



# Département Sciences du Numérique

Parcours Télécommunications & Réseaux

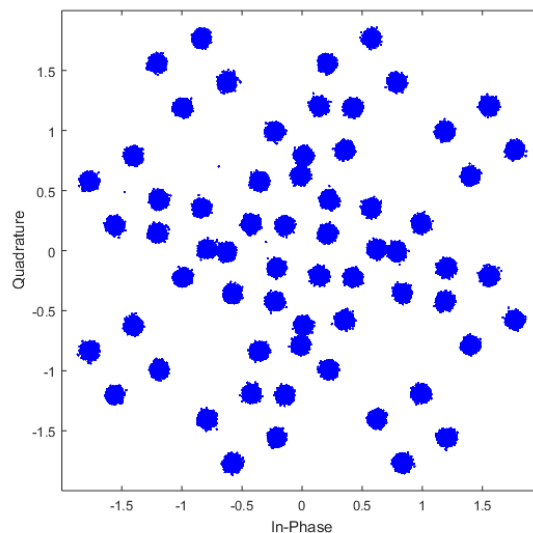
---

## Travaux pratiques

*Egalisation linéaire fréquentielle*

---

*Auteur : C. Poulliat*



Version 1.0 du  
7 décembre 2020



---

## 1 Introduction

Ce TP sera effectué à l'aide du logiciel Matlab. Les thèmes abordés sont :

- Egalisation fréquentielle ZF,
- Egalisation fréquentielle MMSE,



## 2 Transmission sur canal sélectif en fréquence : égalisation temporelle

On considère dans un premier la transmission sur un canal sélectif en fréquence. On considérera le modèle de transmission bande de base. On utilisera à l'émetteur une modulation de type MAQ à  $M$  états avec mapping de type Gray. Le nombre d'états considérés pourra être variable de  $M = 4$  à  $M = 64$  états.

Pour la génération de symboles QAM pour  $M > 4$ , on pourra utiliser `qammod(.)` donnée par

```
function y = qammod(x, M, varargin)
%QAMMOD Quadrature amplitude modulation
%
% Y = QAMMOD(X,M) outputs the complex envelope of the modulation of the
% message signal X using quadrature amplitude modulation. M is the
% alphabet size and must be an integer power of two. The message signal X
% must consist of integers between 0 and M-1. X can be a scalar, a
% vector, a matrix or an array with 3 dimensions.
%
% Y = QAMMOD(X,M,SYMBOL_ORDER) specifies how the function maps an integer
% or group of log2(M) input bits to the corresponding symbol. If
% SYMBOL_ORDER is set to 'gray', then the function uses a Gray-coded
% ordering. If SYMBOL_ORDER is set to 'bin', then the function uses a
% natural binary-coded ordering. If SYMBOL_ORDER is an integer-valued
% vector with M elements, the function uses the ordering specified by
% this vector. This vector must have unique elements in the range [0,
% M-1]. The first element of this vector corresponds to the top-leftmost
% point of the constellation, with subsequent elements running down
% column-wise, from left to right. The last element corresponds to the
% bottom-rightmost point. The default value is 'gray'.
%
% Y = QAMMOD(X,M,...,Name,Value) specifies additional name-value pair
% arguments described below:
%
% 'InputType'          One of the strings: 'integer', or 'bit'. 'integer'
%                      indicates that the message signal is integer
%                      valued between 0 and M-1. 'bit' indicates that the
%                      message signal is binary (0 or 1). In this case,
%                      the number of rows (dimension 1) must be an
%                      integer multiple of log2(M). A group of log2(M)
%                      bits are mapped onto a symbol, with the first bit
%                      representing the MSB and the last bit representing
%                      the LSB. The default value is 'integer'.
%
% 'UnitAveragePower'   A logical scalar value. If true, the QAM
%                      constellation is scaled to average power of 1. If
%                      false, the QAM constellation with minimum distance
%                      of 2 between constellation points is used. The
%                      default value is false.
%
% 'OutputDataType'     Output fixed-point type as a signed, unscaled
%                      numeric type object in MATLAB simulation, and as a
%                      signed, scaled numeric type object during C code or
%                      MEX generation. When this argument is not
%                      specified, if the input datatype is double or
%                      built-in integers, the output datatype is double;
%                      if the input datatype is single, the output
%                      datatype is single. When the input is fixed-point,
%                      this parameter must be specified.
```



```
%  
% 'PlotConstellation' A logical scalar value. If true, the QAM  
% constellation is plotted. The default value is  
% false. The input X is processed and the modulated  
% signal is returned in output Y.  
%  
% Example 1:  
% % 32-QAM modulation. Default: Integer input, Gray coding, minimum  
% % distance of 2 between constellation points  
% x = (0:31)';  
% y = qammod(x, 32);  
%  
% Example 2:  
% % 16-QAM modulation, with LTE specific symbol mapping and constellation  
% % scaled to average power of 1. Default: Integer input  
% x = randi([0, 15], 20, 4, 2);  
% lteSymMap = [11 10 14 15 9 8 12 13 1 0 4 5 3 2 6 7];  
% y = qammod(x, 16, lteSymMap, 'UnitAveragePower', true);  
%  
% Example 3:  
% % 64-QAM modulation with binary mapping, bit input and signed  
% % fixed-point output data type with 16 bits of word length and 10  
% % bits of fraction length. Default: minimum distance of 2 between  
% % constellation points  
% x = randi([0, 1], 10*log2(64), 3);  
% y = qammod(x, 64, 'bin', 'InputType', 'bit', 'OutputDataType', numericitytype←  
(1,16,10));  
%  
% Example 4:  
% % Visualize the constellation for 16-QAM modulation, with gray  
% % mapping, bit input and constellation scaled to average power of 1.  
% x = randi([0, 1], log2(16), 1);  
% y = qammod(x, 16, 'InputType', 'bit', 'UnitAveragePower', true, '←  
PlotConstellation', true);  
%  
% See also QAMDEMOD, MODNORM.
```

