



Master Informatique et Télécommunication

Rapport :

Réalisation d'une chaîne de transmission numérique

Communication Numérique

Réalisé par :

CHETOUANI Yassin

Encadré par :

Prof Jilbab Abdelilah

Table de matières:

Chapitre 1 : Introduction générale.....	6
I. Introduction :	6
Chapitre 2 : Transmission en bande de base.....	8
I. Codage en ligne :	8
I.1 Le code NRZ binaire (Non-Retour à Zéro)	10
I.2 Codage RZ binaire (Retour à Zéro) :	12
I.3 Codage Biphase (code Manchester):	13
I.4 Codage bipolaire :	15
II. Transmission du signal émis :	16
III. Modulation\ Démodulation numérique :	20
III.1 Modulation d'amplitude AM :	21
III.2 Modulation de Phase PM :	22
III.3 Modulation de Fréquence FM :	24
III.4 Démodulation du signal :	25
IV. Récupération du signal du message (symboles) d'origine :	27
Chapitre 3 :	32
Simulation des différentes composantes de la chaine de transmission sous MATLAB.....	32
Chapitre 4 :	42
Simulation et mise en œuvre d'une chaine de transmission sous SIMULINK	42

Table des figures :

Figure 1: Chronogramme du code NRZ binaire.

Figure 2: Densité spectrale de puissance d'un code NRZ binaire.

Figure 3: Chronogramme du code RZ binaire pour $V=1$

Figure 4: Densité spectrale de puissance d'un code RZ binaire.

Figure 5: Chronogramme du code Manchester binaire pour $V=1$

Figure 6: Densité spectrale de puissance d'un code Manchester binaire.

Figure 7 : Chronogramme du code bipolaire RZ.

Figure 8: Signal reçu avec IE

Figure 9: Réponse impulsionnelle d'un filtre d'émission

Figure 10 : a) cosinus surélevé dans le domaine fréquentiel

Figure 10 : b) cosinus surélevé dans le domaine temporel

Figure 11 : Modulation d'amplitude

Figure 12: Exemple d'une modulation ASK

Figure 13: spectre d'un signal ASK modulé par un signal numérique **non** filtré

Figure 14 : Modulation de phase

Figure 15: Le diagramme de constellation pour une modulation BPSK

Figure 16: Exemple d'une modulation PSK

Figure 17: Modulation FSK

Figure 18: Exemple d'une modulation FSK

Figure 19: Spectre d'un signal FSK modulé par un signal numérique non filtré

Figure 20: processus de démodulation d'un signal

Figure 21: Structure du récepteur

Figure 22: Représentation des densités de probabilité conditionnelles

Figure 23: Extraction du diagramme d'œil à l'aide d'un oscilloscope

Figure 24 : simulation d'une chaine de transmission sur simulink

Présentation du projet :

Une bonne présentation d'un sujet a toujours facilité la compréhension et éclairé sur les différentes tâches à accomplir dans le cadre d'une réalisation d'un projet. Dans cette partie nous donnerons le contexte de réalisation du projet.

Contexte du projet :

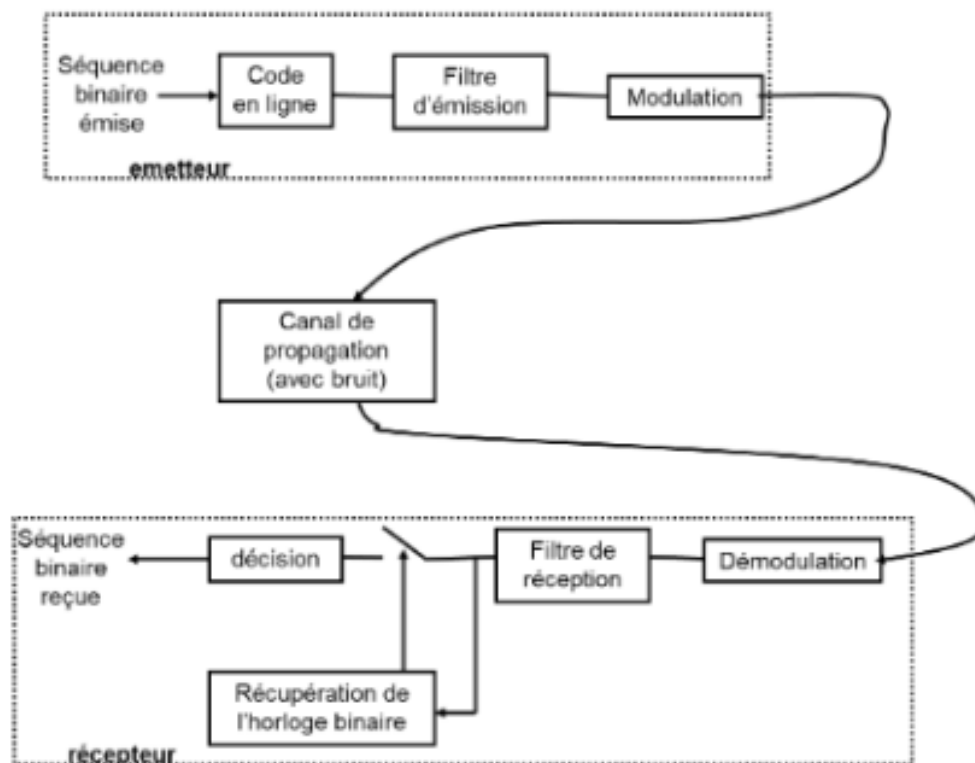
Suite à une formation, de communication numérique, présentée par notre professeur Jilbab Abdelilah. Ce projet vient pour compléter, approfondir et toucher la partie applicative des nouvelles notions acquises.

Cependant, nous sommes chargés dans ce projet de réaliser une application qui reflète le fonctionnement d'une machine de transmission numérique, et de concevoir une IHM facilitant l'utilisation de l'application.

Architecture Applicative :

Le cahier de charge qui nous a été confié prenant en compte les traitements suivants :

- La chaîne de transmission numérique contenant au moins les blocs suivants :



- Effectuer une étude sur les différents outils de construction de la chaîne de transmission.
- Concevoir une IHM facilitant l'utilisation de l'application.

Choix du logiciel de programmation :

Vue que le choix du logiciel de programmation n'était pas imposé, alors on s'est trouvé devant un multi choix (Matlab, Python, LabVIEW, Java, ...).

Notre choix est posé sur Matlab car il nous permet de résoudre certains problèmes liés au traitement du signal et l'information, en essayant de simuler le système. Ainsi nous permet de représenter graphiquement notre système, la chaîne de transmission numérique, à l'aide de « Simulink » qui est un logiciel de simulation numérique qui se trouve dans Matlab et qui nous donne une flexibilité d'insérer des blocs en les déplaçant dans notre champ de travail.



Chapitre 1 : Introduction générale

I. Introduction :

La communication correspond à l'échange d'information (sous forme de signal) entre un émetteur et un récepteur à l'aide d'un canal de transmission.

Une chaîne de transmission d'information est l'ensemble des dispositifs permettant le transport d'une information d'un lieu à un autre.

Le schéma synoptique d'une chaîne de transmission numérique est donné à la figure suivante :



Source : émet un message numérique sous la forme d'une suite d'éléments binaires.

Encodeur : Consiste à transformer le message en une séquence d'éléments binaires mutuellement indépendantes et prenant la valeur 0 et 1. Ainsi éliminer certains éléments binaires non significatifs et corriger les erreurs.

(Former de trois codeurs : -Codeur de source (compression de l'information).

-Codeur de canal (correction des erreurs).

-Codage en ligne (mise en forme des données sous forme d'impulsions).

)

Modulateur : représente chaque symbole par l'état électrique du signal porteur (soit par amplitude, phase ou fréquence) et adapter le spectre du signal au canal.

Canal de transmission :

Démodulateur	}	Sont du côté du récepteur, le démodulateur et décodeur fonctionnent
Décodeur		
		Inversement des fonctions de modulation et codage situées du côté émetteur.

Destinataire : Reçoit et retrouve le message de l'émetteur.

Chaque canal de transmission est caractérisé par trois caractéristiques principales :

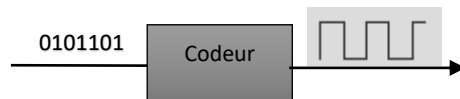
- La probabilité d'erreur P_e par bit.
- L'occupation spectrale du signal.

- La complexité du récepteur.

Pour évoluer la qualité du canal il faut varier les valeurs de ces paramètres dans le sens d'optimisation, en minimisant leurs valeurs.

Chapitre 2 : Transmission en bande de base

La transmission d'une information numérique passe par la création d'un signal au départ qui sera numérisé (convertie de l'analogique au numérique). Ce signal va être traité par un encodeur dans le but de transformer une séquence de bits en une séquence de symbole électrique.



Cette phase d'encodage est considérée comme un premier traitement pour construire une chaîne de transmission en bande de base, où le signal ne subit pas de transposition de fréquence et se réduit à un signal simple codé.

Avant l'injection des informations dans un canal, un codage est nécessaire. Le choix d'un code est fonction du spectre de ce code, la bande passante disponible du bruit et interférence du canal, du contrôle de la performance, de la tenue en horloge et la fiabilité de la réalisation.

Critères de choix d'un code en ligne :

- ✚ Rapidité de modulation
- ✚ Sensibilité au bruit
- ✚ Occupation spectrale
- ✚ Récupération d'horloge en réception (lié au spectre du code : il est bon que celui-ci présente une raie à la fréquence d'horloge)

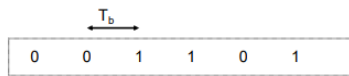
Un canal de transmission est caractérisé par :

- ✓ Débit binaire en bit/s : $D_{bit/s} = \frac{1}{T_b}$
- ✓ Débit de symbole ou rapidité de modulation en baud :
$$R_{(baud)} = \frac{1}{T_s} = \frac{1}{n \cdot T_b} = \frac{D_{bit/s}}{n} = \frac{D_{bit/s}}{\log_2 M}$$

I. Codage en ligne :

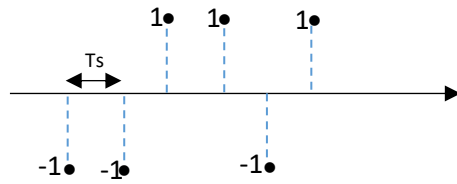
Considérons la transmission d'un message constitué par une suite d'éléments binaires a_k émis aux instants kT_b indépendants et identiquement distribués (i-i-d) sur l'alphabet $\{0,1\}$,

avec: $P_0 = P_r\{a_k = 0\}$ et $P_1 = P_r\{a_k = 1\} \quad \forall k \quad (P_0 = P_1)$



Message binaire

On associe à chaque bit, un symbole électrique a_k puis on les multiplie par le peigne de dirac pour obtenir des impulsions d'amplitude a_k (qui vont exciter le prochain filtre).



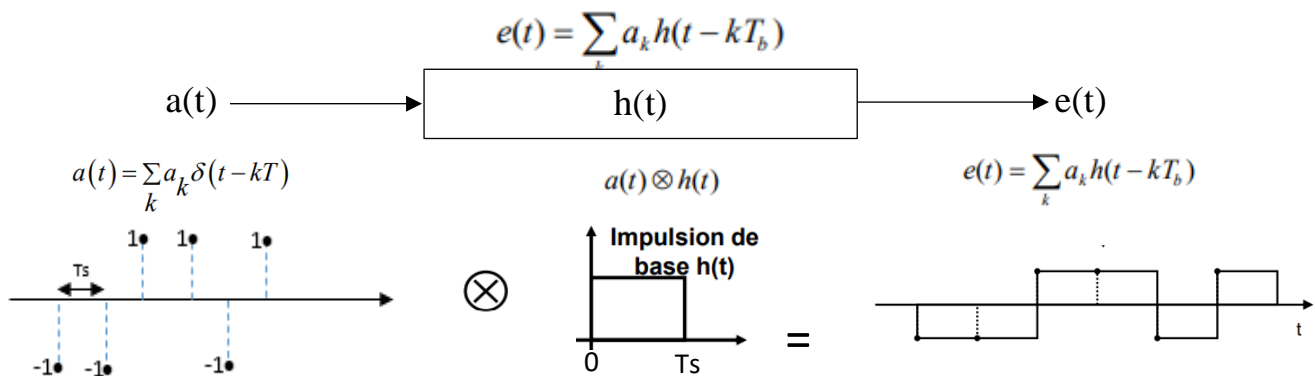
Séquence des symboles Ex : $\{-$

$$a(t) = \sum_k a_k \delta(t - kT)$$

Comme le codeur en ligne nous permet de générer une signal $e(t)$ continue, alors pour chaque élément binaire du message, un signal de durée (tel que : m peut prendre n'importe quelle valeur dépend de la modalité)

Ex : $m = \{-1, 1\}$

Signal d'émission $e(t) = \begin{cases} \text{Si } a_k = -1 & \text{émission du signal } e_{-1}(t - kT_b) \\ \text{Si } a_k = 1 & \text{émission du signal } e_1(t - kT_b) \end{cases}$

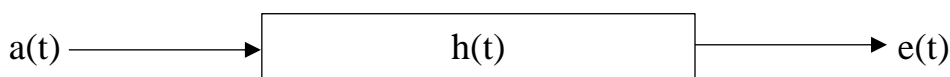


Choisir $h(t)$ afin d'avoir des propriétés particulières sur le spectre du signal numérique émis :

- spectre avec une composante continue (alimentation)
- occupation spectrale (largeur lobe principal)
- présence de raie à la fréquence symbole (régénération de l'horloge symbole à la réception)

La densité spectrale de puissance du code en ligne :

On a :



Alors : $e(t) = a(t) \otimes h(t)$

On sait bien que si on passe au domaine fréquentiel l'opération de convolution se transforme en produit, on obtient l'expression de la densité spectrale de puissance $\gamma_e(f)$ du signal de sortie $e(t)$:

$$\gamma_e(f) = \gamma_a(f) |H(f)|^2$$

Avec : $S(f)$ est la transformée de Fourier de $h(t)$ et $\gamma_a(f)$ est la densité spectrale des symboles qu'on peut la calculer à l'aide de la relation suivante :

$$\gamma_a(f) = \frac{\sigma_a^2}{T} + \frac{2\sigma_a^2}{T} \sum_{k=1}^{+\infty} \Gamma'_a(k) \cos 2\pi k f T + \frac{m_a^2}{T^2} \sum_{-\infty}^{+\infty} \delta(f - \frac{k}{T})$$

m_a : La moyenne, qui peut être calculer par la relation : $m_a = E[a_k]$

σ_a^2 : La variance, qui peut être calculer par la relation : $\sigma_a^2 = E[(a_k - m_a)^2]$

$R_a(k)$: la fonction d'autocorrélation normalisée des symboles centrés, qui peut être calculer par :

$$\frac{E[(a_n - m_a)(a_{n-k} - m_a)]}{\sigma_a^2} \quad \forall n, k$$

Exemples de codes en ligne :

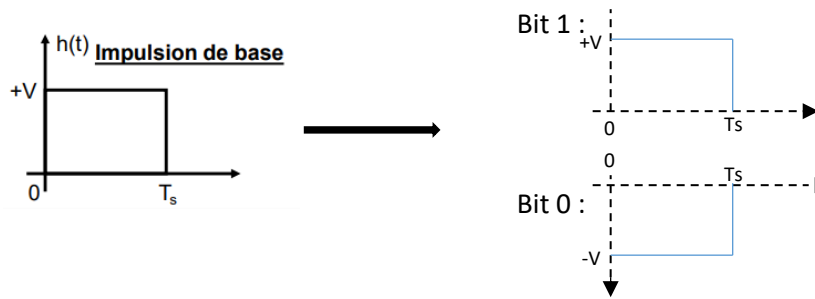
- ❖ Codes en ligne à symboles indépendants (décorrélés)
 - Le code NRZ binaire (Non Retour à Zéro)
 - Le code RZ binaire
 - Le code biphasé binaire (Manchester)
- ❖ Codes en ligne à symboles dépendants (corrélés)
 - Le code bipolaire
 - Le code HDBn

Ces codes se différencient, outre par leur règle, par le spectre du signal électrique qui en résulte. (Après on va traiter les différents codeurs que pour le cas 2-à-2 niveaux).

I.1 Le code NRZ binaire (Non-Retour à Zéro)

C'est un code très simple à manipuler. Il est caractérisé par une fonction de sortie $e(t)$ qui prend $-V$ et $+V$ pendant une période T_s ($T_b = T_s$) tq :

$$\text{La fonction de mise en forme } h(t) : \begin{cases} \text{Si } a_k = 0 \rightarrow e_k(t) = -V \\ \text{Si } a_k = 1 \rightarrow e_k(t) = V \end{cases}$$



Exemple d'un message $m = \{0\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\}$: (pour $V=1$)

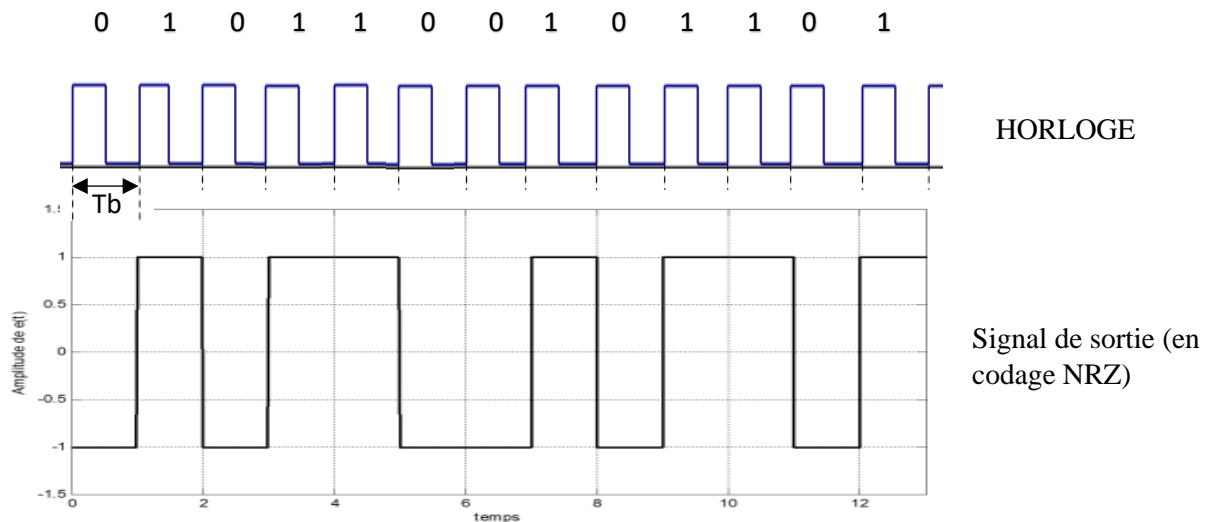


Figure 1: Chronogramme du code NRZ binaire.

