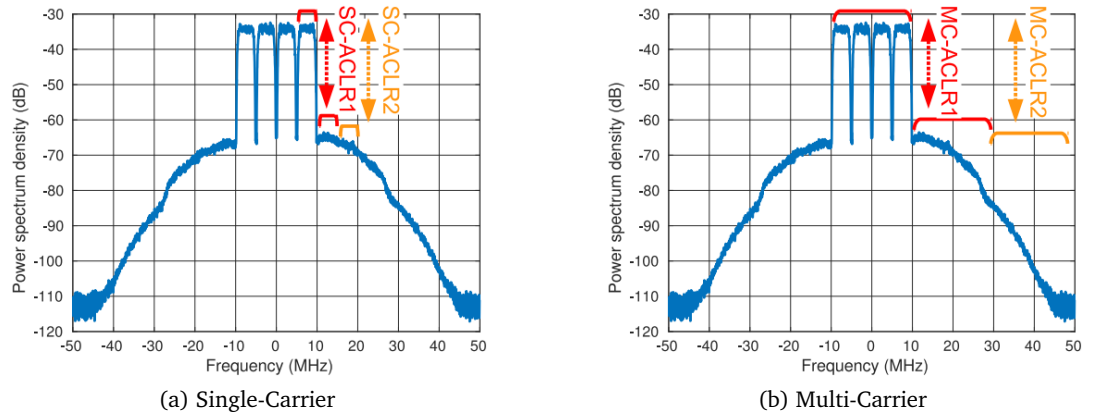


Figure 1.4: Representation of ACPR computation

### Operation and sizing

An appropriate sizing and design of the RF chains are mandatory in order to guarantee the signal integrity at the transmitter output or at the receiver output. Each of the blocks of the transmit and receive chains have a specific function but due to noise and/or nonlinearity they will degrade the quality of the signal (which can be measured with the SNDR). The Table 1.2 describes each block of the transmit and receive chains along with their main characteristics. Please refer to Fig. 1.5 while browsing the items of the list.

