Table des matières

[Introduction 2](#_Toc516825522)

[1. Émetteur 4](#_Toc516825523)

[1.1. Le message initial (source) 4](#_Toc516825524)

[1.2. Codage du message 5](#_Toc516825525)

[1.3. La mise en forme du signal 6](#_Toc516825526)

[1.3.1. Modulation ASK 7](#_Toc516825527)

[1.3.2. Filtre en cosinus surélevé 7](#_Toc516825528)

[1.3.3. Sur-échantillonnage 10](#_Toc516825529)

[2. Canal 11](#_Toc516825530)

[2.1. Le délai de transmission 11](#_Toc516825531)

[2.2. L’atténuation du signal 11](#_Toc516825532)

[2.3. Le bruit AWGN 12](#_Toc516825533)

[2.4. Résultat final 13](#_Toc516825534)

[3. Récepteur 14](#_Toc516825535)

[3.1. Filtres analogiques 14](#_Toc516825536)

[3.1.1. Filtre de Butterworth 15](#_Toc516825537)

[3.1.2. Séparation des messages 16](#_Toc516825538)

[3.2. Filtre adapté 16](#_Toc516825539)

[3.3. Synchronisation 16](#_Toc516825540)

[3.4. Prise de décision 17](#_Toc516825541)

[4. Taux d’erreur binaire sur un canal 18](#_Toc516825542)

[4.1. Définition 18](#_Toc516825543)

[4.2. Calcul du taux d’erreur binaire théorique en fonction de Eb/n0 19](#_Toc516825544)

[4.3. Calcul du taux d’erreur binaire par l’opération logique XOR 19](#_Toc516825545)

[4.4. Comparaison des deux taux d’erreur binaire 20](#_Toc516825546)

# Introduction

**Énoncé du projet**

Dans le cadre du cours de télécommunication 1ère année de Master, il a été demandé aux élèves de simuler, à l'aide du logiciel Matlab, une chaîne de transmission numérique multi-utilisateurs destinée à fonctionner sur un réseau câblé. Notons que, d’un point de vue strictement fonctionnel, nous ne devons prendre en compte que la couche physique du système. En effet, l'un des émetteurs produit un flux binaire qui doit être transmis par des moyens adéquats à l'un des récepteurs situés sur le réseau. Par conséquent, la signification des bits envoyés, le partage d'une même ressource physique par plusieurs utilisateurs et l'implémentation des différents modules au moyen de composants électroniques (circuits physiques) ne seront pas pris en compte dans le cadre de ce travail. Nous considérons le système comme correct si le flux binaire transmis est reproduit à la sortie du récepteur de destination.

Pour rappel, l'objectif de la simulation est de permettre la variation de différents paramètres du système et d'en observer les réactions. Dans le cas où le modèle de notre chaîne de transmission numérique refléterait convenablement le système physique (réduction au maximum du nombre d'erreurs), nous pourrions analyser les performances de celui-ci. Ainsi, nous pourrions réaliser des gains de temps et de coûts considérables tout en évitant l'implémentation du système physique.

Par conséquent, la puissance d'émission nécessaire ou la bande-passante du signal transmis seront des paramètres importants à prendre en compte dans le cadre de ce travail. En effet, tout écart des performances par rapport à la théorie, lors du développement d'un système de télécommunication, ne peut être négligé.

* En d'autres mots, la simulation est donc une excellente option pour se rapprocher au plus près possible des résultats réels tout en évitant de devoir mettre en place un système physique.

**Étapes du projet**

Afin de réaliser une simulation complète, nous devons découper le travail en trois étapes selon le principe d’une transmission numérique. Chronologiquement, nous avons donc réalisé :

1. L’émetteur ;
2. Le canal ;
3. Le récepteur ;

**Topologie du réseau**

Suivant les consignes données, nous considérons un canal multi-utilisateur (ou, plus simplement, un câble partagé) auquel un ensemble de K modules sont rattachés en différents points. Chaque module est géré par un utilisateur déterminé et ceux-ci bénéficient de N ressources physiques, caractérisées par une bande de fréquences déterminée et découpée en deux niveaux :

* Un premier niveau d’accès multiple, implémenté au niveau de la couche physique par la méthode de réparation en fréquences ;
* Comme K > N, un second niveau d’accès multiple, implémenté au niveau du protocole. Plusieurs utilisateurs devront en effet se partager ces ressources. Ce second niveau ne concernant pas la couche physique, il ne sera pas abordé dans le cadre de ce travail. Par conséquent, nous considérons toujours que K N lors de nos simulations.

Ainsi, nous utilisons une bande de fréquences déterminée sur le canal partagé ou « canal fréquentiel » :

Chaque module aura à tout moment la possibilité d’effectuer deux opérations simultanées :

* La transmission ou non d’un signal sur un des N canaux fréquentiels ;
* La réception d’un signal sur l’ensemble des N canaux fréquentiels.

