

Techniques de transmission et traitement du signal :

Simulation d’une chaîne de transmission numérique

Réalisé par :

MALOULI Asmae  
TARDIF Félix

Table des matières

[I. Introduction 3](#_Toc103352053)

[II. Topologie du réseau 4](#_Toc103352054)

[III. Emetteur 5](#_Toc103352055)

[Séquence initiale 5](#_Toc103352056)

[Codage des bits 5](#_Toc103352057)

[Critère de Shanon 6](#_Toc103352058)

[Génération des Formes d’onde 7](#_Toc103352059)

[Forme générale du signal transmis 9](#_Toc103352060)

[IV. Canal 10](#_Toc103352061)

[Caractéristique du canal de transmission 10](#_Toc103352062)

[Génération du signal bruité 10](#_Toc103352063)

[Addition de plusieurs signaux bruités 10](#_Toc103352064)

[Forme générale du signal sur le canal 10](#_Toc103352065)

[V. Récepteur 11](#_Toc103352066)

[VI. Conclusion 12](#_Toc103352067)

# Introduction

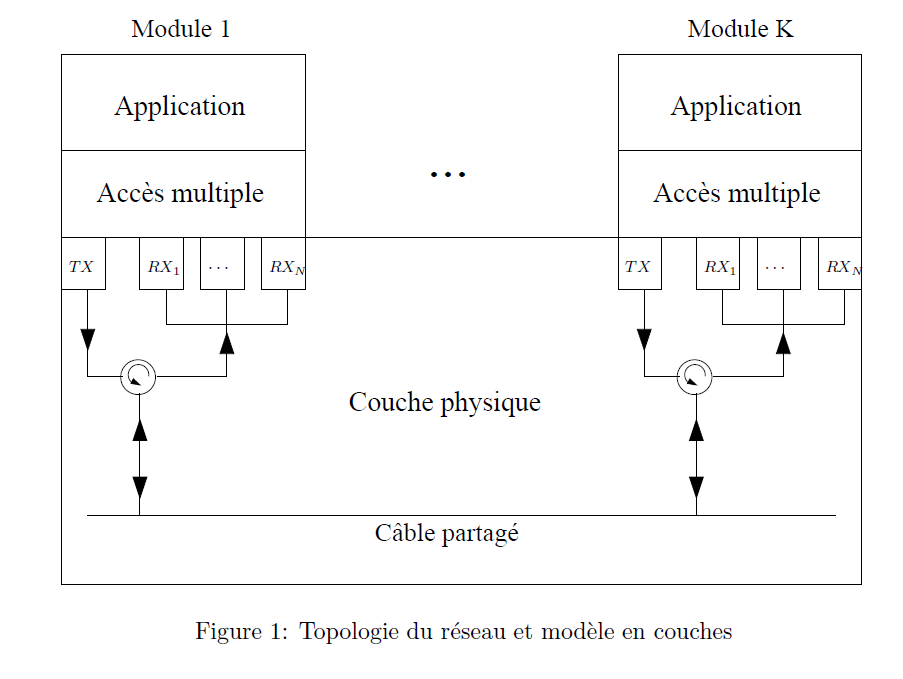
Dans le cadre du cours de Télécommunication 2, il nous est demandé de réaliser un projet en binôme en utilisant le logiciel Matlab. L’objectif de ce projet est de simuler une chaîne de transmission numérique.

Cette simulation sera divisée en 3 parties : l’émetteur, le canal et le récepteur.

…

# Topologie du réseau

Voici la topologie de la chaine de transmission que l’on désire implémenter dans notre simulation :



* 1 câble partagé entre les différents utilisateurs. Il fera office de **Canal de Transmission** pour notre réseau.
* **K** **modules** répartis spatialement sur le canal. Un module est géré par un seul utilisateur.
* Chaque module dispose d’1 seul **Canal d’émission** [TX].
* Chaque module dispose de **N Canaux de réception** [RX(1) 🡪 RX(N)].
* Un niveau d’**accès multiple** et un niveau **applicatif**, dont on ne s’occupera pas lors de notre simulation.
* Objectif de la simulation :
  + Un **module** **K** transmet un signal sur le **câble partagé**, à destination d’un **Canal Fréquentiel N** défini.
  + Chaque **Canal Fréquentiel** reçoit un signal en provenance du **câble partagé**. Le **module** associé au **Canal** décode le signal si ce dernier lui est destiné.