Densité spectrale de puissance du code NRZ binaire :

On sait bien que les symboles $a_k = \{0,1\}$ sont équiprobables donc la moyenne m_a est nulle et la variance σ_a^2 égale à 1. Ainsi que $H(f) = TV \frac{\sin(\pi f t)}{\pi f t}$, avec $V=1$.

On aura :

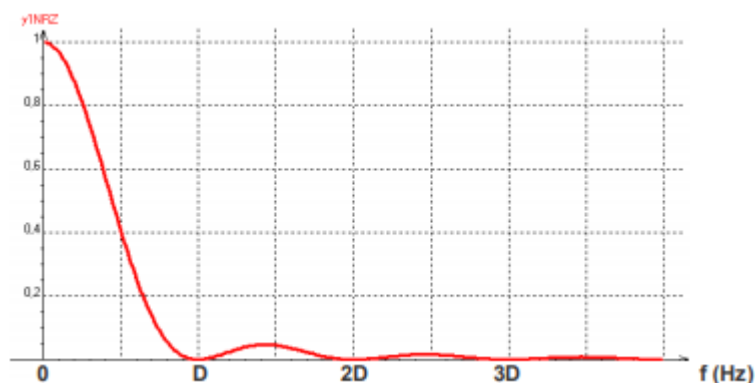
$$\gamma_e(f) = V^2 T_b \frac{\sin^2(\pi f t)}{(\pi f t)^2}$$


Figure 2: Densité spectrale de puissance d'un code NRZ binaire.

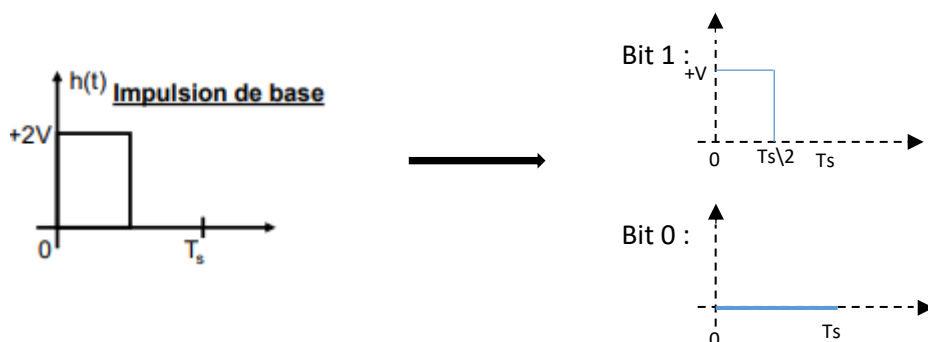
- Lobes de largeur $\Delta f = D$ Le lobe principal renferme 91% de la puissance du signal.
- Absence de composantes aux fréquences multiples du débit, d'où difficulté pour régénérer l'horloge de synchronisation à la réception.

- Forte densité de puissance en continu : Transmission impossible par le biais de transformateurs d'isolement.

I.2 Codage RZ binaire (Retour à Zéro) :

Il est caractérisé par une fonction de sortie $e(t)$ qui prend soit 0 ou $+V$ tq :

La fonction de mise en forme $h(t)$:
$$\begin{cases} \text{Si } a_k = 0 \rightarrow e_k(t) = 0 \\ \text{Si } a_k = 1 \rightarrow e_k(t) = \begin{cases} V, \forall t[0, T_b/2[\\ V, \forall t[T_b/2, T_b[\end{cases} \end{cases}$$



Exemple d'un message $m = \{0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1\}$: (pour $V=1$)

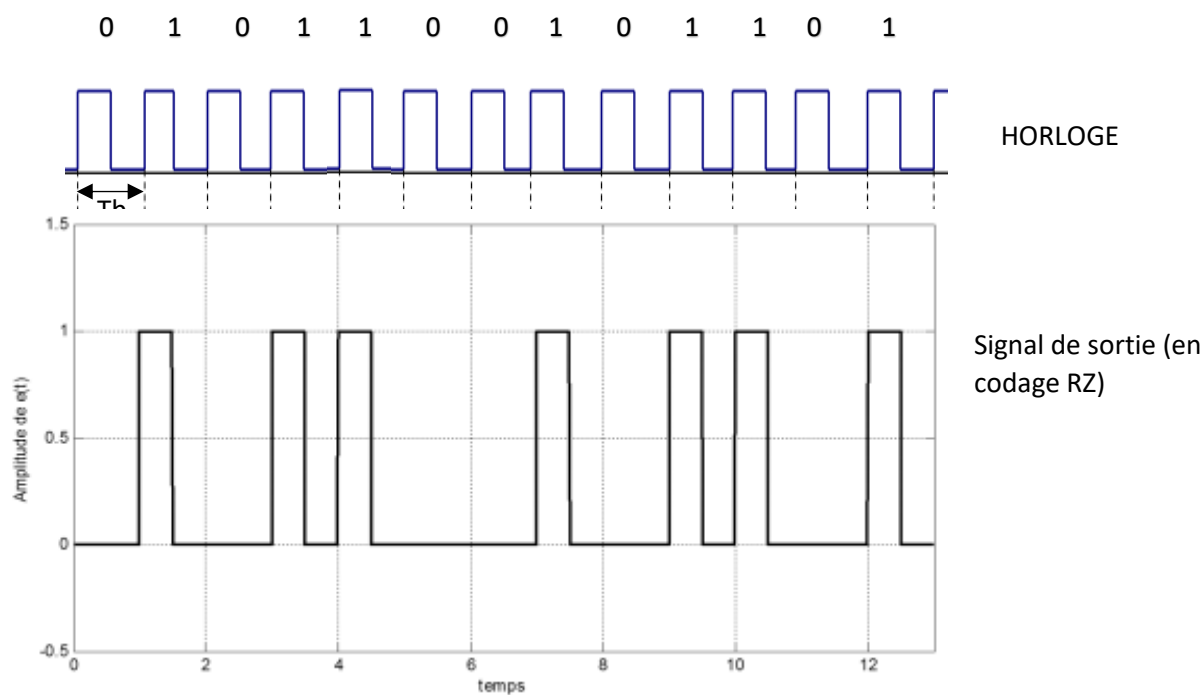


Figure 3: Chronogramme du code RZ binaire pour $V=1$

Densité spectrale de puissance du code RZ binaire :

On sait bien que les symboles $a_k = \{0,1\}$ sont équiprobables donc la moyenne m_a égal 1/2 et la variance σ_a^2 égale à 1/4. Ainsi que $H(f) = \frac{1}{2}T V \frac{\sin(\frac{1}{2}\pi f t)}{\frac{1}{2}\pi f t}$,

Avec $V=1$.

On aura : $\gamma_e(f) = \frac{V^2 T_b}{4} (\text{sinc}^2(\pi f T_s / 2))^2 + \frac{V^2}{4} \sum_{k=-\infty}^{+\infty} (\text{sinc}^2(\pi f T_b / 2))^2 \cdot \delta(f - k / T_b)$

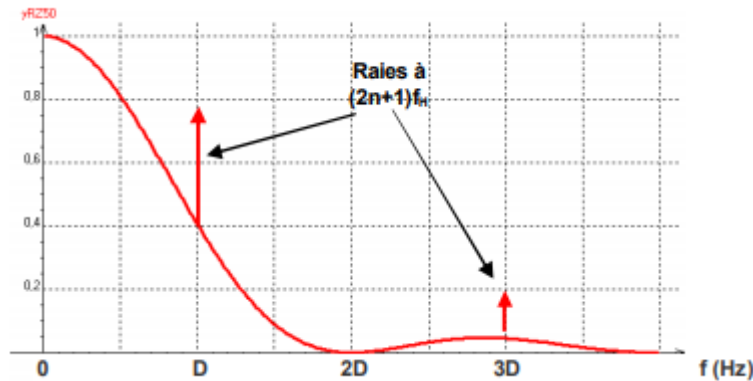


Figure 4: Densité spectrale de puissance d'un code RZ binaire.

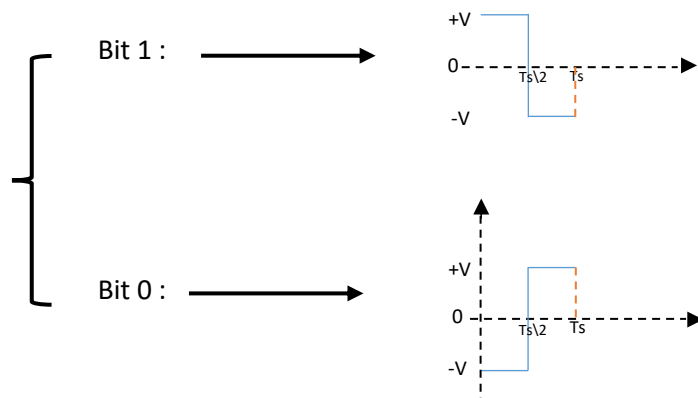
- Lobes de largeur $\Delta f = 2D$
- Le lobe principal renferme 90% de la puissance du signal.
- Spectre mixte, comportant des raies aux fréquences multiples de la fréquence d'horloge, d'où facilité de récupération de cette horloge à la réception.
- Comme le codage NRZ, la présence d'une composante continue interdit l'utilisation de transformateurs d'isolement dans le dispositif de transmission.

I.3 Codage Biphase (code Manchester):

Il est caractérisé par une fonction de sortie $e(t)$ qui prend $-V$ et $+V$ (alternativement) pendant une période une demi période $T_s/2$ ($T_b=T_s$) tq :

La fonction de mise en forme $h(t)$:

$$\left\{ \begin{array}{l} \text{Si } a_k = 0 \rightarrow e_k(t) = \begin{cases} -V & ; \forall t[0, T_b/2[\\ +V & ; \forall t[T_b/2, T_b[\end{cases} \\ \text{Si } a_k = 1 \rightarrow e_k(t) = \begin{cases} +V & ; \forall t[0, T_b/2[\\ -V & ; \forall t[T_b/2, T_b[\end{cases} \end{array} \right.$$



Exemple d'un message $m = \{0\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\}$: (pour $V=1$)

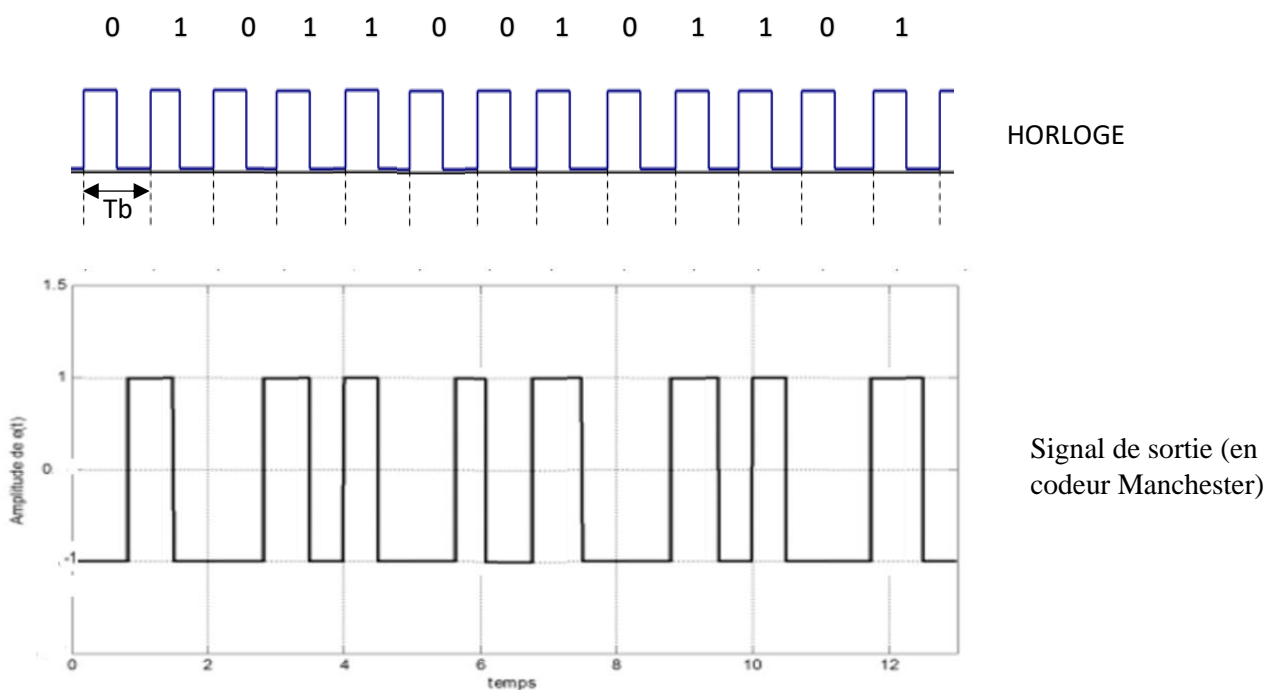


Figure 5: Chronogramme du code Manchester binaire pour $V=1$

Densité spectrale de puissance du code Manchester binaire :

On sait bien que les symboles $a_k = \{0,1\}$ sont équiprobables, indépendants de moyenne m_a nulle et la variance σ_a^2 égale à 1. Ainsi que $H(f) = T V \cdot \sin(\frac{1}{2} \pi f t) \cdot (\frac{\sin(\pi f t)}{\pi f t})$, avec $V=1$.

On aura :

$$\gamma_e(f) = V^2 T_b \cdot \sin^2(\frac{1}{2} \pi f t) \cdot \frac{\sin^2(\pi f t)}{(\pi f t)^2}$$

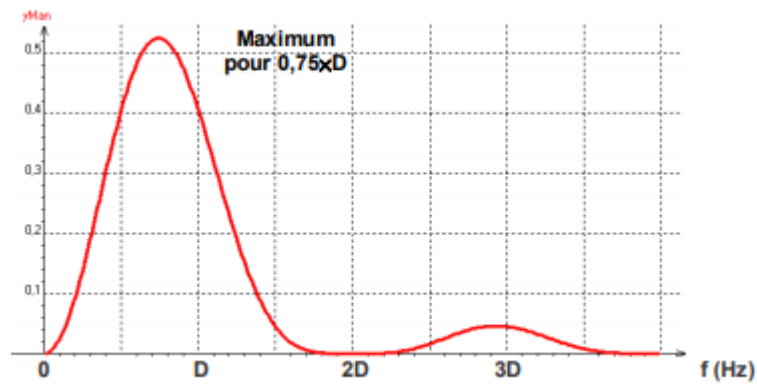


Figure 6: Densité spectrale de puissance d'un code Manchester binaire.

- Lobes de largeur $\Delta f = 2D$
- Le lobe principal renferme 85% de la puissance du signal.
- Pas de composante continue.
- Récupération de l'horloge plus aisée.

I.4 Codage bipolaire :

Il est caractérisé par une fonction de sortie $e(t)$ qui prend soit 0 (si l'élément a_k est égale à 0) ou V sur une demi période (si l'élément a_k est égale à 1) tq :

La fonction de mise en forme $h(t)$:

$$\begin{cases} \text{Si } a_k = 0 \rightarrow e_k(t) = 0 \\ \text{Si } a_k = 1 \rightarrow e_k(t) = \begin{cases} \pm V & ; \forall t[0, Tb/2[\\ 0 & ; \forall t[Tb/2, Tb[\end{cases} \end{cases}$$

Exemple d'un message $m = \{0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1\}$: (pour $V=1$)

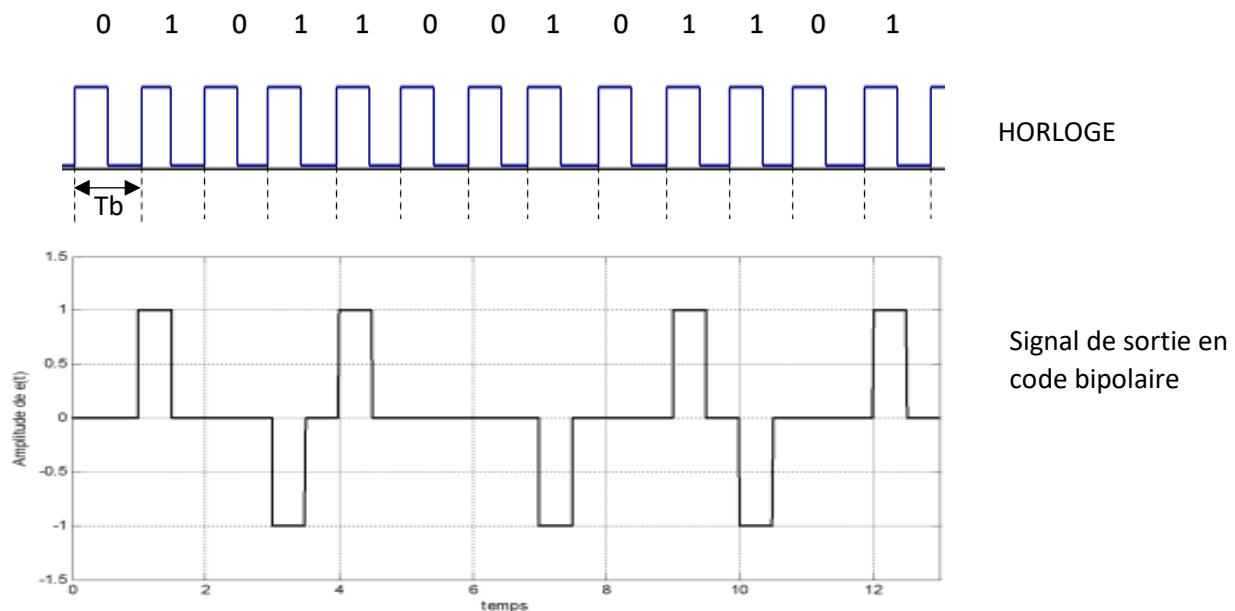


Figure 7 : Chronogramme du code bipolaire RZ.

