

COMMUNICATION NUMERIQUE

MINI-PROJET

REALISATION D'UNE CHAINE DE TRANSMISSION NUMERIQUE

Réalisé par :

ABDALLAOUI MAAN
Amine

Sous la direction de :

Pr. JILBAB Abdelilah

I. Table des matières

INTRODUCTION.....	4
LA CHAÎNE DE TRANSMISSION NUMÉRIQUE.....	5
I. LA PARTIE ÉMETTRICE	5
1. La source du message	5
2. La séquence binaire émise	5
3. Génération des symboles.....	5
4. Modulation en bande de base « Le codeur en ligne »	6
a. Critères de choix d'un code en ligne :.....	6
b. Codage NRZ binaire (Non-Retour à Zéro)	6
i. La densité spectrale de la puissance du signal NRZ.....	7
c. Codage RZ (Retour à Zéro)	8
i. La densité spectrale de la puissance du signal RZ	9
d. Codage Manchester (Biphase)	10
i. La densité spectrale de la puissance du signal Manchester	12
5. Filtre d'émission	12
a. Interférence entre symboles ⁽²⁾ (IES).....	12
i. Filtre de Nyquist en cosinus surélevé.....	14
➤ Domaine temporel.....	14
➤ Domaine fréquentiel.....	15
ii. Filtre blanchissant	16
b. Diagramme de l'œil	17
6. La modulation.....	17
a. Critères de choix d'une Modulation	18
b. Modulation d'amplitude.....	18
i. Modulation par déplacement d'amplitude (ASK).....	18
➤ Principe	18
➤ Diagramme de constellation.....	19
ii. Modulation tout ou rien (OOK).....	20
iii. Modulation par déplacement de phase (PSK)	21
➤ Principe	21
➤ Diagramme de constellation.....	23
iv. Modulation tout ou rien (FSK)	24
➤ Modulation à phase discontinue	24
➤ Modulation à phase continue.....	25
II. CANAL DE TRANSMISSION	25
III. PARTIE RECEPTRICE	26

SIMULATION AVEC INTERFACE GRAPHIQUE MATLAB (GUI)	27
1. Présentation de GUIDE	27
a. Définition	27
b. Fonctionnement.....	27
2. Notre interface graphique de la chaine de transmission	28
3. Code Matlab utilisé.....	28
4. Exemple de résultat trouve après l'exécution de l'interface	29
REFERENCES	30

TABLES DE FIGURES

Figure 1 : La Chaîne de transmission numérique	4
Figure 2 : Principe de fonctionnement du codeur NRZ.....	6
Figure 3 : Code Matlab du codeur NRZ	7
Figure 4 : La forme du codeur NRZ	7
Figure 5 : DSP du signal NRZ complet	8
Figure 6 : Principe de Fonctionnement du codeur RZ	8
Figure 7 : Code Matlab du codeur RZ.....	9
Figure 8 : La forme du code RZ	9
Figure 9 : DSP du signal RZ complet.....	10
Figure 10 : Principe de fonctionnement du codeur Manchester	10
Figure 11 : Code Matlab du codeur Manchester	11
Figure 12 : La forme du codeur Manchester	11
Figure 13 : DSP du signal Manchester complet	12
Figure 14 : Exemple de l'Interférence entre symboles	12
Figure 15 : Variation de la bande passante B du signal	13
Figure 16 : Cosinus surélevé dans le domaine temporel	14
Figure 17 : Trace du Cosinus surélevé en temporel.....	15
Figure 18 : Cosinus surélevé dans le domaine fréquentiel	15
Figure 19 : Code Matlab du filtre Cosinus Surélevé.....	16
Figure 20 : Les traces de filtre en cosinus surélevé avec alpha de 0 a 1	16
Figure 21 : Forme de diagramme de l'oeil	17
Figure 22 : Exemple d'un codage binaire ASK	19
Figure 23 : diagramme de constellation	19
Figure 24 : La modulation ASK.....	20
Figure 25 : Exemple du codage binaire OOK	20
Figure 26 : La modulation OOK	21
Figure 27 : Exemple pour un codage binaire BPSK	22
Figure 28 : Exemple pour un codage binaire QPSK.....	23
Figure 29 : Modulation BPSK	24
Figure 30 : Modèle de canal gaussien.....	26
Figure 31: Partie réceptrice	26
Figure 32 : Demarrage de GUIDE.....	27
Figure 33 : Interface graphique de la chaîne de transmission	28
Figure 34 : Exemple de résultat trouve après l'exécution de l'interface	29

INTRODUCTION

La modélisation informatique de systèmes de communications est un moyen efficace et rapide pour mettre en lumière les performances et les principales difficultés de conception de ces derniers. Tel que la chaîne de transmission numérique qu'a pour but de minimiser/annuler les erreurs et les pertes d'informations lors du transport des données via une ligne de transmission, cette étape a une grande importance lors de développement d'un système de Télécommunication,

Les travaux pratiques, réalisés avec du matériel électronique, sont généralement coûteux, surtout lorsqu'il s'agit de mettre en œuvre des techniques de transmission numérique. Avec le développement de logiciels de simulation tels que Matlab/Simulink, LabVIEW, Python ou JAVA. Il est maintenant facile de simuler une chaîne de transmission complète : acquisition des signaux- émission - canal de transmission - réception.

Ce projet consiste à mettre en œuvre une étude d'une chaîne de transmission numérique qui va contenir les blocs suivants :

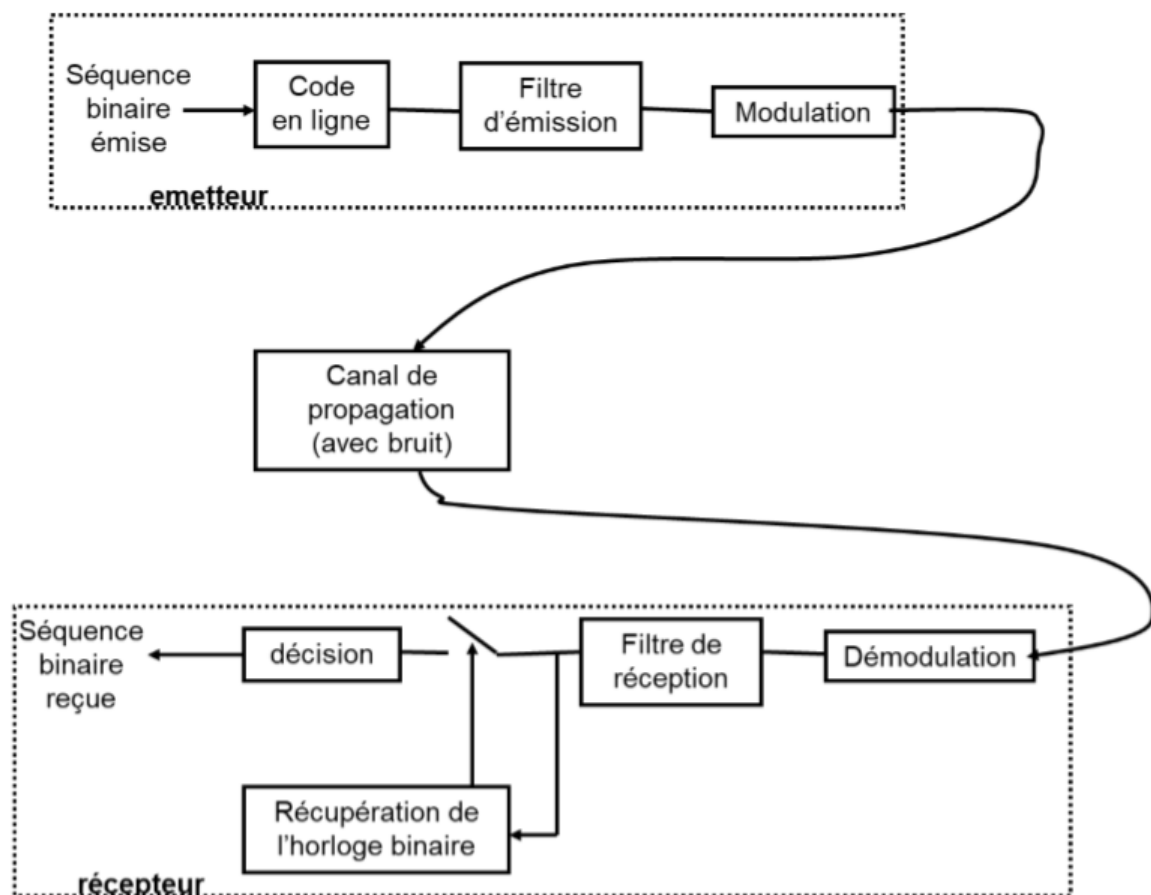


Figure 1 : La Chaîne de transmission numérique

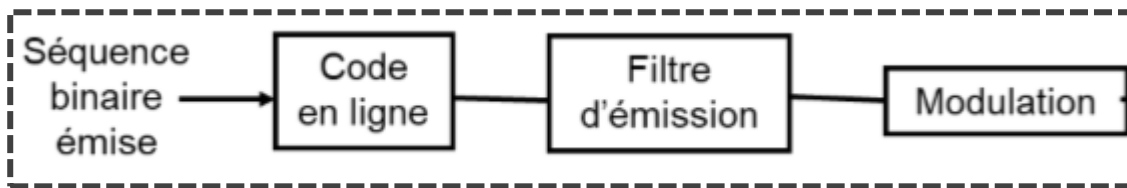
La Chaîne de transmission numérique

Notre chaîne de transmission est constituée généralement de trois parties principales :

- La partie émettrice
- Le canal de communication
- La partie réceptrice

I. La partie émettrice

Cette partie est constituée sous la forme suivante :



1. La source du message

La création d'un signal analogique qui peut être message texte, music, vidéo ou parole.

Les messages textuels sont formés de séquence de caractères alphanumériques. Durant une transmission numérique, ces caractères sont d'abord codés en une séquence de bits, appelée « bit stream » (chaîne de bits) ou signal en bande de base.

2. La séquence binaire émise

C'est le résultat d'une transformation de la forme analogique à la forme numérique nommée la quantification. Ils existent plusieurs méthodes de quantification qui convertissent et compriment le signal en bits.

Un groupe de k bits forme un symbole appartenant à un ensemble finit de symboles, appelé alphabet, formé de $M = 2^k$ symboles. Un système utilisant un alphabet de dimension M est dite système M -aire. On suppose que les données sont émises périodiquement.

On note T_b la durée d'un bit et T la durée d'un symbole. Dans ce cas, la durée d'un symbole T est égale à k fois la durée d'un bit, $T = k.T_b$.

Le débit des symboles est appelé débit baud, ou vitesse de modulation.

On trouve aussi le codeur de canal qui a comme but d'ajouter une redondance pour protéger les informations contre les erreurs introduites par un canal de communication bruité.

