

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](#_Toc103378279)

[II. Topologie du réseau 4](#_Toc103378280)

[III. Emetteur 5](#_Toc103378281)

[Séquence initiale 5](#_Toc103378282)

[Codage des bits 5](#_Toc103378283)

[Critère de Shanon 6](#_Toc103378284)

[Génération des Formes d’onde 7](#_Toc103378285)

[Forme générale du signal transmis 9](#_Toc103378286)

[IV. Canal 10](#_Toc103378287)

[Caractéristique du canal de transmission 10](#_Toc103378288)

[Génération du signal bruité 10](#_Toc103378289)

[Forme générale du signal sur le canal 11](#_Toc103378290)

[V. Récepteur 12](#_Toc103378291)

[VI. Conclusion 13](#_Toc103378292)

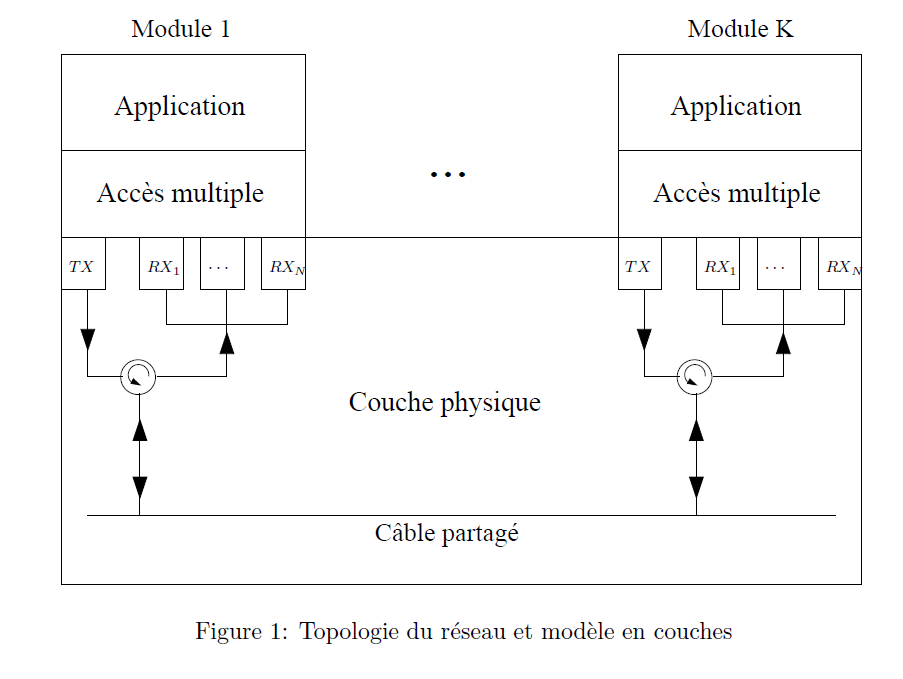
# Introduction

Le projet consiste à simuler une chaîne de transmission numérique multi-utilisateurs destinée à fonctionner sur un réseau filaire. Seule la couche physique du système sera manipulée, et d'un point de vue strictement fonctionnel : un flux binaire présenté à l'un des émetteurs doit être transporté par un moyen approprié vers un récepteur situé sur le réseau. Le système sera considéré comme correct si le flux binaire transmis est reproduit à la sortie du récepteur cible.  
La simulation se fera sur le logiciel Matlab, mais ne prendra pas en compte certains aspects, notamment : la signification des bits transmis et les applications desservies par le système de transmission ; les protocoles d'accès multiples permettant à plusieurs utilisateurs de partager les mêmes ressources physiques sur le réseau.  
La tâche d'une chaîne de transmission numérique est de transmettre le signal en minimisant le nombre d'erreurs et de pertes.

D'autres paramètres sont tout aussi importants, tels que la puissance de transmission requise ou la bande passante. C'est pourquoi l'étude des performances de la chaîne de transmission est une étape essentielle dans le développement des systèmes de télécommunication, car les écarts par rapport aux performances théoriques peuvent être importants.  
Cependant, la production de prototypes est coûteuse et prend du temps, et il est clair que des simulations informatiques de la chaîne de transmission complète peuvent réduire les coûts et fournir une bonne estimation des résultats réels.  
Nous allons commencer par l'émetteur analogique, puis le canal et enfin le récepteur. Mais dans ce rapport, seule la partie transmission sera abordée, puis la partie canal et quelques concepts théoriques du récepteur seront abordés.