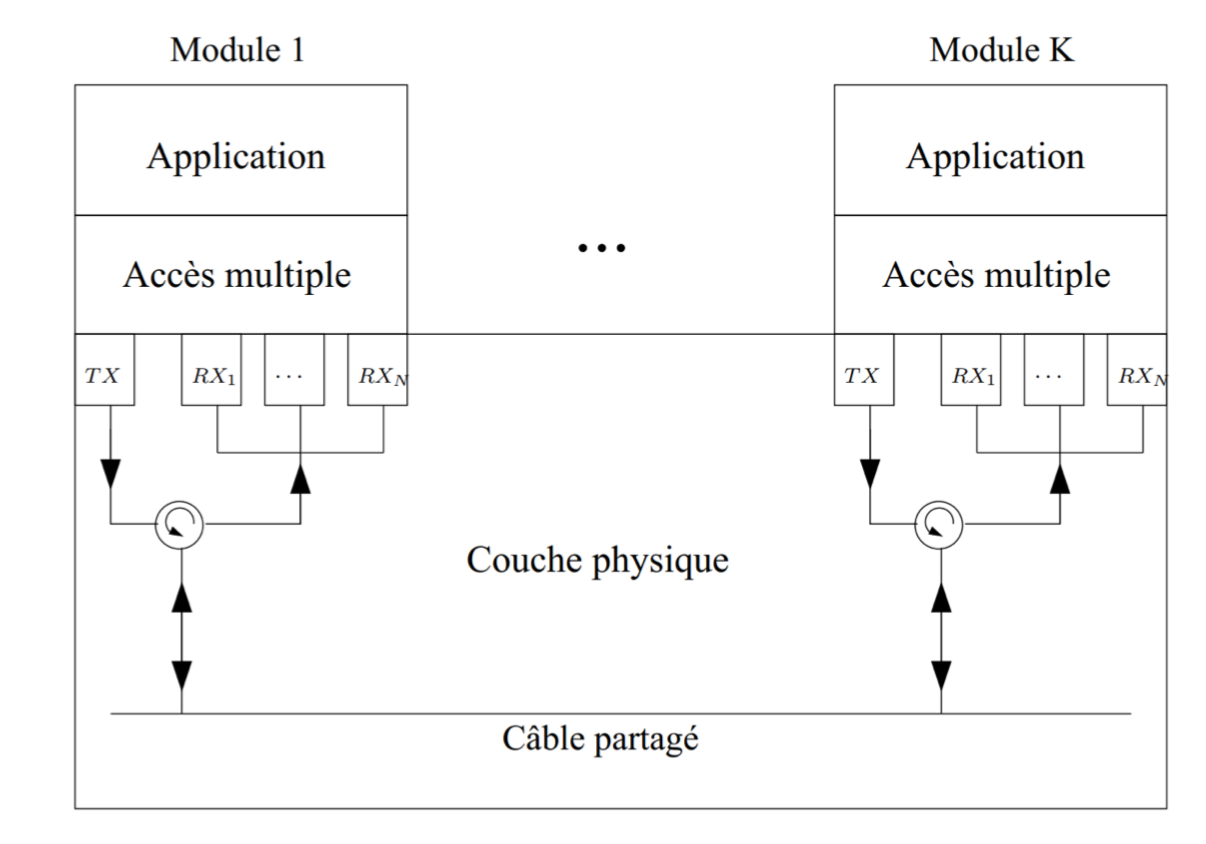


Figure 1: Topologie du réseau avec son modèle en couches

Nous attirons l’attention sur le fait que la rédaction de ce rapport s’appuie essentiellement sur les notions théoriques vues au cours de télécommunications 1ère année master (année 2017-2018). Ainsi, ce rapport reprend certaines figures tirées des slides du cours (c’est-à-dire du livre de référence « *Communications Systems* » de *Simon Haykin*) et constitue une bonne synthèse de la matière parcourue.

# Émetteur

Dans le domaine des télécommunications, comme partout ailleurs, pour que 2 interlocuteurs puissent communiquer ensemble, chacun doit être capable **d’émettre l’information** qu’il souhaite transmettre. Sans cela, aucun échange de données ne peut exister. C’est donc cette notion qu’il convient de définir en premier lieu. Le schéma-bloc (figure 2) ci-dessous reprend les notions générales à définir concernant l’émetteur :



Figure 2 - schéma bloc présentant les différents éléments intervenant dans l'émetteur.

Le schéma-bloc représenté à la figure 3 ci-dessous est celui utilisé pour la réalisation de l’émetteur dans le projet.



Figure 3 - Schéma bloc de l'émetteur utilisé (avec les différentes notations utilisées).

## Le message initial (source)

Un message est une suite de données/d’informations variant au cours du temps : on peut donc représenter un message par une fonction du temps. Selon la source, cette fonction peut être un signal analogique ou numérique. L’énoncé de l’exercice stipule que le message est directement émis sous forme numérique, c’est-à-dire sous forme binaire.

La propagation du message au travers du canal de communication peut prendre différentes formes : signal électrique, signal électromagnétique, signal lumineux, … Dans notre cas, la transmission de l’information se réalise via un support physique (câble partagé), nous utiliserons donc un signal électrique. Ce signal électrique est représenté par 2 états correspondant aux bits d’informations (par exemple : 5V-GND ⬄ 1-0).

* *Ainsi, nous considérons que le message est directement défini sous* ***forme binaire*** *(suite de 0 et 1). Chaque message, généré aléatoirement, possède une* ***taille fixe M*** *(en bits) et un* ***débit R*** *(en bits/s), soit une durée de secondes par bits.*

L’émetteur peut émettre de l’information sur le canal à n’importe quel instant : on parle de communication **asynchrone**. Dans ce type de communication, il est de rigueur de définir un format de trame. Ce format contient généralement :

* 1 bit de *start* pour prévenir le récepteur que le message est en cours d’envoi ;
* 8 bits (ou plus) représentant les données à transmettre ;
* 1 bit de *stop* pour indiquer que les données ont été transmises ;
* 1 bit de parité pour détecter les éventuelles erreurs de transmission ;
* ...

Dans notre cas, il n’est pas demandé d’implémenter de protocole de communication et nous ne devons donc pas définir de format de trame. Nous définirons néanmoins une séquence de synchronisation pour permettre au récepteur de synchroniser son instant d’échantillonnage avec l’émetteur.

* *Ainsi, la trame sera constituée d’une suite de 5 bits à ‘1’ (constituant la* ***séquence de synchronisation****) suivi d’une suite aléatoire de bits ‘0’et ‘1’ (constituant le message).*

La figure 4 ci-dessous est tirée du logiciel MATLAB. Elle représente le message dans sa forme « initiale », c’est-à-dire contenant une séquence de synchronisation suivi de données aléatoires.

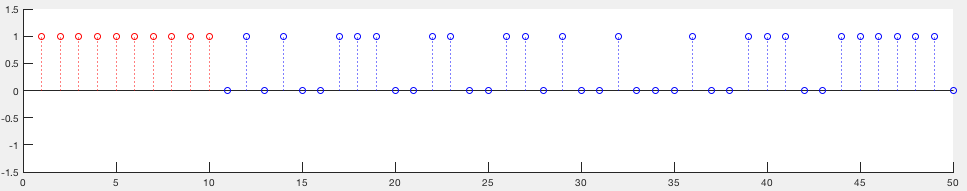


Figure 4 - Message originale (séquence synchronisation de 10 bits en rouge – message de 40 bits en bleu).

## Codage du message

Avant que le signal ne soit envoyé sur le canal de communication, celui-ci doit être codé. On distingue 2 types de codage : le codage de source et le codage de canal.

* Le but principal du *codage de source* est de **comprimer au maximum la source d’information** pour permettre une **communication efficace**. Ainsi, le codage de source transforme le signal binaire d’entrée en une forme la plus économique possible en termes de nombres de bits.
* Le but principal du *codage de canal* est de **protéger le système contre les erreurs dues au bruit** pour permettre une **communication fiable**. Ainsi, le codage de canal ajoute des informations permettant de reconstituer le message initial malgré les erreurs éventuelles causées par le bruit de canal.
* Selon l’énoncé du projet, le codage utilisé est simple : il s’agit d’un codage de lignes et plus spécifiquement du codage bipolaire NRZ. On transforme la séquence de bits de départ en nouvelle séquence comme ceci :

Après l’étape du codage, on a généré une nouvelle séquence discrète. En théorie, chaque élément de cette séquence est appelé ***symbole*** est peut-être associé à un ou plusieurs bits issus de la source d’informations. De cette manière, après le codage, on définit un second débit, le débit des symboles, qui est le nombre de symboles transmis par unité de temps : on parle de *bauds*. Dans notre cas, on utilise un codage binaire (2 niveaux), représentés par les symboles 1 et -1. Ainsi, à 1 bit correspond 1 symbole. Autrement dit, la taille du message avant et après le codage n’a pas changée (taille M). De même, le débit est resté identique (débit R).