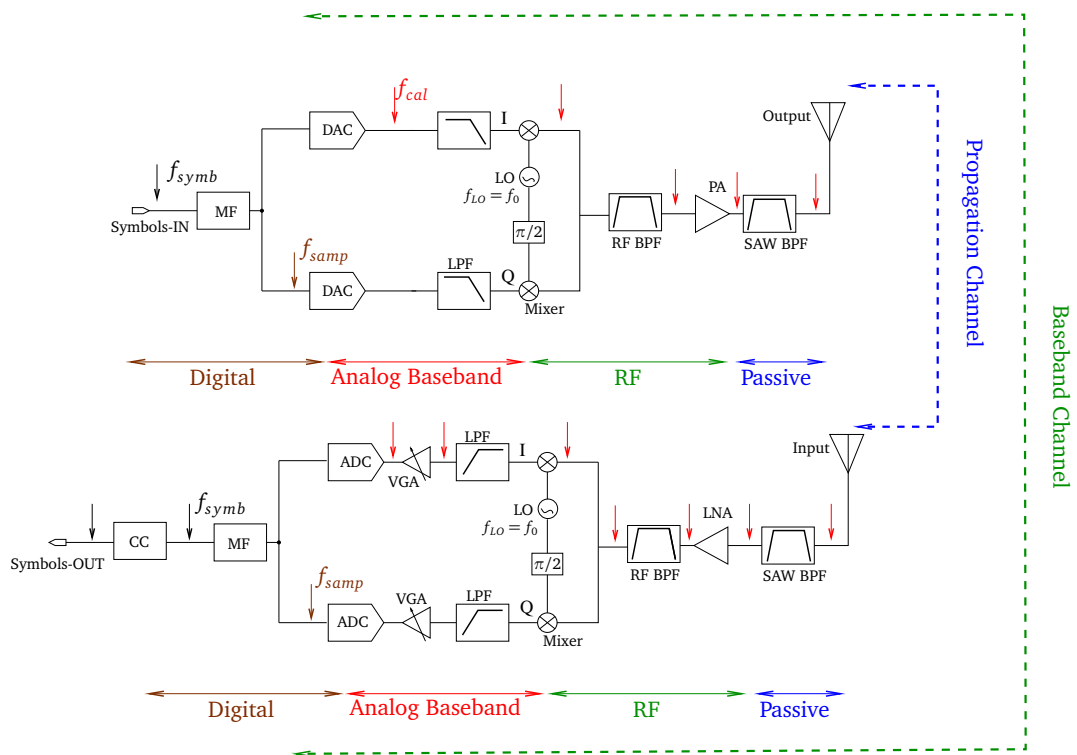


Figure 1.5: Full transmission chain

Block	Function	Main non idealities	Specification
<b>Transmitter</b>			
Digital to Analog Converter (DAC)	Converts the signal from digital to analog	Quantization Noise, Thermal Noise, nonlinearity due to mismatch	Number of bits, fs, Full Scale
Reconstruction filter	Filter DAC replicas to avoid poluting adjacent chan.	Thermal Noise	Order, Bw Filter approximation
Up-Mixer	Up-mixes the signal around the LO frequency	Phase noise, nonlinearity	IIP2
Power Amplifier (PA)	Amplifies the signal before transmtion	Non linearity	Gain, P1dB, IIP3
<b>Receiver</b>			
Low Noise Amplifier (LNA)	Amplifies the signal at the receiver input	Noise	Gain, noise figure
Down-Mixer	Down-mixes the signal from the LO frequency to DC	Phase noise, nonlinearity	IIP2
Anti-Alias filter (AAF)	Filter interferers before the sampling to avoid aliasing	Thermal Noise	Order, Bw Filter approximation
Baseband Gain	Amplifies the signal to occupy the full scale of the ADC	Thermal noise	Gain, Noise Figure
Analog to digital Converter (ADC)	Converts the signal from analog to digital	Quantization noise, Thermal Noise	Number of bits, fs, Full Scale

Table 1.2: *Blocks of the transmit and receive chains with their main characteristics*

Associated deliverable: [D2 on page 16](#).

## 1.5 Optical point-to-point communications

We assume that the basestation forwards aggregated information to a remote server in the cloud, as light over an optical fiber. The information forwarded may be either the users' data after extracting it from the radio channel (decode-and-forward architecture), or the raw radio waveform (quantize-and-forward architecture). The latter saves cost on the RF hardware and signal processing at the basestation, but puts a heavier strain on the optical communication system. Since this is part of the access network, where operating and capital expenses (OpEx and CapEx) are shared by fewer users, the need to keep costs low is emphasized. Must you pay for a heftier optical system, or can you improve your digital communications techniques to accomodate a cheaper one, without losing on processing costs?

### 1.5.1 Issues with optical fiber

We can assume that the fiber used is a standard singlemode fiber according to the G.652 standard. The main physical impairments for a signal traveling over such a fiber are:

- Attenuation: the signal loses power exponentially with propagation length. The minimum

---

attenuation is about 0.2 dB/km, for a signal in the C band (carrier frequency around  $193 \pm 3$  THz, corresponding roughly to 1530 – 1560 nm in wavelength); attenuation will be higher outside that range.

- Dispersion: a monochromatic wave propagates as  $e^{i(\omega t - \beta z)}$ , but  $\beta$  varies with  $\omega$ , and is usually approximated by a second- or third-order Taylor series around the carrier frequency:

$$\beta(\omega) = \beta_0 + \beta_1(\omega - \omega_0) + \frac{1}{2}\beta_2(\omega - \omega_0)^2 + \frac{1}{6}\beta_3(\omega - \omega_0)^3 + \dots$$

You will measure  $\beta_2$  over a standard fiber in the C band and use it in simulation. Dispersion can be alleviated in the O band (around 1300 nm), where it is lowest, at the cost of a higher attenuation.

- Polarization: the direction of the optical field is not conserved in the optical fiber due to random stressing and twisting. This is not an issue with systems based on direct detection (see below) but requires coherent-detection systems to be MIMO.
- Other physical impairments, chiefly nonlinearity, can be neglected for access and even metropolitan networks; but they can be considered as a limit on the maximum optical power that can be injected into the fiber: about 10 dBm in the C band; or 0 dBm in the O band.

### 1.5.2 Issues with optical transmission systems and devices

Apart from the optical fiber itself, the optical or opto-electronic elements used in a transmission system are lasers, modulators, photoreceivers and filters or demultiplexers. Depending on their arrangement, they can be classified into the following categories:

- Laser: converts an electrical current into optical power. Semiconductor lasers typically have a static characteristic  $P(I)$  with a threshold of  $I_{\text{th}} \simeq 20$  mA, above which it is linear up to about 5 times  $I_{\text{th}}$ , reaching powers up to about 10 dBm. In dynamic regimes, the response has a resonant frequency of around 10 GHz, variable with the DC current. (See TELECOM203 lecture and tutorial on semiconductor lasers.) In addition, lasers emit intensity and phase noises, and their optical frequency varies with the bias current, temperature and other external factors.
- Direct modulation: information is imparted onto the optical signal by modulating the bias current of a laser. This solution is a very simple and cheap implementation of intensity modulation. However, a phase and frequency modulation is also imparted; and at modulation frequencies in excess of about 1 GHz, the impulse response of the laser must be taken into account.
- External modulation: information is imparted onto the light from a CW laser by a separate voltage-controlled modulator. Different modulator types operate on optical intensity, phase or both. Typical bandwidths are on the order of 20–50 GHz, though power amplifiers are usually required to produce enough voltage at frequencies above a few GHz.
- Photoreceiver, direct detection: converts optical power into electrical current, with a typical sensitivity of 1 A/W at 1550 nm. Adds a white noise roughly proportional to the temperature (thermal noise).

