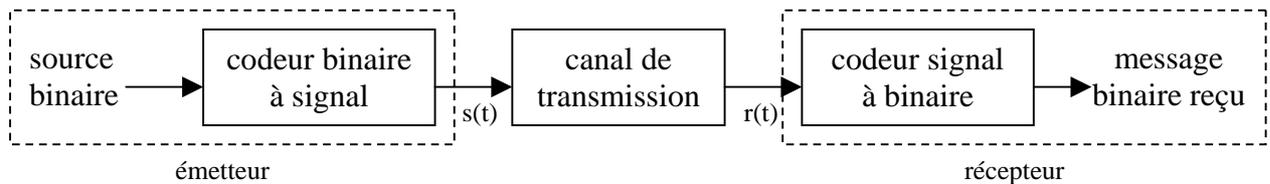


Travaux Pratiques de Théorie de l'Information : **Communication numérique****I. Transmission en bande de base**

La figure ci-dessous donne le schéma typique d'un système de communication numérique en bande de base.



Le message binaire issu de la source est transformé en une suite de signaux électriques de façon à « transcrire » le message numérique dans le domaine analogique. On appelle cette opération *codage binaire à signal*. La densité spectrale de puissance du signal ainsi obtenu est concentrée autour de la fréquence zéro, ce qui explique son utilisation sur les canaux de transmission de type passe-bas. Le récepteur doit extraire le message numérique du signal reçu, généralement déformé et bruité par le canal.

I.1. Codage à 2 niveaux

Un signal binaire est disponible dans le fichier *binnaire.mat*. Le codeur binaire à signal doit faire associer aux valeurs binaires 0 et 1 des signaux électriques (et analogiques) $m_0(t)$ et $m_1(t)$. Le signal émis dans le canal s'écrit :

$$s(t) = \sum_k m_\beta(t - kT_e)$$

où β vaut 0 ou 1 (suivant la valeur binaire) et T_e est le débit des symboles. Plusieurs schémas de codage sont envisageables suivant le choix des signaux $m_0(t)$ et $m_1(t)$.

- 1) Ecrire un programme Matlab permettant de générer le signal $s(t)$ en utilisant les codes unipolaires NRZ et RZ. Noter que nous simulons une transmission analogique à l'aide d'un système entièrement numérique. Le passage en temps continu est donc remplacé par une augmentation importante de la fréquence d'échantillonnage. Pour ce TP, on peut utiliser un facteur de suréchantillonnage 32.
- 2) Tracer les représentations temporelles et fréquentielles de chacun des deux signaux générés. Conclusion ?

I.2. Codage à 4 niveaux

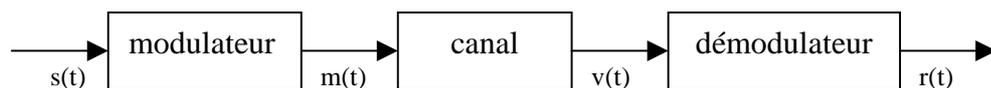
On souhaite maintenant coder le message binaire en regroupant les bits par 2 et en utilisant 4 niveaux d'amplitude pour représenter les 4 combinaisons possibles 00, 01, 10, et 11. On associe donc à chaque couple de bits successifs un signal électrique, $m_\beta(t) = a_\beta \Pi_{T_e}(t)$, $\beta=0,1,2,3$, où $\Pi_{T_e}(t)$ est une porte de largeur T_e . Cette association peut être effectuée en utilisant soit le code naturel, soit le code Gray (voir le tableau suivant).