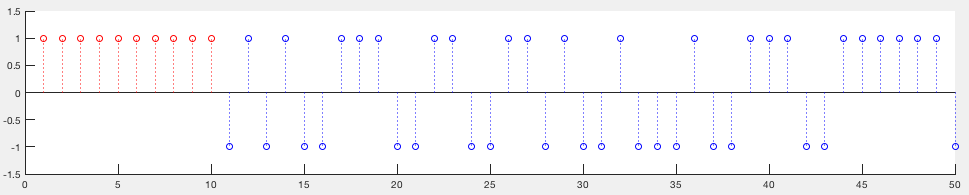
La figure 5 ci-dessous reprend le codage réalisé sur la même séquence de bits que la figure 1.

Figure 5 - Message codé (en code polaire NRZ).

## La mise en forme du signal

Notre message initial binaire a été transformé en nouvelle séquence de -1 et 1 grâce au *codage*. Si on se réfère à la figure 2, l’étape qui suit le codage dans l’émission du message concerne la mise en forme de ce dernier. On distingue 2 points dans la mise en forme des données :

1. *La modulation* : La modulation transforme la séquence binaire en signal continu selon une porteuse oscillant à une certaine fréquence. Cette fréquence de la porteuse représente ainsi le canal sur lequel les données seront transmises.
2. *Le filtre de mise en forme* : le filtre de mise en forme limite la bande passante du signal émis. Ceci permet ainsi de délimiter les différents canaux sur le médium de transmission.

Le signal continu en sortie de bloc de mise en forme doit avoir pour équation :



Où :



* représente donc la fonction de mise en forme du signal. Elle est composée du ***filtre de mise en forme***  ainsi que la ***porteuse*** . Nous noterons que la fréquence de la porteuse, représentant le canal sur lequel l’information va être envoyée, est définie par . Enfin, un **amplificateur de gain A** permettra d’ajuster la puissance du signal au niveau prévu.

La porteuse est clairement définie : il s’agit d’une fonction en cosinus.

En revanche, concernant le filtre de mise en forme, il faut définir la fonction . Ce filtre de base est un filtre de type « racine de Nyquist », c’est-à-dire tel que représente un filtre de Nyquist. Dans l’énoncé du projet, il est précisé que ce filtre doit être un **filtre en cosinus surélevé**.

### Modulation ASK

La modulation ASK (**A**mplitude **S**hift **K**eying) établit la correspondance entre les symboles et la forme d’onde caractérisée par :

Où

( représentant le gain en puissance souhaité).

Cette modulation permet ainsi, via une porteuse sinusoïdale, d’émettre l’information sur une canal particulier fonction de la fréquence choisie **.**

La figure ci-dessous représente le principe de la modulation ASK. On remarque bien que l’amplitude de la porteuse varie selon les symboles transmis (dans ce cas-ci 0 ou 1).

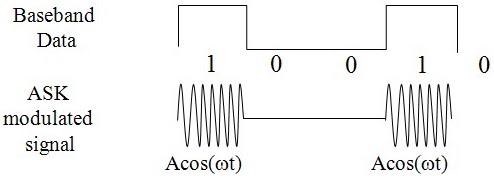
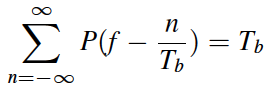
****

Figure 6 - Principe de la modulation ASK.

### Filtre en cosinus surélevé

Pourquoi utiliser un filtre en cosinus surélevé ? Pour expliquer le filtre en cosinus surélevé, nous avons besoin d’introduire le critère de Nyquist afin de définir le filtre de Nyquist idéal.

***Le critère de Nyquist*** est une expression mathématique exprimée dans le domaine fréquentielle (on peut aussi la définir dans le domaine temporel mais c’est préférable dans le domaine fréquentiel) comme ceci :



Cette expression indique que si le canal global p(t), composé de la mise en cascade du filtre d’émission, du canal physique, et du filtre de réception, a un spectre P(f) satisfaisant le critère de Nyquist (expression ci-dessus), alors les échantillons du canal enregistrés à une cadence 1/Tb sont exemptés d’interférence entre symboles.

* En conclusion, ce critère permet d’assurer une réception parfaite des données émises en l’absence de bruit (or, comme on le verra dans la section suivante, notre canal induit toujours du bruit).

***Le filtre de Nyquist idéal*** est un filtre dont la représentation spectrale est rectangulaire. Il est caractérisé par une bande passante W :

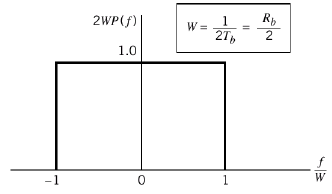


Figure 7 - Filtre de Nyquit idéal représenté dans le domaine fréquentiel.

Dans le domaine temporel, cette impulsion rectangulaire correspond à la fonction sinus cardinal (obtenue par transformée de Fourier inverse) :

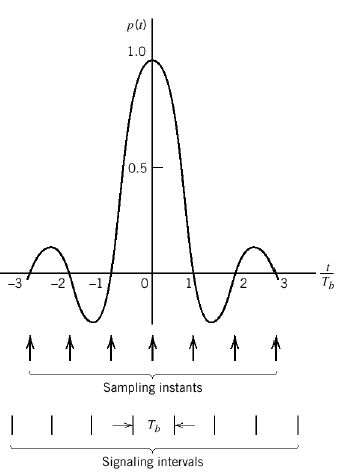


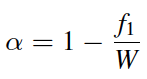
Figure 8 - Filtre de Nyquist idéal représenté dans le domaine temporel.

Cependant, il y a 2 grands inconvénients à ce filtre de Nyquist idéal :

1. Premièrement, un tel filtre est *irréalisable en pratique*. En effet, lorsqu’on visualise la forme rectangulaire dans le domaine spectrale, on ne peut pas délimiter la bande passante du filtre de façon aussi abrupte.
2. Deuxièmement, si l’on visualise le domaine temporel, l’impulsion p(t) décroît lentement, ce qui qui provoque une *dégradation importante des performances en cas d’erreurs de timing*.

* La solution pour résoudre ces 2 inconvénients est : le ***filtre en cosinus surélevé***. Ce filtre est constitué d’une partie plate et d’une partie dite de « rolloff » de forme sinusoïdale.

Pour distinguer ces 2 parties (plate et de rolloff), on définit le ***facteur de rolloff (.*** Ce facteur représente la bande passante en excès par rapport à W (qui correspond à la bande passante du filtre de Nyquist idéal). Son expression est la suivante :



Avec

Dès lors, au plus augmente, au plus la bande passante totale s’élargit. De cette manière, le 1er inconvénient cité précédemment est résolu. En effet, la bande passante total a pour expression mathématique :



La figure 9 ci-dessous représente les différentes formes obtenues en faisant varie le facteur de rolloff.