Pour la détection on pourra utiliser la fonction `qamdemod(.)`, après correction du biais éventuel le cas échéant.



### 3 Transmission sur canal sélectif en fréquence : égalisation fréquentielle

#### SC/FDE

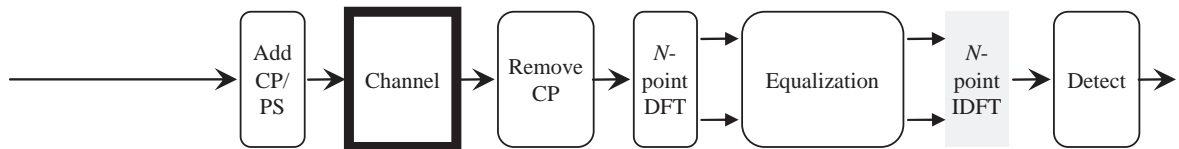


FIGURE 1 – Chaîne de transmission SCFDE

On considère tout d'abord une transmission sur un canal sélectif en fréquence avec égalisation fréquentielle dont le schéma est donné figure 1. On considérera le modèle de transmission bande de base. On utilisera une modulation de type MAQ à  $M$  états avec mapping de Gray. Le nombre d'états considérés devra être variable de  $M = 4$  à  $M = 64$  états.

Le travail à réaliser est le suivant :

1. Mettre en oeuvre les éléments en émission permettant une égalisation fréquentielle au récepteur (Préfixe cyclique).
2. Réaliser l'égalisation fréquentielle en utilisant les critères 'Zero-Forcing' (ZF) et Minimum Mean Square Error (MMSE).
3. Calculer et visualiser le taux d'erreur binaire.

#### 3.1 Egalisation fréquentielle et démodulation

En réception, le signal est alors égalisé dans le domaine fréquentiel. On utilisera pour ce faire les fonctions `fft` et `ifft` de Matlab dont les spécifications sont :

```
%FFT(X,N) is the N-point FFT, padded with zeros if X has less
%than N points and truncated if it has more.
NFFT=128;
Y = fft(y,NFFT);

%IFFT(X,N) is the N-point inverse transform.
y=ifft(Y,NFFT);
```

On rappelle que les égaliseurs ZF et MMSE ont pour expressions,  $\forall k = 1, \dots, NFFT - 1$

$$W_{zf}[k] = \frac{H^*[k]}{|H[k]|^2}$$

$$W_{mmse}[k] = \frac{H^*[k]}{|H[k]|^2 + (E_s/N_0)^{-1}}$$

où  $H[k], \forall k \in [1, NFFT]$  est la séquence obtenue par  $fft(h(n), NFFT)$ .