3. Génération des symboles

Le signal numérique émis $e(t)$ est obtenu de façon équivalente avec un filtre d'émission de réponse impulsionnelle $h_e(t)$ excité par le train d'impulsions de Dirac transportant les symboles a_k :

$$e(t) = \sum_{k=1}^{+\infty} a_k h(t - kT_s)$$

4. Modulation en bande de base « Le codeur en ligne »

La manière la plus simple de transmettre un signal est d'utiliser deux niveaux de tension pour représenter les deux états binaires 0 et 1. La présence d'une tension peut, par exemple, représenter un 1 tandis que l'absence d'une tension représente un 0.

On peut distinguer les principales catégories suivantes pour le codage linéaire de signaux PCM:

- Non-Retour à Zéro (NRZ)
- Retour à Zéro (RZ)
- Codage Manchester

a. 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)

b. Codage NRZ binaire (Non-Retour à Zéro)

Le codage NRZ ⁽¹⁾ est le moyen le plus simple de coder une information : il consiste tout simplement à transformer les 1 en +V et les 0 en -V (V étant une Valeur de tension continue), et ce de manière synchrone.

Représentation mathématique :

$$\begin{cases} e(i)=0 \Rightarrow s(i) = -V \\ e(i)=1 \Rightarrow s(i) = +V \end{cases}$$

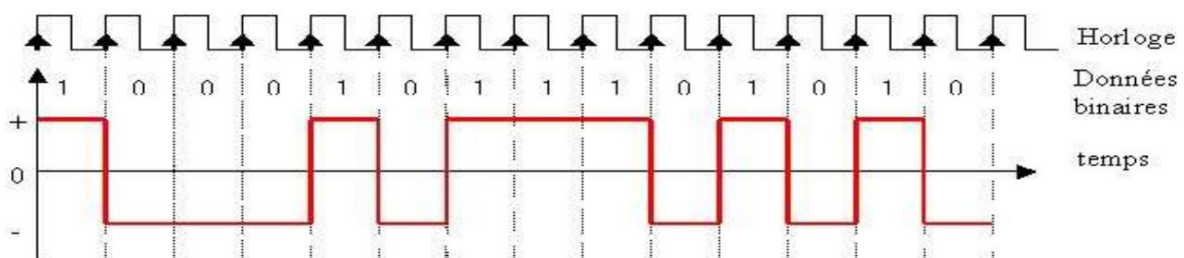


Figure 2 : Principe de fonctionnement du codeur NRZ

Lorsque le front montant de l'horloge capte la présence d'un 1 en entrée, le signal de sortie prend la valeur +V jusqu'au front montant suivant. Si l'information suivante est un 0, alors le signal prendra la valeur -V et ainsi de suite (figure 2). Ces tensions matérialiseront ainsi l'information binaire.

```
function [nrz,dsp_NRZ,t,f] = NRZ(X)
N=length(X);
Tb = 1;
nrz=[];
for i= 1:1:N
    if(X(i)==0)
        Y=-1*ones(1,100);
    else
        Y=ones(1,100);
    end
    nrz=[nrz Y];
end
t=Tb/100:Tb/100:100*N*(Tb/100);

%*-**-**-**-*- Affichage du codeur NRZ *-**-**-**-*-
figure;
subplot(2,1,1);plot(t,nrz);
title('Codeur NRZ');
ylim([-1.5 1.5])
```

Figure 3 : Code Matlab du codeur NRZ

La commande exécutée dans Command Window :

```
>> NRZ([1 0 1 0 0 1 1 0 1 0])
```

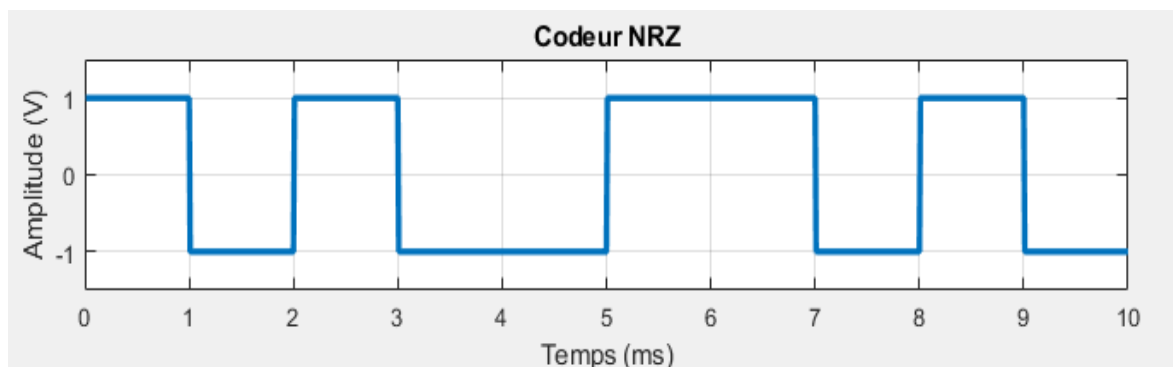


Figure 4 : La forme du codeur NRZ

i. La densité spectrale de la puissance du signal NRZ

Code Matlab utilisé :

```
%*-**-**-**-*- Calcul du DSP du codeur NRZ *-**-**-**-*-
f=-N*1/2*Tb:0.01:N*1/2*Tb-0.01;
dsp_NRZ=0.002*fftshift(fft(nrz));
```

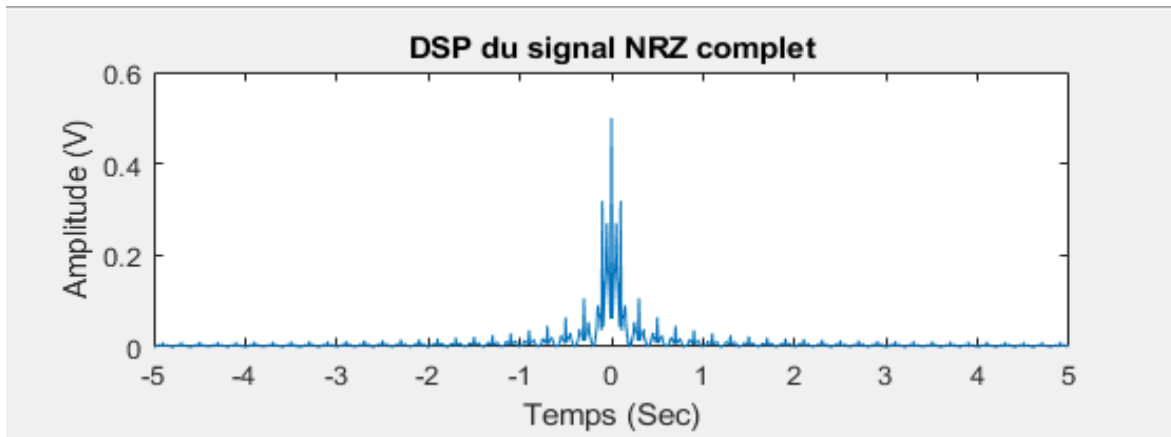



Figure 5 : DSP du signal NRZ complet

En analysant les figure 4 et 5 du codeur NRZ et sa DSP on constate qu'il se caractérise :

- Annulation du spectre tous les k/T_s (pas de synchronisation directe)
- Largeur du lobe principal : $2/T_s$
- Lobes secondaires : bande passante infini

c. Codage RZ (Retour à Zéro)

Le codage RZ⁽¹⁾ fut le premier codage exploité grâce à sa facilité d'utilisation et de réalisation. En effet, pour obtenir ce codage, il suffit de réaliser une opération logique ET (AND) entre l'information à transmettre et le signal de l'horloge. Par conséquent, si un 1 est présent il y aura alors le signal de l'horloge en sortie (amplifié pour atteindre +V au maximum). Si par contre, il y a un 0 logique, alors le signal de sortie aura la valeur nulle (0V).

Représentation mathématique :

$$\begin{cases} c(i)=0 \Rightarrow 0 \\ c(i)=1 \Rightarrow \begin{cases} s(i)=+V \\ s(i+\frac{T}{2})=0 \end{cases} \end{cases}$$

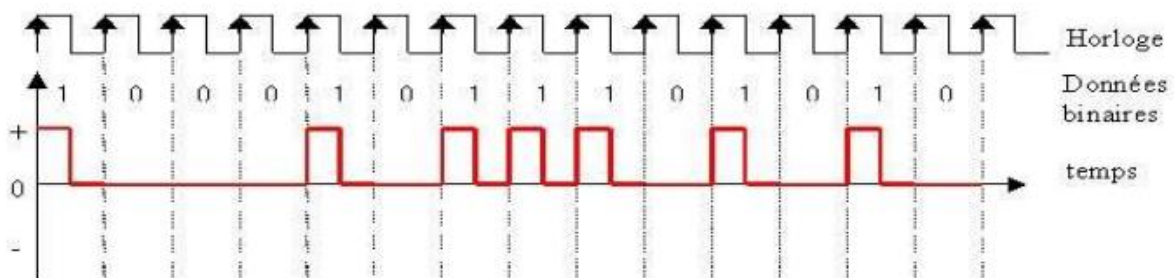


Figure 6 : Principe de Fonctionnement du codeur RZ

Malgré la simplicité de codage, le code RZ n'est plus guère utilisé. En effet, RZ signifie « retour à zéro », et le principal défaut de ce code est justement cette tension nulle. Puisqu'une information est codée sur le niveau 0 Volt, le récepteur ne peut pas établir une différence entre le bit 0 et une coupure de transmission.

```
function [rz,dsp_RZ,t,f] = RZ(X)
N=length(X);
Tb = 1;
rz=[];
for i= 1:1:N
    if(X(i)==0)
        Y=zeros(1,100);
    else
        p2=zeros(1,50);
        p1=ones(1,50);
        Y = [p1 p2];
    end
    rz=[rz Y];
end
t=Tb/100:Tb/100:100*N*(Tb/100);

%*-*-*-*-*-*-*-*-*- Affichage du codeur RZ *-*-*-*-*-*-*-*-*-
plot(t,rz,'linewidth',2);
xlabel('Temps (ms)');
ylabel('Amplitude (v)');
title('Codeur RZ');
ylim([-0.5 1.5]);grid on;
```