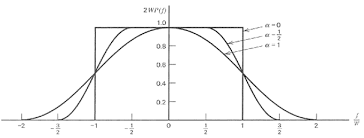
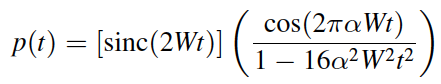


Figure 9 - Représentation fréquentielle du filtre en cosinus surélevé.

Pour comprendre comment le second inconvénient est résolu, nous devons nous intéresser au domaine temporel. Dans le domaine temporel, l’équation du filtre en cosinus surélevé est la suivante :



Dans cette équation :

* Le 1er facteur est un sinus cardinal. Ce facteur représente donc l’impulsion de Nyquist idéale. Ce qui permet d’assurer le passage par 0 aux instants adéquats.
* Le 2ème facteur est caractérisé par une décroissance en . Ce qui permet de réduire l’importance (hauteur) de la queue de l’impulsion. Ceci permet donc d’être plus robuste face aux erreurs de timing.
* Conclusion : La bande passante requise et la robustesse à l’encontre des erreurs de timing augmentent quand le facteur de rolloff évolue depuis 0 vers 1. Dans le cas particulier où le facteur de rolloff vaut 1, la robustesse à l’encontre des erreurs de timing est optimale cependant le prix à payer est que la bande passante est dans ce cas 2 fois plus élevée comparativement au filtre de Nyquist idéal.

### Sur-échantillonnage

Le filtre de Nyquist a comme inconvénient d’être difficile à implémenter de façon analogique.

* Pour résoudre ce problème, une solution est d’échantillonner notre signal sn(t) au moyen d’un **FIR** (filtre numérique à réponse impulsionnelle finie) et dont les coefficients sont programmables.

Pour ce faire, nous définissons un facteur de suréchantillonnage β respectant le critère de Shannon afin de construire un signal β fois plus rapide que la débit initiale des symboles (bauds). Rappelons que pour respecter le critère de Shannon, il faut que la fréquence d’échantillonnage soit deux fois plus grande ou égale à la fréquence maximale du signal (ici sn(t)).

Si on considère que l’on prend le canal se situant à la fréquence la plus élevée et que le facteur de roll-off est également pris le plus élevé, c’est-à-dire valant 1 : Alors dans ce cas, nous trouvons que le facteur de suréchantillonnage β vaut : .

Où N représente le nombre de canaux possibles.

Nous avons donc en sortie du FIR une séquence d’échantillons dont la cadence est :

Enfin, nous construisons notre signal analogique continu par interpolation des échantillons à l’aide d’un DAC (Digital to Analog Converter).

# Canal

Notre émetteur étant opérationnel, nous devons maintenant être en mesure de transmettre l’information au(x) destinataire(s). Pour ce faire, nous avons besoin d’un **support de transmission**. Dans le domaine des télécommunications, ce support est appelé « médium ». Il peut se présenter sous différentes formes : câbles physiques, ondes électromagnétiques, … Dans notre cas, le signal sera transmis sur un câble physique (cuivre). Dans cette section, nous allons aborder les caractéristiques essentielles d’un canal de communication afin de simuler de façon aussi réaliste que possible la transmission des données sur ce canal.

Le canal possède différentes caractéristiques néfastes pour le signal émis et à destination du récepteur. En effet, le câble peut être la source de distorsions linéaires dues à l’effet de peau et aux réflexions multiples des signaux sur les terminaisons non-adaptées. Nous ferons l’hypothèse que ces distorsions sont négligeables si la bande passante étroite est de l’ordre de 1Khz. Les effets néfastes pris en compte se réduiront alors aux 3 points suivants :

* Le délais de transmission ( << ) ;
* L’atténuation du signal (;
* L’ajout du bruit blanc gaussien AWGN qui se note n(t) (et de densité spectrale N0/2).

Le signal transporté sur le câble aura alors pour expression :

*(Modèle du canal)*

## Le délai de transmission

Chaque canal doit être retardé d’un délais qui lui est propre. Pour réaliser le délai de transmission :

* Nous calculons un délais de déphasage aléatoire dans un intervalle [0, ].
* Nous créons une boucle *for* pour ajouter en début de chaque signal, le nombre de zéros nécessaire en fonction du délais de déphasage précédemment calculé afin de simuler le décalage temporel (zéro padding). Pour que la matrice contenant les signaux soit respectée (avoir le même nombre de lignes et de colonnes) , il ne faut pas oublier de combler le vide en ajoutant des zéros après le signal.

## L’atténuation du signal

Dans cette partie, nous devons ajouter une atténuation sur le signal avec un gain de < 1. Pour se faire :

* Nous calculons le facteur d’affaiblissement aléatoire qui est compris entre 0 et 1 (1 non compris) et nous le multiplions par notre matrice calculée au point 2.1.

## Le bruit AWGN

Nous allons réaliser dans cette partie l’ajout du bruit blanc gaussien n(t) de densité spectrale bilatérale qui est ajouté à tout le signal traversant le canal. Pour se faire, nous utilisons simplement la fonction MATALB « *awgn*() » qui permet de créer du bruit blanc Gaussien[[1]](#footnote-1). Ce bruit est ensuite ajouté à la matrice calculée au point 2.2 en mettant en paramètres cette même matrice et la valeur du SNR (signal to noise ratio) souhaité. Pour rappel, le SNR est le rapport entre la puissance du signal (c’est-à-dire l’énergie par bit) et la densité spectrale de puissance du bruit :

Où

* est l’énergie par bit (la puissance du signal divisé par le débit binaire) :
* est la densité spectrale de puissance du bruit (la puissance du bruit pour une bande de 1Hz). Cette puissance de bruit peut se calculer en multipliant avec la bande du signal.