- Coherent detection: allows access to the optical field's complex amplitude, instead of only the intensity. A local laser (or local oscillator, LO) is used to interfere with the received signal. If the optical field is  $Ae^{i\phi}$ , then 2 interference paths with different phase shifts sent to a pair of balanced photodetectors yield  $A\cos\phi$  and  $A\sin\phi$ . A full implementation also requires 2 polarization channels, for a total of 8 photodetectors. In theory, this also requires the optical frequency of the LO to be locked to the signal's carrier frequency; this is exceedingly hard to ensure, so in practice the LO is left unlocked, which results in an  $e^{i\Omega t}$  phase shift on the signal with a random frequency offset  $\Omega$ . Signal processing techniques allow to track this phase shift up to a few 100 MHz.
- Optical filters and demultiplexers: allow parts of the optical spectrum to be blocked or separated. Unlike their electronic counterparts, they can be assumed to have a typical insertion loss of about 5 dB, bandwidths greater than a few nm (a few 100 GHz), and slopes no sharper than 20 dB/nm, at least for devices cheap enough to be used in access networks.

Associated deliverable: [D3 on page 17](#).

## 1.6 Network communications

We will first consider the performance between the nodes and the basestation in a standard cellular network.

### 1.6.1 Cellular Network

We consider  $K$  active users. In order to manage the multiple access between these users, we consider a very simple technique: the so-called Time-Division Multiple Access (TDMA). Each TDMA superframe is divided into  $K$  frames, one per user. Each frame is of length  $N$ . The communication is done on the bandwidth  $B$  and the carrier frequency is denoted by  $f_0$ .

For each user, we have a certain Quality of Service (QoS) described as follows:

- The rate of each user  $\ell$  is fixed and denoted by  $R_\ell$ . If needed, we may consider four user modes:
  - $R_\ell = 40$  kbit/s (2G-GPRS/EDGE)
  - $R_\ell = 400$  kbit/s (3G-WCDMA/UMTS)
  - $R_\ell = 4$  Mbit/s (3G-HSPA/H+)
  - $R_\ell = 40$  Mbit/s (4G)
- The Frame Error Rate (FER) –as soon as a bit in a frame is wrong, the entire frame is assumed to be not well detected and discarded–. The access control will reject the user if its FER is greater than a pre-defined threshold  $F_{\max}$ . We will consider  $F_{\max} = 10^{-2}$ .
- The power used for each transmitted symbol, or equivalently the symbol energy  $E_s$ , is upper-bounded by  $E_{\max}$ . The access control will reject the user if its symbol energy is greater than  $E_{\max}$  in order to satisfy the rate constraint  $R_\ell$  and the FER constraint  $F_{\max}$ . We will consider that the maximum average power (when the device is active) is  $P_{\max} = 20$  dBm.

---

In order to compute a network, we need to locate the nodes around the basestation according to a certain model. We will use a uniform distribution of the  $K$  users within a square whose side measures 2 km and the basestation is located in the center of the square. In practice, the square corresponds to a cell. The different figures that will be defined in D4 have to be averaged over various realizations of the users' locations.

Given one realization of the users' locations, find the corresponding pathloss between each user and the basestation. Given the pathloss, you have to select randomly with a uniform distribution one channel impulse shape out of the three channels described in deliverable [D1 on the next page](#). The channel impulse response  $h$  has to be rescaled according to the pathloss given by the Friis equation where the antenna's gains are equal to 1.

You have to implement this scheme with the following values: bandwidth  $B = 30$  MHz, including roll-off  $\rho = 0.5$  and carrier frequency  $f_0 = 2.4$  GHz. We will also consider that  $N_0 = -174$  dBm/Hz.

### 1.6.2 Cloud Radio Access Network

Now we focus on the whole system where the basestation plays a role of relay from the user to the remote server. Actually, we apply the Decode and Forward (DF) approach where the basestation attempts to decode all the streams coming from the nodes associated with it (so by decoding the users assigned on this cell). Then these streams are re-encoded to go into the optical fiber.

Associated deliverable: [D4 on page 18](#).

## 1.7 Octave program

Your simulator will be implemented in Octave, an open-source environment and language for scientific computing mostly compatible with Matlab<sup>®</sup>. It is not explicitly forbidden to use Matlab for development (using the school's or your own legally-purchased licenses); but your code will be tested in Octave<sup>2</sup>, and so are the functions and code samples that will be provided to you.

Depending on the specific deliverable, you will usually implement simple models for various parts of the system. Then you will be provided with functions for more complex models, or if you didn't succeed in implementing the simple models correctly but need to move on to the next deliverables. (Your grade will of course depend on your results.)

---

<sup>2</sup>Except for the hardware tests of the AD-FMCOMMS3-EBZ Board which, for now, requires the use of Matlab

## 2 Deliverables

For all deliverables, unless otherwise indicated, you will be expected to produce a short report detailing the results of your assigned studies; and demonstrate the programs that you used to obtain said results. Both must conform to the guidelines and good practices outlined in chapter 4.

Most deliverables have both minimum requirements and extra-credit questions. While you are not required to address the latter in each deliverable, you are expected to do at least some of the extra credit over the whole project. Grading rules are specified section 3.2.

### D1: Simulation of system with ideal hardware

For this deliverable, the Signal-to-noise Ratio (SNR) is defined as the  $E_b/N_0$  where  $E_b$  is the energy consumed to send one information bit. We remind that  $N = 100$ .