Figure 7 : Code Matlab du codeur RZ

La commande exécutée dans Command Window :

```
>> RZ([1 0 1 0 0 1 1 0 1 0])
```

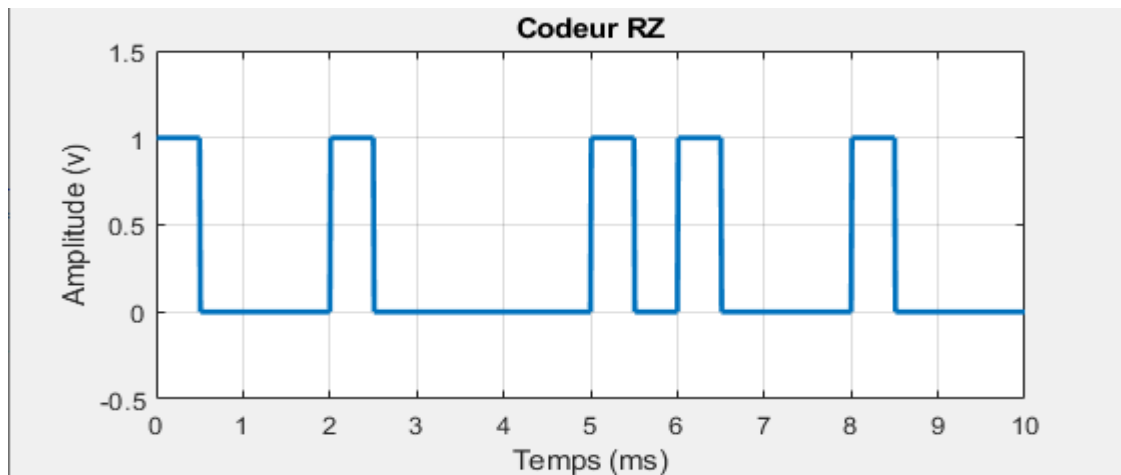


Figure 8 : La forme du code RZ

i. La densité spectrale de la puissance du signal RZ

Code Matlab utilisé :

```
%*-*-*-*-*-*-*-*-*- Calcul du DSP du codeur NRZ *-*-*-*-*-*-*-*-*-
f=-N*1/2*Tb:0.01:N*1/2*Tb-0.01;
dsp_RZ=0.002*fftshift(fft(rz));
```

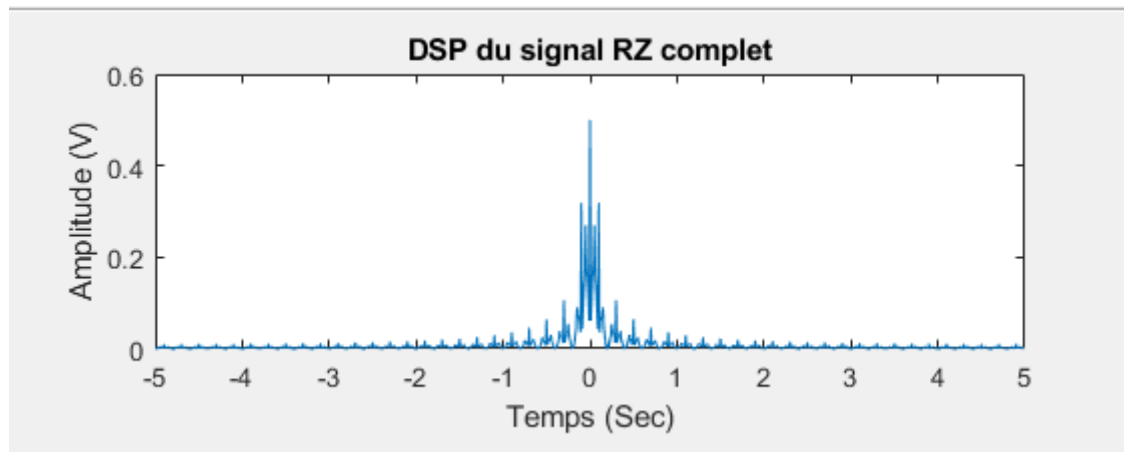


Figure 9 : DSP du signal RZ complet

En analysant les figure 4 et 5 du codeur RZ et sa DSP on constate qu'il se caractérise :

- Spectre de raie (synchronisation directe possible).
- Largeur du lobe principal : $4/T_s$.
- Lobes secondaires : bande passante infinie.

d. Codage Manchester (Biphase)

Pour obtenir le code Manchester ⁽¹⁾, il suffit de réaliser l'opération logique OU exclusif (XOR) entre l'horloge et l'information binaire à transcrire.

Le signal de sortie aura dès lors la forme suivante : un front montant en $2T$ (T représentant la période de l'horloge) pour un 1 logique et un front descendant en $2T$ pour un 0.

Représentation mathématique :

$$\begin{cases} e(i)=0 \Rightarrow \begin{cases} s(i)=+V \\ s(i+\frac{T}{2})=-V \end{cases} \text{ ou } \downarrow \\ e(i)=1 \Rightarrow \begin{cases} s(i)=-V \\ s(i+\frac{T}{2})=+V \end{cases} \text{ ou } \uparrow \end{cases}$$

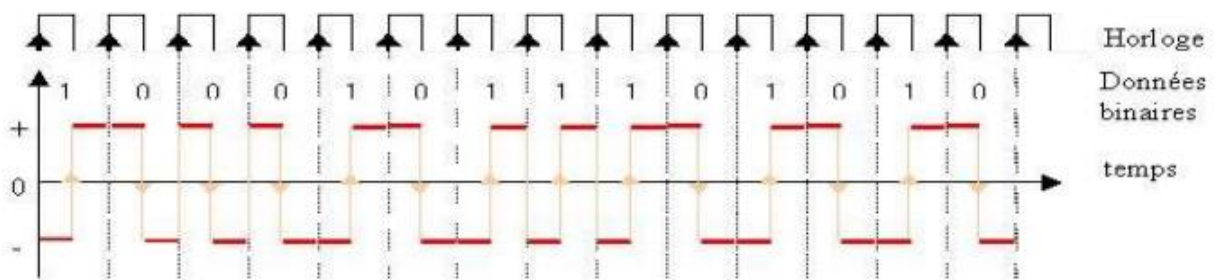


Figure 10 : Principe de fonctionnement du codeur Manchester

Lors de l'envoi de longues séries de 0 ou de 1, la synchronisation entre émetteur et récepteur est toujours assurée puisqu'il y a un perpétuel changement d'état, on dit qu'il est autoporteur de l'horloge. De plus, l'information binaire étant représentée non pas par un niveau, mais par une transition, cela augmente la fiabilité du codage et de la transmission. La réalisation technique est aussi relativement simple. Cependant, le codage Manchester présente deux inconvénients : le débit de transmission doit être deux fois plus élevé que pour un codage NRZ. Pour obtenir un débit de 1Gbit/s, il faudrait une fréquence de 1Ghz, ce qui est bien trop élevé pour être physiquement stable.

L'autre inconvénient est le câblage : une inversion de polarité entraîne une inversion des informations binaires. Ce pourquoi le code Manchester différentiel a été inventé.

Code Matlab du codeur en ligne Manchester :

```
function [manch,dsp_Manchester,t,f] = Manchester(X)
N=length(X);
Tb = 1;
manch=[ ];
for i= 1:1:N
    if (X(i)==0)
        p1=-1*ones(1,50);
        p2=ones(1,50);
        Y=[p1 p2];
    else
        p3=-1*ones(1,50);
        p4=ones(1,50);
        Y = [p4 p3];
    end
    manch=[manch Y];
end
t=Tb/100:Tb/100:100*N*(Tb/100);
%*-*-*-*-*-*-*-*-* Affichage du codeur Manchester *-*-*-*-*-*-*-*-*
plot(t,manch,'linewidth',2);
xlabel('Temps (ms)');
ylabel('Amplitude (v)');
title('Codeur Manchester');
ylim([-1.5 1.5]);grid on;
```

Figure 11 : Code Matlab du codeur Manchester

La

commande exécutée dans Command Window :

```
>> Manchester([1 0 1 0 0 1 1 0 1 0])
```

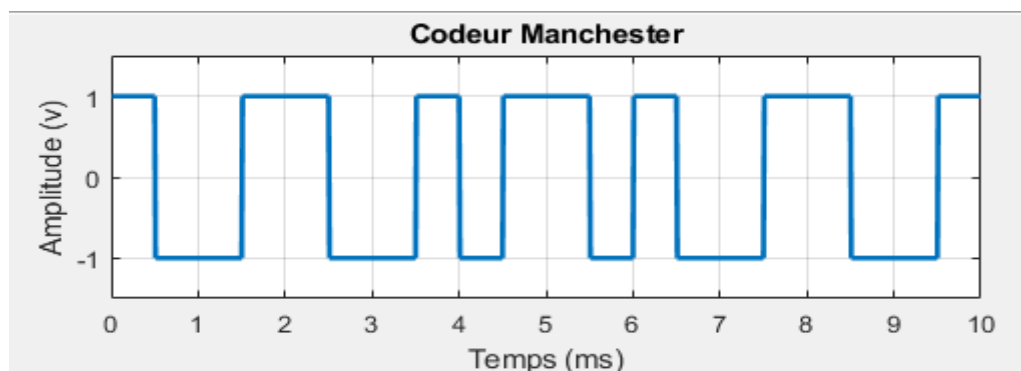


Figure 12 : La forme du codeur Manchester

i. La densité spectrale de la puissance du signal Manchester

Code Matlab utilisé :

```
%*-*-*-*-* Calcul du DSP du codeur NRZ *-*-*-*-*
f=-N*1/2*Tb:0.01:N*1/2*Tb-0.01;
dsp_Manchester=0.002*fftshift(fft(manch));
```

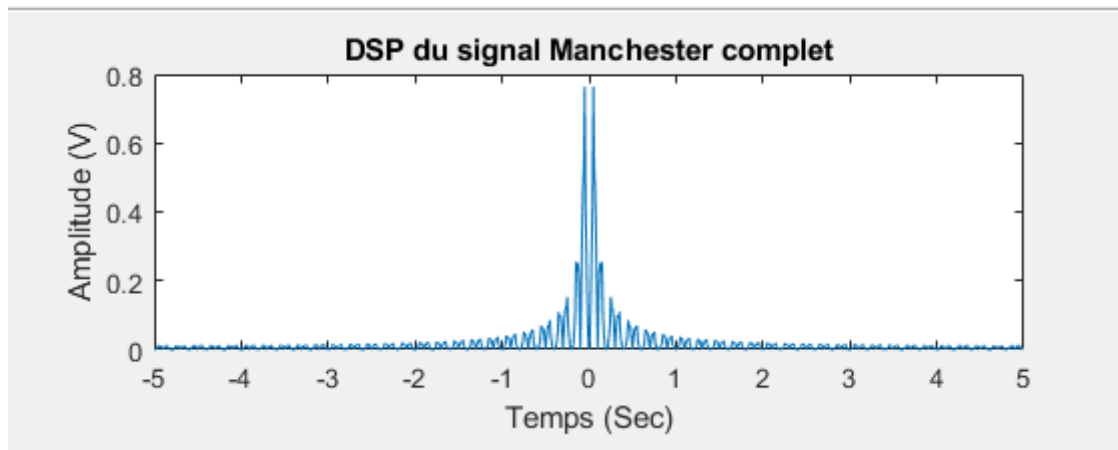


Figure 13 : DSP du signal Manchester complet

En analysant les figure 12 et 13 du codeur Manchester et sa DSP on constate qu'il se caractérise :

- Spectre non nul à la fréquence symbole(synchronisation directe possible).
- Largeur du lobe principal : $4/T_s$
- Lobes secondaires : bande passante infinie.
- Absence de composante continue.

5. Filtre d'émission

a. Interférence entre symboles⁽²⁾ (IES)

- Chevauchement entre symboles successifs conduisant à une erreur binaire.

Ce phénomène se produit si l'amplitude de l'impulsion soumise à échantillonnage en réception dépend à l'instant de décision de symboles voisins.

- Les retards des canaux et le phénomène de multi-trajet conduit à faire apparaître de l'IES.