# 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) à 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.

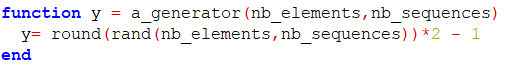


Figure géneration des séquences ak

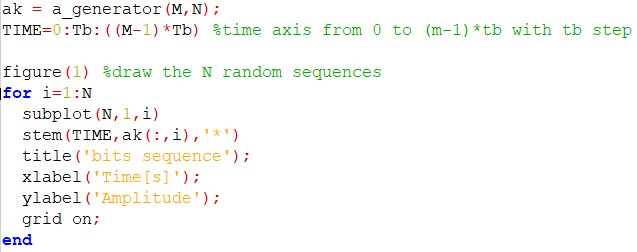


Figure Affichage des séquences

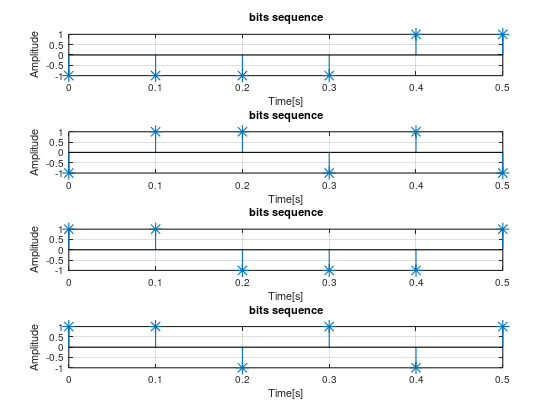


Figure 3 représentation temporelle de la séquence ak

## 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.

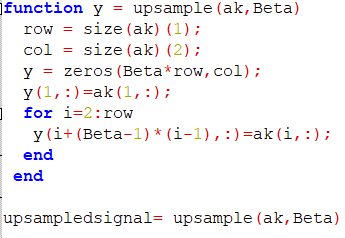


Figure 4 fonction de surechantillonage d’une séquence

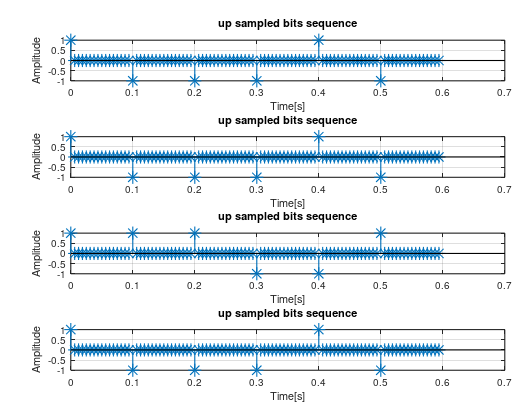


Figure 5 représentation temporelle des séquences suréchantionnées

## Génération des Formes d’onde

Le choix des différentes formes d’onde devrait 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.