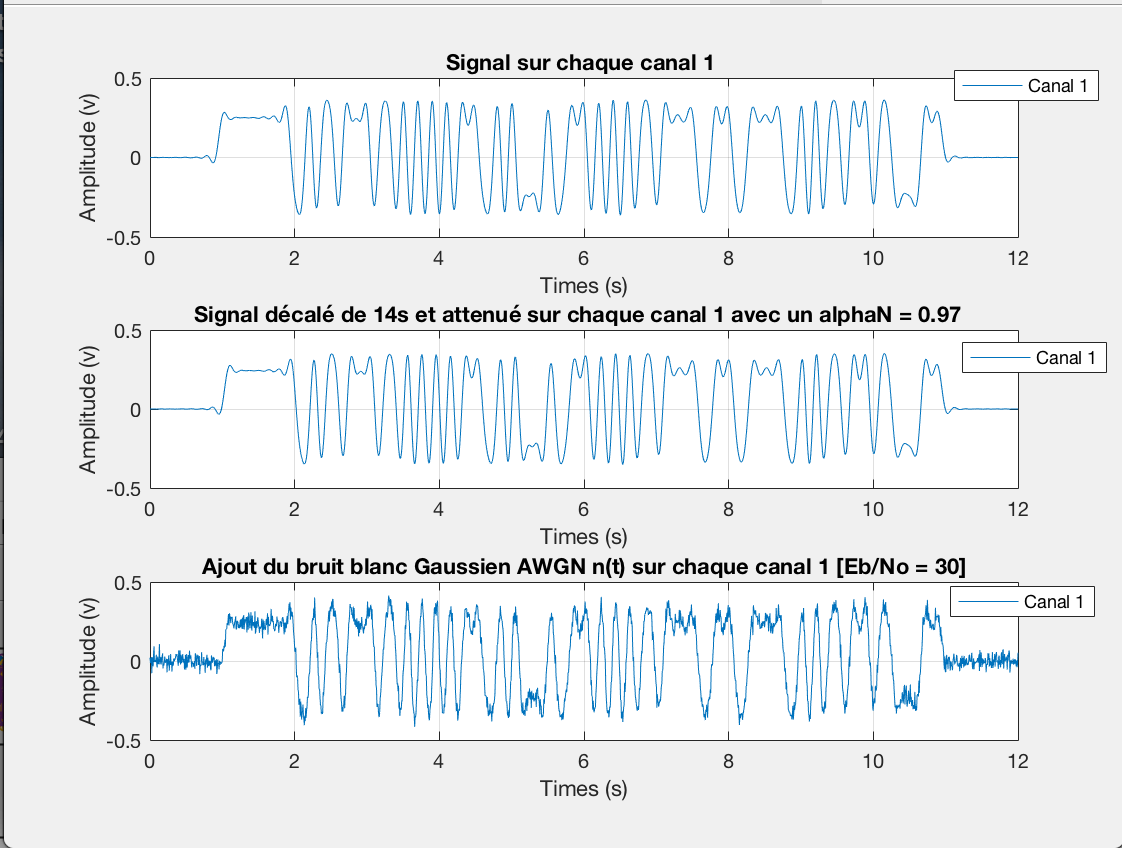


Figure 10: Atténuation, retard et bruit blanc sur le signal du canal 1

Nous avons remarqué après plusieurs essais que plus le signal s’atténuait, plus le bruit se faisait ressentir sur le signal.

## Résultat final

Nous additionnons enfin les différents signaux sur les différents (addition de chaque élément par colonne) afin de n’obtenir qu’un seul signal : celui qui sera transmis sur le câble.

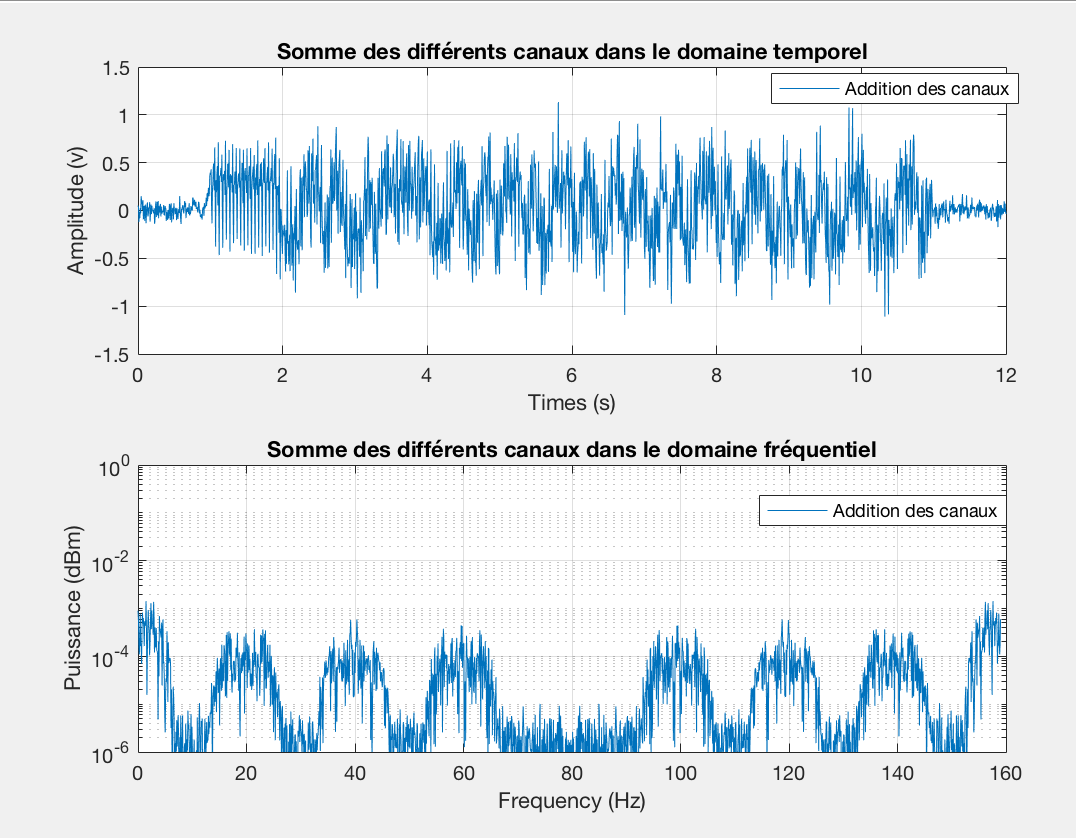


Figure 11: Somme des différents canaux dans le domaine temporel et fréquentiel

# Récepteur

Dans la section 1, nous avons défini les caractéristiques de l’émetteur afin *d’émettre* une information sur un seul des N canaux fréquentiels. Dans la section 2, nous avons aborder les caractéristiques du canal. Ainsi, nous avons remarqué que le canal apporte du bruit, de l’atténuation et du délai : ces éléments viennent polluer le signal initialement émis. Dans cette section, nous allons voir comment les **données** sont **réceptionnées** sur un canal en particulier (parmi les N canaux disponibles). Pour ce faire, nous définissons les caractéristiques d’un **récepteur**.

La figure X ci-dessous présente le schéma-bloc de la partie réceptrice. Nous allons définir chacun des termes présents sur ce schéma.



Figure 12 - Schéma bloc de la partie réceptrice.

## Filtres analogiques

Le signal transitant sur le médium est un signal analogique. Plus précisément, ce signal est formé de la somme des N signaux envoyés par les N émetteurs auquel est venu se greffer les effets parasites du canal (atténuation, délai, bruit). Ainsi, la première étape à réaliser au niveau du récepteur est la séparation du signal en N composantes fréquentielles correspondant aux N bandes sur lesquels les émetteurs ont éventuellement transmis de l’information.

* La séparation des N canaux est réalisée à l’aide de **filtres analogiques**. Pour N canaux, nous aurons donc N filtres analogiques. Nous noterons que le 1er filtre sera de type ***passe-bas*** tandis que les N-1 autres filtres seront de type ***passe-bande***.

