
Travaux pratiques de Communications Numériques

ENSEIRB-MATMECA – T1

Benoît ESCRIG et Guillaume FERRE

COMPTE-RENDU : un compte-rendu est à rendre à l'encadrant, par voie électronique, au plus tard 15 jours après la dernière séance de travaux pratiques. Le format retenu est le format pdf. Le compte-rendu ne devra pas excéder 15 pages. Les deux codes MATLAB seront également joints au courrier électronique. La notation portera essentiellement sur les commentaires et les interprétations des différents résultats.

Programmation MATLAB

Les deux programmes MATLAB à rendre devront scrupuleusement respecter le canevas fournis à la fin de ce document.

Notion de fréquence normalisée et paramètres de simulation

Par défaut, la fréquence d'échantillonnage F_s du logiciel MATLAB est de un ($F_s=1$)¹. Ainsi, les quantités représentant des fréquences (ou des débits) devront être normalisées par rapport à la fréquence d'échantillonnage utilisée. Dans toute la suite, il ne sera fait référence qu'aux fréquences normalisées. De même, la période d'échantillonnage T_s du logiciel MATLAB est de un ($T_s=1$). Ainsi, les quantités représentant des temps, comme les périodes, seront normalisées par rapport à la période d'échantillonnage utilisée.

	Valeur réelle	Valeur normalisée
Fréquence d'échantillonnage	10 MHz	$F_s=1$
Période d'échantillonnage	0,1 μ s	$T_s=1$
Fréquence porteuse	2 MHz	$f_c=0,2$
Débit binaire	1 Mbit/s	$D_b=0,1$
Période binaire	1 μ s	$T_b=10$

Emetteurs numériques : modulations numériques

L'objectif de cette partie est de générer des signaux issus de modulations numériques (voir Figure 1) et de comparer les représentations en temps et en fréquence de plusieurs modulations.

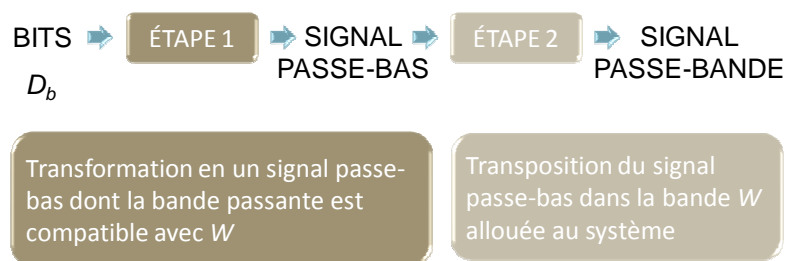


Figure 1 : génération de modulations numériques – schéma de principe

Cette partie commence par l'étude de la modulation de phase BPSK (Binary Phase Shift Keying). Puis, l'étude est généralisée aux autres modulations de phase, aux modulations d'amplitude et aux modulations d'amplitude en quadrature.

¹ Le « s » de F_s fait référence à « sampling » (échantillonnage).

Génération d'une modulation numérique de phase BPSK

La génération de la modulation se fonde sur le schéma de principe de la Figure 2.

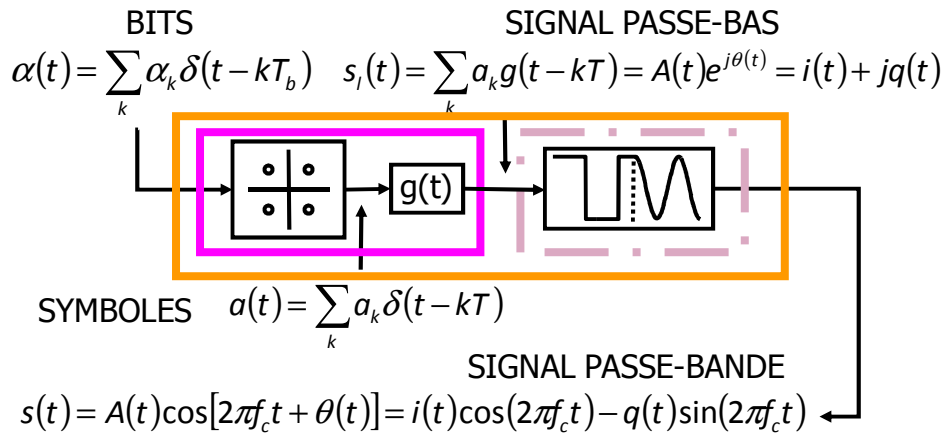


Figure 2 : génération de modulations numériques - schéma détaillé

Génération d'une séquence aléatoire de bits α_k

Générer une séquence aléatoire de N_b bits.