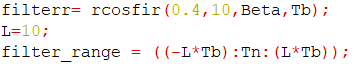


Figure 6 génération du filtre rcosfir

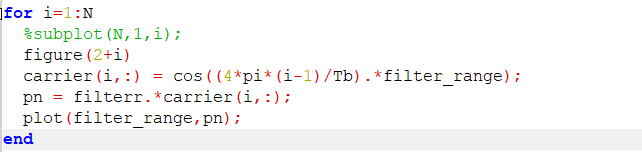


Figure 7 génération des porteuses

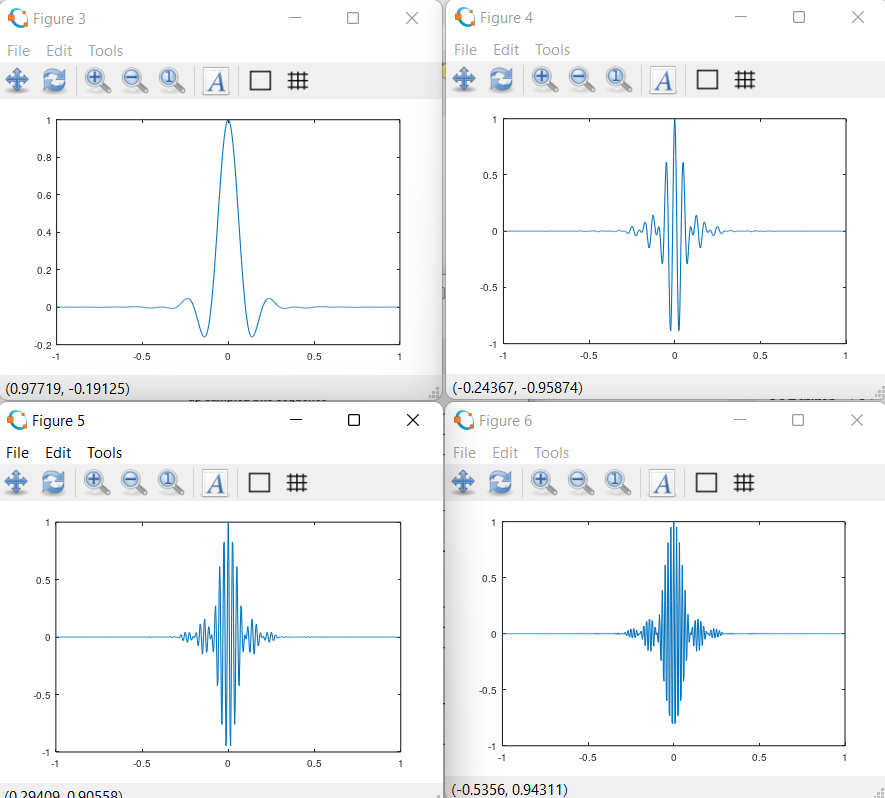


Figure 8 représentation fréquentielle des 4 porteuses

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.

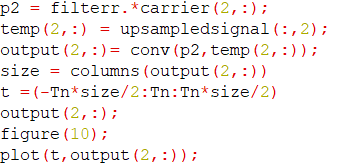


Figure 9 réalisation de la convolution

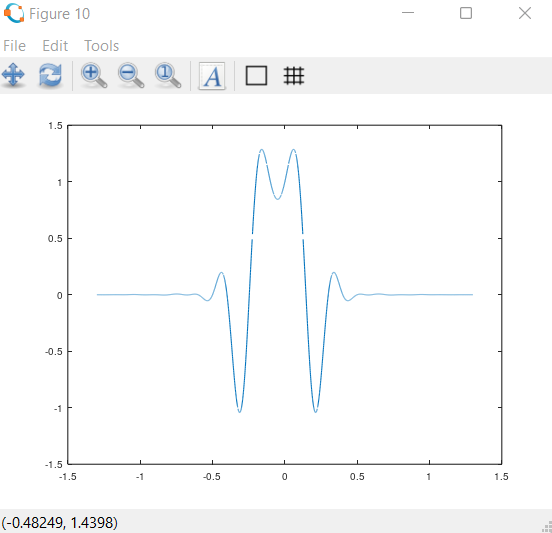


Figure 10 résultat de la convolution

## 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 sur l’ensemble des signaux transmis sur le canal 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.

On observe à la figure 13 le résultat d’une telle transformation sur un signal quelconque.

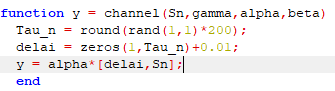


Figure : Fonction channel

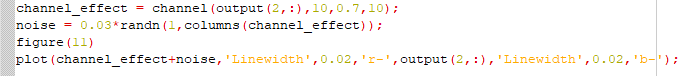


Figure : Utilisation de la fonction channel et génération du bruit

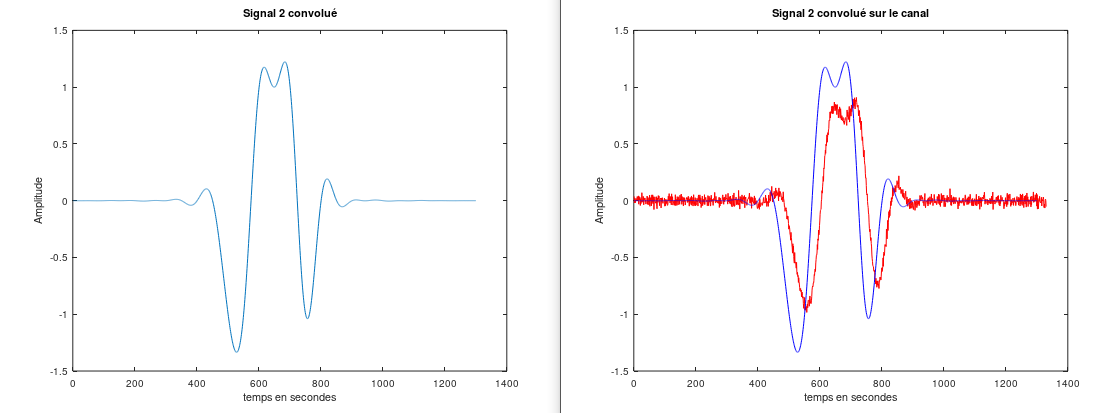


Figure : Graphique montrant l'effet du canal de transmission sur un message transmis

Le signal rouge sur la figure de droite présente un exemple de bruit appliqué sur un seul signal atténué et déphasé. Il faut noter que c’est juste un exemple car le bruit sera appliqué sur la sommation de tous les signaux passants sur ce canal et pas sur chacun.