# Emetteur

L’émetteur a pour fonction de transmettre un signal analogique susceptible de se propager convenablement sur le canal physique, occupant la bande de fréquences voulue, et avec le niveau de puissance voulu, en fonction des bits d’information fournis à son entrée.

Intéressons-nous d’abord aux différentes caractéristiques de l’émetteur.

## Séquence initiale

Message à transmettre : Trame de **M bits**  *.*

Débit binaire : **R bits/s**.

Temps par bit : seconde par bit.

Pour notre simulation, nous choisissons N = 4 séquences (pour 4 canal fréquentiel N différents) et M = 6 bits à transmettre.

## Codage des bits

Codage : **Symboles** où .

Temps par symbole : seconde par symbole.

Sur Matlab, nous décidons de générer et coder cette séquence telle que représentée sur la Figure 1 et sur la Figure 2. Tout d’abord nous créons une fonction qui génère une matrice de N colonnes représentant le nombre de séquences et de M lignes représentant le nombre de symboles dans chaque séquence. Enfin nous l’affichons sur un graphe et cela donne le résultat représenté sur la Figure 3.

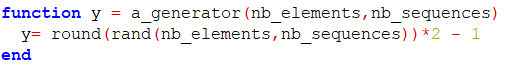


Figure 1 géneration des séquences ak

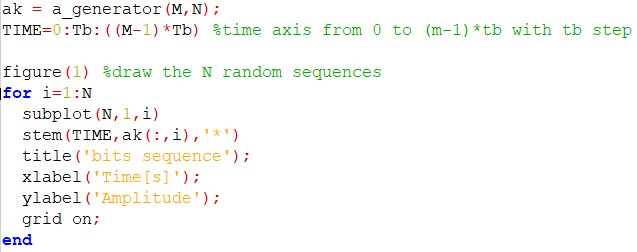


Figure 2 Affichage des séquences

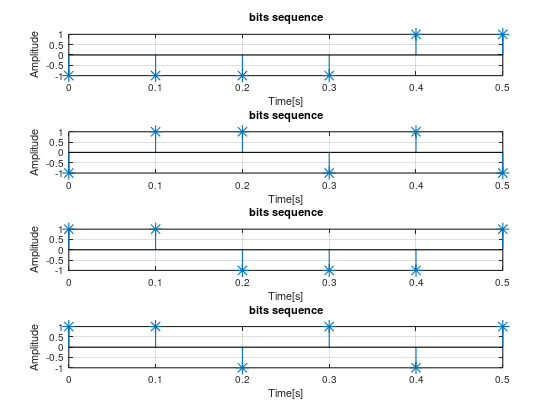


Figure représentation temporelle de la séquence ak

## Critère de Shanon

Les signaux à transmettre doivent être suréchantillonnés selon le critère de Shanon.  
Sur Matlab, nous créons une fonction qui réalise le suréchantillonnage du signal, en ajoutant une multitude de zéro entre les différents symboles transmis. Cette fonction est représentée sur la Figure 4. Sur la Figure 5, on observe la même séquence que sur la Figure 3, mais suréchantillonnée.