Nom de la variable : `bits`

Fonction MATLAB utilisée : `randint()`

Paramétrage : $N_b=100$

Conseil de programmation : compléter la partie initialisation au début du code MATLAB. L'objectif est de supprimer au maximum les valeurs numériques à l'intérieur du programme. Par exemple, pour générer la séquence de bits, ne pas écrire `randint(1,100)` mais plutôt `randint(1,Nb)` en définissant $Nb=100$ dans la partie initialisation.

Génération de symboles BPSK α_k

Coder la séquence de bits en une séquence de symboles BPSK, grâce à la règle de codage suivante : $\alpha_k=-1$ si $\alpha_k=0$ et $\alpha_k=+1$ si $\alpha_k=1$. Le cardinal de l'alphabet des symboles est, M . Dans le cas de la BPSK, il vaut deux ($M=2$). Représenter les symboles avec l'instruction `scatterplot()`.

Nom de la variable : `symbols`

Génération du signal $a(t)$

Générer un peigne de Dirac, de débit D , débit symbole, dont les amplitudes correspondent aux symboles BPSK.

Nom de la variable : `Dirac_comb`

Rappel : $D=D_b/\log_2(M)$.

Indication : générer un vecteur nul de durée NT où N représente le nombre de symboles, avec $N=N_b/\log_2(M)$, et T désigne la période symbole ($T=1/D$), avec l'instruction `zeros(1,NT)`. Puis, placer, à intervalles de temps réguliers T , les valeurs des symboles BPSK stockées dans `symbols` avec une instruction du type `Dirac_comb(1:T:NT)=symbols`.

Mise en forme des symboles BPSK :) et $s_i(t)$

Filtrer le peigne de Dirac par un filtre porte d'amplitude unité et de durée T pour générer le signal passe-bas.

Indication : la réponse du filtre à une impulsion sera stockée dans le vecteur g et vaut donc `ones(1, T)`.

Nom de la variable : $s1$

Fonction MATLAB à utiliser : `filter()`

Paramétrage : g

Transposition en fréquence : génération du signal $s(t)$

Transposer le signal autour de la fréquence porteuse f_c pour générer le signal en bande transposée.

Nom de la variable : s .

Indication : la méthode proposée repose sur les relations de la Figure 2. Dans l'instruction qui suit, le vecteur temps va de 0 à $NT-1$ mais il peut aller de 1 à NT , ou même de $T1$ à $T1+NT-1$ avec $T1$ entier. Le tout est de conserver la même convention tout au long du programme MATLAB.

```
s=real(s1).*cos_carrier-imag(s1).*sin_carrier où
  o cos_carrier=cos(2*pi*fc*[0:N*T-1]) et où
  o sin_carrier=sin(2*pi*fc*[0:N*T-1])
```

Remarque sur l'hypothèse de signaux à bande étroite : l'hypothèse de signaux à bande étroite (f_c très grande devant D) permet de faire un certain nombre de simplifications dans les calculs sur les signaux numériques. Ici, cette hypothèse n'est pas vérifiée. En effet, pour réduire la taille des vecteurs générés et ainsi réduire le temps des simulations, cette hypothèse a été abandonnée. Cependant, pour que les résultats établis en cours restent valides, cette hypothèse a été remplacée par une autre hypothèse, à savoir que $f_c=kD$, où k est un entier non nul.

Visualisation des résultats

Représentations en temps

Représenter les signaux suivants :

- Le signal passe-bande,
- Les parties réelles du peigne de Dirac des symboles et du signal passe-bas,
- Les parties imaginaires du peigne de Dirac des symboles et du signal passe-bas.

L'axe temporel sera exprimé en multiples de la période symbole T .