La densité spectrale de puissance du code bipolaire RZ est donnée par :

$$H(f) = VT. \sin(\pi fT). \text{sinc}\left(\frac{fT}{2}\right)$$

On aura : $\gamma_e(f) = \frac{V^2T}{4} \cdot \sin^2(\pi fT). \text{sinc}^2\left(\frac{fT}{2}\right)$

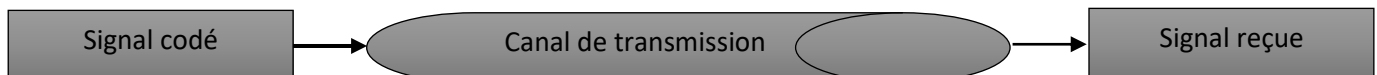
Remarque :

-Le codage le plus mal adapté est le NRZ puisque la naissance est concentrée dans les basses fréquences. Le codage biphase à un spectre particulièrement large et ne peut être envisagé sur les supports en large bande (exemple : câble coaxial, câble croisé). On préfère en général le code bipolaire qui pour une même largeur de bande permet un débit au moins 2 fois plus élevé.

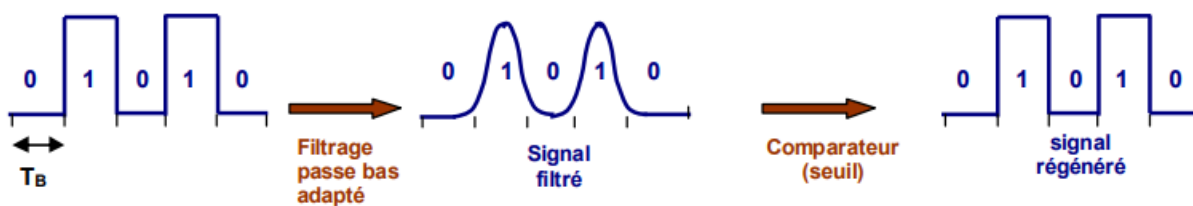
-L'essentiel de l'information est concentrée dans la lobe principale (on peut la vérifier par le calcul de la puissance du lobe principal : 96% de l'information se concentre dans ce lobe) : L'idée est de filtrer par un filtre passe bas respectant le 1er lobe et atténuant au maximum les suivants.

II. Transmission du signal émis :

Après la formation (codage) du signal et de parcourir les différents organes de la chaîne de transmission :



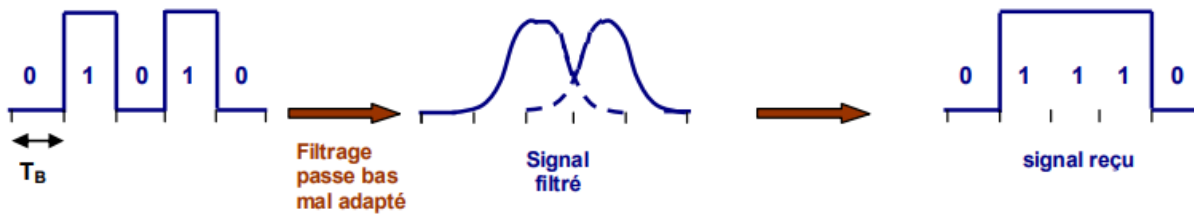
Comportement de la chaîne de transmission :



La récupération des symboles se fait par comparaison des valeurs du signal par un seuil.

Mais le problème n'est pas si simple, car si le signal émis n'est pas adapté vu que la bande passante d'un canal de transmission est toujours limitée, ce qui induit un étalement dans le temps des impulsions acheminées dans ce canal. Ce défaut entraîne un

recouvrement entre l'impulsion correspondante à des périodes d'échantillonnage adjacentes.



Alors, si le filtre n'est pas adapté au débit du canal, on aura des interférences entre symboles qui va créer une confusion entre les bits donc avoir un message erronément reçu.

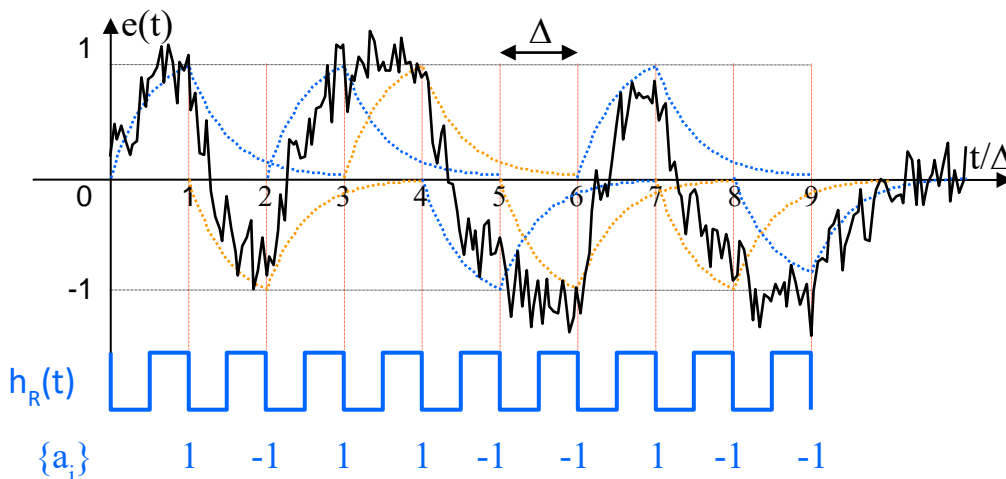
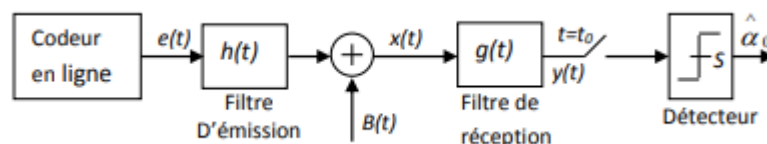


Figure 8: Signal reçu avec IE

Solution :

Nous allons cette fois-ci ajouter un filtre d'émission qui va nous permettre de réduire la puissance du bruit et diminution de la largeur de la bande du signal émis.

Conception des filtres d'émission et de réception :



La meilleure façon d'éviter l'interférence entre symboles d'occurrence est l'utilisation d'une impulsion similaire à celle représentée sur la figure ci-dessous: Il est tout à fait évident que de cette façon, dans le cas idéal où le moment du signal coïncide exactement avec celui de la transmission, les symboles interfèrent jamais entre eux depuis le pic de chaque impulsion, qui est toujours en principe coïncide avec la valeur championnat du récepteur, vient toujours de tomber à un point où tous d'autres impulsions sont exactement zéro, au-delà du code de signal.

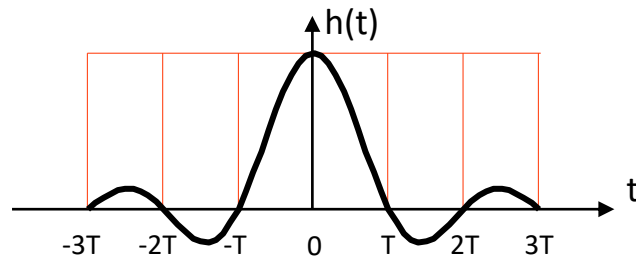
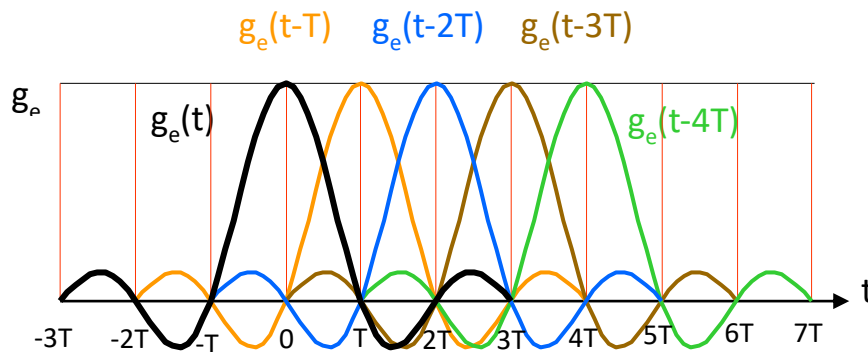


Figure 9: Réponse impulsionnelle d'un filtre d'émission

L'interférence entre les symboles est liée à la bande passante limitée du canal qui engendre un étalement temporel des symboles élémentaires. Alors, pour assurer une bonne régénération des symboles (absence totale de IES) Nyquist a imposé une condition de non interférence sur le filtre :

Condition temporelle de non interférence entre les symboles



La réponse temporelle fait bien apparaître des zéros situés à des multiples de la période de symbole :

Pour réduire l'interférence inter symbole à son minimum, il conviendra d'échantillonner le signal à ces instants.

Critère de Nyquist temporal :

Pour permettre une transmission sans IES, la bande passante du canal de propagation doit vérifier : $B \geq \frac{1}{2T}$;

Avec B : la largeur de la bande du canal

T : la période entre deux symboles successifs

Critère de Nyquist fréquentiel:

Le spectre de la réponse impulsionnelle admet le point : $(\frac{1}{2T}, \frac{G(0)}{2})$

La périodisation du spectre d'impulsion de base, cadencé avec un pas égal à la fréquence d'échantillonnage, devrait donner lieu à une tendance parfaitement constante.

Le signal de bande minimale fait observer que cette condition est la Sinc. Il y a bien sûr d'autres tendances, certes moins pure, mais à certains égards, mieux qu'une bande passante minimale, répondant à cette contrainte. La raison pour laquelle Sinc est en fait peu utilisée dans la pratique réside dans la mauvaise immunité aux erreurs de synchronisation introduites tant par l'émetteur que la phase d'échantillonnage de récepteur. Filtre utilisé dans les normes telecoms : **filtre en cosinus surélevé**

Expression du cosinus surélevé :

$$G(f) = \begin{cases} \frac{T_s}{2} \left\{ 1 - \sin\left[\frac{\pi}{2\alpha}(2|f|T_s - 1)\right] \right\} & ; \frac{1-\alpha}{2T_s} \leq |f| \leq \frac{1+\alpha}{2T_s} \\ T_s & ; 0 \leq |f| \leq \frac{1-\alpha}{2T_s} \\ 0 & ; \text{ailleurs} \end{cases}$$

Avec : T : la durée d'un symbole

α : Coefficient de retombée

Représentation graphique du cosinus surélevé :

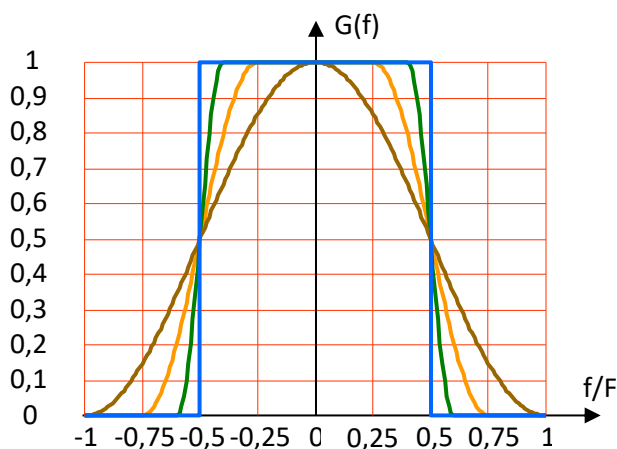


Figure 10 : a) cosinus surélevé dans le domaine fréquentiel

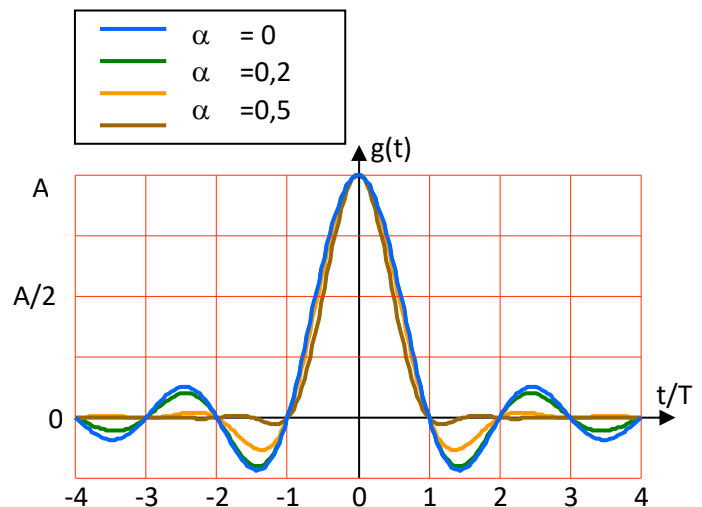


Figure 10 : b) cosinus surélevé dans le domaine temporel

Remarque :

On cas réel à la sortie du codeur en ligne, le signal est continué. Par contre le filtre de Nyquist doit être excité par des impulsions (le cas qu'on a étudié est idéal). Alors pour résoudre le problème on utilise un filtre blanchissant.

Filtre blanchissant :

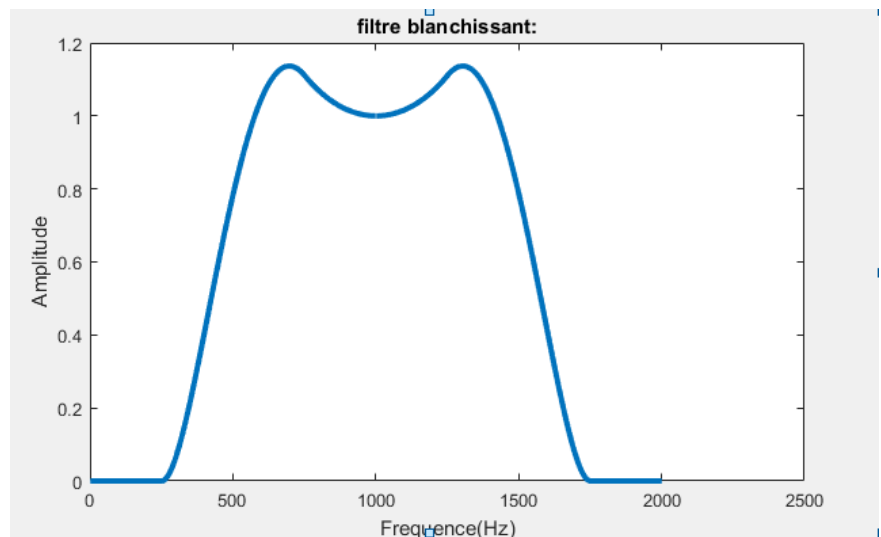
C'est un filtre qui transforme un signal réel continue en une suite d'impulsions. Alors, c'est la fonction inverse de la fonction de mise en forme et qui vérifie la relation : $H(f) * G_e(f) = 1$

$H(f)$: filtre de mise en forme

$G_e(f)$: Filtre blanchissant

Le filtre blanchissant est représenté comme ci-dessous :

$$G_e(f) = \frac{1}{T_b} \frac{\pi f T_b}{\sin(\pi f T_b)}$$



Pour ajouter plus d'amélioration sur ma chaine de transmission on ajoute un bloc de MODULATION (du côté émetteur) et de DEMODULATION (du côté récepteur) qui nous permet d'assurer une transmission d'un signal de basse fréquence et qui a pour objectif d'adapter le signal à émettre au canal de transmission.

III. Modulation\ Démodulation numérique :

La modulation consiste à transformer un signal sinusoïdal connu appelé porteuse par le signal à transmettre.

Un signal sinusoïdal est entièrement défini par trois caractéristiques qui sont :

- son amplitude A
- sa fréquence f
- sa phase θ

A partir de ces trois paramètres on peut définir trois types de modulation :

- Modulation d'Amplitude (AM)
- Modulation de Phase (PM)
- Modulation de Fréquence (FM).

III.1 Modulation d'amplitude AM :

La modulation d'amplitude consiste à modifier l'amplitude de la porteuse.

La plus simple technique pour avoir une modulation ASK est la multiplication du signal numérique par une porteuse.

Soit un signal carré : $s(t) = \sum_{k=0}^{+\infty} a_k \text{rect}(t - kT_s)$

La porteuse : $p(t) = A_k \cdot \cos(2\pi f_0 + \varphi)$

Le signal qui résulte de la modulation : $A_k \cdot \sum_{k=0}^{+\infty} a_k \text{rect}(t - kT_s) \cdot \cos(2\pi f_0 + \varphi)$

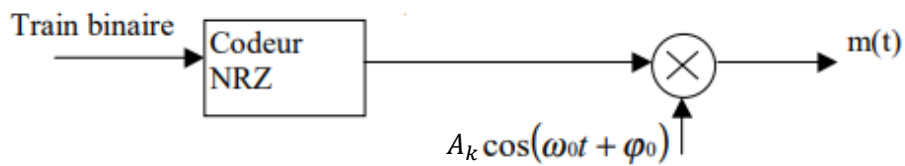


Figure 11 : Modulation d'amplitude

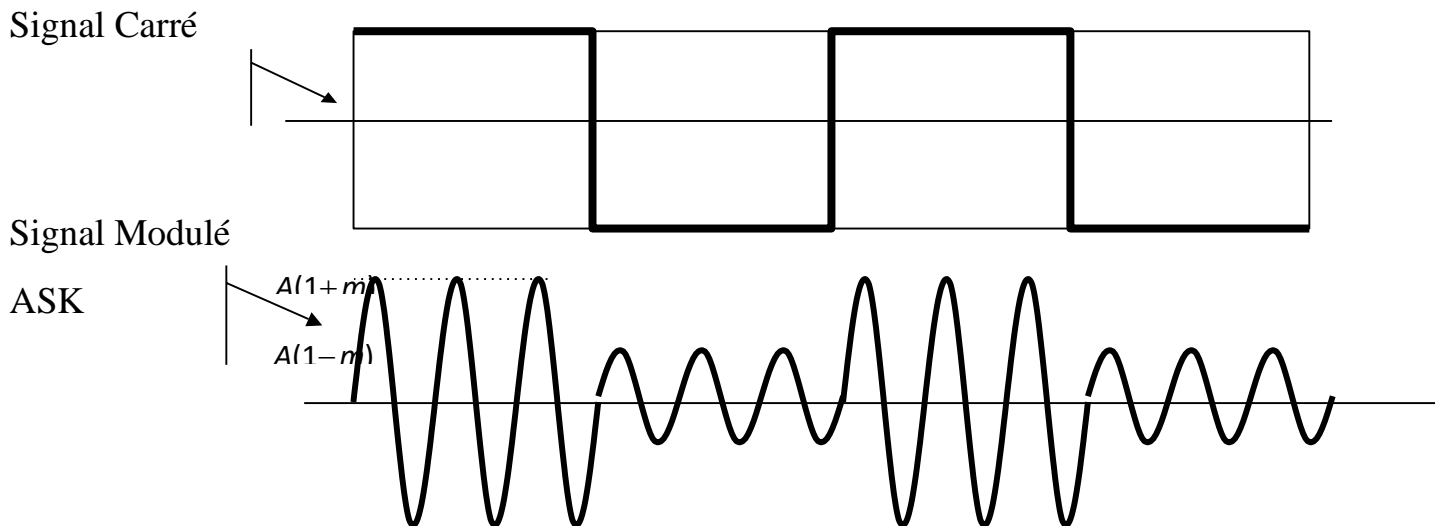
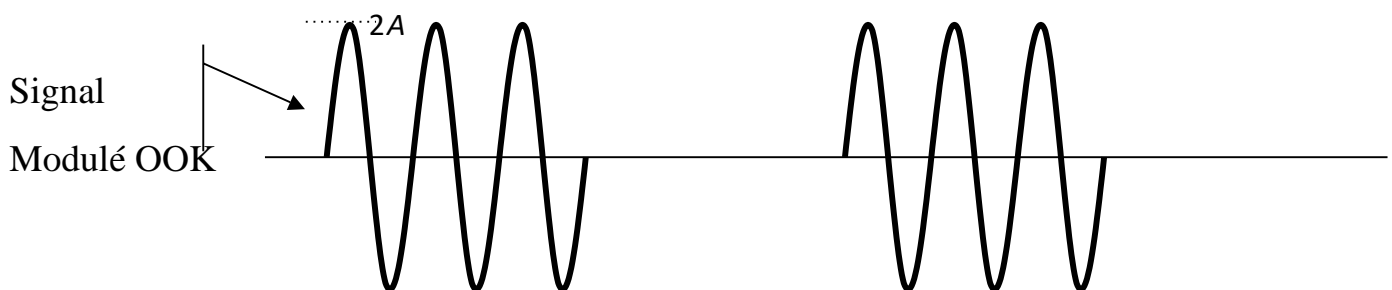


Figure 12: Exemple d'une modulation ASK

Cas particulier : Modulation OOK où $(M+2)$



En Modulation d'amplitude, le spectre du signal modulé est symétrique par rapport à la raie de la porteuse et les deux bandes latérales ont la même forme que le signal BF.