**Un filtre idéal** serait caractérisé par une transmittance unitaire dans la bande passante étudiée et une transmittance nulle en dehors de cette bande. Cependant, ce filtre idéal n’est pas réalisable en pratique. On peut néanmoins approcher les caractéristiques d’un filtre idéal à partir de filtres réels.

**Un filtre réel** est principalement caractérisé par son ordre, noté . Au plus son ordre est élevé, au plus on se rapproche d’un filtre idéal. Les principaux filtres analogiques réalisables physiquement sont les filtres de Butterworth, de Chebyshev, de Bessel et enfin les filtres elliptiques. Outre l’ordre du filtre, nous retrouvons également d’autres caractéristiques propres au dimensionnement de ces filtres analogiques. Nous citerons ainsi les fréquences de coupures à 3 dB, le niveau d’oscillation dans la bande passante et l’affaiblissement hors-bande.

* Pour la réalisation de ce projet, nous utiliserons un **filtre de Butterworth**.

Soit n le nombre de filtres correspondant au nombre de canaux à distinguer. Si on note r(t), le signal reçu à l’entrée du récepteur, alors le signal filtré par le nième récepteur s’écrit :



Où fn(t) représente la réponse impulsionnelle du filtre analogique correspondant au nième canal fréquentiel.

### Filtre de Butterworth

Le filtre de Butterworth a été choisi comme filtre analogique pour la réalisation du récepteur.

Dès lors, sur MATLAB, une question se pose : Comment déterminer la réponse impulsionnelle de chaque filtre analogique ? On procède en 3 étapes :

1. Nous utilisons la fonction ‘*butter*’ pour déterminer les coefficients de la fonction de transfert de chaque filtre. En effet,

[a, b] = butter(n, wn) nous retourne les coefficients a et b de la fonction de transfert Fn(f) (car réponse impulsionnelle notée fn(t)) du filtre de Butterworth du nième ordre selon la fréquence de coupure normalisée Wn.

*Remarque* : Nous aurons donc besoin de connaître, au préalable, la fréquence de coupure (notée fc) des N filtres analogiques.

1. Connaissant les N\*2 fréquences de coupure, nous pouvons déterminer les différents coefficients correspondants (a et b) de chaque fonction de transfert Fn(t). Connaissant ces coefficients, nous pouvons exprimer la fonction de transfert de chaque filtre agissant sur un canal particulier. En effet, ceci est rendu réalisable grâce à la fonction ‘*freqz’*.
2. Connaissant les fonctions de transfert Fn(t) de chaque filtre, nous pouvons déterminer les réponses impulsionnelles correspondantes fn(t) en utilisant la fonction ‘*ifft’*. Cette fonction réalise la transformée de Fourier inverse.

La figure X ci-dessous représente les réponses fréquentielles de nos N filtres (dans le cas où N = 3) :

### Séparation des messages

Pour rappel, les N filtres de Butterworth, définis selon les N canaux fréquentiels, sont utilisés afin de restituer les messages correspondant aux N canaux concernés.

En effet, connaissant la réponse impulsionnelle de chaque filtres analogiques, nous sommes en mesure de **distinguer les différents messages** transporter sur le signal (sur le médium). Ainsi, si nous souhaitons obtenir le message correspondant au canal 2 (exemple), nous devons simplement réaliser la **convolution** du signal (émis sur le médium) avec la réponse impulsionnelle du filtre 2.

La figure X ci-dessous permet de visualiser les différents signaux réceptionnés par les différents filtres de Butterworth. Pour ce faire, il a suffi de faire la convolution entre le signal présent sur le canal et la réponse impulsionnelle des différents filtres analogiques (de Butterworth).

Nous avons donc ici 3 canaux, c’est-à-dire 3 messages (signaux) *visibles* dans le domaine temporelle et dans le domaine fréquentiel. Chaque canal est caractérisé par une fréquence centrale différente et ne laisse donc passer qu’une certaine gamme de fréquences délimitée par la réponse impulsionnelle du filtre (bande passante). De cette manière, nous pouvons voir sur la figure ci-dessous que nos 3 canaux sont émis sur des fréquences centrales distinctes et qu’ils possèdent tous une bande passante qui leur est propre.