Conseil : prédire le résultat. Cela vaut pour tous les TPs. Lorsqu'il est demandé de tracer une courbe (représentation en temps, représentation en fréquence, courbe de BER), il faut toujours anticiper le résultat qui va être affiché. Ce n'est qu'en comparant le résultat obtenu avec le résultat escompté qu'il est possible de décider si le résultat de la simulation est correct ou non.

Indication : pour générer l'échelle d'un axe entre les valeurs $T0$ et $T1$ sur N_{samp} échantillons, il est possible d'utiliser `linspace(T0, T1, N_samp)`. L'avantage de cette instruction est qu'elle fonctionne aussi avec des valeurs négatives et des réels.

Indication : pour la représentation des peignes de Dirac, il est parfois intéressant de ne représenter que les valeurs non nulles et ce pour éviter de surcharger la figure. Lorsque les abscisses des valeurs non nulles d'un peigne de Dirac noté `vector` sont dans le vecteur `time`, il est possible de ne représenter que ces valeurs là

par l'instruction `plot(time,vector(time),'ro')`. Le marker choisi ici pour la représentation est un rond ('o') rouge ('r').

Consignes pour toute la suite des travaux pratiques : toujours donner une légende explicite aux figures. Différencier les courbes par un tracé différent (en prévision d'une impression sur une imprimante noir et blanc).

Représentation en fréquence

Tracer les trois densités spectrales de puissance correspondant aux trois signaux suivants :

- Le peigne de Dirac des symboles,
- Le signal passe-bas à la sortie du filtre de mise en forme,
- Le signal passe-bande à la sortie du modulateur.

Les densités spectrales de puissance seront tracées entre $-F_s/2$ et $F_s/2$ en utilisant une échelle logarithme². Expliquer la forme des densités spectrales de puissance.

Indication : la densité spectrale de puissance d'un signal donné est obtenue par le module au carré de sa transformée de Fourier (FFT pour Fast Fourier Transform)³, divisé par le nombre de points de la transformée de Fourier, noté `nfft`.

Indication : pour limiter les axes de la figure entre `xmin` et `xmax` pour l'axe des abscisses et `ymin` et `ymax` pour l'axe des ordonnées, il est possible d'utiliser l'instruction `axis([xmin xmax ymin ymax])`.

Rappel : le calcul de la FFT est optimisé pour un nombre de points de la FFT multiple d'une puissance de deux. Il est conseillé de choisir, comme nombre de points de la FFT, la puissance de deux immédiatement supérieure au nombre d'échantillons du signal. Pour cela, utiliser la fonction `nextpow2()`.

Compte-rendu : les figures correspondant aux représentations en temps et en fréquence, et les explications.

Généralisation à d'autres modulations linéaires

Adapter le précédent programme aux trois modulations suivantes : QPSK (Quaternary Phase Shift Keying), 4-PAM (Pulse Amplitude Modulation) et 16-QAM (Quadrature Amplitude Modulation). Expliquer les changements observés au niveau des représentations en temps et en fréquence des signaux par rapport à la modulation BPSK.

Compte-rendu : la comparaison des représentations en temps et en fréquence.

Récepteurs numériques

L'objectif de cette partie consiste à étudier le récepteur optimal pour les canaux AWGN (Additive White Gaussian Noise). Le récepteur est optimal au sens où il minimise le BER (Bit Error Rate). Dès lors que le canal n'est plus AWGN, le récepteur proposé n'est plus optimal (voir Figure 3).

² Le recentrage des densités spectrales de puissance se fait par la fonction `fftshift()`.

³ Sous MATLAB, l'échelle des fréquences par défaut va de 0 à 1 où 1 représente la fréquence d'échantillonnage.

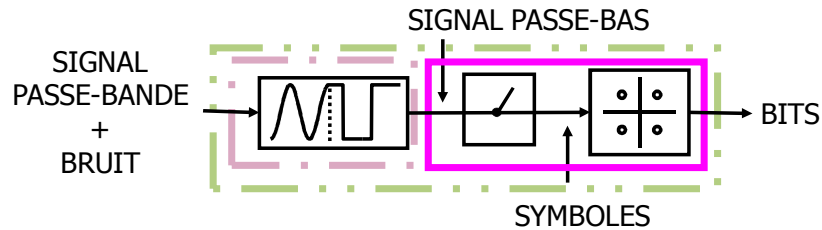


Figure 3 : réception de modulation numériques - schéma de principe

Cette partie commence par l'étude de la modulation de phase BPSK. Puis, l'étude est généralisée aux autres modulations de phase, aux modulations d'amplitude et aux modulations d'amplitude en quadrature.