#### Coding part

- After calculating the generator polynomial of the considered BCH codes, implement the standard coding operation (systematic shape), using the coding method provided in the Appendix.
- Implement an optimal ML decoding for both codes.
- Plot the Bit Error Rate (BER) versus  $E_b/N_0$  of the uncoded BPSK case and both BCH codes (modulated with BSPK) for a AWGN channel. The range of BER could be between 0 and 10dB. Compare the empirical coding gains with the theoretical ones.

**Modulation part** We consider three channels for the communication. Once one channel is selected, it is applied over all the frames.

- Channel 1:  $\mathbf{A} = [1, 0.1, 0.1, 0.1, 0.1]$ ,  $\boldsymbol{\tau} = [0, 0.5, 1, 1.5, 2] \times T_s$ ,
- Channel 2:  $\mathbf{A} = [1, 0.8, 0.6, 0.4, 0.2]$ ,  $\boldsymbol{\tau} = [0, 0.5, 1, 1.5, 2] \times T_s$ ,
- Channel 3:  $\mathbf{A} = [1, 0.8, 0.8, 0.8, 0.8]$ ,  $\boldsymbol{\tau} = [0, 0.5, 1, 1.5, 2] \times T_s$ ,

We remind that the global continuous-time filter  $\tilde{h}$  is given by

$$\tilde{h}(t) = \sum_{k=0}^M A_k \tilde{g}(t - \tau_k)$$

where  $\tilde{g}(t) = \overline{g(-t)} \star g(t)$  is a Nyquist filter, and more precisely, a raised cosine with roll-off  $\rho = 0.5$ .

---

In practice, according to Eqs. (1.1)-(1.2), we only need to implement the discrete-time version of  $\tilde{h}$ , denoted by  $h$ , and given by

$$h(m) = \tilde{h}(mT_s) = \sum_{k=0}^M A_k \tilde{g}(mT_s - \tau_k).$$

Finally,  $h$  is normalized so that  $\sum_{m=-L}^L \|h(m)\|^2 = 1$ .

- Display  $h$  for channel 1, channel 2, and channel 3. We consider that  $T_s = 0.05\mu\text{s}$ .
- Plot the BER versus the SNR for each channel with three different equalizers (threshold, ZF, DFE) when BPSK is used. Typically consider a range of SNR between 0 and 20dB.
- Plot the BER versus the SNR for each channel with three different equalizers (threshold, ZF, DFE) when 16-QAM is used. Typically consider a range of SNR between 0 and 30dB.

### Extra credits

- Apply coding scheme into the frequency-selective channel configurations given by the three channels. Once again, the channel is kept constant over the frames.
- Apply an Automatic Request (ARQ) scheme with at most 1 retransmission. A retransmission is run for the frame if this frame has not been correctly decoded previously. A frame is perfectly decoded if each information bit is correctly decoded. As figure of merit, we consider the throughput (number of information bits belonging to correctly decoded frame over the number of transmitted bits). Plot the throughput versus the SNR without ARQ and with one-retransmission ARQ. Test the uncoded case, first, and if time available, the coded case. For all the simulations, use the channel 2 with the three receivers (threshold, ZF, and DFE).

## D2: Radio-frequency system and hardware modeling

In this section, you will use hardware devices such as AD-FMCOMMS3-EBZ Board<sup>1</sup> and various small base station power amplifiers. The following tasks and deliverables should not be executed in the order they appear.

- Provide a description of each blocks of the AD-FMCOMMS3-EBZ Board with the support of slides.
- Measure the noise factor of the AD-FMCOMMS3-EBZ Board for different frequencies.
- Plot the measured AM/AM and efficiency characteristics of the PA allocated to your group.
- Plot the measured SNR at the receiver as a function of the distance between the transmitter and the receiver:
  - in the 3B corridor and in the garden of the school;
  - with and without the PA allocated to your group.

---

<sup>1</sup><https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms3-ebz>



- Extra credit: plot the BER alongwith the SNR plots.
- Extra credit: Provide the complex polynomial model of the PA allocated to your group from the measurements.
- Extra credit: demonstrate the possibility of listening broadcast radio frequencies

**Focus point** For each focus point course, the basic performance plots of the studied block must be provided.

- Focus point PA: With a Vector Analyzer Network (VNA), measure the gain, IP1 and the AM/AM and AM/PM characteristics. (See specific subject for the corresponding lab session.) Results are to be used in the model of distortion due to the non-linearities of the PA to estimate the IM3 and ACPR.
- Focus point LNA: With a Vector Analyzer Network (VNA) and Spectrum analyzer, measure the gain and the Noise Figure NF. (See specific subject for the corresponding lab session.)

### D3: Optical point-to-point system performance

**Focus point** measure the dispersion of a standard optical fiber. (See specific subject for the corresponding lab session.) Results are to be used in the simulation parameters.