Voici quelques exemples d'allures de spectres ASK :

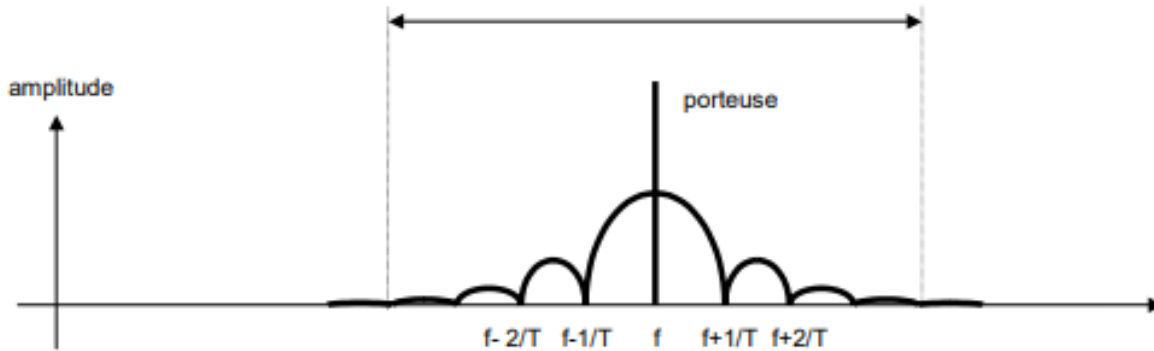
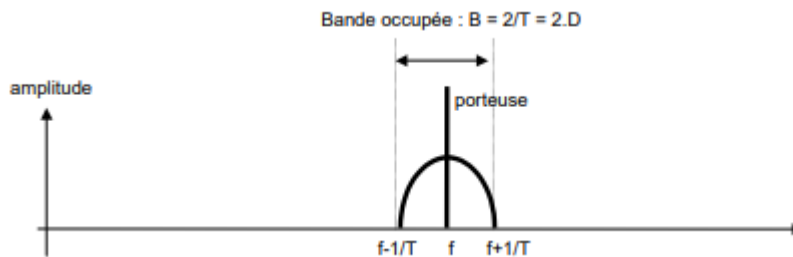


Figure 13: spectre d'un signal ASK modulé par un signal numérique non filtré

Remarque :

Lorsque le signal numérique n'est pas filtré, le signal modulé ASK occupe en théorie une bande infinie, ce qui est inacceptable dans la pratique. On est donc amené dans la pratique à limiter la bande du signal numérique par un filtre passe bas (simple ou Nyquist)



III.2 Modulation de Phase PM :

La modulation d'amplitude consiste à modifier la phase de la porteuse. Elle est très la plus utilisée car elle est simple à mettre en œuvre.

Soit un signal carré : $s(t) = \sum_{k=0}^{+\infty} a_k \text{rect}(t - kT_s)$

La porteuse : $p(t) = A \cdot \cos(2\pi f_0 t + \varphi)$

Le signal qui résulte de la modulation : $A \cdot \sum_{k=0}^{+\infty} a_k \text{rect}(t - kT_s) \cos(2\pi f_0 t + \varphi)$

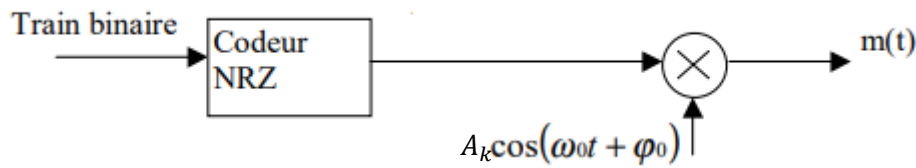


Figure 14 : Modulation de phase

Pour une modulation de phase numérique binaire BPSK on prend : $a_k = \{0,1\}$ et $\varphi = \{0, \pi\}$

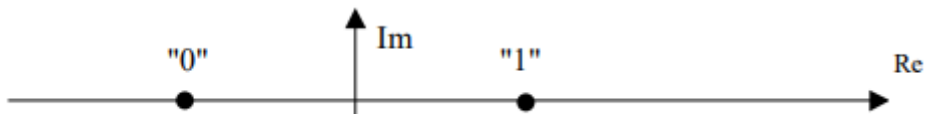


Figure 15: Le diagramme de constellation pour une modulation BPSK

Ce type de modulation donne une porteuse présentant des sauts de π à chaque changement de l'information binaire.

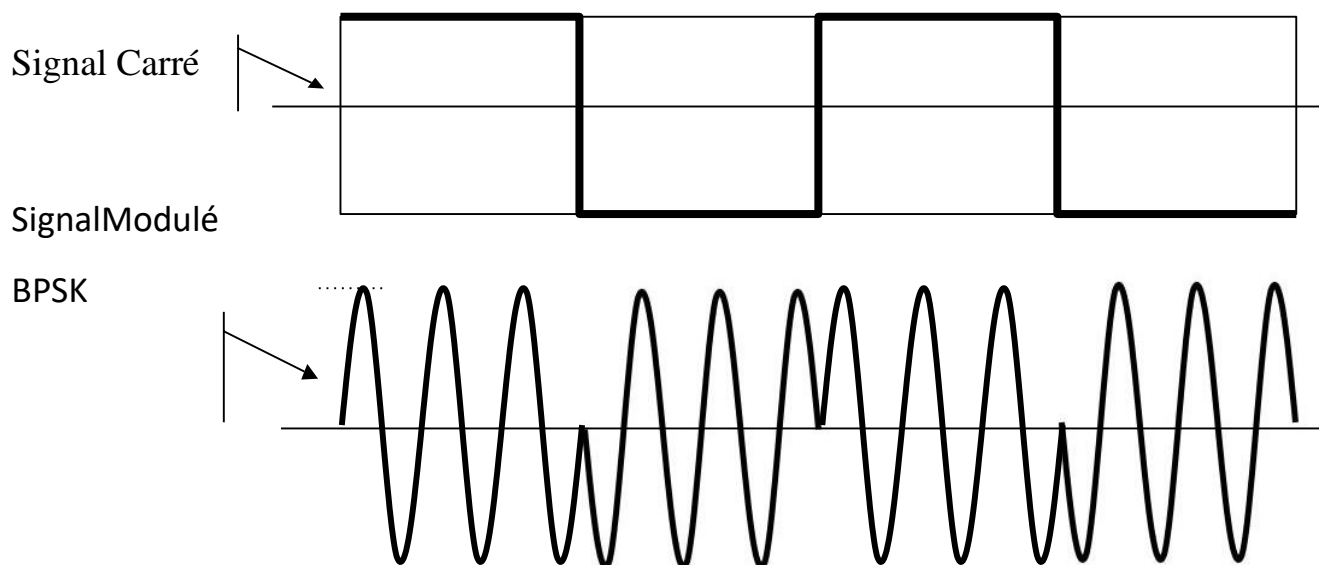
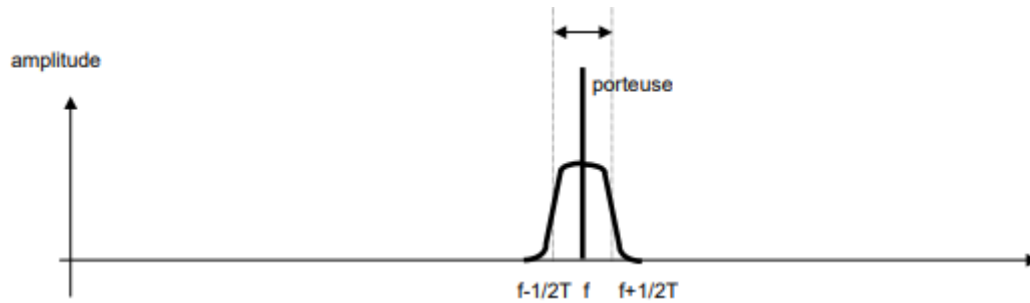


Figure 16: Exemple d'une modulation PSK

La porteuse modulée aura un spectre du même type qu'en ASK . Alors cette fois on filtre le signal par un filtre de Nyquist qui donne cette fois une largeur deux fois plus faible :



III.3 Modulation de Fréquence FM :

La modulation de fréquence est une modulation très largement utilisée, en particulier dans le domaine des télécommunications car il est plus robuste que les autres modulations qui permet de transmettre un message dans des condition difficiles. L'information est portée par une modification de la fréquence de la porteuse.

Mais la modulation FSK, comme d'ailleurs la modulation ASK, ne convient plus si on désire faire transiter des informations numériques avec un débit élevé sur un canal de largeur.

Pour la modulation FSK la porteuse est modulée en fréquence par le signal numérique, c'est-à-dire qu'elle saute d'une fréquence f_0 (pour le « 0 ») à une fréquence f_1 (pour le « 1 »).

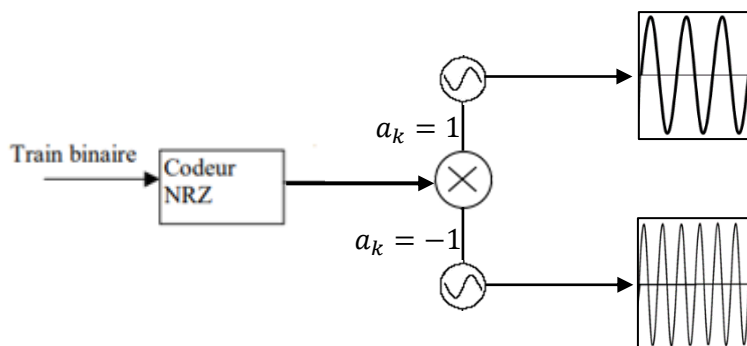


Figure 17: Modulation FSK

Signal
Carré



Signal Modulé

BPSK

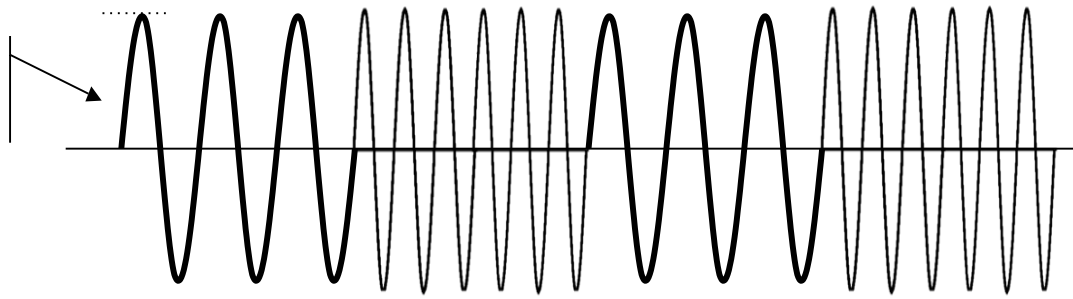


Figure 18: Exemple d'une modulation FSK

Le spectre de ce signal modulé est en $\text{sinc}(x)$ autour des deux fréquences f_0 et f_1 .

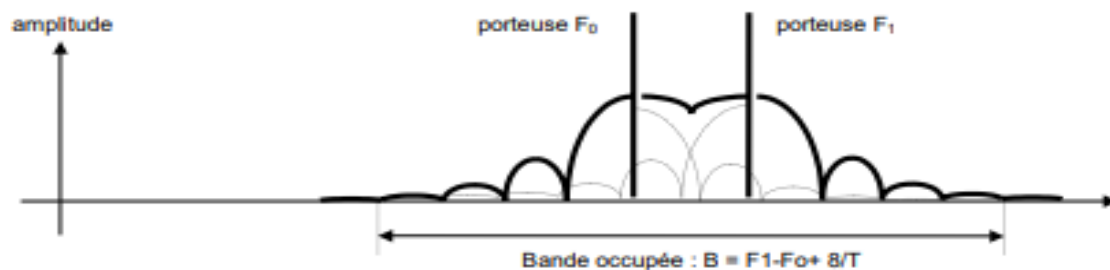


Figure 19: Spectre d'un signal FSK modulé par un signal numérique non filtré.

III.4 Démodulation du signal :

Pour la phase de modulation, presque tous les types de démodulation seront identique aux modulations. Alors, multiplier le signal reçu avec le même signal de la porteuse qui est utilisé au niveau de la modulation.

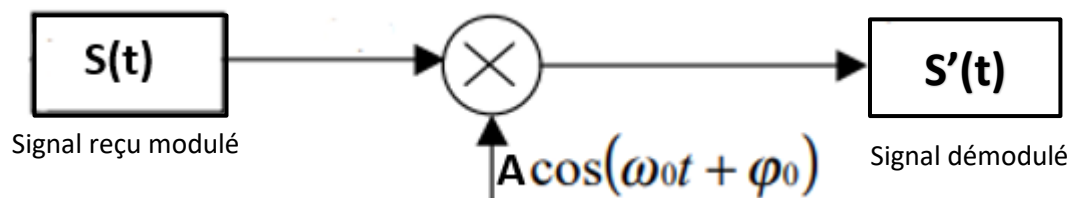


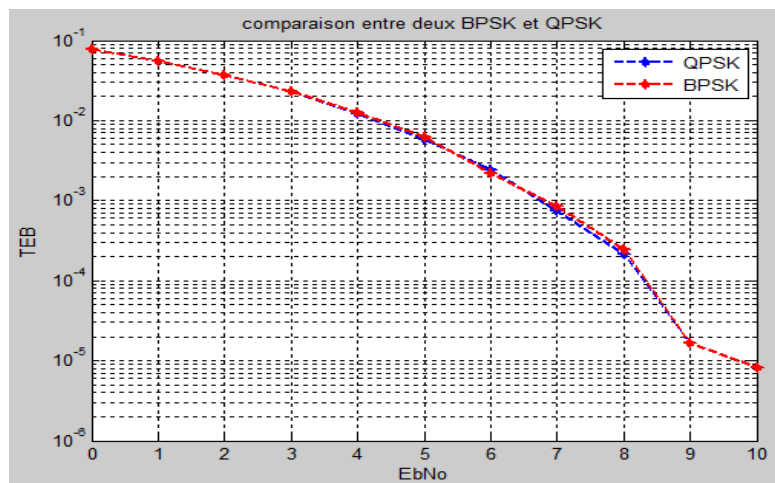
Figure 20: processus de démodulation d'un signal

Comparaison entre les différentes modulations :

Voici un tableau qui résume l'efficacité spectrale théorique maximale pour les différents types de modulation.

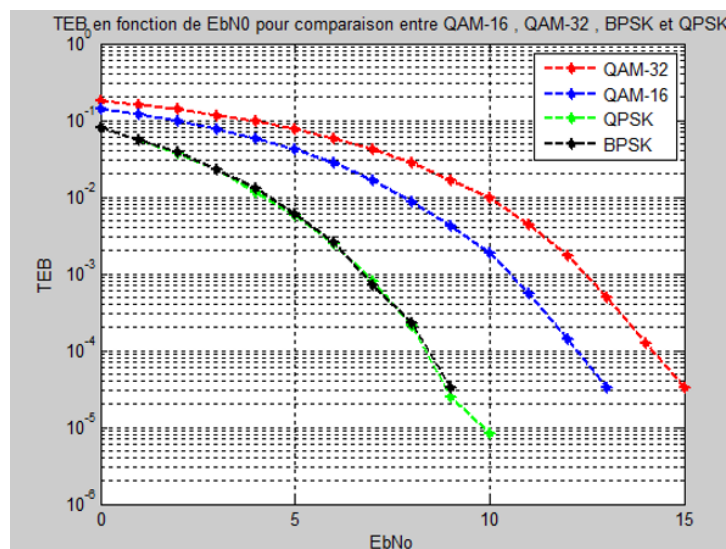
Modulation	Efficacité bits/s/Hz
MSK	1
BPSK	1
QPSK	2
16 QAM	4
32 QAM	5
64 QAM	6
256 QAM	8

- Comparaison entre la modulation BPSK et QPSK :



L'augmentation de M réduit la distance entre symboles adjacents sur la constellation et cela dégrade naturellement les performances.

- Comparaison entre les différentes modulations qu'on déjà traite :



Il est clair que dans ce cas, plus le niveau de bruit est important, le taux d'erreur Binaire et plus faible, plus le nuage de points (dans la constellation) sera étendu et plus la décision sera entachée d'erreur.

IV. Récupération du signal du message (symboles) d'origine :

Comme on peut bien remarquer que la structure du récepteur est presque symétrique de celle de l'émetteur.