Récepteur pour modulation BPSK

Retour en bande de base

Le retour en bande de base se fait sur deux voies. Le signal reçu est multiplié par deux porteuses pures de la forme $\cos[2\pi f_0 t + \varphi]$ et $-\sin[2\pi f_0 t + \varphi]$ où $f_0 = f_c$ et $\varphi = 0$. Pour les modulation BPSK, cela permet de traiter les modulations dont les symboles sont de la forme $\exp[j\vartheta]$ et $\exp[j(\vartheta + \pi)]$.

Filtrage adapté

La suppression des lobes secondaires en $-2f_c$ et $+2f_c$, et la maximisation du rapport signal à bruit aux instants de prise de décision, se font grâce au filtrage adapté. Filtrer le signal (après le retour en bande de base) par un filtre h adapté au filtre de mise en forme.

Nom de la variable : y

Fonction MATLAB à utiliser : `filter()`

Paramétrage : h

NB : si le filtre de mise en forme est de la forme $g(t)$, alors la réponse impulsionnelle du filtre adapté est de la forme $g^*(\tau - t)$ où τ est un retard servant à compenser les retards introduits par les différentes étapes de filtrage. La permutation de la réponse impulsionnelle du filtre de mise en forme se fait par la fonction `flipplr()`.

Sous-échantillonnage et prise de décision

Sous-échantillonner au rythme symbole T , le signal y à la sortie du filtre adapté et stocker les échantillons dans `dec_samples`. Représenter les échantillons avec l'instruction `scatterplot()`. Décoder les symboles reçus et stocker les bits estimés dans `estimated_bits`. Comparer les bits émis `bits` et les bits reçus `estimated_bits`. Conclure.

NB : tenir compte des retards introduits par le filtre de mise en forme et le filtre adapté (temps de propagation de groupe) dans le sous-échantillonnage. En particulier, le premier échantillon, étant donné la forme des filtres (porte de durée T), est prélevé à l'instant T .

Indication : un facteur $\frac{1}{2} E_g$ a été introduit par la transmission, où E_g désigne l'énergie du filtre de mise en forme. Ce facteur doit être pris en compte lors de la démodulation.

Visualisation des résultats

Représentations en temps

Superposer, sur la même figure, les signaux suivants :

- Le peigne de Dirac représentant les symboles émis,

- Le signal passe-bas à la sortie du filtre de mise en forme,
- Le signal passe-bas à la sortie du filtre adapté,
- Le peigne de Dirac correspondant aux échantillons prélevés toutes les périodes T .

L'axe temporel sera exprimé en multiples de la période symbole T . Une figure sera pour la partie réelle et une autre pour la partie imaginaire. Commenter.

Représentation en fréquence

Superposer, sur la même figure, les densités spectrales de puissance, entre $-F_s/2$ et $F_s/2$, du signal passe-bande émis et du signal à la sortie du filtre adapté. Commenter.

Compte-rendu : les représentations en temps et en fréquence, ainsi que les commentaires.

Réception des autres modulations linéaires

Adapter le programme précédent aux trois modulations suivantes : QPSK, 4-PAM et 16-QAM. Expliquer les différences et les similitudes avec le cas BPSK.

Compte-rendu : programme MATLAB correspondant à la chaîne de transmission de la QPSK, noté `QPSK_chain.m`, les figures et les explications.

Performances des transmissions numériques en présence d'un canal AWGN

L'objectif de cette partie est d'établir les performances des transmissions numériques en présence d'un canal AWGN. Dans un premier temps, les performances d'une transmission BPSK seront établies ; puis celles d'autres modulations linéaires.

Simulation d'un canal AWGN

L'hypothèse de canal AWGN consiste à supposer que l'ensemble des perturbations subies par le signal transmis par l'émetteur peut se modéliser sous la forme d'une seule et même source de bruit placée entre l'émetteur et le récepteur. L'ensemble des perturbations comprend notamment les imperfections des équipements électroniques d'émission et de réception (au niveau composants, connectique), les perturbations apportées par le milieu de transmission (câble, air, etc) et les perturbations ou interférences dues à la présence d'autres systèmes et/ou d'autres utilisateurs du système.