- Implement a simulator function to model propagation in an optical fiber with given attenuation (dB/km) and dispersion ( $\beta_2, \beta_3$ ). You can use the `fft` function to work in the spectral domain; you will determine the conditions under which this is legitimate (e.g. length of null sequences at the beginning and end of the sample vector representing the signal). Check its effects on a short pulse.
  - Extra credit: also check it by reproducing the work you did in the lab session.
- Using the given functions `TX_optical_em1` and `RX_photodetector`, study the BER of an optical system back-to-back (without fiber) using OOK intensity modulation and direct detection at  $R_b = 2.5$  and 10 Gbit/s; deduce the required optical powers at the photodetector in each case to get a BER lower than  $10^{-3}$  and  $10^{-6}$ , without equalization or error correction.
- Model the same system over up to 100 km of fiber. Show that the BER increases abruptly after a certain length, even if you maintain a constant power at the RX.
  - Extra credit: study the dependence of  $L_{\max}$  in  $R_b$ . Compare it with a simple analytical model based on how much pulses broaden due to dispersion.
  - Extra credit: propose and study solutions to reach 100 km at 10 Gbit/s with an uncorrected BER lower than  $10^{-3}$ .
- Using the given function `TX_optical_dm1`, study the performance of a directly-modulated optical system at 1 Gbit/s, back-to-back and over 20 km.

---

## D4: Network simulation and global performance

We define the rejection rate as the number of unserved users due to a constraint outage over the total number of users denoted  $K$ . The constraint outage comes from the power outage which occurs when the required power to do the transmission is larger than  $P_{\max} = 20$  mW for ensuring a FER of  $F_{\max} = 10^{-2}$ .

We consider that all the users use the same mode and the same receiver. The  $K$  users are uniformly distributed within a square whose side measures 2km and the basestation is located in the center of the square.

We remind you of the following values:  $B = 30$  MHz, including roll-off  $\rho = 0.5$ , the carrier frequency  $f_0 = 2.4$  GHz and  $N_0 = -174$  dBm/Hz.

- Compute the rejection rate at the basestation versus  $K$  for 8 configurations defined by a mode (one out of the four rate values  $R_\ell$ ) and a receiver (one out of the two equalizers: ZF or DFE) by following these steps:
  - To evaluate the presence or absence of a constraint outage, build a Look-Up Table given the minimum required  $E_s/N_0$  (in dB) at the receiver side leading to a FER of  $F_{\max}$  for each constellation (BPSK or 16-QAM), each channel (one of the three models defined in D1), and each receiver (ZF or DFE).
  - Evaluate the maximum  $E_s$  (according to the  $P_{\max}$  constraint) and deduce the maximum available  $E_s/N_0$  (in dB) at the transmitter side.
  - Given a configuration, select randomly a set of users' locations (for a given  $K$ ) and find the corresponding pathloss between each user and the base station. Then, select randomly with a uniform distribution one channel impulse response, apply the corresponding pathloss and count the number of users in outage. Average over a lot of realizations. Finally plot the rejection rate versus  $K$ . Take  $K$  within the interval  $[1, K_{\max}]$  where  $K_{\max}$  depends on the user mode (for instance, if  $R_\ell = 40$  kbit/s, then  $K_{\max} = 2000$ ).
- Design an optical-fiber transmission system connecting the basestation to the remote server, depending on  $K$  and the server distance (20 km or 100 km away).

**Extra credit** We consider that we have two adjacent cells having  $\tilde{K}$  users each using the same band as the cell of interest (as in 3G or 4G). We consider a 2G-mode for any user and a ZF receiver.

- What is the loss of performance (in terms of FER) if the multi-cell interference is not managed and seen as an additional noise? We assume  $\tilde{K} = K$ .
- Assuming that  $\tilde{K} = 0$  (so adjacent cells are empty). Then the basestation of adjacent cells will help the cell of interest by also decoding the users of the cell of interest and by offloading the resulting frames into the remote server. Then the remote server decodes three versions of the same frame. By applying a majority vote based decoder on each bit at the remote server, what is the gain in performance (in terms of FER).

## 3 Practical information

### 3.1 Planned schedule

Semester S2, Period P3	TH1–2	Simulation with ideal hardware
	TH3–4	Simulation with ideal hardware
	Deliverable <a href="#">D1</a>	
	TH5–6	DAC + ADC
	TH7–8	DAC + ADC Filtering at RX and TX
	TH9–10	Mixer at RX and TX
	TH11–12	Lab sessions: optics (TELECOM203 + focus on fiber dispersion)
	TH13–14	Power amplifier + low-noise amplifier
	TH15–16	Lab sessions: optics (TELECOM203 + focus on fiber dispersion)
Semester S2, Period P4	TH17–18	Integration + optimization
	Deliverable <a href="#">D2</a>	
	TH19–20	Optics: dispersion in fiber
	TH21–22	Lab sessions: optics (TELECOM203 + focus on fiber dispersion)
	TH23–24	Optics: dispersion in fiber
	TH25–26	Optics: direct vs external modulation
	Deliverable <a href="#">D3</a>	
	TH27–28	Network and global performance
	TH29–30	Network and global performance
	TH31–32	Presentation + code review Deliverable <a href="#">D4</a>

Each deliverable is due on the day before the session that follows it. You will then set a meeting with professors in the following days to review your results and code. For the final deliverable, this review will take place in the same session as your presentation.