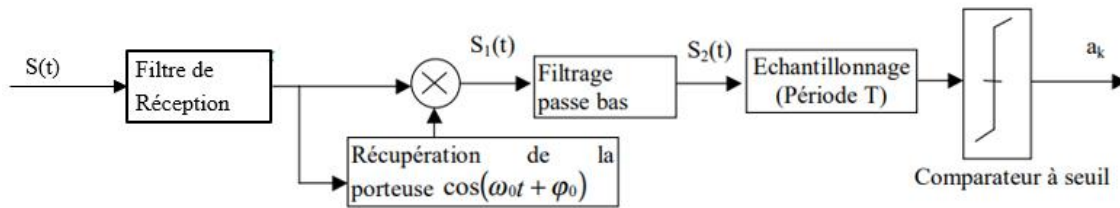


Figure 21: Structure du récepteur

On distingue les étages suivants :

- La réception du signal.
- Filtrage du signal reçu à l'aide d'un filtre de réception qui construit d'un filtre blanchissant en série avec le filtre demi-Nyquist.
- La démodulation en quadrature pour la récupération des signaux $i(t)$ et $q(t)$.
- La digitalisation des signaux $i(t)$ et $q(t)$ (échantillonnage).
- La restitution du train binaires.

La plus grande difficulté dans le récepteur est la récupération de fréquence d'horloge permettant une récupération correcte des données. Cette difficulté est minimisée par le cosinus surélevé (qui est connu sous le nom le simple nom de filtre de Nyquist).

Après la récupération des symboles il est nécessaire vérifier la performance du canal qui reflète sa qualité.

Alors, pour cela on utilise deux paramètres de référence :

- Le calcul du TEB (taux d'erreurs binaire)
- Diagramme de l'Œil

Taux d'erreur binaire :

Soit l'échantillon $y(t_0)$ prélevé en sortie du dernier stage dans la partie de réception.

Conditionnellement à une valeur de l'élément binaire $a_0(0 \text{ ou } 1)$:

L'expression de la probabilité d'erreur P_e peut s'écrire :

$$P_e = p_0 \int_S^{+\infty} P_{Y/\alpha_0=0}(y) dy + p_1 \int_{-\infty}^S P_{Y/\alpha_0=1}(y) dy$$

Où $P_{Y/\alpha_0=1}(y)$ et $P_{Y/\alpha_0=0}(y)$ représentent respectivement les densités de

probabilité conditionnelles de la variable aléatoire gaussienne Y lorsque l'élément binaire a_0 est égal à 1 ou à 0. Tel que :

$$P_{Y/\alpha=1}(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - r_1(t_0))^2}{2\sigma^2}\right)$$

$$P_{Y/\alpha=0}(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - r_0(t_0))^2}{2\sigma^2}\right)$$

Les deux densités de probabilité conditionnelles sont représentées ci-dessous :

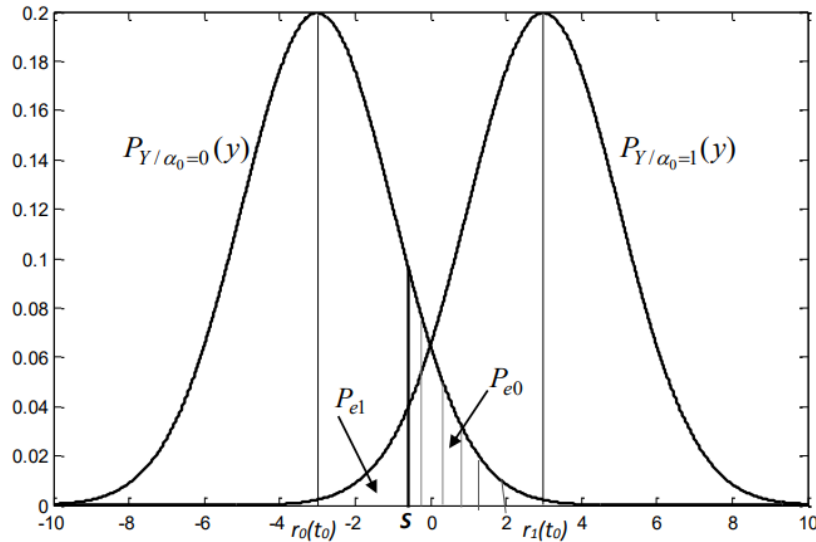


Figure 22: Représentation des densités de probabilité conditionnelles

La probabilité d'erreur Pe a finalement pour expression :

$$P_e = \frac{1}{\sqrt{2\pi\sigma^2}} \left[p_0 \int_S^{+\infty} \exp\left(-\frac{(y - r_0(t_0))^2}{2\sigma^2}\right) dy + p_1 \int_{-\infty}^S \exp\left(-\frac{(y - r_1(t_0))^2}{2\sigma^2}\right) dy \right]$$

Pour choisir un seuil optimal il faut minimiser la probabilité d'erreur, par conséquent on calcule : $\frac{dP_e}{dS} = 0$

On trouve le seuil optimal : $S_{opt} = \frac{r(t_0) + r(t_1)}{2} + \sigma^2 \ln \frac{p_0}{p_1}$

Lorsque $p_0 = p_1 = 1/2$, on trouve : $S_{opt} = \frac{r(t_0) + r(t_1)}{2}$

Et la probabilité d'erreur devient : $P_e = \frac{1}{2} \operatorname{erfc}\left(\frac{r(t_1) - r(t_0)}{2\sqrt{2}\sigma}\right)$

Où $r_m(t)$ est la repense du filtre de réception au signal reçu

$erfc(x)$ est la fonction d'erreur complémentaire définie par:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} \exp(-u^2) du$$

Diagramme de l'Œil

C'est un diagramme représenté dans le domaine temporel qui nous permet de juger visuellement :

- D'estimer l'interférence entre les symboles et donc d'évaluer la marge de bruit ;
- De régler l'égaliseur pour minimiser l'interférence entre les symboles ;
- D'ajuster la phase de l'horloge récupérée pour échantillonner aux instants idéaux

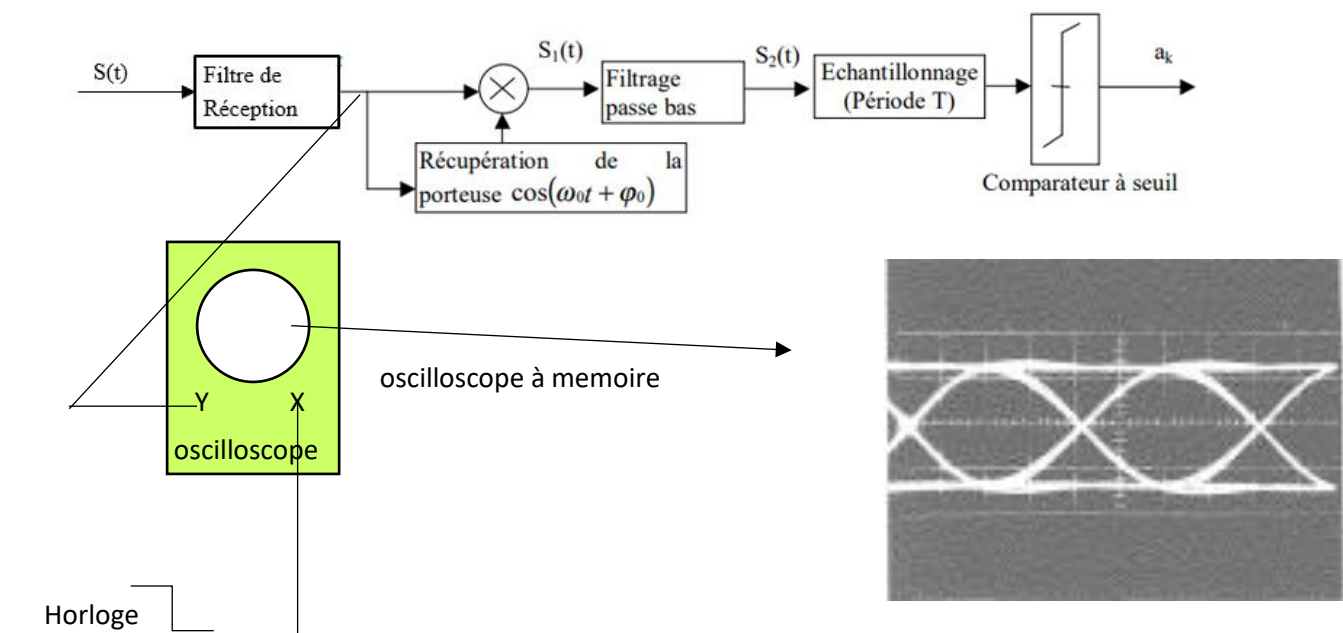
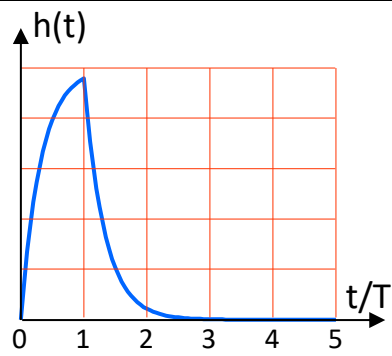


Figure 23: Extraction du diagramme d'œil à l'aide d'un oscilloscope

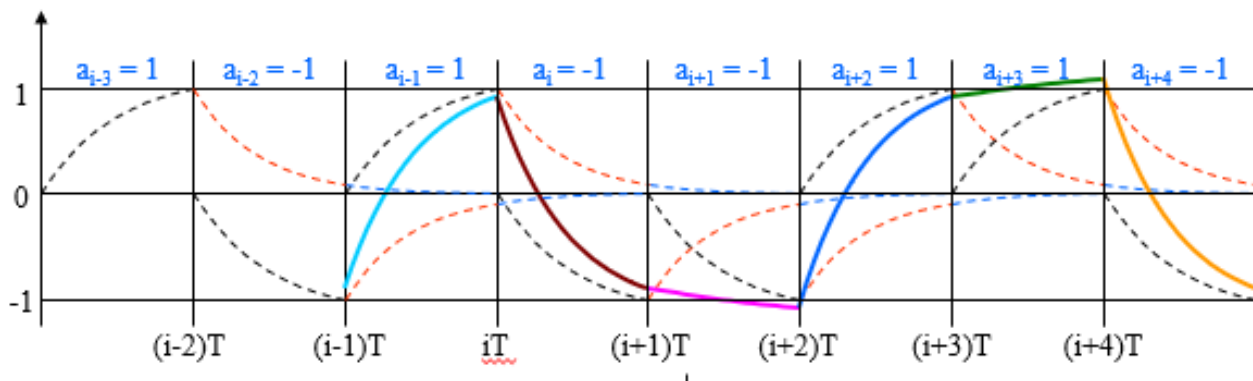
Construction du diagramme d'œil :

Soit la $h(t)$ réponse impulsionnelle pour chaque symbole :

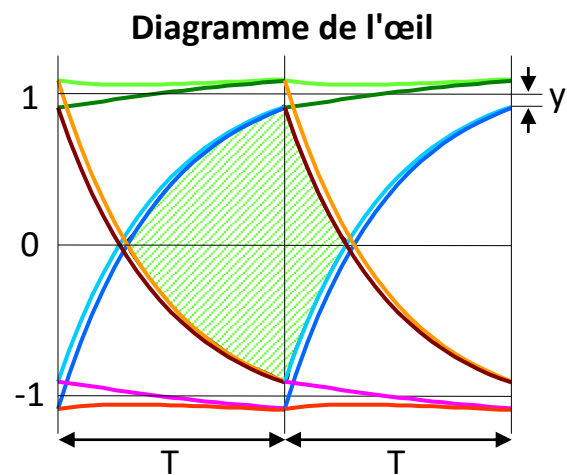


A l'aide de l'oscilloscope à mémoire on mémorise le signal reçu pour chaque période d'une durée de symbole, on aura :

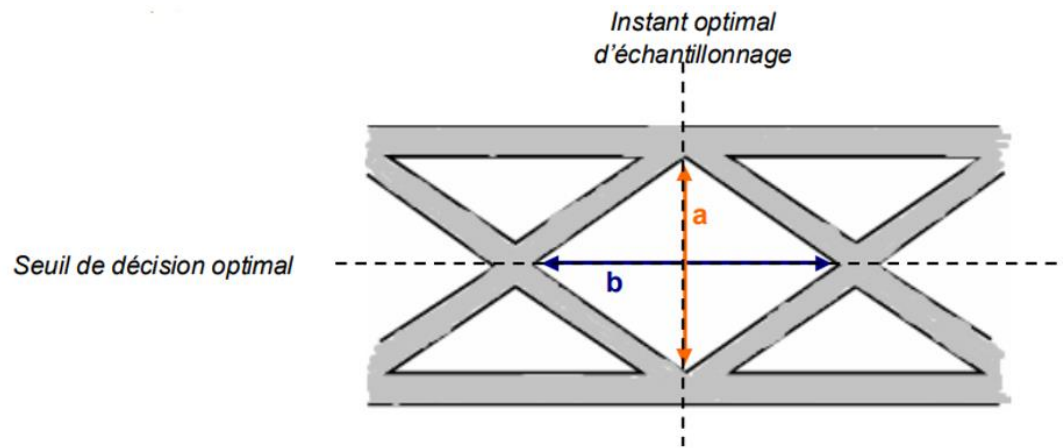
Pour un signal ci-dessous :



On aura un diagramme d'œil comme suivant :



Propriétés du diagramme d'œil :



a : Ouverture verticale (Immunité au bruit)

b : Ouverture horizontale (Immunité au déphasage par rapport à l'horloge)

Chapitre 3 :

Simulation des différentes composantes de la chaîne de transmission sous MATLAB

Codeur en ligne:

Fonction codeur NRZ : Cette fonction nous permet de générer un code NRZ à partir d'une séquence de bits binaire.

```
function [t,x] = nrz(bits, e)

    T = length(bits)/e; % longueur de tout la sequence de bits
                        % e =Nbr de bits par seconds
                        % bits = sequence de bits

    n = 1000;
    N = n*length(bits);
    dt = T/N;
    t = 0:dt:T;
    x = zeros(1,length(t)); % signal de sortie

    for i = 0:length(bits)-1
        if bits(i+1) == 1
            x(i*n+1:(i+1)*n) = 1;
        else
            x(i*n+1:(i+1)*n) = -1;
        end
    end
end
```

Fonction codeur RZ : Cette fonction nous permet de générer un code RZ à partir d'une séquence de bits binaire.

```
function [t,x] = rz(bits, e)

    T = length(bits)/e; % longueur de tout la sequence de bits
                        % e =Nbr de bits par seconds
                        % bits = sequence de bits

    n = 1000;
    N = n*length(bits);
    dt = T/N;
    t = 0:dt:T;
    x = zeros(1,length(t)); % signal de sortie

    for i = 0:length(bits)-1
        if bits(i+1) == 1
            x(i*n+1:(i+0.5)*n) = 1;

        else
            x(i*n+1:(i+1)*n) = 0;
        end
    end
end
```


Fonction codeur Manchester : Cette fonction nous permet de générer un code Manchester à partir d'une séquence de bits binaire.

```
function [t,x] = manchester(bits, e)

T = length(bits)/e; % logeur de tout la sequence de bits
                    % e =Nbr de bits par seconds
                    % bits = sequence de bits

n = 1000;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t)); % signal de sortie

for i = 0:length(bits)-1
    if bits(i+1) == 1
        x(i*n+1:(i+0.5)*n) = 1;
        x((i+0.5)*n+1:(i+1)*n) = -1;
    else
        x(i*n+1:(i+0.5)*n) = -1;
        x((i+0.5)*n+1:(i+1)*n) = 1;
    end
end
```

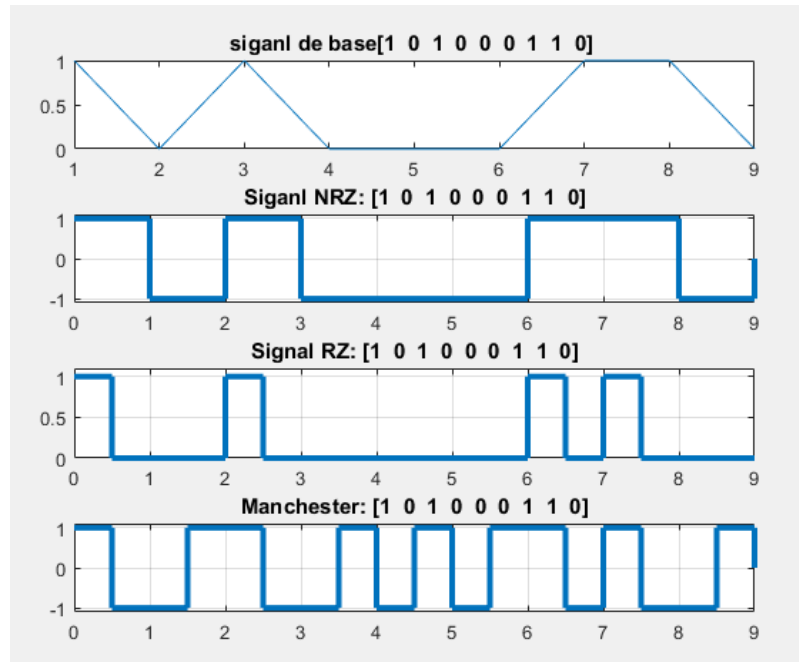
code démonstration :

```
clear all
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Siganl de base %%%%%%%%%%%
bits = [1 0 1 0 0 0 1 1 0];
e = 1; % Nbr de bits par seconds
figure;
subplot(4,1,1);
plot(bits);
title(['siganl de base[' num2str(bits) ']' ])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Signal coder %%%%%%%%%%%
Tb=0.1;
[t,s] = nrz(bits,e); %% Codage NRZ
subplot(4,1,2);
plot(t,s,'LineWidth',3);
axis([0 t(end) -1.1 1.1])
grid on;
title(['Siganl NRZ: [' num2str(bits) ']' ]);

[t,s] = rz(bits,e); %% Codage RZ
subplot(4,1,2);
plot(t,s,'LineWidth',3);
axis([0 t(end) 0 1.1])
grid on;
title(['Signal RZ: [' num2str(bits) ']' ]);

subplot(4,1,3);
[t,s] = manchester(bits,e); %% Codage RZ
plot(t,s,'LineWidth',3);
axis([0 t(end) -1.1 1.1])
grid on;
title(['Manchester: [' num2str(bits) ']' ]);
```

Résultat :



Filtre d'émission :