Le phénomène d'IES a pour conséquence de réduire la marge de tolérance entre les niveaux échantillonnés en réception. En présence de bruit, il entraîne une augmentation de la probabilité d'erreur de décision.

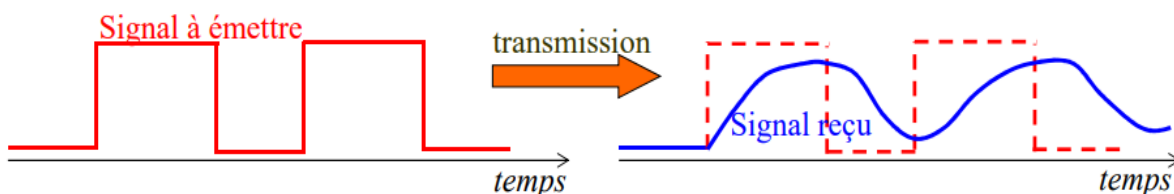


Figure 14 : Exemple de l'Interférence entre symboles

Conditions indispensables pour les transmissions numériques :

L'interférence inter symboles doit être nulle.

Pour annuler l'IES, l'effet des symboles précédents doit s'annuler à l'instant d'échantillonnage = condition de Nyquist dans le temps. Pour ceci la bande passante du canal de propagation doit vérifier :

$$B \geq \frac{1}{2T_s}$$

$$g(KT_s) = 0 \iff k \neq 0$$

$$g(0) = 1$$

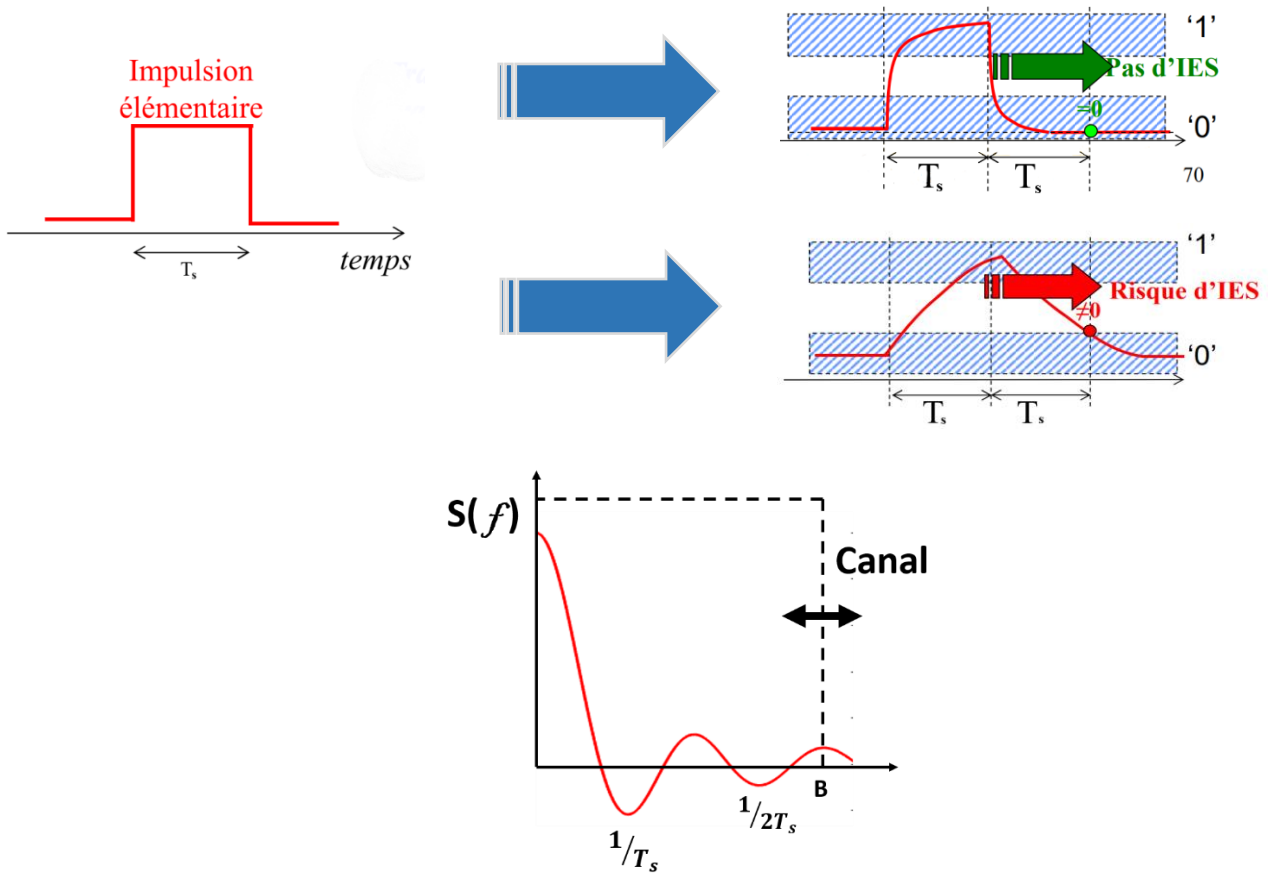


Figure 15 : Variation de la bande passante B du signal

Au fur et à mesure que l'on diminue la bande passante B du canal, on constate une augmentation de la distorsion du signal avec, aux instants de décision

Filtre utilisé dans les normes télécoms : filtre en cosinus surélevé

i. *Filtre de Nyquist en cosinus surélevé*

➤ *Domaine temporel*

$$g(t) = \frac{\sin\left(\frac{\pi t}{T_s}\right)}{\left(\frac{\pi t}{T_s}\right)} \frac{\cos\left(\frac{\pi \alpha t}{T_s}\right)}{1 - \left(\frac{\pi \alpha t}{T_s}\right)^2}$$

Figure 16 : Cosinus surélevé dans le domaine temporel

Le paramètre α est appelé le coefficient de retombée (roll-off factor), parfois le facteur d'arrondi $\alpha = 0$ correspondant au cas "idéal" (filtre rectangulaire), $\alpha = 1$ correspondant au filtre avec réponse fréquentielle en cosinus. La largeur spectrale est donnée par : $B = \frac{1+\alpha}{2T_s}$

Code Matlab du filtre Cosinus Surélevé dans le domaine temporel :

```
function [g0,g05,g1,t] = CosSurelveTemp(a)
fs=4;
Ts=1;
t=-fs:1/120:fs;
s=(sin(pi.*t)/Ts)./( (pi.*t)/Ts);
c=(cos(a*pi.*t)/Ts)./(1-((2*a.*t)/Ts).^2);
g0=s.*c;
g05=s.*c;
g1=s.*c;
```

A executer :

```
figure;
[g,t] = CosSurelveTemp(0);
plot(t,g,'r');hold on;
[g,t] = CosSurelveTemp(0.5);
plot(t,g,'b');hold on;
[g,t] = CosSurelveTemp(1);
plot(t,g,'g');hold on;
legend('a=0','a=0.5','a=1')
title('Filtre en cosinus sureleve');
xlabel('Temps ms') ;
```

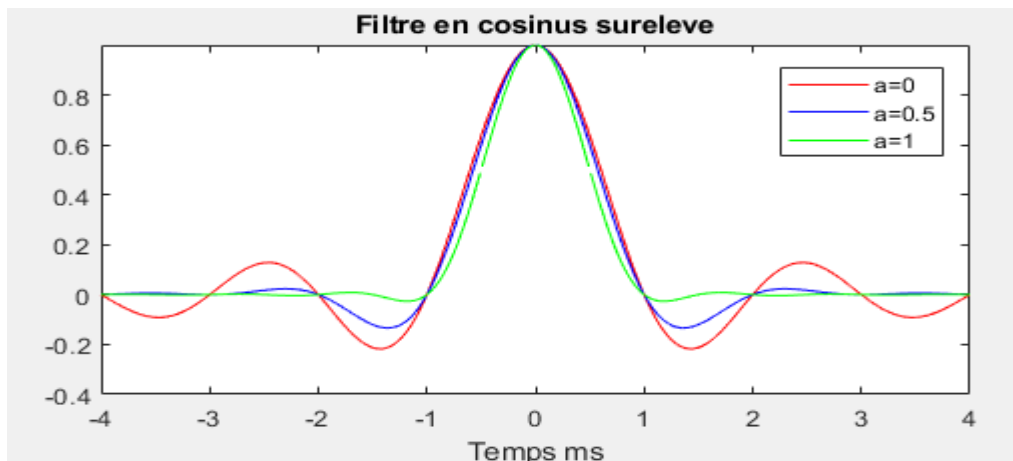


Figure 17 : Trace du Cosinus surélevé en temporel

Le paramètre α est appelé coefficient d'arrondi ou facteur de retombée (roll-off).

Le cas $\alpha = 0$, purement théorique, correspond à un filtre dont la réponse impulsionnelle est en sinus cardinal. Cette forme de réponse possède des lobes secondaires importants à décroissance lente, propriété défavorable pour IES.

Pour $\alpha = 1$, la réponse impulsionnelle présente des passages par zéro qui s'effectuent avec une période égale à $T_s/2$.

➤ Domaine fréquentiel

En calculant la transformée de fourrier du filtre en cosinus surélevé $g(t)$ on trouve la relation en domaine fréquentiel suivante :

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

Figure 18 : Cosinus surélevé dans le domaine fréquentiel

Avec

- α = le facteur de retombe ou Roll-off.
- T_s = la période d'un symbole.

Code Matlab du filtre Cosinus Surélevé :

```
a=0; %coefficient de retombé
Ts=0.1; %Période d'un symbole
f= -1/Ts:0.001:1/Ts; %génération de vecteur de
fréquence

%cosinus surélevé (qui vérifie le critère de Nyquist)
g=zeros(1,length(f));
for i=1:length(f)
    if (( abs(f(i))>=0 ) && ( abs(f(i))< ((1-a)/(2*Ts)) ) )
        g(i)=Ts;
```



```

else if(( abs(f(i))<= ((1+a)/(2*Ts))) && (abs(f(i))>=((1-
a)/(2*Ts))) )
    g(i)=(Ts/2)*(1-sin((pi*(2*abs(f(i))*Ts-1)/(2*a))));
end
end
end
end

figure;
plot(f,10.*g,'linewidth',2);grid on
ylim([0 1.2]);
legend('a = 0');
title('Filtre en cosinus surélevé');
xlabel('fréquence (kHz)');

```

Figure 19 : Code Matlab du filtre Cosinus Surélevé

Résultat des traces des filtre en cosinus surélevé pour les trois différentes valeurs du facteur de retombe $\alpha = 0/0.5/1$:

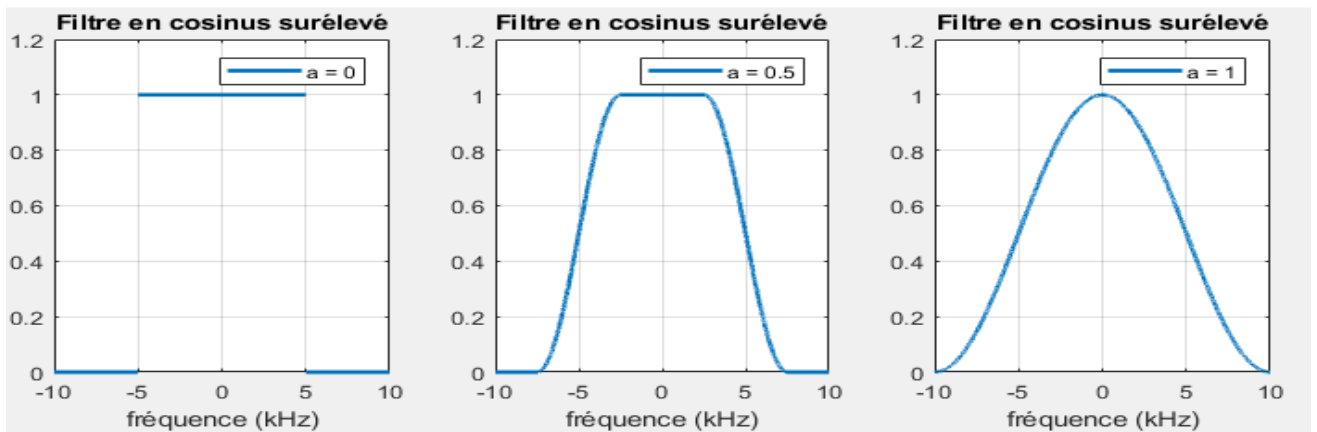


Figure 20 : Les traces de filtre en cosinus surélevé avec alpha de 0 à 1

D'après l'analyse de la figure 15 on remarque que la largeur spectrale prend comme valeur $\frac{1+\alpha}{2T_s}$

Pour une valeur nulle de α on trouve le cas idéal (un filtre passe bas idéal).

NB : Quel que soit notre α le filtre en cosinus surélevé est une solution idéale pour éviter les interférences entre symbole.

ii. Filtre blanchissant

La présence de ce filtre dans le filtre d'émission a une grande importance a cause de son rôle qui consiste a transformer un signal continue a des impulsions à travers le calcul effectué sur le codeur en ligne (transforme de fourrier inverse)



$$g(t) = \text{rect}_{T_b}(t) \rightarrow G(f) = T_b \text{sinc}(fT_b)$$

Alors le filtre blanchissant sera la fonction réciproque de $G(f)$ notée

$$G_1(f) = \frac{1}{T_b} \frac{\pi f T_b}{\sin(\pi f T_b)} \Rightarrow G_1(f) \times G(f) = 1$$

Le filtre d'émission a pour rôle la réduction de la bande passante du signal émis qu'il a un spectre infini et avec un inconvénient présenté dans la génération des interférences entre symboles

Le filtre de réception consiste à réduire la puissance du bruit ajouté au signal émis par le canal de transmission en utilisant un filtrage passe bas

b. Diagramme de l'œil

Le diagramme de l'œil permet de contrôler visuellement la quantité d'IES.

Superposition des tracés d'un signal aléatoire reçu sur un multiple de la durée du symbole.

Les performances du canal de transmission sont lues à travers les ouvertures verticales et horizontales de l'œil.

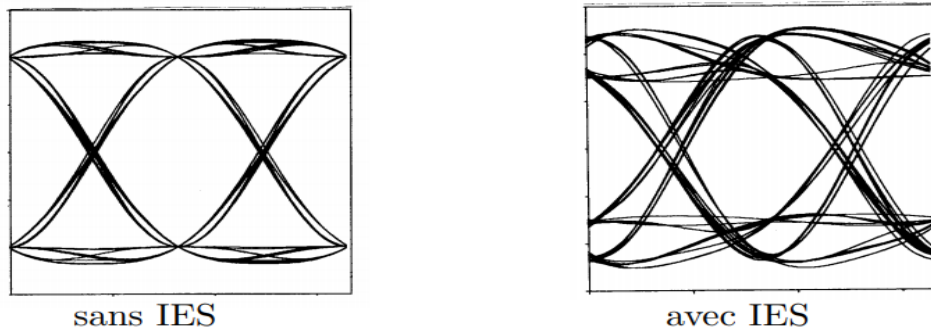


Figure 21 : Forme de diagramme de l'œil

6. La modulation

La modulation⁽³⁾ consiste à utiliser le signal modulant pour faire varier l'un des paramètres d'une onde sinusoïdale qu'on désigne par porteuse. La porteuse doit avoir une fréquence très supérieure à la fréquence la plus élevée du signal modulant à transmettre.

Le rôle de la modulation est de maîtriser le spectre du signal émis :

- Adaptation au canal physique.
- Adaptation de la bande passante.
- Maîtrise du débit.
- Multiplexage.

Et donc, La modulation consiste à modifier un ou plusieurs paramètres d'une onde porteuse $s(t) = A \sin(\omega_0 t + \varphi_0)$, centrée sur la bande de fréquence du canal :

- Son amplitude A . \rightarrow Modulation AM.

- Sa fréquence. $f_0 = \frac{\omega_0}{2\pi} \rightarrow$ Modulation FM.
- Sa phase $\varphi_0 \rightarrow$ Modulation PM.

Le message transmis est une suite de symboles pris dans un alphabet de taille $M = 2^n$ (symboles M-aires, ou binaires pour $M = 2$).

La rapidité de modulation, en bauds, est le nombre de changement d'états par seconde : $R = \frac{1}{T}$

Le débit binaire est le nombre de bits transmis par seconde : $D = 1/T_b$.

Pour un alphabet M-aire, $M = 2^n$, $T = nT_b$, soit $D = nR$.

NB : il est préférable d'utiliser des signaux sinusoïdaux comme porteur à cause de l'occupation d'une partie affine dans le spectre si on parle sur la partie positive, il n'a qu'une raie, c'est une particularité des signaux sinusoïdaux.

a. Critères de choix d'une Modulation

On cherche à :

- Minimiser l'occupation spectrale du signal modulé pour un débit binaire donné.
- Minimiser la probabilité d'erreur pour des conditions de transmissions données.
- Efficacité spectrale : le rapport du nombre de bits transmis par second (avec une certaine probabilité d'erreur pour un canal donné) avec la largeur de bande utilisée.
- Efficacité en puissance : le rapport E_b / N_0 nécessaire pour obtenir une probabilité d'erreur donnée pour un canal particulier (E_b : énergie moyenne par bit d'information, $N_0/2$: Densité Spectrale d'un bruit blanc).

b. Modulation d'amplitude

i. Modulation par déplacement d'amplitude (ASK)

➤ Principe

La modulation ASK (amplitude Shift Keying) s'obtient en remplaçant chaque symbole de la bande de base par une impulsion sinusoïdale de fréquence fixe f_c , de durée T_s , et dont l'amplitude dépend du niveau a_k associé au symbole.

Le signal ASK s'écrit donc : **$S(t) = e(t) \cdot p(t)$**

Avec :

- $e(t)$ = signal numérique émis.
- $p(t)$ = signal de la porteuse.

$$S(t) = \left[\sum_{k=0}^{+\infty} a_k h(t - kT_s) \right] \cdot A \sin(2\pi f_c t) = i(t) A \cos(2\pi f_c t)$$

($i(t)$ peut correspondre à un codage binaire ou M-aire)

Exemple pour un codage binaire 2-aire :

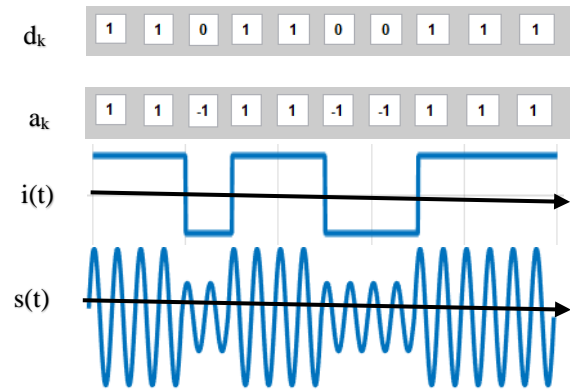


Figure 22 : Exemple d'un codage binaire ASK

➤ **Diagramme de constellation**

Le diagramme de constellation permet de représenter tous les états possibles de la modulation. Pour une modulation ASK, on représente les différents états pris par $i(t)$ sur un axe horizontal.

Exemple pour un codage binaire 2-aire précédent :

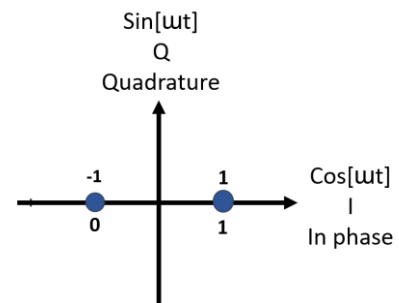


Figure 23 : diagramme de constellation

Code Matlab de la modulation ASK :

```
clc;
close all;
Nb=10;
X=randi([0 1], 1, Nb);
Tb=1;
disp(' le signal binaire à transmettre:');
Disp(X);

%*-**-**-**-**-**-**-**-**-**-**-**-**-**- Modulation ASK *-**-**-**-**-**-**-**-**-**-
A1=10; % Amplitude pour le bit 1
A2=5; % Amplitude pour le bit 0
f=2; % la fréquence de la porteuse
m=[];
t=Tb/100:Tb/100:Tb;
for (i=1:length(x))
    if (x(i)==1)
        y=A1*sin(2*pi*f*t2);
    else
        y=A2*sin(2*pi*f*t2);
    end
    m=[m y]; % le signal modulé
end
```

```
t2=Tb/100:Tb/100:Tb*length(x);

%***** AFFICHAGE *****
plot(t2,m);
xlabel('Temps(Sec)');
ylabel('Amplitude(V)');
title('Modulation ASK');
```

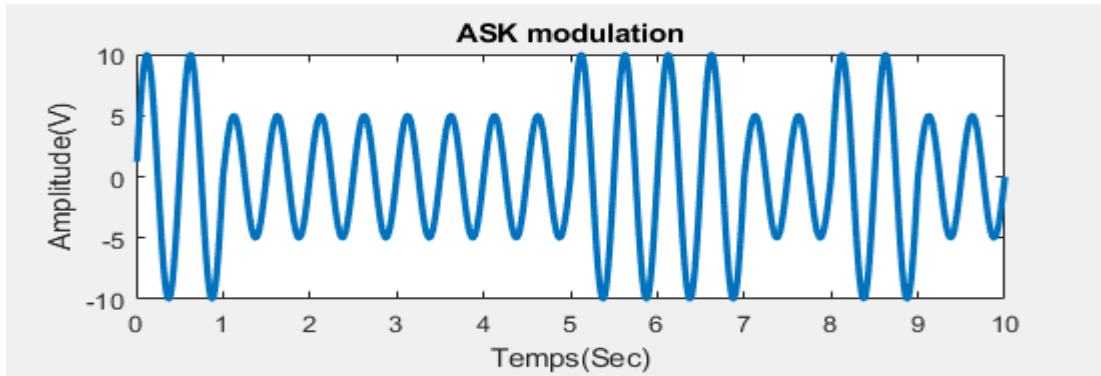


Figure 24 : La modulation ASK

ii. Modulation tout ou rien (OOK)

La modulation OOK (On Off Keying) est un cas particulier de modulation ASK, dans le cas où le signal en bande de base $i(t)$ est un signal binaire RZ (état bas $\rightarrow 0$; état haut $\rightarrow +V$).