## Forme générale du signal sur le canal

Avec :

= Facteur d’atténuation du signal.

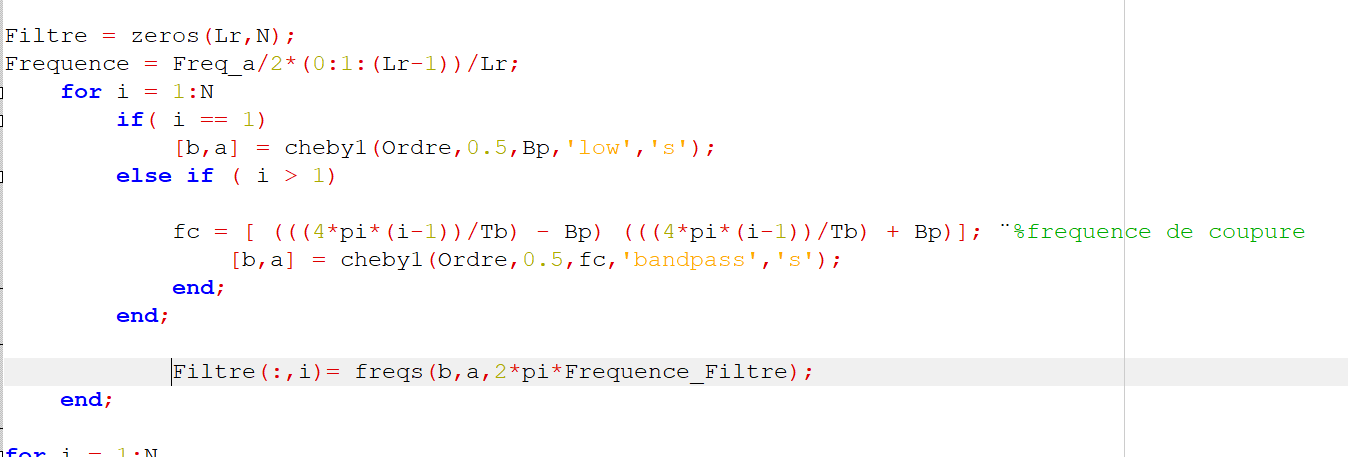
= Signal transmis décalé dans le temps.

= Bruit blanc gaussien aléatoire.

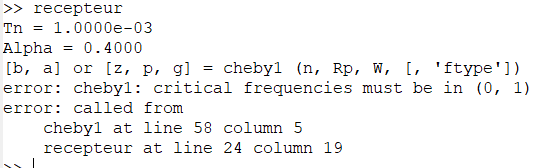
# Récepteur

Le signal reçu correspond à la somme de tous les messages. Le but est de diviser ce signal en selon différentes bandes de fréquences.

Pour cela nous avons essayé de manipuler la fonction cheby1 :



Malheureusement, nous n'avons pas pu tester cette fonction à cause de l’erreur suivante qu’on n'a pas pu résoudre :



# Conclusion

Notre souhait était d’arriver au bout du projet mais compte tenu du temps à notre disposition nous n’avons pas pu atteindre les objectifs ; nous n'avons pas pu faire beaucoup dans la partie “récepteur” à cause de certaines erreurs qui nous ont un peu bloqués.

Néanmoins ce fût pour nous une expérience très enrichissante car nous avons énormément appris lors de ce projet :   
Premièrement par rapport aux cours de Telecom : Cela nous a permis de mettre en pratique certains des concepts vus lors des séances théoriques, de mieux les comprendre et dans une certaine mesure d’apprendre à les implémenter “en vrai” ou, dans notre cas, de les simuler informatiquement.

En second lieu au niveau du logiciel Matlab en lui-même, un logiciel que l’on n'a pas toujours l’habitude d’utiliser mais qui se retrouve dans les plus performant pour la simulation informatique. Nous avons donc eu l’occasion de découvrir de nouvelles fonctionnalités présente sur le logiciel.