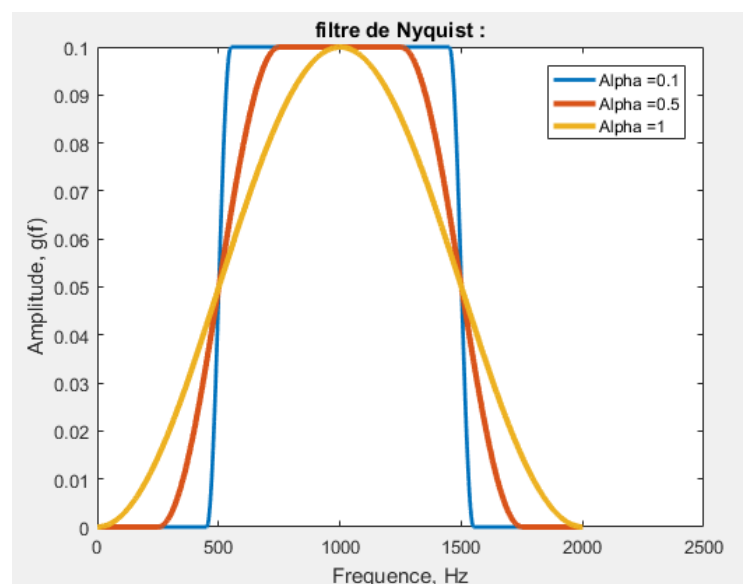
Filtre de Nyquist : il est construit à partir de la fonction cosinus surélevé $\text{cosinus_surelve}(a)$ est le coefficient de retombé.

```
function y=cosinus_surelve(a)
%a est le coefficient de retombé
Ts=0.1; %durée d'un symbole ici on a Tb=Ts
f= -1/Ts:0.01:1/Ts; %génération de vecteur de fréquence

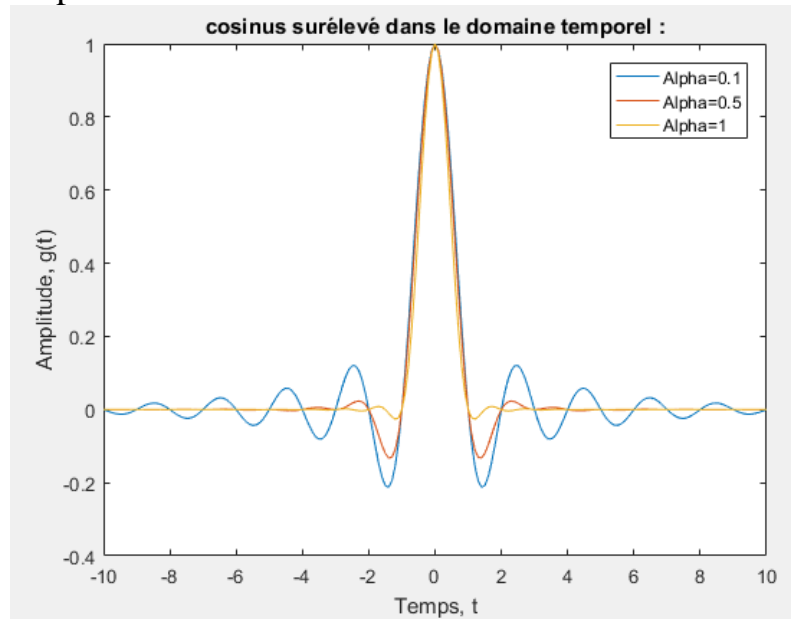
y=zeros(1,length(f));
for i=1:length(f)
    if(( abs(f(i))<= ((1+a)/(2*Ts))) && (abs(f(i))>=((1-a)/(2*Ts))) )
        y(i)=(Ts/2)*(1-sin((pi*(2*abs(f(i))*Ts-1)/(2*a))));
    elseif (( abs(f(i))>0 ) && ( abs(f(i))< ((1-a)/(2*Ts))) )
        y(i)=Ts;
    else y(i)=0;
    end
end
end
```

Résultat :

```
Tb=1;
f= -1/Tb:0.01:1/Tb;
g=cosinus_surelve(0.1);
g1=cosinus_surelve(0.5);
g2=cosinus_surelve(1);
figure;
plot(g,'LineWidth',2);
hold on
plot(g1,'LineWidth',3);
plot(g2,'LineWidth',3);
legend('Alpha =0.1','Alpha =0.5','Alpha =1')
title('filtre de Nyquist :');
xlabel('Frequence, Hz')
ylabel('Amplitude, g(f)')
```

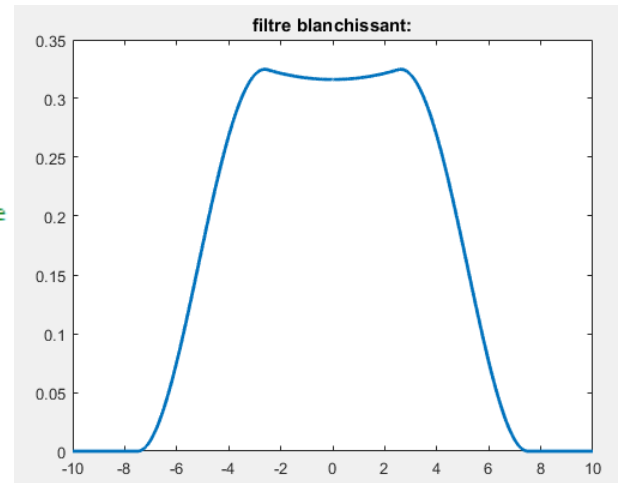


Dans le domaine temporel :



Filtre de Blanchissant : on trace la fonction du filtre blanchissant à l'aide de l'inverse de la fonction Sinc à l'intérieur de la bande passante du canal et le cosinus surélevé aux extrémités.

```
Tb=0.1;
f= -1/Tb:0.01:1/Tb;
g=cosinus_surelve(0.5);
%l'inverse de fonction sinc
%(inverse de la de la fonction de mise en forme de
    %filtre du codeur en ligne)
y=(pi*f)./sin(pi*f*Ts);
G=y.*g; %filtre de blanchissant
figure;
plot(f,X, 'LineWidth',2);
title('filtre blanchissant:');
```



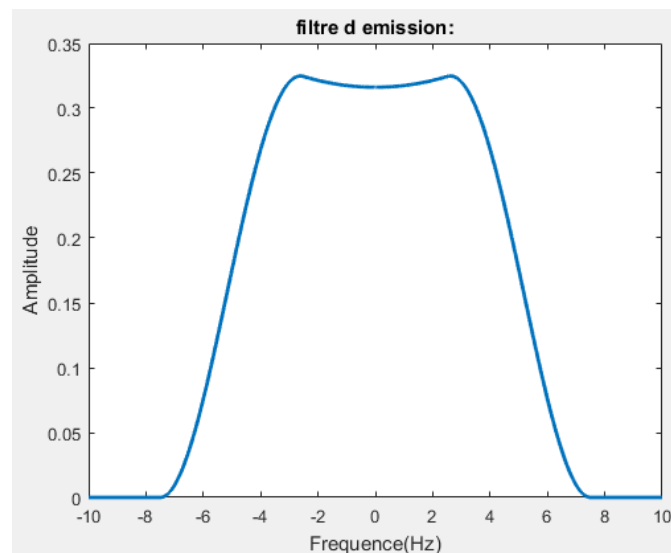
Résultat de combinaison de la fonction du filtre demi-Nyquite et le filtre blanchissant pour construire le filtre blanchissant :

```

clear all
clc
Tb=0.1;
f= -1/Tb:0.01:1/Tb; %génération de vecteur de fréquence
%cosinus surélevé :
Cos_slv=cosinus_sureleve(0.5);
%l'inverse de fonction sinc : (fonction de mise en forme de filtre du
%codeur en ligne)
Sin_inv=(pi*f)./sin(pi*f*Tb);
% filtre d'émission
demi_Nyq=sqrt(cosinus_sureleve(1));
%filtre blanchissant :
F_blan=Sin_inv.*Cos_slv; %filtre de blanchissant
%Filtre d'émission :
F_emis=F_blan.*demi_Nyq; %le trace à la sortie de filtre d'émission
figure
plot(f,F_emis,'LineWidth',2)
title('filtre d emission:');
xlabel('Frequence(Hz) ');
ylabel('Amplitude');

```

Résultat :



Modulation :

Modulation PSK : ex : $M=2$ (BPSK)

```

clear all
clc
bits = [1 0 1 0 0 0 1 1 0]; %sequence binaire
e = 1; % Nbr de bits par seconds
Tb=0.1;
fc=0.8; %frequence de la porteuse
V=1;
[t,s] = nrz(bits,e); %gereneration d'un code NRZ
A=1;
x=A.*cos(2*pi*fc*t); %la porteuse
BPSK_signal=x.*s; % traçage du signal modulé

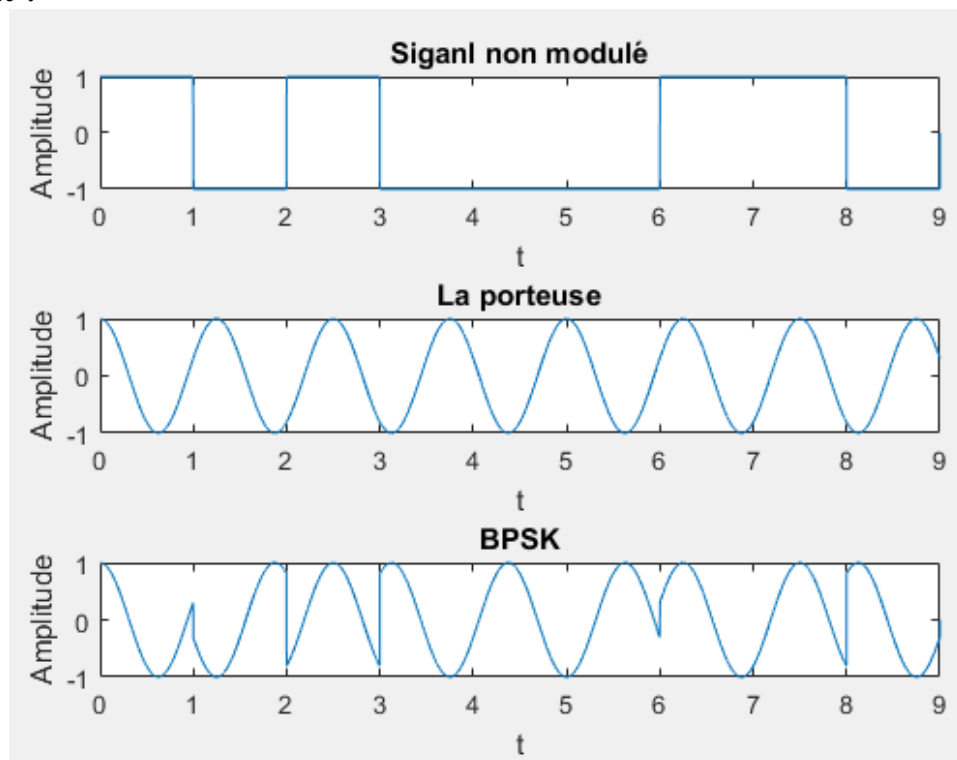
figure;
subplot(3,1,1);
plot(t,s);
xlabel('t');
ylabel('Amplitude');
title('Signal non modulé')

subplot(3,1,2);
plot(t,x);xlabel('t');
ylabel('Amplitude');
title('La porteuse')

subplot(3,1,3);
plot(t,BPSK_signal);
xlabel('t');
ylabel('Amplitude');
title('BPSK');

```

Résultat :



Modulation FSK : ex : $M=2$ (B-FSK)

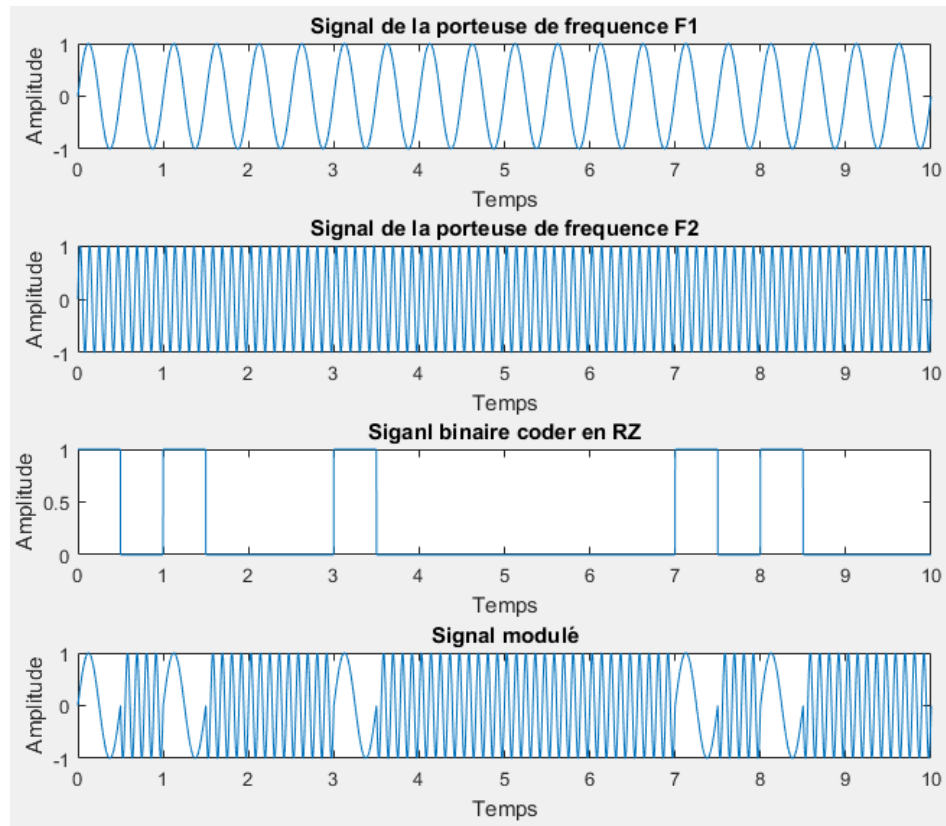
```
clear all
fc1=2;
fc2=9;
bits=[1 1 0 1 0 0 0 1 1 0] ;
e = 1;
[t,x] = rz(bits,e);

c1=sin(2*pi*fc1*t);% la porteur de frequence F1
c2=sin(2*pi*fc2*t);% la porteur de frequence F2

for i=0:length(x)-1 %Creation du signal modulé
    if x(i+1)==0
        mm(i+1)=c2(i+1);
    else
        mm(i+1)=c1(i+1);
    end
end

subplot(4,1,1);
plot(t,c1)
xlabel('Temps')
ylabel('Amplitude')
title('Signal de la porteuse de frequence F1')
subplot(4,1,2)
plot(t,c2)
xlabel('Temps')
ylabel('Amplitude')
title('Signal de la porteuse de frequence F2')
subplot(4,1,3)
plot(t,x)
xlabel('Temps')
ylabel('Amplitude')
title('Siganl binaire coder en RZ')
subplot(4,1,4)
plot(t,mm)
xlabel('Temps')
ylabel('Amplitude')
title('Signal modulé')
```

Résultat :



Modulation ASK : ex : $M=2$ (B-ASK)

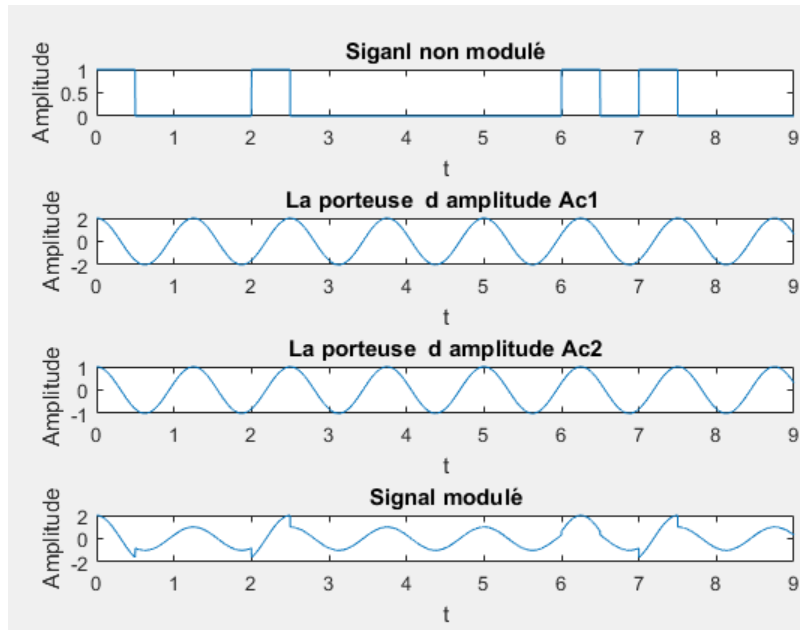
```
clear all
clc
bits = [1 0 1 0 0 0 1 1 0]; %sequence binaire
e = 1; % Nbr de bits par seconds
Tb=0.1;
fc=0.8; %fréquence de la porteuse
V=1;
[t,x] = rz(bits,e); %gereneration d'un code RZ
Ac1=2; % Amplitude pour le bit 1
Ac2=1; % Amplitude pour le bit 0
c1=Ac1*cos(2*pi*fc*t);% la porteur d'amplitude Ac1
c2=Ac2*cos(2*pi*fc*t);% la porteur d'amplitude Ac2
for i=0:length(x)-1 %Creation du signal modulé
    if x(i+1)==0
        mm(i+1)=c2(i+1);
    else
        mm(i+1)=c1(i+1);
    end
end
figure;
subplot(4,1,1);
plot(t,x);
xlabel('t');
ylabel('Amplitude');
title('Signal non modulé');
subplot(4,1,2);
plot(t,c1);
xlabel('t');
ylabel('Amplitude');
title('La porteuse d amplitude Ac1 ');
```

```

subplot(4,1,3);
plot(t,c2);
xlabel('t');
ylabel('Amplitude ');
title('La porteuse d amplitude Ac2')|
subplot(4,1,4);
plot(t,mm);
xlabel('t');
ylabel('Amplitude');
title('Signal modulé');

```

Résultat :