## 4 Transmission multi-utilisateurs

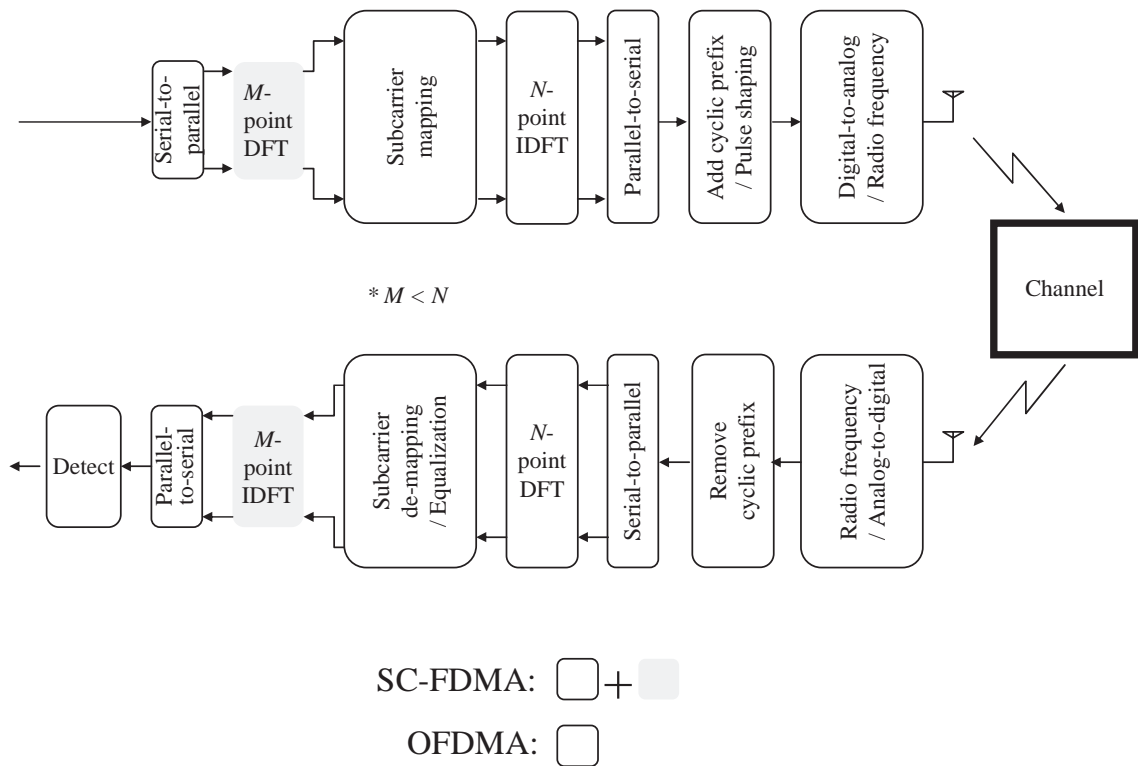


FIGURE 2 – Schémas SC-FDMA vs OFDMA

On considère maintenant l'extension au cas multi-utilisateurs et on souhaite implémenter les systèmes SC-FDMA dans le cadre de deux utilisateurs. La structure de ces schémas est donnée figure 2. On prendra soin de prendre des canaux différents pour chaque utilisateur. On pourra tester différents type de mapping comme montré sur la figure 3. Comparer pour les schéma l'évolution de l'amplitude des signaux (module) au cours du temps pour le cas 4-QAM. Que pouvez vous en déduire ?

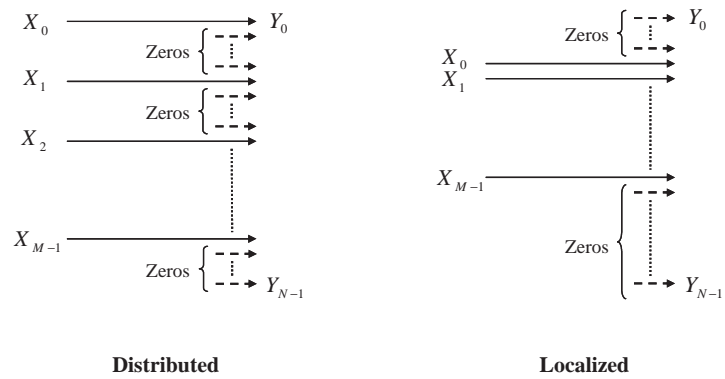


FIGURE 3 – Mappings fréquentiels possibles