### 3.2 Grading

- 15 % for each deliverable, 20 % for the final presentation, 20 % for at least 4 “extra-credit” questions (see below).
- In general, each deliverable will be graded as:
  - 25 % results successfully obtained,
  - 25 % interpretation and exploitation of the results,
  - 25 % explanation of the models and mathematical techniques used,

- 
- 25 % quality of the code and of the report.
  - Each deliverable includes questions marked “extra credit”, which count towards the extra-credit grade instead of the deliverable itself. A minimum of 4 such questions must be answered over the whole project to get a perfect grade.

## 4 Appendix: guidelines and good practices

### 4.1 Good practices: reports

#### 4.1.1 The one rule guide to making good charts

Making good charts is child's play. However, when it actually comes to producing one by ourselves, major failures usually occur<sup>1</sup>.

Here is a minimalist guide on good practices for drawing a good chart<sup>2</sup>. We will limit ourselves to the elements that are really essential to a good layout. (The advanced reader who wants to improve on the topic is encouraged to read Loren Shure's excellent "Making Pretty Graphs" posts [1])

If there is a single one idea to remember from this mini-guide, it is "**MAKE LEGENDS**". This applies to the three elements that make up the figure: its **title**, its **axes** and its **plot lines**.

Let's walk you through the step-by-step process of making a good chart.

We have a table of values, derived from measurements or calculations.

```
mois = [1:12];
temp_minimale_paris = [-4.3 9 5.9 9.3 7.4 15.6 14.6 16.7 13.8 13 5 0.8];
temp_minimale_papeete = [25.5 25 26.2 25.3 23.9 23.9 24.3 23 21.7 21.5 25.5 25];
```

We draw the basic graph (the one that must absolutely not shown to a third person because it is unusable!):

```
plot(mois,temp_minimale_paris)
hold on
plot(mois,temp_minimale_papeete)
hold off
```

Now we go to the essential step of this mini guide: **MAKE LEGENDS** ! This includes :

1. To put a title to the chart (ideally that states the goal of the chart)
2. To put a legend for all axes, ie to indicate for each axis the name of the associated quantity as well as its unit, in parenthesis
3. To put a legend for the different lines

Here are the associated command lines:

```
% Title of the graph
title('A Year of minimal temperatures on the 1st of the Month in 2017')
% Axes' names
xlabel('Month number')
ylabel('Temperature (deg. C)')
% Lines' names
legend('Paris (France)', 'Papeete (Fr. Polynesia)')
```

<sup>1</sup>which can have direct impacts on grades...

<sup>2</sup>which increases the probability of high grades...