L'hypothèse de blancheur du bruit signifie que les échantillons de bruit ne sont pas corrélés les uns aux autres. Cette hypothèse est justifiée par le fait que la valeur d'un échantillon d'une source de bruit particulière n'est pas conditionnée par les valeurs des échantillons des autres sources de bruit. Par exemple, si les défauts de connectique de l'émetteur sont modélisés par une source de bruit, les valeurs aléatoires de cette source de bruit ne sont pas conditionnées par les valeurs de la source de bruit modélisant les perturbations liées au milieu de transmission, et inversement.

Test : tracer la fonction d'auto-corrélation d'un bruit blanc gaussien de moyenne nulle, de variance unité, de 1000 échantillons. Commenter.

L'hypothèse relative au caractère gaussien de la source de bruit vient du théorème de la limite centrale : toute combinaison de N variables aléatoires indépendantes et identiquement distribuées tend vers une loi gaussienne lorsque N tend vers l'infini. Il faut également tenir compte du fait que toute combinaison linéaire de variables aléatoires gaussiennes suit une loi gaussienne. Ainsi, lorsque toutes les sources de bruit sont référencées et classées selon leur densité de probabilité, il est possible d'obtenir, pour chaque classe, une

variable aléatoire globale suivant une loi gaussienne qui, une fois combinée aux variables aléatoires globales associées aux autres classes, donne également une loi gaussienne.

Test : tracer l'histogramme d'une variable aléatoire uniformément répartie sur (0,1) comprenant 1000 réalisations. Puis, tracer l'histogramme de la somme de 2, 3 et 10 variables aléatoires. Tracer l'histogramme d'un bruit AWGN. Commenter.

Compte-rendu : les courbes d'auto-corrélation, l'histogramme d'un bruit blanc et les commentaires.

Génération d'un bruit AWGN et caractéristiques du rapport E_b/N_0

Ajouter un bruit blanc gaussien à un signal QPSK de sorte que le rapport E_b/N_0 soit de 3 dB.

Indication : l'énergie moyenne transmise pour un symbole, E_s , est de la forme

$$E_s = \frac{E_g}{2} \sigma_a^2 \quad \text{avec} \quad E_g = \int_{-\infty}^{+\infty} |g(t)|^2 dt \quad \text{et} \quad \sigma_a^2 = E[|a_k|^2]$$

où E_g est l'énergie du filtre de mise en forme et σ_a^2 est la variance des symboles, à condition que ces derniers soient centrés et où les a_k sont les symboles de la constellation. L'énergie moyenne E_b d'un bit est de la forme $E_s/\log_2(M)$. Le bruit blanc gaussien doit être généré avec une variance $N_0/2$.

Indication : si E_b/N_0 dB désigne le rapport E_b/N_0 en décibels alors E_b/N_0 lin = $10^{(E_b/N_0\text{dB}/10)}$ représente le même rapport mais en échelle linéaire. Par la suite, en connaissant l'énergie moyenne transmise par bit, E_b , il suffit de calculer N_0 par $N_0 = E_b / E_b/N_0$ lin.

Représentation en temps

Sur une première figure, superposer le signal reçu sans bruit et le signal reçu avec bruit. Sur une autre figure, superposer le signal à la sortie du filtre adapté dans le cas sans bruit et dans le cas avec bruit. Tracer l'histogramme des échantillons aux instants de prise de décision. Commenter.

Représentation en fréquence

Sur une même figure, représenter les densités spectrales de puissance des trois signaux suivants : le signal émis, le signal reçu et le bruit. Commenter.

Recommencer les représentations en temps et en fréquence pour un rapport E_b/N_0 de 10 dB. Commenter.

Compte-rendu : les figures et les commentaires.