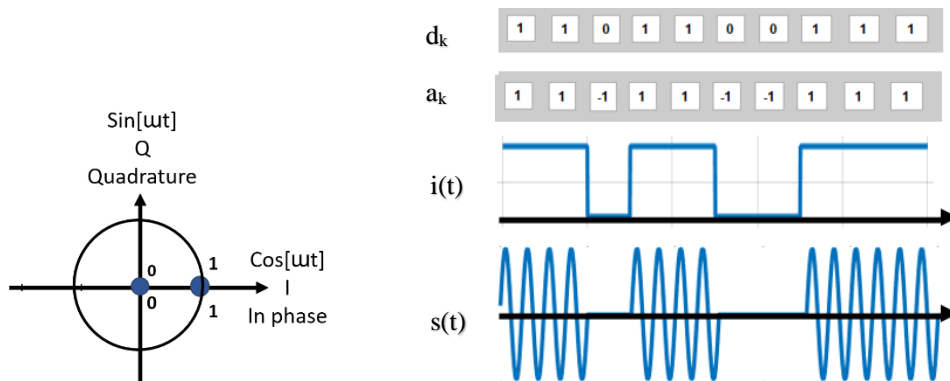


Figure 25 : Exemple du codage binaire OOK

$$S(t) = \left[\sum_{k=0}^{+\infty} a_k h(t - kTs) \right] \cdot A \sin(2\pi f_c t) = i(t) A \cos(2\pi f_c t)$$

Code Matlab de la modulation OOK :

```
clc;
close all;
Nb=10;
X=randi([0 1], 1, Nb); % Séquence binaire
Tb=1; % période d'un bit
disp(' le signal binaire à transmettre:');
Disp(X);
%***** Modulation OOk *****
A=10; % Amplitude de la porteuse
f=2; % la fréquence de la porteuse
t=Tb/100:Tb/100:Tb;
```

```

sc=A.sin(2*pi*fc*t); % le signal porteur
m=[];
for (i=1:length(x))
    if (x(i)==1)
        y=1*A*sin(2*pi*f*t2);
    else
        y=0*A*sin(2*pi*f*t2);
    end
    m=[m y]; % le signal modulé
end
%*-*-*-*-*-*-*-*-*-* AFFICHAGE *-*-*-*-*-*-*-*-*-*
t2=Tb/100:Tb/100:Tb*length(x);
plot(t2,m);
xlabel('Temps(Sec)');
ylabel('Amplitude(V)');
title('Modulation OOk ');

```

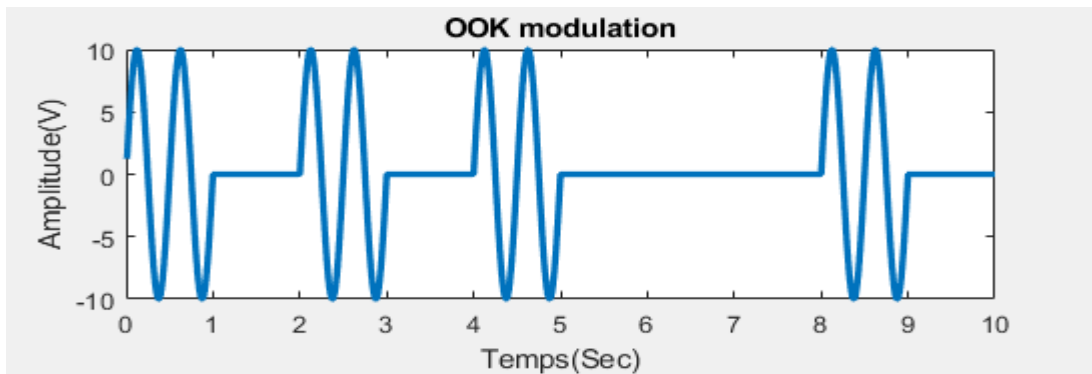


Figure 26 : La modulation OOk

iii. Modulation par déplacement de phase (PSK)

Le phase-shift keying (ou PSK, soit « Modulation par déplacement de phase ») désigne une famille de formes de modulations numériques qui ont toutes pour principe de véhiculer de l'information binaire via la phase d'un signal de référence (porteuse), et exclusivement par ce biais.

Comme pour toute technique de modulation numérique, la phase en question ne peut prendre qu'un nombre fini de valeurs. Chacune de ces valeurs représente un unique nombre binaire, dont la taille (et donc la quantité d'information transmise) dépend du nombre de valeurs possibles pour la phase. Généralement, pour une modulation PSK donnée, les nombres binaires représentés sont tous de même taille.

➤ Principe

A partir d'un signal en bande de base $u(t) = \sum_{k=0}^{+\infty} d_k h(t - kT_s)$, on remplace le symbole (bit) d_k par une fonction cosinus dont la phase φ_k dépend de la valeur de d_k . On obtient un signal modulé en phase:

$$s(t) = \sum_{k=0}^{+\infty} A \cos(\omega_p t + \varphi_k) h(t - kT_s)$$

Les formes de PSK les plus utilisées sont la BPSK (ou 2-PSK : deux valeurs de phase possibles), la QPSK (ou 4-PSK : quatre valeurs de phase possibles) et la DPSK (différentiel-PSK : où l'information est contenue non pas dans une valeur absolue de phase, mais dans le déphasage entre deux signaux successifs).

BPSK est la forme la plus simple du PSK. Elle utilise deux phases qui sont séparées de 180° ; on l'appelle également 2-PSK. Cette modulation est la plus robuste de toutes les PSK car il faut une grande déformation du signal pour que le démodulateur se trompe sur le symbole reçu.

Cependant on ne peut moduler qu'un seul bit par symbole (voir figure 27), ce qui est un inconvénient pour les applications qui nécessitent un débit binaire élevé.

Exemple pour un codage binaire 2-aire :

(BPSK)

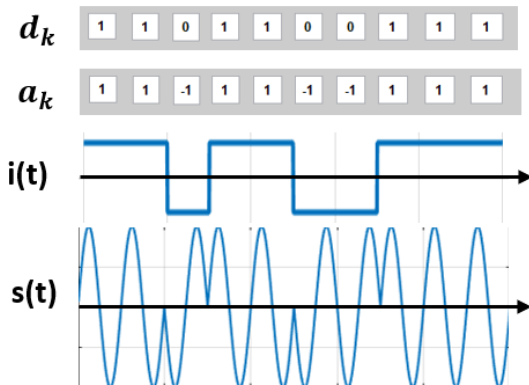
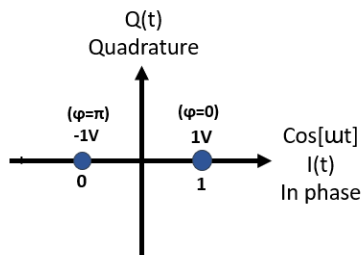


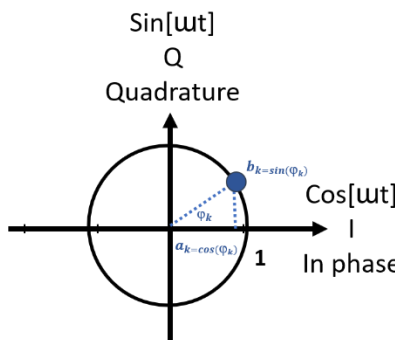
Figure 27 : Exemple pour un codage binaire BPSK

Exemple pour un codage binaire 4-aire (QPSK) :

$$s(t) = \sum_{k=0}^{+\infty} A \cos(\omega_p t + \varphi_k) h(t - kT_s)$$

$$s(t) = \sum_{k=0}^{+\infty} \cos(\varphi_k) h(t - kT_s) A \cos(\omega_p t) - \sum_{k=0}^{+\infty} A \sin(\varphi_k) h(t - kT_s) A \sin(\omega_p t)$$

Le principe de la modulation QPSK est de grouper l'envoi de bits deux par deux.



On pose :

$$a_k = \cos(\varphi_k) \quad \text{Et} \quad i(t) = \sum_k a_k h(t - kT_s) \quad \text{Modulant de phase}$$

$$b_k = \sin(\varphi_k) \quad \text{Et} \quad q(t) = \sum_k b_k h(t - kT_s) \quad \text{Modulant de quadrature}$$

Le signal modulé s'écrit alors : $s(t) = i(t) A \cos(\omega_p t) + q(t) A \sin(\omega_p t)$

(Ce qui est l'équivalent de 2 modulations ASK)

Exemple pour une modulation QPSK (4 états) :

- Déterminer les différentes valeurs possibles de a_k et b_k
- Dessiner les signaux $i(t)$ et $q(t)$ pour la séquence suivante : 01 11 01 01 10 01

$$\begin{aligned}
 d_k = 00 & \rightarrow \varphi_k = \frac{\pi}{4} \rightarrow \begin{cases} a_k = \cos\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \\ b_k = \sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \end{cases} \\
 d_k = 11 & \rightarrow \varphi_k = -\frac{\pi}{4} \rightarrow \begin{cases} a_k = \cos\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \\ b_k = \sin\left(\frac{\pi}{4}\right) = -\frac{1}{\sqrt{2}} \end{cases} \\
 d_k = 01 & \rightarrow \varphi_k = \frac{3\pi}{4} \rightarrow \begin{cases} a_k = \cos\left(\frac{\pi}{4}\right) = -\frac{1}{\sqrt{2}} \\ b_k = \sin\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \end{cases} \\
 d_k = 10 & \rightarrow \varphi_k = -\frac{3\pi}{4} \rightarrow \begin{cases} a_k = \cos\left(\frac{\pi}{4}\right) = -\frac{1}{\sqrt{2}} \\ b_k = \sin\left(\frac{\pi}{4}\right) = -\frac{1}{\sqrt{2}} \end{cases}
 \end{aligned}$$

➤ *Diagramme de constellation*

Le diagramme de constellation d'une modulation PSK comporte 2 axes :

- Un axe horizontal I où l'on porte les différentes valeurs de $i(t)$.
- Un axe vertical Q où sont portées les valeurs prises par $q(t)$.