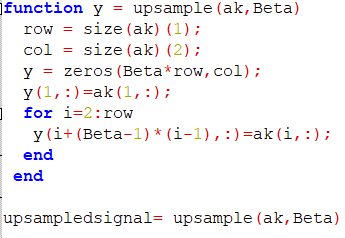


Figure fonction de surechantillonage d’une séquence

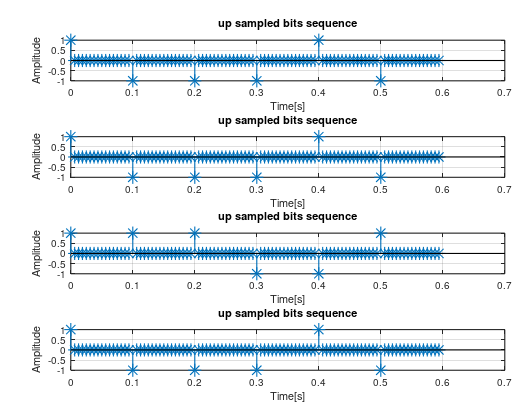


Figure représentation temporelle des séquences suréchantionnées

## Génération des Formes d’onde

Le choix des différentes formes d’onde devra donc permettre de positionner le signal s(t) dans une bande de fréquences correspondant à l’un des N canal fréquentiel disponible.

Pour notre simulation, nous utiliserons une porteuse sinusoïdale telle que :

, *.*

Avec :Fréquences centrales

Le filtre de Nyquist choisi est un filtre en cosinus surélevé avec un facteur de descente compris entre 0 et 1.  
Sur Matlab, on utilise la fonction « rcosfir » pour réaliser ce filtre (Figure 6). Il faut ensuite ramener ce filtre à N différentes fréquences pour pouvoir transmettre chaque séquence indépendamment sur leur propre canal fréquentiel (Figure 7). La Figure 8 montre nos 4 formes d’ondes ainsi générées.