Performances d'une modulation BPSK dans un canal AWGN

Tracer la courbe donnant le BER en fonction du rapport E_b/N_0 , pour un rapport allant de 0 à 6 dB. Pour chaque valeur du rapport E_b/N_0 envoyer 1000 bits. Superposer à la courbe issue des simulations, la courbe théorique

$$P_b = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E_b}{N_0}}$$

Commenter. Modifier le code MATLAB pour obtenir des courbes complètes. Superposer aux courbes précédentes les courbes d'écart-type sur l'estimation de la probabilité d'erreur par le BER, c'est-à-dire $BER \pm \sigma_{BER}$. Commenter.

Indication : le BER est un estimateur de la probabilité d'erreur P_b avec les caractéristiques suivantes

$$m_{BER} = E[BER] = P_b$$

$$\sigma_{BER}^2 = E[BER^2] - m_{BER}^2 = \frac{P_b(1-P_b)}{N_b}$$

La précision de la mesure de la probabilité d'erreur est quantifiée par l'erreur relative quadratique moyenne, ϵ^2 , définie par

$$\epsilon^2 = \frac{\sigma_{BER}^2}{m_{BER}^2} = \frac{(1-P_b)}{N_b P_b} \quad \text{avec} \quad \epsilon^2 = \frac{1}{N_b P_b} \quad \text{si} \quad P_b \ll 1$$

Tracer la précision de la mesure en fonction du rapport E_b/N_0 .

Indication : la quantité $N_b P_b$ représente le nombre d'erreurs commises au cours de la simulation.

Le calcul du BER consistait jusqu'à présent à envoyer un nombre de bits arbitrairement grand pour toutes les valeurs du rapport E_b/N_0 et à observer un certain nombre d'erreurs. Ainsi, par rapport à l'expression suivante du BER

$$BER = \frac{N_e}{N_b}$$

où N_e représente le nombre d'erreurs observées, la procédure précédente consistait à fixer N_b et à compter N_e . La procédure va maintenant être modifiée. Au lieu de fixer N_b , c'est N_e qui va être fixé. Expliquer l'impact de cette procédure sur la précision de la mesure.

Etablir une nouvelle courbe de BER avec la méthode suivante. Pour chaque valeur du rapport E_b/N_0 , envoyer des blocs de N_b bits, avec N_b assez faible (exemple : $N_b=100$), et accumuler ces erreurs dans une variable intermédiaire N_e . Envoyer autant de blocs de N_b bits jusqu'à ce que N_e soit supérieur à un seuil $N_{e_{max}}$ fixé ici à 100 ($N_{e_{max}}=100$). Le BER estimé sera alors donné par

$$BER = \frac{N_e}{N_b \times N_{block}}$$

où N_{block} représente le nombre total de blocs de N_b bits envoyés.

Commenter les résultats et expliquer les avantages de cette méthode par rapport à la précédente en se fondant sur la notion de précision de la mesure. Tracer également les courbes BER +/- σ_{BER} où σ_{BER} est maintenant calculé en fonction de ϵ^2 . Expliquer et commenter.

Compte-rendu : les courbes de BER, les commentaires et explications.

Performances des autres modulations linéaires

Généraliser les résultats précédents aux cas d'une modulation QPSK, d'une modulation 4-PAM et d'une modulation 16-QAM. Visualiser la dispersion gaussienne des échantillons prélevés à la sortie du filtre adapté. Puis, tracer les courbes de BER. Tester les programmes avec et sans le codage de Gray. Commenter et conclure.

Compte-rendu : le programme de performances pour la modulation QPSK, nommé QPSK_perf.m, les courbes de BER et les commentaires.

Approfondissement

Certaines notions de communications numériques vont maintenant être approfondies par le biais d'expérimentations.

Impact de différents paramètres sur les représentations en temps et en fréquence des signaux

Pour la modulation BPSK, et pour chaque paramètre dans le tableau qui suit, changer la valeur du paramètre et expliquer les changements au niveau des représentations en temps et/ou en fréquence des signaux.

Paramètre	Valeur initiale	Nouvelle valeur
Filtre de mise en forme	ones(1,T)	sqrt(8.5/10)*[0:5,4:-1:1]
Nombre de bits émis	100	1000
Rayon de la constellation	1	2
Phase à l'origine	0	$\pi/4$
D_b	0,1	0,05
f_c	0,2	0,3
N_{fft}	$2^{\text{nextpow2}(N_{pts})}$ (i)	128

- (i) La formule permet de récupérer la puissance de deux immédiatement supérieure au nombre d'échantillons du signal, noté N_{pts} .