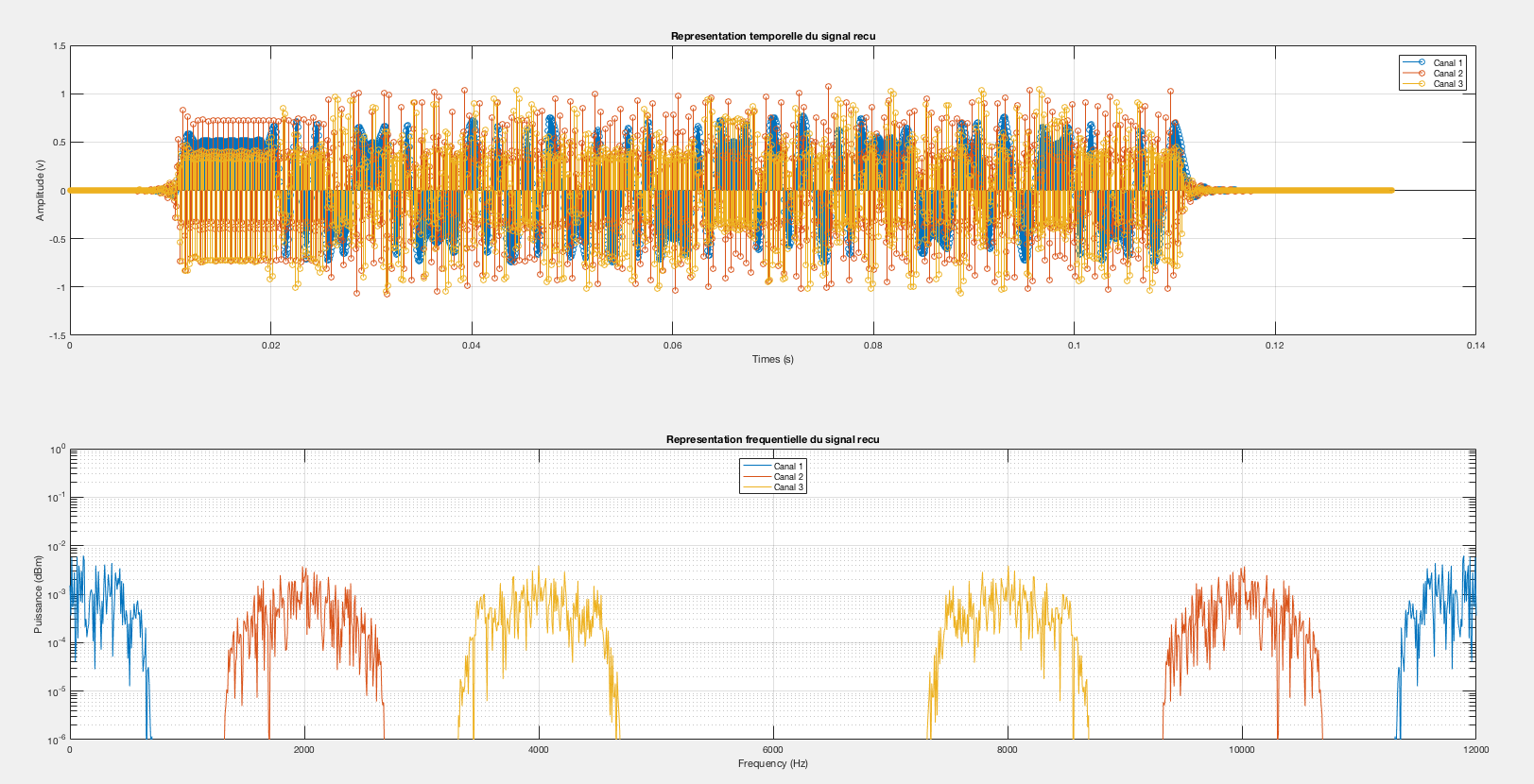


Figure 13 - Représentation temporelle et fréquentielle de messages réceptionnés après passage dans le filtre analogique.

## Filtre adapté

**Un filtre adapté** est un filtre linéaire, optimal pour maximiser le rapport signal sur bruit. En effet, en maximisant ce rapport, nous nous assurons que le message n’est pas noyé pas dans le bruit. Ces filtres sont souvent utilisés dans la détection d’un signal inconnu (bruit) en corrélation avec un signal déjà connu (message). Pour réaliser un filtre adapté, nous devons :

1. Numériser le signal via un ADC. Pour ce faire, le signal analogique en sortie du filtre de Butterworth est échantillonné. L’étape qui suit est alors la quantification.
2. Ensuite, lorsque le signal est numérisé, nous réalisons un filtrage numérique au moyen d’un FIR. Le FIR utilisé sera identique à celui de l’émetteur.

La figure X ci-dessous permet de mieux comprendre l’utilité des filtres adaptés. En effet, sur cette figure est représenté le signal avant filtrage adapté et après filtrage adapté. Nous constatons que le signal après filtrage adapté ne comporte plus de bruit.

## Synchronisation

Un rôle essentiel du récepteur concerne la synchronisation des messages. En effet, lors de son passage dans le canal, le signal a été parasité : L’un de ses parasites concerne le délai. Ce délai n’est pas connu du récepteur. Or, si nous souhaitons éviter les problèmes liés aux interférences entre symboles (ISI), il faut que l’échantillonnage des symboles soit réalisé correctement. Sinon, si nous prenons nos échantillons de façon décalée, les symboles risquent de ne plus être reconnaissable (à cause de l’interférence).

* Le ***moyen utilisé*** pour que le récepteur puisse capturer un échantillon tous les Tb instants sans créer d’ISI est de repérer le début du signal grâce à une séquence de démarrage, appelée **séquence de synchronisation**. Ainsi, comme nous l’avons vu pour l’émetteur, celui-ci introduit une séquence de démarrage de 10 bits. Cette séquence est connue du récepteur qui saura alors déterminer le début du message.

En pratique, cette synchronisation est réalisé via une corrélation croisée. Ceci permet de mesurer la similitude des 2 signaux par rapport à leurs déplacements relatifs dans le temps. En effet, lorsqu’un signal est reçu, nous savons que l’émetteur l’a fait passer au travers d’un filtre en cosinus surélevé. De cette manière, dans le récepteur, si nous prenons la séquence de démarrage connue et la passons dans le même filtre et qu’ensuite nous faisons une corrélation croisée, nous sommes alors en mesure de trouver le décalage entre les deux signaux.

## Prise de décision

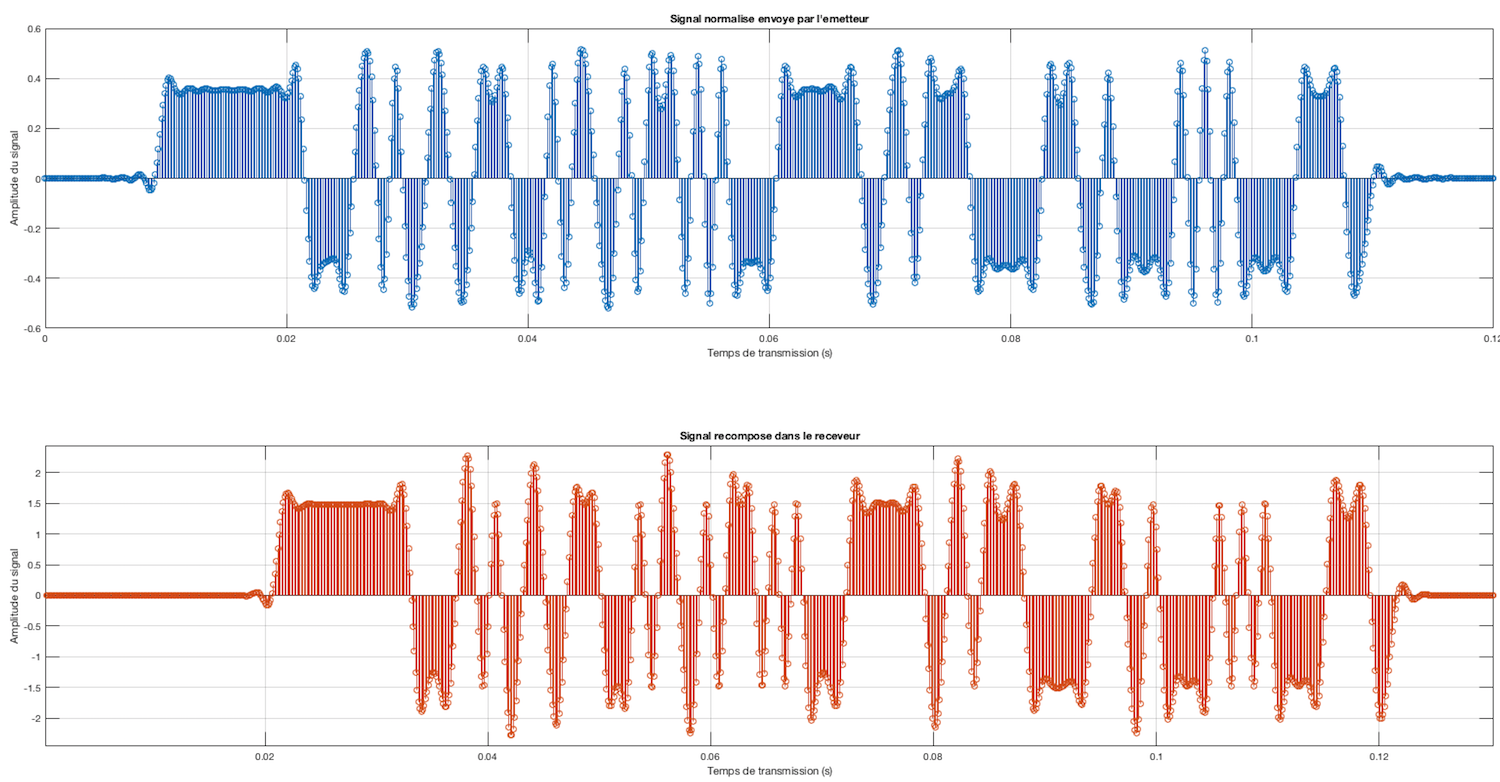
La toute dernière étape concerne la prise de décision. Ce dernier bloc va donc déterminer selon la séquence de bits obtenues après synchronisation et filtrage, si le bit doit être décodé en 0 ou 1 afin de retrouver le message initial.