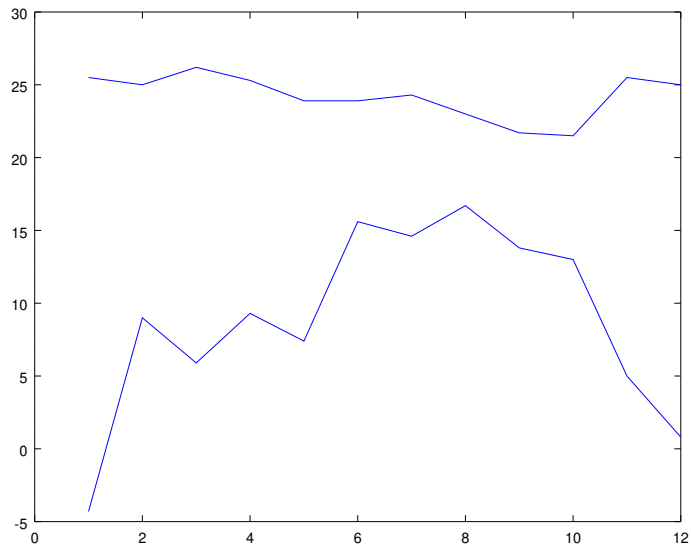


Figure 4.1: Displayed figure at the screen: example of bad figure

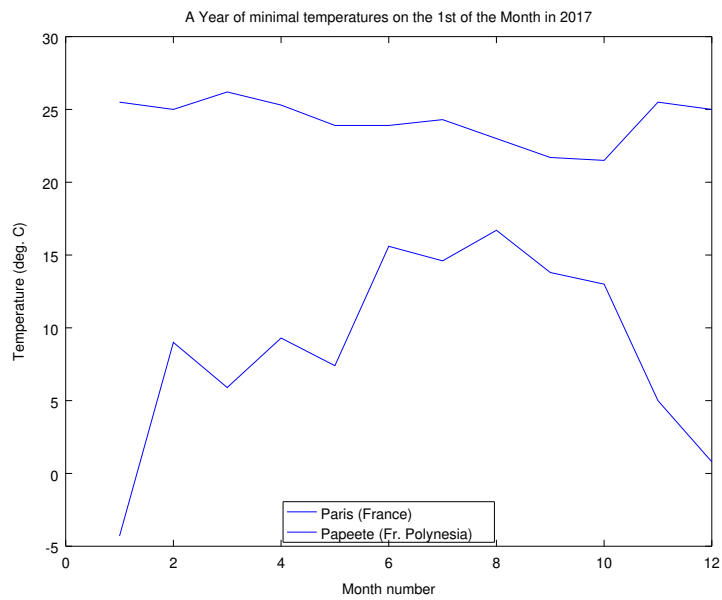


Figure 4.2: Displayed figure at the screen: example of somewhat better figure

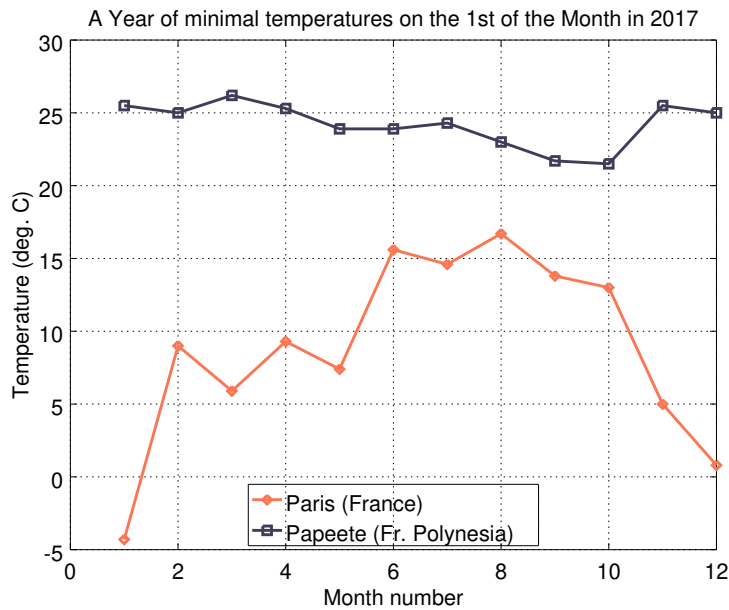


Figure 4.3: Displayed figure at the screen: example of good figure

Now we can go to the aesthetic part (which is strongly recommended):

```
% Display grid
grid on
% Get plot handles for each line
line_hdl=get(gca,'children')
% Change markers shape
set(line_hdl,{'marker'},{'s','d'})
% Change color lines
set(line_hdl,{'color'},{[64,61,88]/256;[252,119,83]/256})
% Change linewidth
set(line_hdl,'linewidth',2)
% Set Fontsize of text elements
set(gca,'fontsize',15)
set(get(gca,'title'),'fontsize',15)
set(get(gca,'xlabel'),'fontsize',15)
set(get(gca,'ylabel'),'fontsize',15)
```

Comment : the last commands for changing the font size can be implemented with the following one-liner command :

```
set(findall(gcf,"-property", 'fontsize'),'fontsize',15)
```

It is advisable to configure your *startup* file `~/.octaverc`<sup>3</sup> so that the font size and the linewidth have the proper default values<sup>4</sup>.

<sup>3</sup><https://octave.org/doc/v4.2.0/Startup-Files.html>

<sup>4</sup><https://octave.org/doc/v4.2.0/Managing-Default-Properties.html>

---

## 4.2 Good practices: programming

As with any software development project, following good practices will help ensure that your results are accurate, avoid or solve programming mistakes, and save you time in the long run. Here are some recommended guidelines, further documented in [2, 3].

### 4.2.1 Use a version control system

Don't share code by email or a mere folder such as our school's Owncloud service, let alone Dropbox®. You *will* get confused between versions. Git repositories will be provided. Use them as the reference version of your code; use the `git merge` functionality to merge your version with that of your collaborators; use branches to experiment with code without breaking the reference version.

If you are not familiar with Git, you may refer to the first-year lecture on the topic [4] (in French), based on SmartGIT, a graphical wrapper around Git; or to any of the many tutorials available online.

### 4.2.2 Document everything, and keep it up-to-date

If you don't see the need to make your code easy to understand to other people, consider that yourself next month will be another person. You *will* waste time trying to understand what you had in mind when you wrote last month's code if you're not careful. Additionally, your teachers will evaluate your code, and will be in a better mood if they don't have to warp their mind around this undocumented optimization that you thought was so clever while pulling an all-nighter.

Comments in Octave begin with `%`. An whole line of comments usually begins with `%%`. Your text editor/development environment will indent them accordingly (see section 4.2.3). Use them to document:

- Functions: the first few lines of comments in a function are displayed when you type `help <function>` in Octave. Use this to tell the reader what your function does, both in short and in detail; what are the expected parameters and return values, what they represent and any assumption you make; and for more complex functions, some examples and non-obvious corner cases.
- Blocks of code: you don't have to add a comment to every line of code, nor to write trivialities such as:

```
x = x + 1 % Add 1 to x.
```

Rather, isolate meaningful blocks of code and document what the code achieves.

```
%% Count total symbols sent, print status every million symbols.
symbols_sent = symbols_sent + N;
if (~ mod(symbols_sent, 1e6))
    fprintf('%d sent, %d errors... ', symbols_sent, errors_dfe);
end
```

Finally, keep this documentation religiously up to date. You may waste a lot of time before realizing that a section of the code doesn't do what the comments say, because someone forgot to update them. Always keep them synchronized.