Compte-rendu : explications de l'influence des différents paramètres sur les représentations en temps et en fréquence des signaux⁴.

Estimation de la densité spectrale de puissance

Pour la modulation BPSK, moyenner 1 ($N_{stat}=1$) puis 100 ($N_{stat}=100$) estimations des densités spectrales de puissance pour les trois signaux suivants : le peigne de Dirac des symboles, le signal passe-bas à la sortie du filtre de mise en forme, le signal passe-bande à la sortie du modulateur. Tracer les densités spectrales de puissance estimées et les densités spectrales de puissance théoriques. Commenter.

Compte-rendu : les représentations en fréquence et les commentaires.

Comparaison des modulations : efficacité spectrale

En se fondant sur les représentations en fréquence, les modulations numériques sont maintenant comparées selon leur efficacité spectrale. Pour des valeurs de M de 2, 4, 8 et 16, donner l'occupation spectrale de la modulation numérique. En déduire l'efficacité spectrale et remplir le tableau suivant. Commenter et conclure sur l'efficacité spectrale des modulations numériques.

	M-PAM				M-PSK				M-QAM			
	2	4	8	16	2	4	8	16	2	4	8	16
Occupation spectrale												
Efficacité spectrale												

Compte-rendu : le tableau ci-dessus rempli, les commentaires et les conclusions.

⁴ Il est possible d'ajouter des figures pour illustrer une explication.

Impact de différents paramètres sur la réception des signaux numériques

Impact de la synchronisation

Changer la fréquence porteuse du récepteur. Pour cela, rajouter 1% de f_c à la valeur de f_c . Sur une même figure, superposer deux signaux : l'un correspondant à la sortie du filtre adapté pour une fréquence porteuse à la réception égale à f_c , et l'autre avec la nouvelle valeur de la fréquence porteuse. Commenter et conclure.

Changer la phase de la porteuse du récepteur. Faire varier la phase de 0 à $\pi/2$ par pas de $\pi/8$ (la valeur de la fréquence porteuse doit reprendre sa valeur initiale). Sur une même figure, superposer les signaux à la sortie du filtre adapté, correspondants aux quatre valeurs de la phase. Commenter et conclure.

Compte-rendu : les figures, les commentaires et les conclusions.

	Ancienne valeur	Nouvelle valeur
f_0	f_c	$f_c + 1\% \times f_c$
Φ	0	0, $\pi/8$, $\pi/4$, $3\pi/8$, $\pi/2$

Impact du filtre adapté

Des versions sous-optimales du récepteur vont maintenant être implantées, toujours pour une modulation BPSK, afin d'appréhender les éléments caractéristiques de l'architecture optimale.

Tester les combinaisons filtre de mise en forme – filtre adapté référencées dans le tableau ci-dessous. Commenter et conclure.

	Filtre à l'émission	Filtre à la réception
Filtre porte de durée T et d'amplitude unité	$g = \text{ones}(1, T)$	$h = \text{fliplr}(g)$
Filtre aléatoire	$g = \text{randn}(1, T)$	$h = \text{fliplr}(g)$
Filtre non adapté	$g = \text{ones}(1, T)$	$h = [0:T/2 \quad T/2-1:-1:1]$
Filtre non inversé	$g = \text{randn}(1, T)$	$h = g$

Compte-rendu : les commentaires et les conclusions.

Comparaison de l'efficacité en puissance des modulations numériques

Envoyer un signal tel que l'énergie moyenne émise par bit soit unité. Mesurer le BER pour un rapport signal à bruit de 3 dB et remplir le tableau suivant. Commenter et Conclure.

Modulation	M-PAM				M-PSK				M-QAM			
M	2	4	8	16	2	4	8	16	2	4	8	16
BER pour un E_b/N_0 de 3dB												

Compte-rendu : le tableau rempli, les commentaires et les conclusions.