* Pour prendre la décision, nous réalisons simplement le procédé inverse du codage dans l’émetteur. Ainsi :

La figure X ci-dessous compare la représentation temporelle d’un message envoyé sur un canal particulier (en bleu) et ce même message à la sortie du récepteur. Nous pouvons constater les différents effets néfastes du canal sur le message de réception (délai, atténuation).

## Simplification

N’étant parvenu à réaliser les étapes 3.2. à 3.3. sur MATLAB, nous avons contourné le problème de cette façon :



# Taux d’erreur binaire sur un canal

## Définition

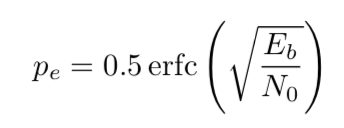
Pour rappel, si le récepteur n’est pas parfaitement synchronisé avec l’émetteur, celui-ci ne sera pas en mesure de connaître parfaitement la position des symboles de Tb. Par conséquent, nous aurons une erreur (ISI) sur l’interprétation à la sortie et le récepteur comptera plus de symboles échantillonnés que l’émetteur lui avait envoyé.

Dans la section précédente, nous avons parlé de la règle de décision qui permet au récepteur de déterminer si le symbole reçu correspondra à un « 0 » ou à un « 1 ». Pour cela, il compare le signal échantillonné réceptionné et le compare avec une valeur de seuil λ (qui vaut « 0 » dans notre cas). Si la valeur est supérieur à ce seuil, alors le symbole échantillonné vaut 1, autrement il vaut 0. Cependant, à cause du bruit, il arrive que la prise de décision soit erronée. Dès lors, on distingue 2 types d’erreur :

* Erreur de type 1 : choix du symbole 1 au lieu de 0.
* Erreur de type 2 : choix du symbole 0 au lieu de 1.

Nous devons alors jouer sur la valeur de seuil λ pour minimiser le taux d’erreur qui est dus au bruit additif, aux interférences entre les canaux/symboles et aux erreurs de synchronisation.

Rappelons que le taux d’erreur binaire théorique pour un canal binaire symétrique et un filtre adapté s’illustre par :



Où :

* Pe représente la probabilité d’erreur moyenne. Autrement dit, c’est la probabilité d’être en présence d’un bit 0 et de choisir 1, additionné à la probabilité d’être en présence d’un bit 1 et de choisir 0. Par conséquent, c’est le taux d’erreur théorique binaire sur les canaux que l’on doit détecter.
* erfc() représente la fonction d’erreur de Gauss qui est symétrique par rapport à l’origine des axes et dont le domaine est compris entre + 1 et -1.
* Ebreprésente l’énergie du signal par bit.
* N0 représente la densité spectrale de puissance du bruit N0.
* On peut voir dans la formule que Pe dépend uniquement du rapport Eb/N0

## Calcul du taux d’erreur binaire théorique en fonction de Eb/n0

Dans le point 2.3, nous avons défini que le SNR était de 30 afin de générer le bruit AWGN.

Nous obtenons alors un taux d’erreur théorique de X

Nous pouvons interpréter cette réponse et dire qu’il y a une erreur toutes les X bits envoyés.

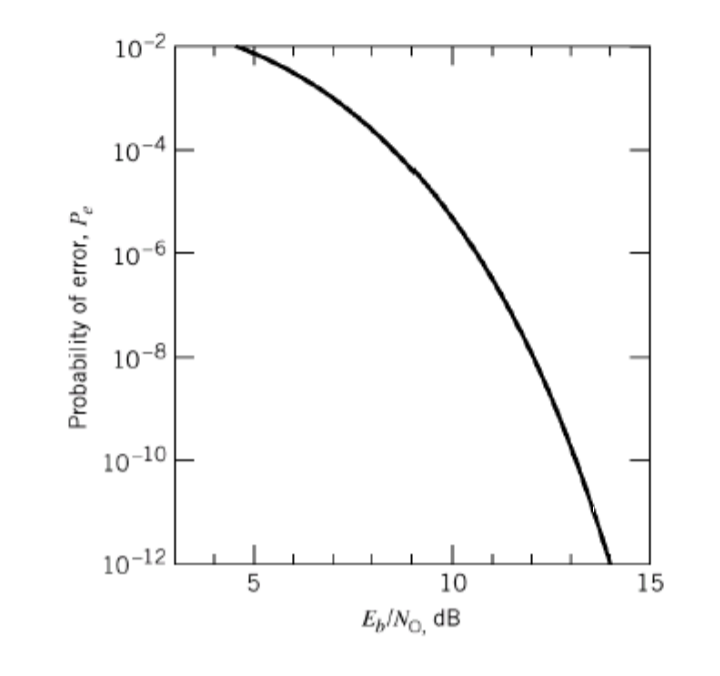
Conditions :

* Nombre de canal : 4
* Nombre de bit envoyés pour un canal : X

La taux d’erreur obtenu à la sortie du simulateur pour des faibles valeurs Eb/No est illustré à la figure suivante :

[graph matlab]

Comparaison avec la courbe théorique :



On constate ainsi que plus le rapport Eb/No augmente, plus la probabilité de faire une erreur diminue et inversement.

## Calcul du taux d’erreur binaire par l’opération logique XOR

Nous allons maintenant calculer le taux binaire par l’opération XOR entre la trame envoyée et reçue afin d’y étudier l’évolution des erreurs. Le XOR est représenté par une matrice de même taille que la trame envoyée ou reçue qui indique :

* 1 🡪 s’il y a une différence entre la trame envoyée et reçue ;
* 0 🡪 si tous les bits sont identiques.

Ensuite, il suffit d’additionner toutes les différences, soit la somme de tous les 1 pour ensuite la diviser par longueur de la trame représentant notre message. Nous obtenons alors le taux d’erreur binaire.

## Comparaison des deux taux d’erreur binaire

* Expliquer comment vous avez ajusté la variance des échantillons de bruit pour obtenir le rapport eb/No souhaité.

1. AWGN peut être symbolisé par une variable aléatoire de distribution normale et de moyenne nulle. [↑](#footnote-ref-1)