Combinaison de bits	a_β (codage naturel)	a_β (codage Gray)
00	0	0
01	1	1
10	2	3
11	3	2

- 1) Quel est l'avantage du code Gray par rapport au code naturel ?
- 2) Ecrire un programme Matlab pour générer le signal $s(t)$ en utilisant le codage à 4 niveaux et chacun des codes naturel et Gray. Tracer les représentations temporelles et fréquentielles.
- 3) On va modéliser le canal de transmission par un générateur du bruit blanc, gaussien, et additif, $n(t)$. Le signal reçu en entrée du récepteur s'écrit alors $r(t)=s(t)+n(t)$.
 - (a) Pour chacun des signaux $s(t)$ obtenus avec le codage naturel et Gray, générer les signaux $r(t)$ correspondants.
 - (b) Ecrire un programme Matlab pour reconstituer les signaux binaires à partir des signaux $r(t)$.
 - (c) En comparant les signaux binaires reconstitués et le signal binaire d'origine, calculer le taux d'erreur pour différentes valeurs du rapport signal à bruit¹. Comparer les taux d'erreur des codages naturel et Gray. Conclusion ?

II. Transmission avec modulation

On va maintenant considérer la transmission du signal dans les canaux de type passe-bande. Cela nécessite un modulateur qui transforme le signal électrique passe-bas $s(t)$ en un signal passe-bande $m(t)$. Le signal modulé, déformé et bruité par le canal et reçu en entrée du récepteur, est ramené en bande de base par un démodulateur.



- 1) Effectuer les modulations ASK et PSK des signaux à 4 niveaux (codage naturel et Gray), générés dans la partie I.2.2, en utilisant les fonctions Matlab *askmod* et *pskmod* (voir annexe). Tracer les représentations temporelles et fréquentielles et commenter.
- 2) On va modéliser le canal de transmission par un générateur du bruit blanc, gaussien, et additif, $n(t)$. Le signal reçu en entrée du démodulateur s'écrit alors $v(t)=m(t)+n(t)$. Générer ce signal pour chacun des signaux modulés de la question précédente.
- 3) Démoduler les signaux $v(t)$, en utilisant les fonctions Matlab *askdemod* et *pskdemod* (voir annexe) pour obtenir les signaux $r(t)$ correspondants. Tracer les représentations temporelles et fréquentielles.
- 4) Reconstituer les signaux binaires à partir des signaux $r(t)$.
- 5) Calculer les taux d'erreur pour différentes valeurs du rapport signal à bruit. Conclusion ?

¹ Pour chaque valeur du RSB, répéter l'expérience une centaine de fois et calculer la moyenne du taux d'erreur.

III. Annexe

1) Fonction ASKMOD

```
[y, t]=askmod(x,Fe,F0)
% Modulation ASK (Amplitude Shift Keying)
%
% [y, t]=askmod(x,Fe,F0)
% x: signal d'entrée, en bande de base
% Fe: taux des symboles
% F0: fréquence de modulation
% t: temps
% y: signal modulé
%
% les symboles 0,1,2 et 3 sont respectivement associés aux amplitudes
% 0, 1, 2 et 3 d'un cosinus de fréquence F0
```

2) Fonction PSKMOD

```
[y, t]=pskmod(x,Fe,F0)
% Modulation PSK (Phase Shift Keying)
%
% [y, t]=pskmod(x,Fe,F0)
% x: signal d'entrée, en bande de base
% Fe: taux des symboles
% F0: fréquence de modulation
% t: temps
% y: signal modulé
%
% les symboles 0,1,2 et 3 sont respectivement associés aux phases
% 0,  $\pi/2$ ,  $\pi$  et  $3\pi/2$  d'un cosinus de fréquence F0
```

3) Fonction ASKDEMOMD

```
y=askdemod(x,Fe,F0)
% Demodulation ASK (Amplitude Shift Keying)
%
% y=askdemod(x,Fe,F0)
% x: signal d'entrée, passe-bande
% Fe: taux des symboles
% F0: fréquence de modulation
% y: Amplitude du signal demodulé et sous-échantillonné de Fe
```

4) Fonction PSKDEMOMD

```
y=pskdemod(x,Fe,F0)
% Demodulation PSK (Phase Shift Keying)
%
% y=pskdemod(x,Fe,F0)
% x: signal d'entrée, passe-bande
% Fe: taux des symboles
% F0: fréquence de modulation
% y: phase du signal demodulé et sous-échantillonné de Fe (entre  $-\pi$  et  $\pi$ )
```