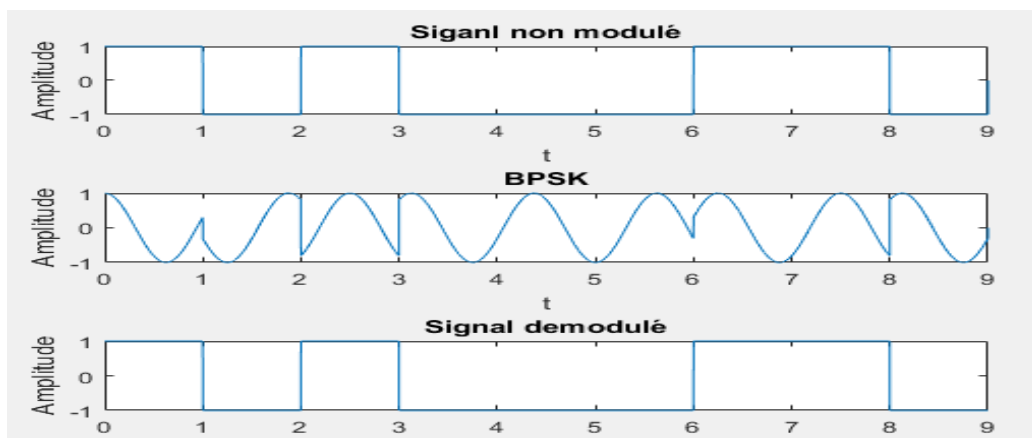
Démodulation :

Démodulation PSK :

```

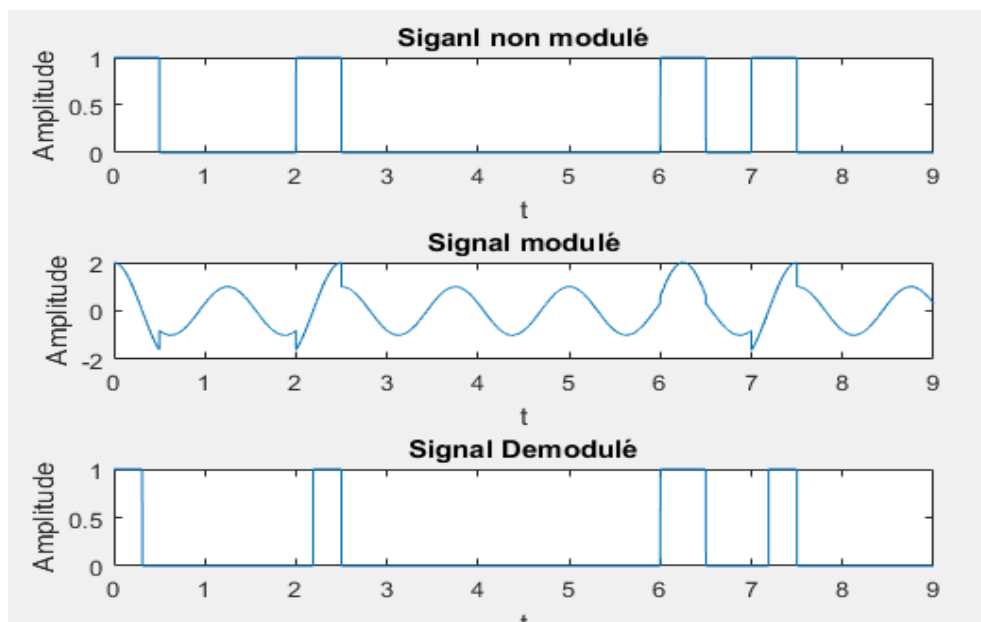
BPSK_signal_demo = BPSK_signal.*x ;
for i=0:length(BPSK_signal_demo)-1 %Reconstruction du signal
    if BPSK_signal_demo(i+1)<=0 %Comparaison au seuil (zeros) et regeneration
        mm(i+1)=-1; %du signal
    else
        mm(i+1)=1;
    end
end
end

```



Demodulation ASK:

```
%*****Demodulation : *****  
BASK_signal_demo = mm.*x ;  
for i=0:length(BASK_signal_demo)-1 %Reconstruction du siganl  
    if BASK_signal_demo(i+1)<=0 %Comparaison au seuil (zeros) et regeneration  
        m(i+1)=0; %du siganl  
    else  
        m(i+1)=1;  
    end  
end
```



Chapitre 4 :

Simulation et mise en œuvre d'une chaîne de transmission sous SIMULINK

Introduction :

Après d'avoir présenté, théoriquement ainsi que pratiquement sur Matlab, les différentes composantes qui construisent une chaîne de transmission. Dans cette partie on va essayer de combiner et rassembler les différentes pièces qui construisent le puzzle de la chaîne de transmission.

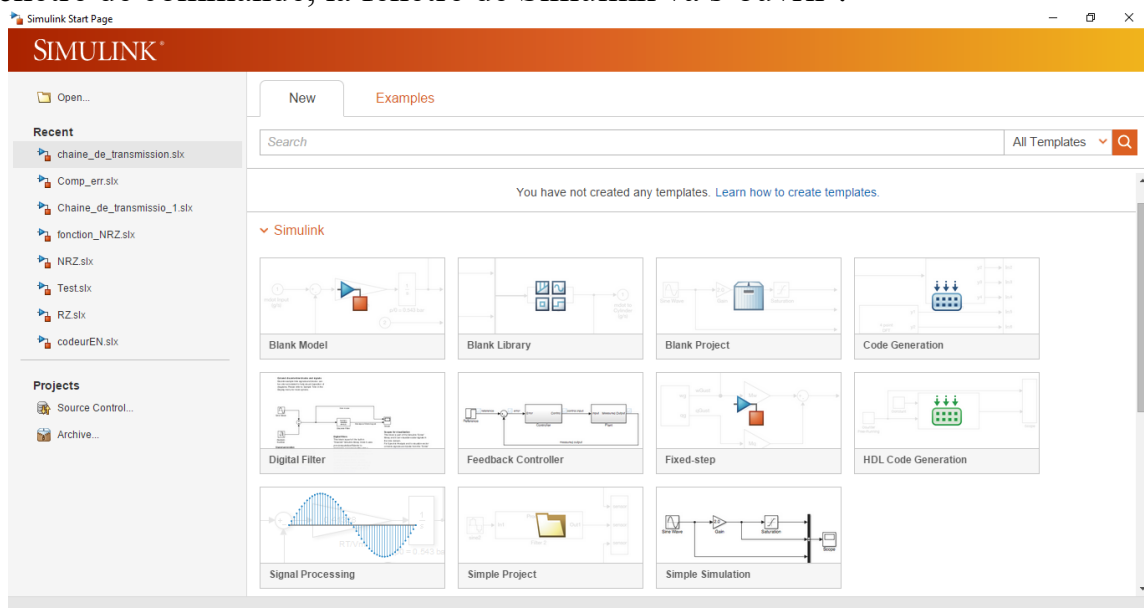
Premièrement, on va essayer de présenter notre espace de travail, qui est SIMULINK, qui va nous aider à établir les connections entre les blocs par un simple paramétrage.

En second temps, on va essayer de présenter le paramétrage des différents blocs ainsi que l'enchaînement.

Présentation de l'espace de travail « SIMULINK » :

Simulink est l'extension graphique de MATLAB permettant de représenter les fonctions mathématiques et les systèmes sous forme de diagramme en blocs, et de simuler le fonctionnement de ces systèmes.

Pour démarrer SIMULINK, Il suffit de taper une simple commande « simulink » dans la fenêtre de commande, la fenêtre de Simulink va s'ouvrir :



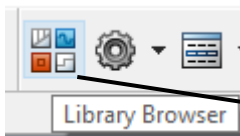
CONSTRUCTION D'UN DIAGRAMME SIMULINK

Pour commencer, dans le menu File, on choisit New - Model. Une fenêtre de travail Untitled s'ouvrira.

Ouvrir les collections de blocs en cliquant dessus (double). Faire glisser dans la fenêtre de travail les blocs dont on a besoin pour construire le diagramme. Faire des liaisons entre les blocs à l'aide de la souris.

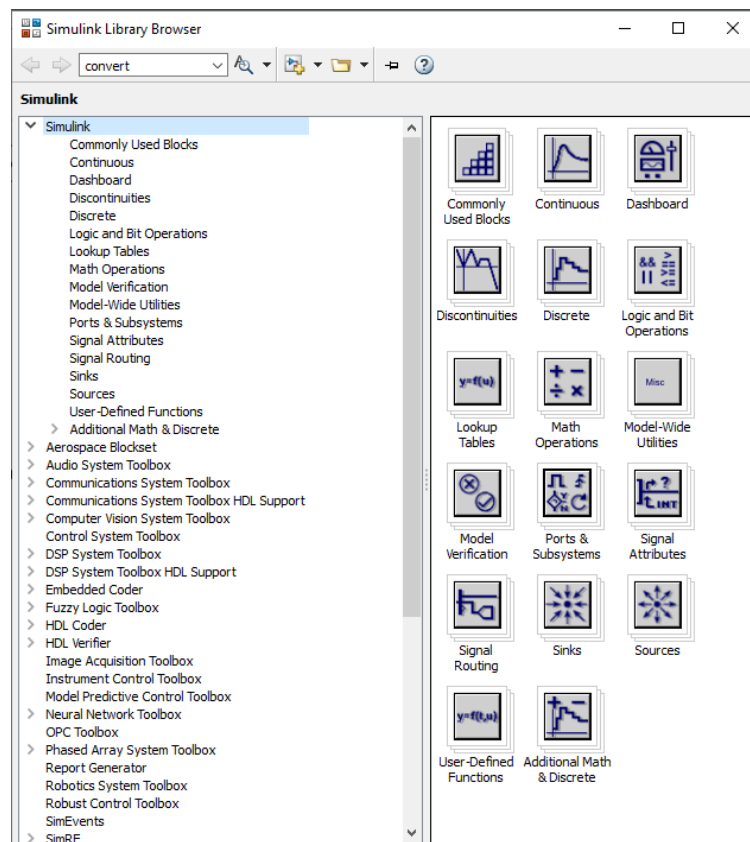
Lorsqu'on clique (double) sur un bloc, une fenêtre de dialogue s'ouvrira. On peut alors changer les paramètres de ce bloc. Une fois terminé, on ferme la fenêtre de dialogue.

Une fois le diagramme terminé, on peut l'enregistrer dans un fichier : dans le menu File, choisir Save As et donner un nom (*.mdl) au fichier.



Cette fenêtre contient des collections de blocs que l'on peut ouvrir en cliquant (double) dessus :

- Sources (Sources de signaux)
- Discrete (Blocs discrets)
- Linear (Blocs linéaires)
- Nonlinear (Blocs non linéaires) Connections (Entrée/sortie, multiplexeur/démultiplexeur, etc.)
- Demos (Démos)
- Blocksets & Toolboxes (Blocksets et toolboxes)
- Ect...



Après de faire une simple présentation de l'interface de simulink, on passe maintenant au choix des blocs et la formation des connexions entre les blocs.

Simulation d'une chaine de transmission :

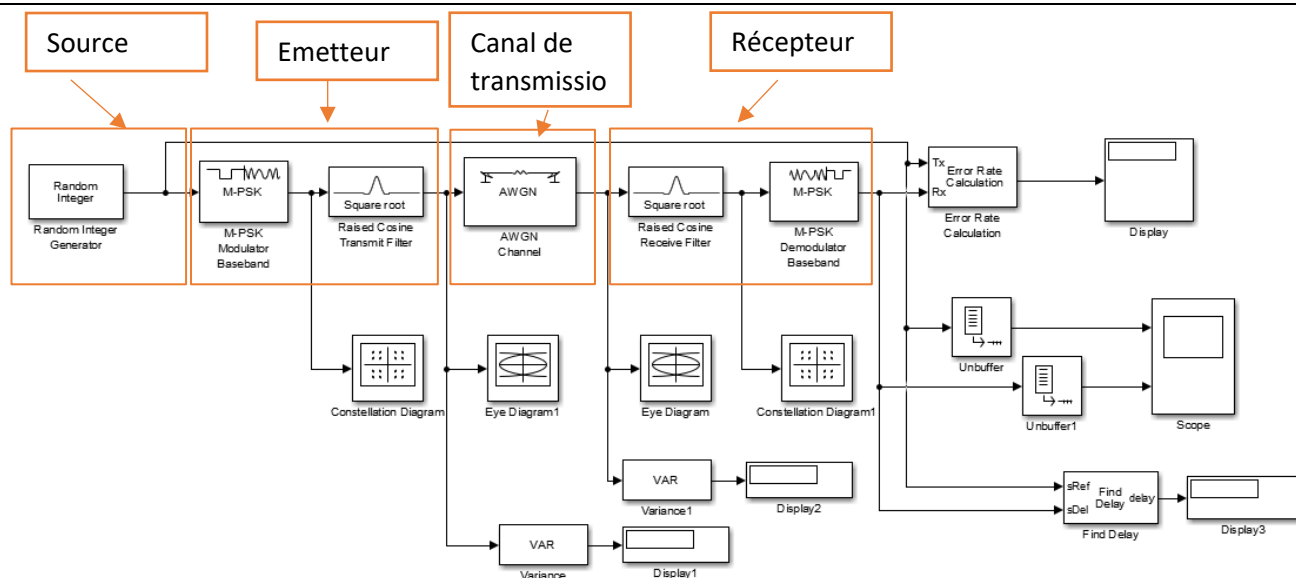
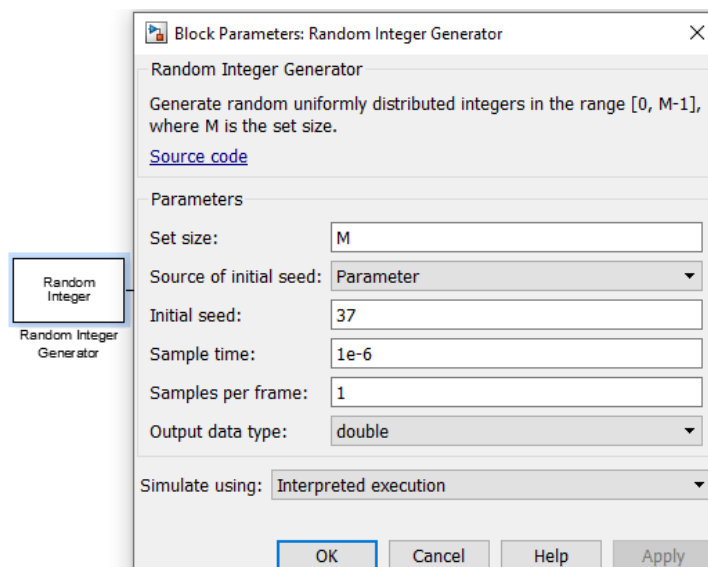


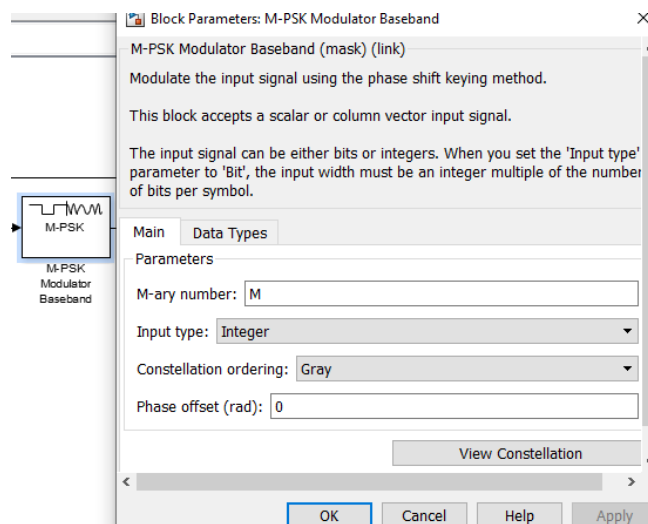
Figure 24 : simulation d'une chaine de transmission sur simulink

Paramétrage des différents blocs :

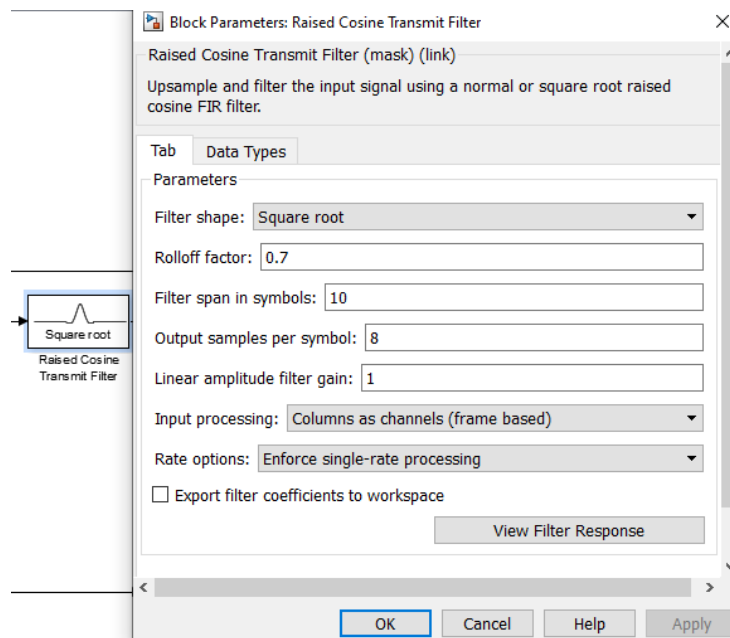
Source : générateur des entiers aléatoire (avec M est ça multiplicité)



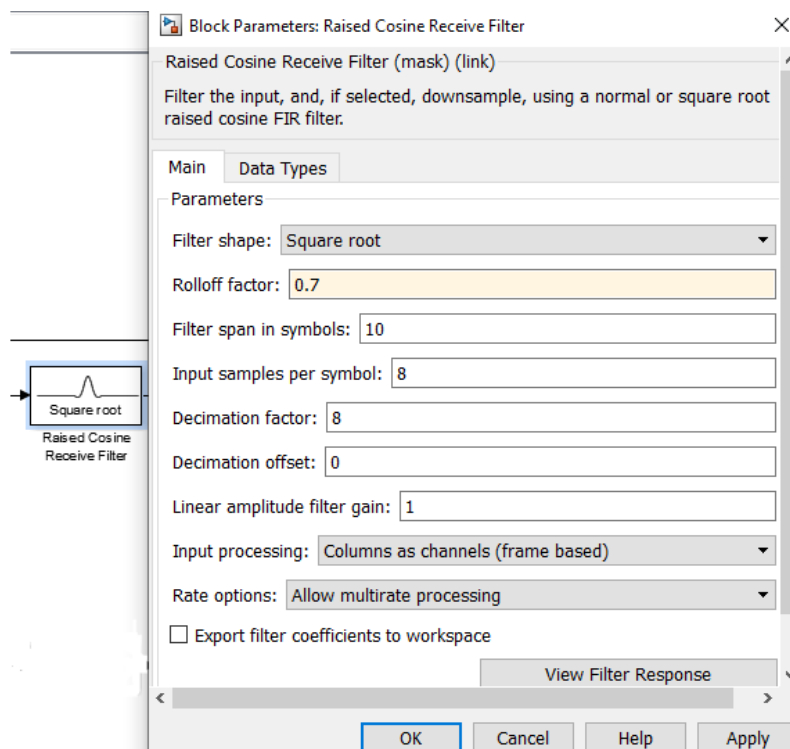
Modulateur : C'est un modulateur de phase M-PSK