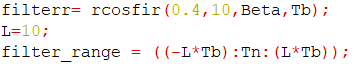


Figure génération du filtre rcosfir

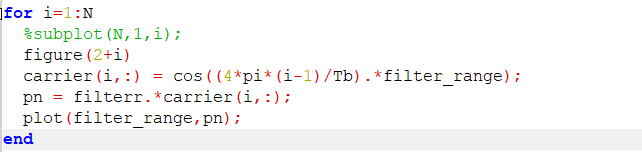


Figure génération des porteuses

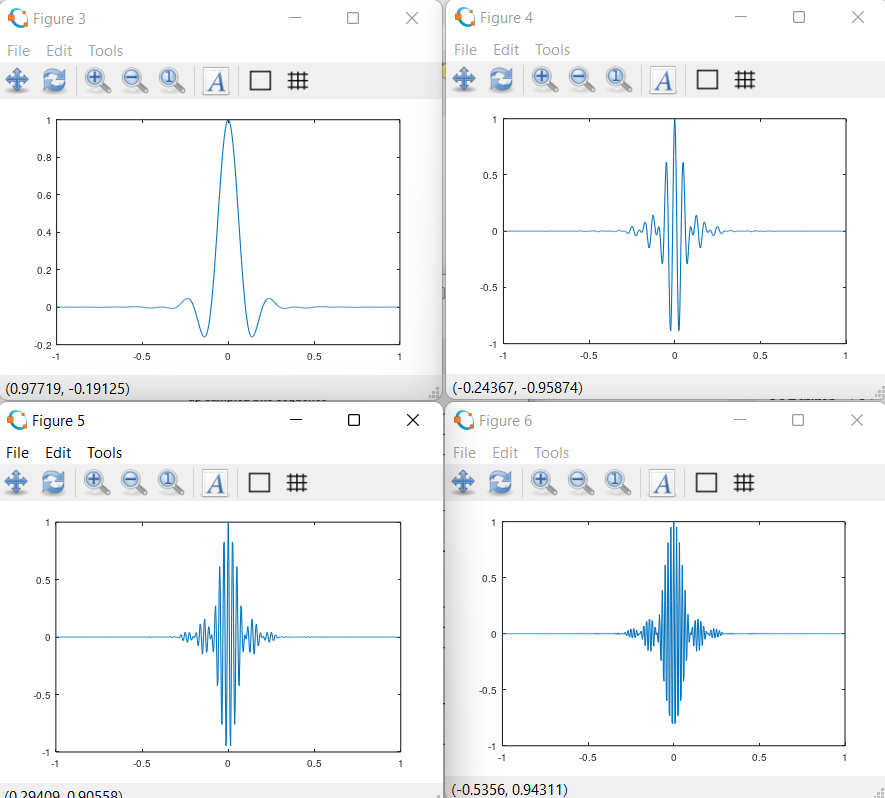


Figure représentation fréquentielle des 4 porteuses

On convolue ensuite la forme d’onde avec le signal suréchantillonné.  
Sur Matlab on utilise la fonction « conv », tel que sur la Figure 9. La Figure 10 montre le résultat pour la 2ème séquence.

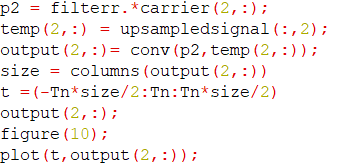


Figure réalisation de la convolution

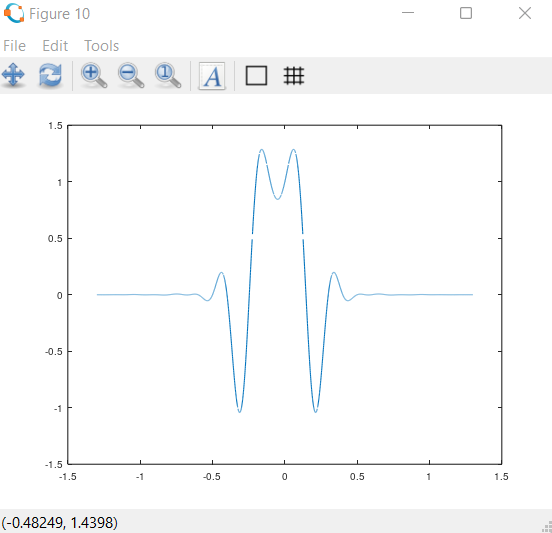


Figure résultat de la convolution

## Forme générale du signal transmis

Avec :

A = Amplitude du signal.

= Symboles transmis.

= Forme d’onde. Dépend directement du **Canal Fréquentiel** **N** sélectionné.

# Canal

Le canal physique (câble) peut être la source de distorsions linéaires sur le signal transmis.  
Pour cette modélisation, nous ferons l’hypothèse que ces distorsions sont négligeables sur une bande de fréquence très étroites (kHz).  
Cependant, le canal de transmission adjoint toujours un **délai** , une **atténuation**  ainsi qu’un **bruit blanc gaussien** .

## Caractéristique du canal de transmission

Densité spectrale bilatérale de  :

## Génération du signal bruité

Pour créer un délai et une atténuation sur un signal , nous avons créé la fonction *channel* (Figure 11).

On rentre notre signal généré précédemment (symbolisé *output(…)*) dans la fonction *channel* pour le retarder et l’atténuer (Figure 12). La variable *channel\_effect* représente :

Où  et dépendent directement du **Canal Fréquentiel** **N** sélectionné.

On ajoute ensuite du bruit au nouveau signal au moyen de la variable *noise*, qui va générer un bruit de valeur aléatoire (Figure 12). La variable *noise* représente :

Où sera propre au **récepteur**  sélectionné.

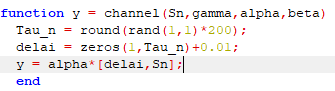


Figure : Fonction channel

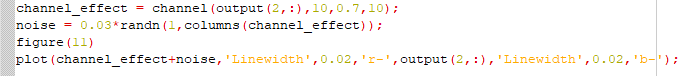


Figure : Utilisation de la fonction channel et génération du bruit

## Addition de plusieurs signaux bruités sur le canal

## Forme générale du signal sur le canal

Avec :

= Facteur d’atténuation du signal.

= Signal transmis décalé dans le temps.

= Bruit blanc gaussien aléatoire.

# Récepteur

# Conclusion