Canevas pour le programme : transmission d'une modulation de phase à 4 états QPSK

```
%-----
% QPSK_chain.m
%-----
```

```

% ENSEIRB-MATMECA Dpt :
% Author :
% Year :
%-----
% QPSK Transmission
%-----
clear all
close all
%-----
% Initialization
%-----
fc=      % Carrier Frequency
Db=      % Bit Rate
Tb=      % Bit Period
Nb=      % Number of Bits
Fs=      % Sampling Frequency
M=       % Size of the Symbol Alphabet
n=       % Number of Bits per Symbol
D=       % Symbol Rate
T=       % Symbol Period
N=       % Number of symbols
g=       % Impulse Response of the Shaping Filter
h=       % Impulse Response of the Matched Filter
Eg=      % Energy of the shaping filter
cos_carrier=cos(2*pi*fc*[0:N*T-1]); % Cosine Carrier
cos_carrier=sin(2*pi*fc*[0:N*T-1]); % Sine Carrier
time_axis=      % Time Axis
nfft=          % Number of Samples for the FFT
freq_axis      % Frequency Axis
first_sample=T; % Delay at the Receiver Side
var_s=         % Variance of the Symbols
Es=           % Average Energy per Symbol
Eb=           % Average Energy per Bit
EbN0dB=       % Eb/N0 Ratio in dB
EbN0=         % Eb/N0 Ratio
N0=           % Noise Parameter
PSD_theo=     % Theoretical Power Spectral Density
              % of the Band-Pass Signal
%-----
% Begin Simulation
%-----
% Emitter
%-----
% Generating bits with randint()
% Generating symbols
% Generating a Dirac Signal
% Generating a Low Pass Signal
% Generating a Band Pass Signal
%-----
% Channel
%-----
% Receiver
%-----
% Recovering the base-band signal
% Mathed Filtering
% Sampling
% Received bits are compared to emitted bits
%-----
% End Simulation
%-----

```

```

%-----
% Figures
%-----

```

Canevas pour le programme des performances

```

%-----
% perf_QPSK_chain.m
%-----
% ENSEIRB-MATMECA Dpt :
% Author :
% Year :
%-----
% BER Performance of a QPSK Transmission
%-----
clear all
close all
%-----
% Initialization
%-----
fc=      % Carrier Frequency
Db=      % Bit Rate
Tb=      % Bit Period
Nb=      % Number of Bits
Fs=      % Sampling Frequency
M=       % Size of the Symbol Alphabet
n=       % Number of Bits per Symbol
D=       % Symbol Rate
T=       % Symbol Period
N=       % Number of symbols
g=       % Impulse Response of the Shaping Filter
h=       % Impulse Response of the Matched Filter
Eg=      % Energy of the shiping filter
cos_carrier=cos(2*pi*fc*[0:N*T-1]); % Cosine Carrier
cos_carrier=sin(2*pi*fc*[0:N*T-1]); % Sine Carrier
time_axis=      % Time Axis
nfft=          % Number of Samples for the FFT
freq_axis      % Frequency Axis
first_sample=T; % Delay at the Receiver Side
var_s=         % Variance of the Symbols
Es=           % Average Energy per Symbol
Eb=           % Average Energy per Bit
EbN0dB=       % Eb/N0 Ratio in dB
EbN0=         % Eb/N0 Ratio
N0=           % Noise Parameter
BER_theo_QPSK= % Theoretical BER Curve
BER=          % Iniatilize the BER vector to zero
nblock=       %Initializing the Number of Blocks
nblocklim=    % Maximum Number of Emitted Blocks
nerror=       % Maximum Number of Bits per Block
%-----
% Begin Simulation
%-----
% For each Eb/N0 Value
for
    % While the Maximum Number of Error has not been reached
    % or While the Maximum Number of Blocks has not been reached
while

```

```

%-----
% Perform the transmission of Nb-Bits Block
%-----
% Emitter
%-----
% Generating bits with randint()
% Generating symbols
% Generating a Dirac Signal
% Generating a Low Pass Signal
% Generating a Band Pass Signal
%-----
% Channel
%-----
%-----
% Receiver
%-----
% Recovering the base-band signal
% Mathed Filtering
% Sampling
% Received bits are compared to emitted bits
% Counting the Number of Errors
% Updating the Number of Blocksnbblock(k)
end
% End of the While Loop
end
% End of the "For each Eb/N0 Value" Loop
BER= % Normalizing the BER
%-----
% End Simulation
%-----

%-----
% Figures
%-----

```