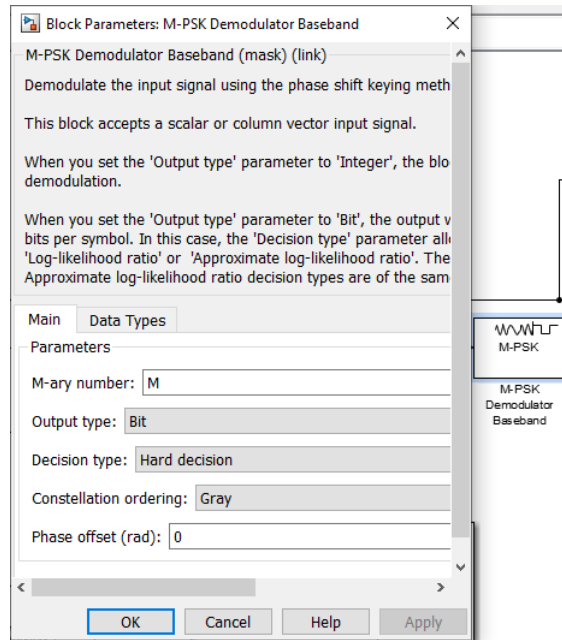
Filtre demi-Nyquist d'émission : (de coefficient de retombé $\alpha=0.7$)



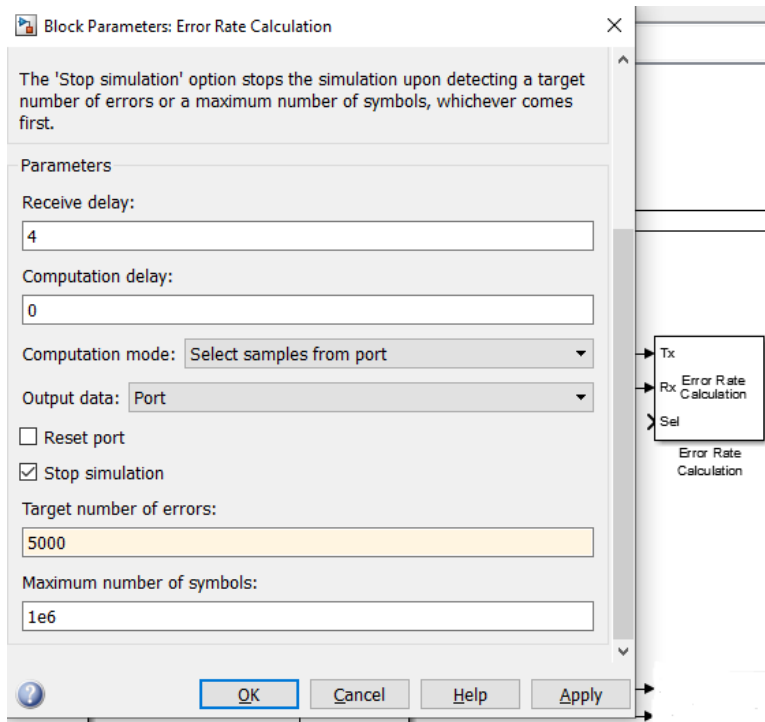
Filtre demi-Nyquist de réception : (de coefficient de retombé $\alpha=0.7$)



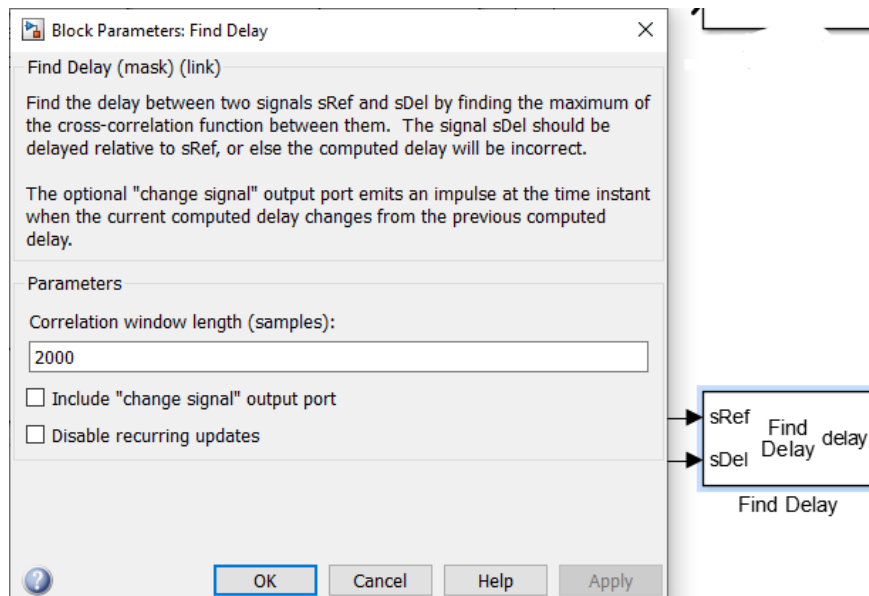
Démodulateur : démodulateur de phase M-PSK



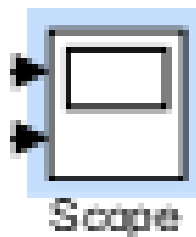
Error rate calculation : Nous permet d'estimer et de vérifier la qualité de notre canal en calculant le TEB



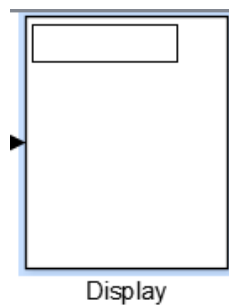
Calculateur du délai entre le signal de sortie et le signal de base qui va nous permettre de le corriger avant l'affichage.



Scope : Affichage des graphes



Display : Affichage numérique



Var : calcule la variance d'un signal en un point de la chaine



Canal de transmission : AWGN

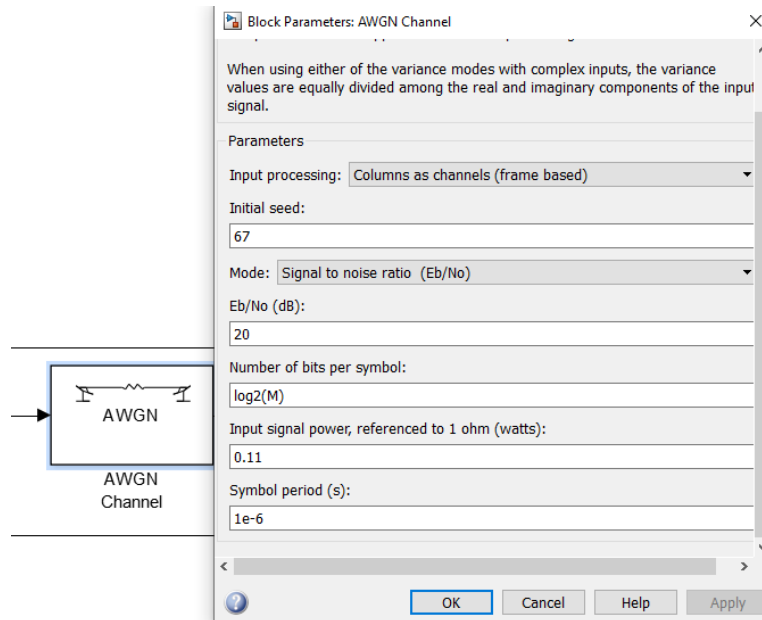


Diagramme de constellation :

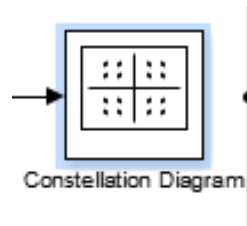
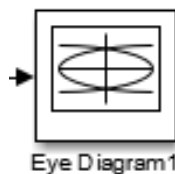
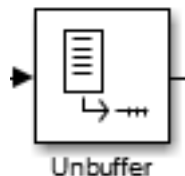


Diagramme d'œil :



Unbuffer ; Transformer une séquence parallèle de data en une séquence en série.



Résultat d'une simulation démonstrative :

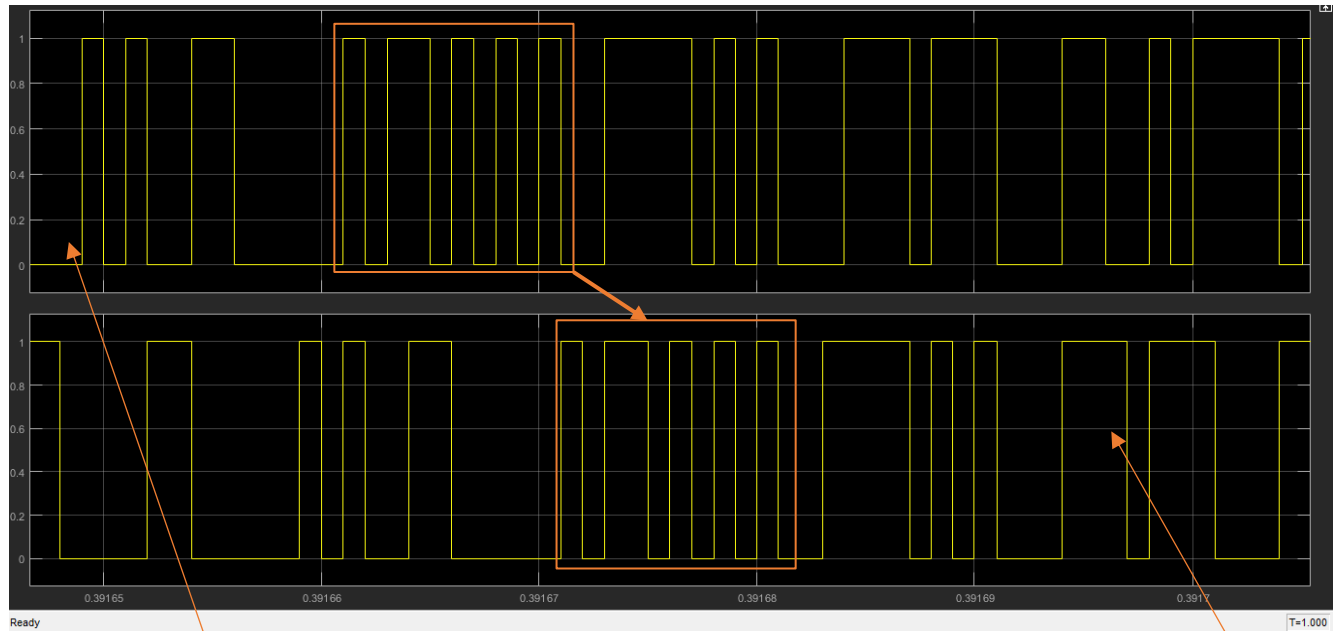
Pour effectuer une simulation il suffit en général de :

- Saisir la durée totale de la simulation (en seconde)
- Lancer la simulation (Une barre indique alors la progression)

Dans le cas où le tracé apparaît sous forme de petits segments de droites il faut réduire le pas de simulation maximal (Max step size). Pour cela : Menu "Simulation"

➔ "Configuration paramètres"

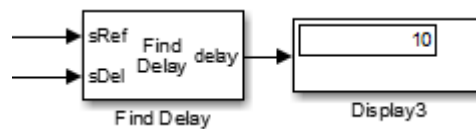
Pour $M=2$:



Signal
d'entrée

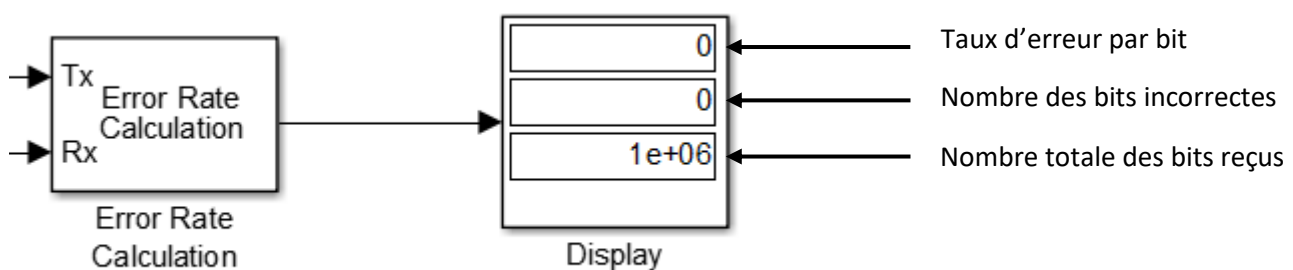
Signal de
sortie

Graphiquement, le signal de sortie est décalé de 10 bits. Cela est vérifié à l'aide du bloc « Find Delay » :

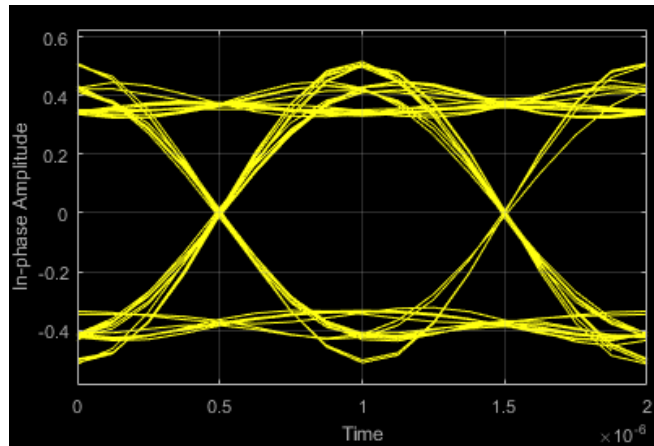


Vérification de la qualité de la chaîne :

Pour vérifier la qualité de notre chaîne le bloc « Error rate calculation » nous donne une idée. Dans notre cas on a :



Ainsi que le diagramme d'œil nous donne une idée sur la qualité du canal :
Avant de passer par le canal :



Après le passage par le canal :

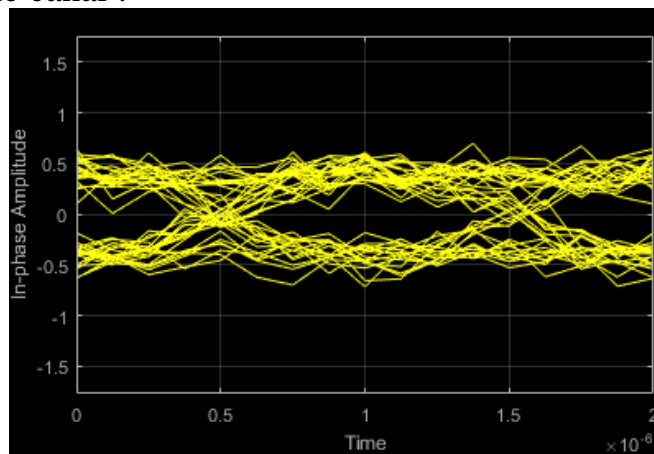
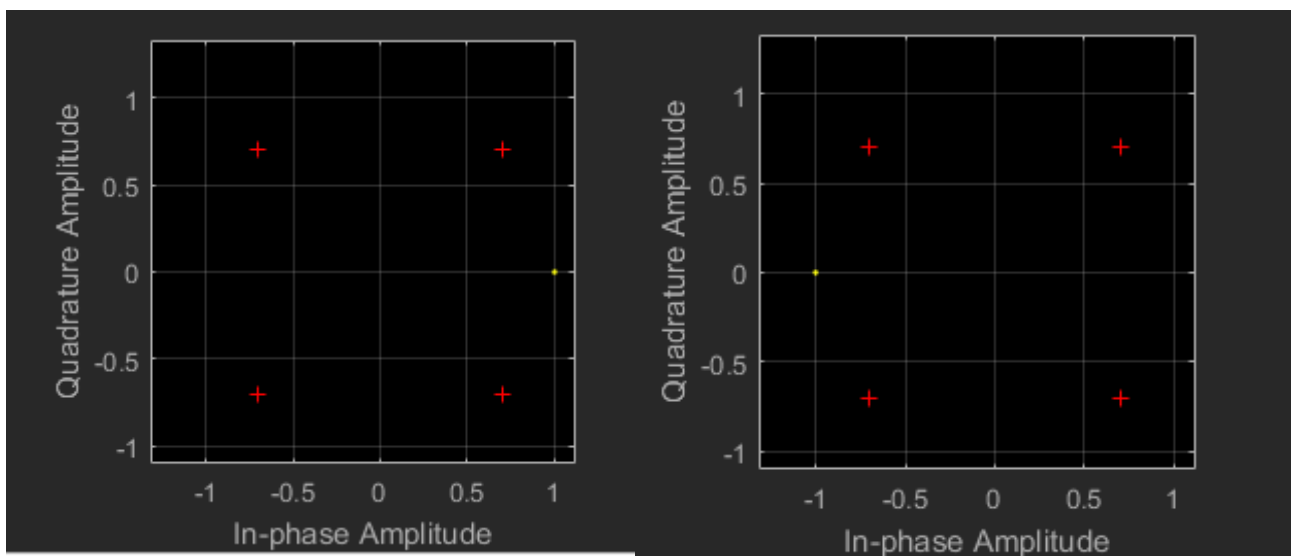


Diagramme de constellation :



Les points prennent les valeurs 1 et -1

Bibliographie:

- [1] Cours : Communication Numérique - A. JILBAB
- [2] Cours : Communication Numérique – Eric Vourc’h - Master 2 *FESUP*
UNIVERSITÉ PARIS SACLAY
- [3] Cours : Transmission Numérique en bande de base - Jamila BAKKOURY –
UNIVERSITÉ HASSAN 2
- [4] Cours de Communication Numérique : Transmission en bande de base -
Boukharouba Abdelhak - *UNIVERSITÉ 8 MAI 1945 GUELMA*
- [5] Outils Mathématiques et utilisation de Matlab - Quentin Glorieux - Université
Pierre et Marie Curie - Paris VI
- [6] Introduction à MATLAB et Simulink - Hoang Le-Huy Professeur - *Département*
de génie électrique et de génie informatique - Université Laval