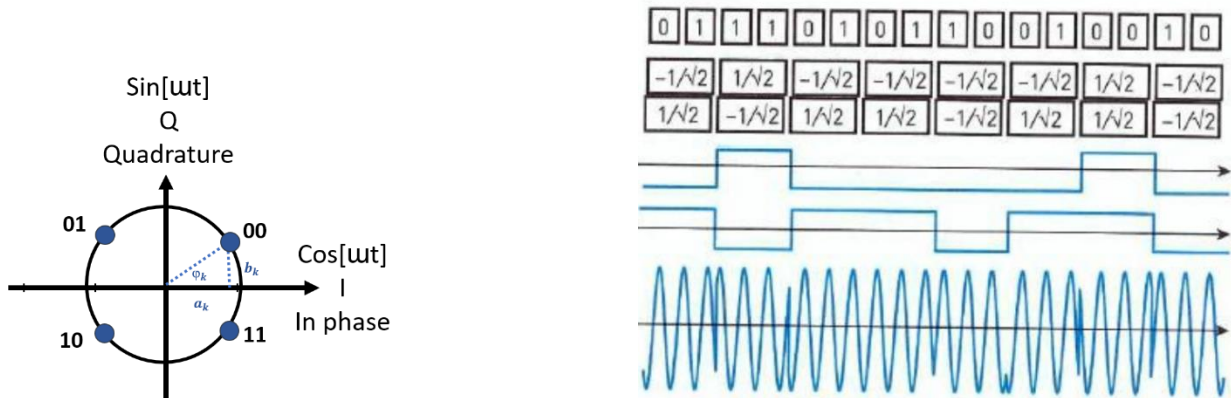


Figure 28 : Exemple pour un codage binaire QPSK

Code Matlab de la modulation BPSK :

```

clc;
close all;
Nb=10;
X=randi([0 1], 1, Nb); % Séquence binaire
Tb=1; % periode d'un bit
disp(' le signal binaire à transmettre: ');

```


Disp(X);

```
%*-*-*-*-*-*-*-*-*-* Modulation OOK *-*-*-*-*-*-*-*-*-*
A=10; % Amplitude de la porteuse
f=2 % la fréquence de la porteuse
t=Tb/100:Tb/100:Tb;
sc=A.sin(2*pi*f*c*t); % le signal porteur
m=X.*sc; % le signal modulé
t2=Tb/100:Tb/100:Tb*length(x);
plot(t2,m);
xlabel('Temps (Sec)');
ylabel('Amplitude (V)');
title('Modulation BPSK');
```

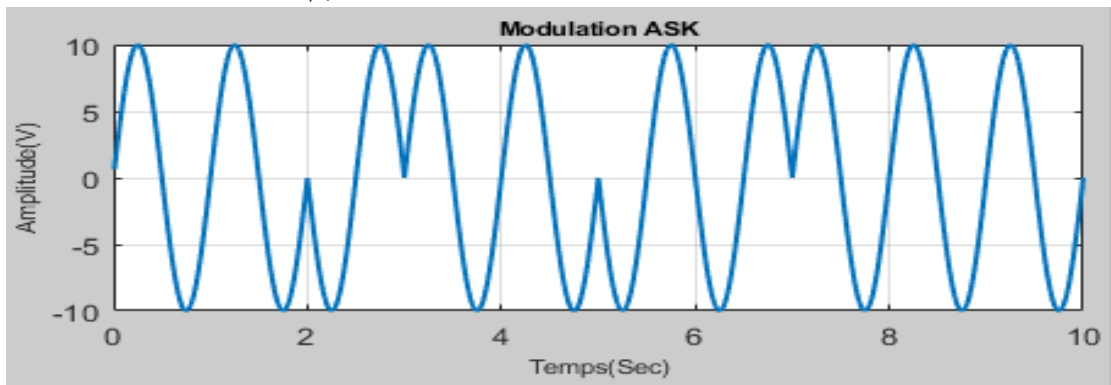


Figure 29 : Modulation BPSK

iv. Modulation tout ou rien (FSK)

On considère une communication binaire de débit R , transmettant les bits d_k ($d_k=0$ ou 1). La modulation FSK (Frequency Shift Keying) consiste à émettre un signal sinusoïdal (la porteuse) d'amplitude A constante, mais dont la fréquence varie selon les valeurs de d_k .

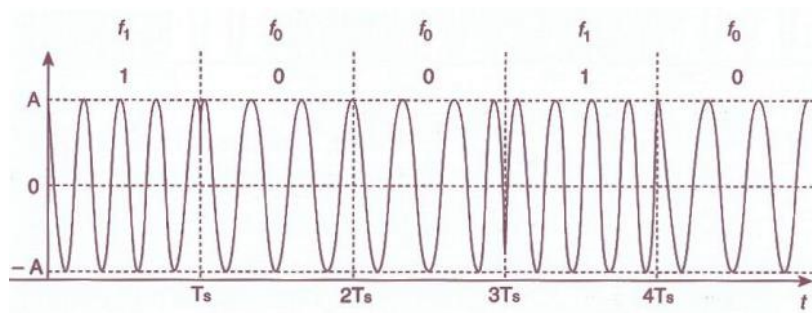
➤ Modulation a phase discontinue

Le signal modulé est de la forme $S(t) = A \cos(2\pi f_c t)$

f_c est la fréquence de la porteuse : elle peut prendre uniquement 2 valeurs : f_0 et f_1 .

$$d_k = 0 \rightarrow f_k = f_0$$

$$d_k = 1 \rightarrow f_k = f_1$$

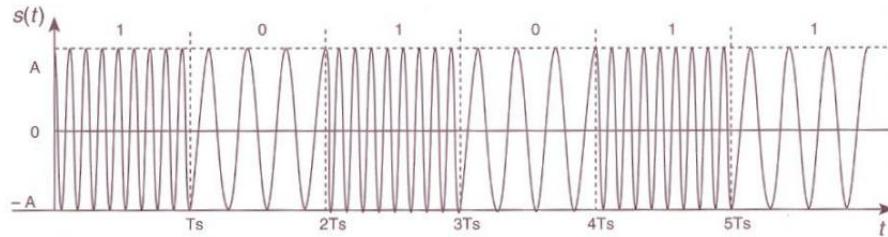


La modulation FSK à phase discontinue peut s'interpréter comme une somme de deux modulations OOK : une à la fréquence f_1 lorsque $d_k = 1$, et l'autre à la fréquence f_0 lorsque $d_k = 0$.

➤ *Modulation a phase continue*

Le signal modulé s'écrit $S(t) = A \cos(2\pi f_p t + \varphi(t))$

f_p est la fréquence centrale de la porteuse. La fréquence instantanée $f(t)$ vaut $f_0 = f_p - \Delta f$ pour $d_k = 0$ et $f_1 = f_p + \Delta f$ pour $d_k = 1$. Contrairement au cas précédent, la fréquence $f(t)$ varie de façon continue entre f_0 et f_1 lors de la transition entre 2 symboles (pour $t = kT_s$) :



II. Canal de transmission

Une transmission d'information se fait toujours à distance, un support physique assure le lien entre la source et le destinataire. Il existe plusieurs principaux supports qui sont couramment utilisés comme média de transmission.

Un signal numérique transmis dans un canal subit une distorsion linéaire due au produit avec la réponse impulsionnelle. et l'ajout d'un bruit additif blanc gaussien (BBAG).

Le premier modèle est le plus utilisé en théorie. Il s'agit du bruit blanc gaussien et additif (AWGN).

Ce modèle implique que le bruit du canal est une variable aléatoire n qui s'additionne au signal modulé transmis.

Cette partie est constituée sous la forme suivante :

**Canal de
propagation
(avec bruit)**

Dans ce cas, la variable n est gaussienne, de moyenne nulle et de variance σ^2 . Ce bruit est avant tout un modèle basé sur les deux suppositions suivantes:

- Tous les bits ont même probabilité, quelle que soit leur position.
- Les erreurs des différentes positions sont indépendantes

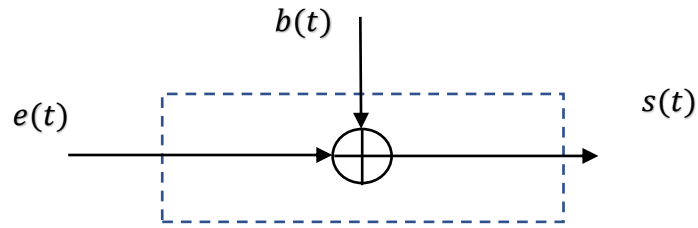


Figure 30 : Modèle de canal gaussien

La principale perturbation subie par le signal transmis est l'addition d'un bruit gaussien. Même en limitant la bande de fréquence du signal reçu, le récepteur doit fonctionner avec un signal utile entaché d'un signal perturbateur dont l'amplitude à un instant donné suit une loi de probabilité gaussienne.

Alors, plus que le bruit a plus de force, le travail du récepteur devient extrêmement délicat.

III. Partie réceptrice

Comme déjà vu que l'émetteur permet de transcrire le message en un signal a fin de le transmettre sur le canal. Inversement, le récepteur doit extraire le message du signal reçu. Pour cela, il procède soit de manière séquentielle en prenant une suite de décisions sur les symboles successifs du message émis dans le cas numérique.

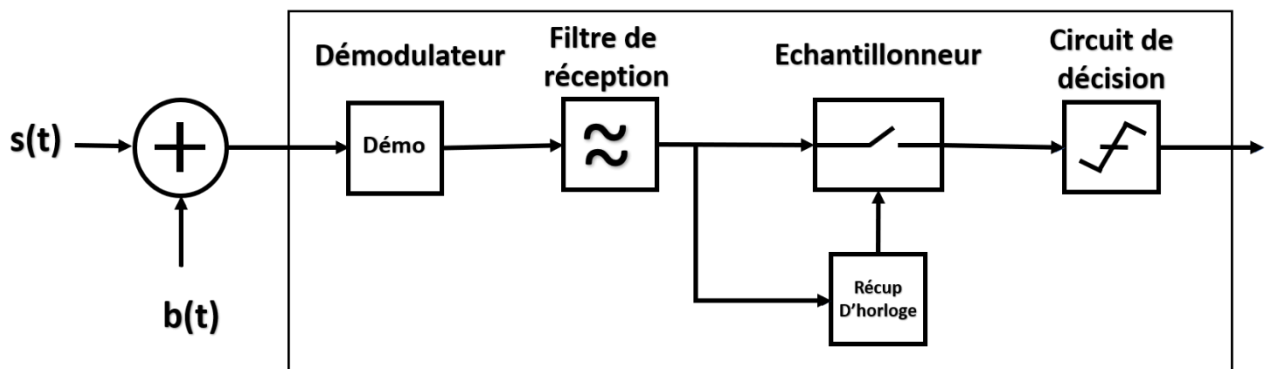


Figure 31: Partie réceptrice

En raison de la différence entre le signal émis et le signal reçu, le fonctionnement de la partie réceptrice est un peu complexe, Un bon fonctionnement du récepteur est lié à l'exploitation des connaissances a priori de la structure du signal émis, mais également des conditions dans lesquelles s'est déroulée la transmission.

Simulation avec interface graphique MATLAB (GUI)

A fin de Réaliser une application qui reflète le fonctionnement d'une chaîne de transmission numérique. Nous sommes demandés de concevoir une IHM facilitant l'utilisation de l'application. Pour ceci nous allons utiliser un environnement de développement d'interfaces graphiques de MATLAB.