---

### 4.2.3 Use spacing and indentation for best legibility

Don't pack your code into the smallest possible space. Make it easy to read. If a formula doesn't fit on a single line, break it into several lines, adding “...” at the end of each unfinished line.

Your text editor/development environment will automatically indent code blocks to provide a visual reference of where it fits. This helps not only to read it but also to write it: if the automatic indentation level surprises you, then perhaps you forgot an end or a closing parenthesis somewhere.

### 4.2.4 Don't repeat yourself

Any piece of data or code should have one single representation in the entire program. If you copy-paste values or code, you *will* forget to update all of them when an update is required. Use functions instead, and pass them parameters. You may group related values in a struct and pass that as a single parameter.

### 4.2.5 Use meaningful variable names

It should always be obvious what a variable represents. The common-sense method is to name them `nb_errors` or `nb_symbols_sent` instead of `n_e`, `n_s`. However, in scientific computing, this may clash with the mathematician's habit of giving variables single-letter names. We recommend that single-letter variable names only be used sparingly:

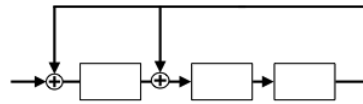
- for a limited number of standard notations shared across the entire program; even then, you may want to index them (as in `X_in`, `X_out`);
- for local variables within a short function.
- Also, beware of loop counters: the traditional notation `i` clashes with Octave's usual notation for  $i = \sqrt{-1}$ ; avoid calling your counter `i`, and/or use the explicit notation `1i` for  $\sqrt{-1}$ .

## 5 Appendix: BCH codes

### 5.1 Encoder for BCH codes

The encoding operations for generating BCH codes can be performed using linear feedback shift registers based on the generator polynomial  $g(x)$ . The main coding operation to handle is polynomial modulo operation.

To illustrate the encoder, we take the example of BCH code (7, 4, 3) where the generator polynomial is  $g(x) = 1 + x + x^3$ . The encoder calculates  $m(x) \bmod g(x)$ . Let  $m(x) = b_0x + b_1x^2 + b_2x^3$



- Initially, the shift register contains all zeros.
- The coefficients of  $m(x)$  are clocked into the shift register one bit coefficient at a time, beginning by the highest coefficient,  $b_2$ , followed by  $b_1$  and so on.

The contents of the shift register are as follows :

Inputs	Shift Register contents		
	0	0	0
$b_2$	$b_2$	0	0
$b_1$	$b_1$	$b_2$	0
$b_0$	$b_0$	$b_1$	$b_2$
0	$b_2$	$b_2 + b_0$	$b_1$

$m(x) \bmod g(x) = b_2 + (b_0 + b_2) + b_1x^2$  contented on the shift registers after 4 clock ticks corresponding to the degree of  $m(x)$ .

### 5.2 Decoder for BCH codes

Use the decoding example for BCH codes from the TELECOM202B course. You have to create a matrix M which will contain the vector representation of element of  $GF(32) = GF(2)[x]/1 + x^2 + x^5$  (last column of the following table). To build matrix M, you can use the same modulo operation program used for encoding.

---

l	$\alpha^l$	Polynomial representation	Vector representation
0	1	1	[10000]
1	$\alpha$	$x$	[01000]
2	$\alpha^2$	$x^2$	[00100]
3	$\alpha^3$	$x^3$	[00010]
4	$\alpha^4$	$x^4$	[00001]
5	$\alpha^4$	$1 + x^2$	[11000]
6	$\alpha^4$	$x + x^3$	[01010]
7	$\alpha^4$	$x^2 + x^4$	[00101]
8	$\alpha^4$	$1 + x^2 + x^3$	[11100]
9	$\alpha^4$	$x + x^3 + x^4$	[01110]
10	$\alpha^4$	$1 + x^4$	[10001]
11	$\alpha^4$	$1 + x + x^2$	[11100]
12	$\alpha^4$	$x + x^2 + x^4$	[01101]
13	$\alpha^4$	$1 + x^3$	[10010]
$\vdots$	$\vdots$	$\vdots$	$\vdots$

## Bibliography

- [1] Loren Shure. *Making Pretty Graphs*. MathWorks. Dec. 2017. URL: <https://blogs.mathworks.com/loren/2007/12/11/making-pretty-graphs/>.
- [2] Greg Wilson et al. “Best practices for scientific computing”. In: *PLoS biology* 12.1 (2014), e1001745. DOI: [10.1371/journal.pbio.1001745](https://doi.org/10.1371/journal.pbio.1001745).
- [3] Michael Robbins. *Good Matlab Programming Practices for the Non-Programmer*. 2001. URL: <http://www.mit.edu/~pwb/cssm/GMPP.pdf>.
- [4] Rémi Sharrock. *TP SmartGIT*. URL: <https://perso.telecom-paristech.fr/bellot/CoursJava/tps/tpgit.html>.