1. Présentation de GUIDE

a. Définition

Le GUIDE ⁽⁴⁾ (Graphical User Interface en anglais) est un environnement de développement d'interfaces graphiques pour MATLAB. Les interfaces graphiques sont programmées de façon à répondre aux événements. Contrairement à la programmation traditionnelle, les événements générés par l'interface utilisateur (qui sont causés par l'interaction de l'utilisateur avec celui-ci) dirigent le comportement du programme.

b. Fonctionnement

Dans GUIDE, chaque événement généré par l'interface graphique génère un appel de fonction. Cette fonction s'appelle : « Callback ». Par exemple, un clic sur un bouton génère un événement qui appelle la fonction liée à ce bouton. La même source d'événements peut générer plusieurs types d'événements. Le bouton peut générer un événement lorsqu'il est appuyé, lorsqu'il est relâché et lors d'un clic. Cliquer sur un bouton génère ainsi les trois événements.

Le code du programme réside principalement dans les fonctions que les sources d'événements appellent. Le programmeur indique au GUIDE les événements pour lesquels il désire effectuer une réponse. GUIDE place une déclaration du Callback dans le fichier *.m pour chaque fonction. Le programmeur ajoute ensuite son code après la déclaration. Le code sera exécuté à partir de la déclaration jusqu'à la ligne précédant la prochaine déclaration.

Pour démarrer GUIDE, taper GUIDE dans la fenêtre de commande de MATLAB.

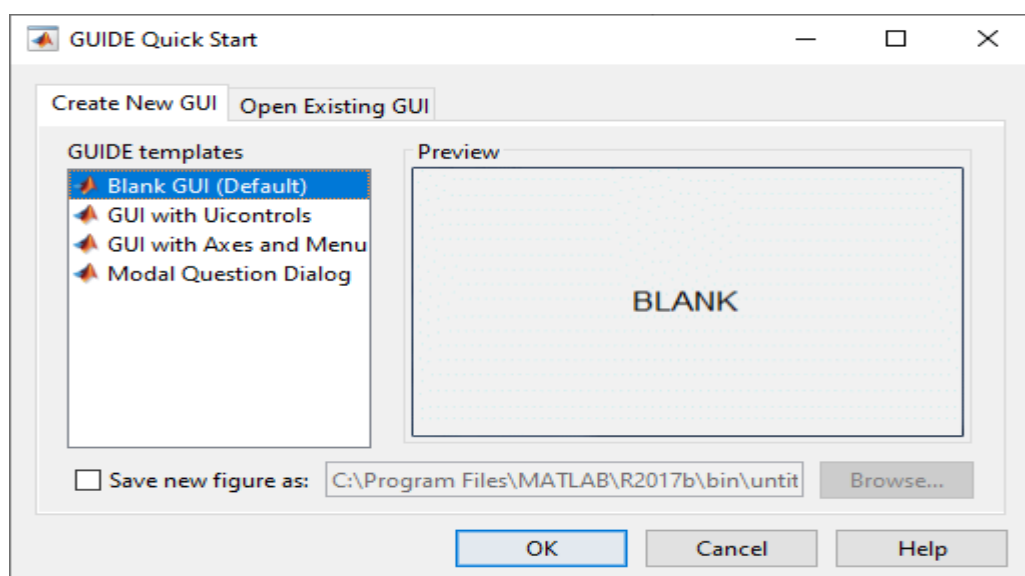


Figure 32 : Démarrage de GUIDE

Une fenêtre « GUIDE Quick Start » s'ouvrira vous donnant le choix entre quatre modèles. Choisir le premier et cliquer sur «Ok», car il n'est pas prédéfini et ne contient aucun objet.

2. Notre interface graphique de la chaine de transmission

Après la création d'une interface graphique de notre chaine de transmission numérique, elle sera de la forme suivante :

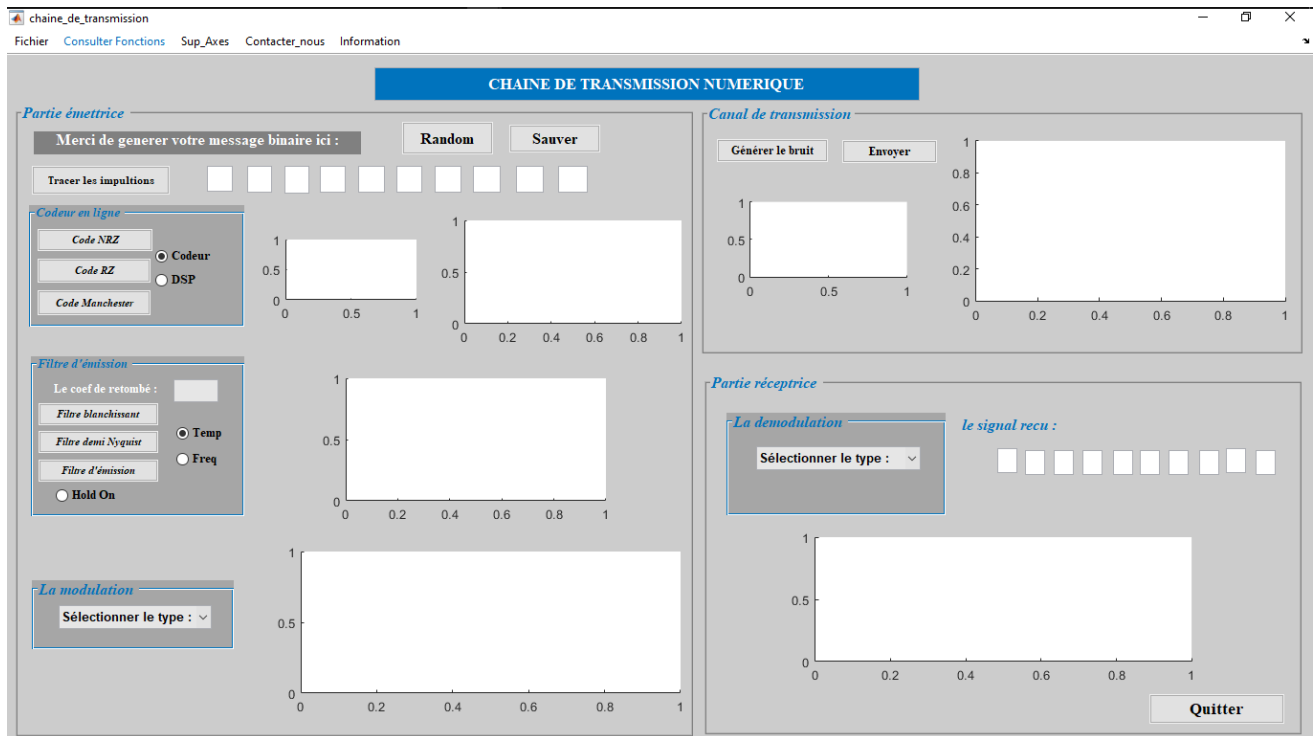


Figure 33 : Interface graphique de la chaine de transmission

3. Code Matlab utilisé

Vous trouverez le code MATLAB ⁽⁵⁾ de l'interface graphique dans le lien de drive suivants (avec tous les codes des fonctions utilisés) :

https://drive.google.com/drive/folders/1cZskRKlyms65q8YmF5ru_iJ2EQhrYZ6N?usp=sharing

4. Exemple de résultat trouve après l'exécution de l'interface

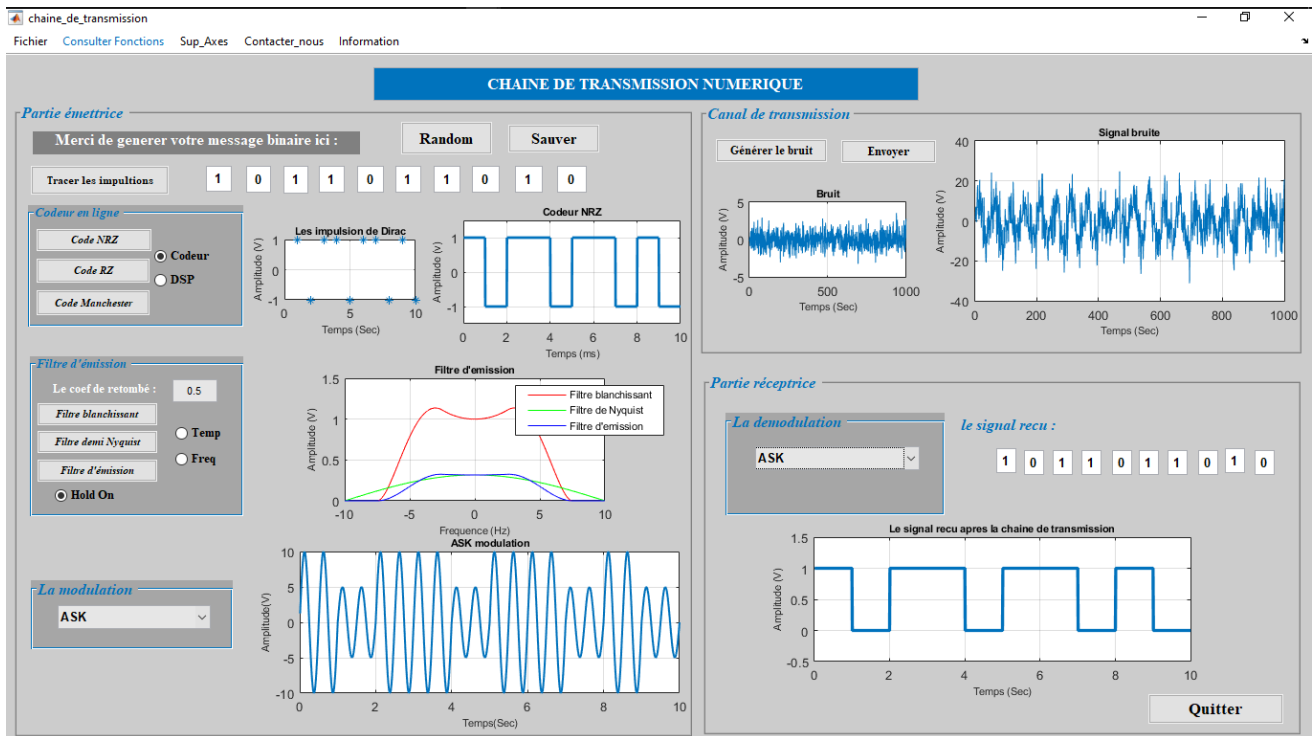


Figure 34 : Exemple de résultat trouve après l'exécution de l'interface

Références

1. **Julien, MICHOT.** *Les différents codages d'une information binaire.* Tours : Institut Universitaire de Technologie de Tours, 2004.
2. **Boyer, Alexandre.** *Canaux de transmission bruités.* Toulouse : Université de Toulouse, 2016.
3. **KADIRA, LHOUSAIN.** Modulation et. <https://www.lewebpedagogique.com>. [En ligne] 2014/2015. <https://www.lewebpedagogique.com/lhoussainkadira/files/2014/12/AM.pdf>.
4. **MatlabPourTous.** [En ligne] Paris, France, 2020. <https://matlabpourtous.com/matlab/les-interfaces-graphiques-sous-matlab/utilisation-de-guide/>.
5. **Amine, ABDALLAOUI MAAN.** Interface graphique GUI d'une chaîne de transmission numérique. *MIT-FSR*. Fes : s.n., 2020.