

Implémentation d'une chaîne de communication à modulation de fréquence

François Horlin
Wireless Communications Group
Service OPERA

Table des matières

1 Chaîne de communication	1
2 Objectifs	2
2.1 Objectifs généraux	2
2.2 Objectifs en termes de connaissances	2
3 Etapes du projet	3
3.1 Implémentation de l'émetteur FM	3
3.2 Implémentation du récepteur FM	4
3.3 Simulation d'un canal idéal bruité	4
3.4 Modulation/démodulation d'une porteuse audio	5
3.5 Implémentation du récepteur FSK	6
3.6 Comparaison de performance	6
4 Annexes	7
4.1 Evaluation de la densité spectrale de puissance par la méthode de Welch	7
4.2 Conversion texte en séquence binaire	8
4.3 Conversion séquence binaire en texte	8
4.4 Conversion image en séquence binaire	8
4.5 Conversion séquence binaire en image	8
4.6 Génération d'un fichier audio	8
4.7 Enregistrement du signal audio	8

1 Chaîne de communication

La Figure 1 illustre la chaîne de télécommunications étudiée dans ce projet. L'objectif est de moduler une porteuse f_c (carrier) en fréquence en fonction de l'information binaire à transmettre et de récupérer cette information au récepteur. La performance de communication est dégradée par la présence de bruit blanc additif Gaussien qui corrompt le signal reçu. Deux types de récepteurs sont comparés : le discriminateur FM et le récepteur FSK par corrélation. Le passage bande de base/bande passante est explicitement implémenté dans le projet.

Il est intéressant de noter que les oscillateurs locaux chargés de générer les fréquences porteuses en émission et en réception n'ont pas de raison d'être synchronisés en phase à-priori. Pour cette

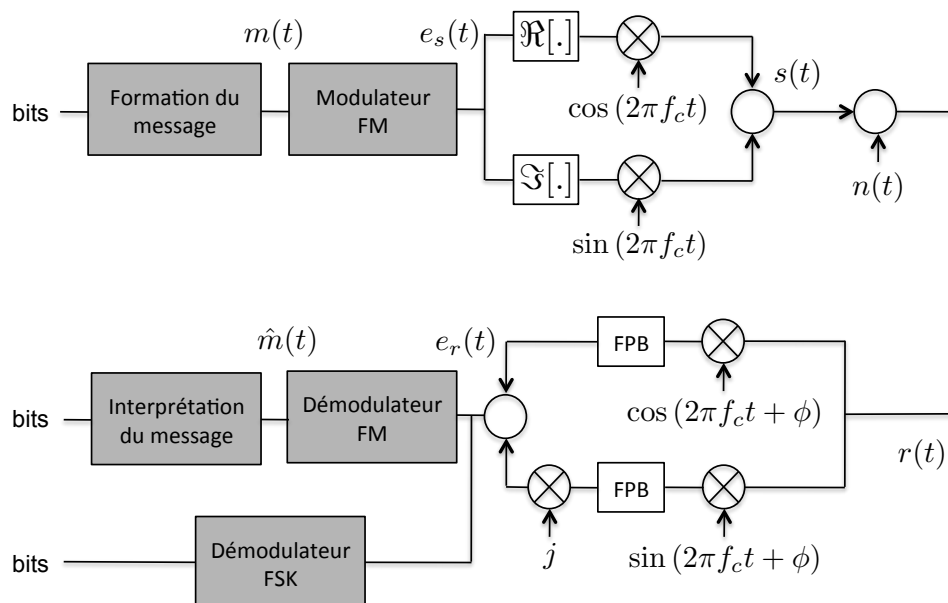


FIGURE 1 – Chaîne de communications FM et FSK

raison, une phase ϕ est introduite dans le modèle mathématique du récepteur. Les récepteurs proposés dans ce projet sont dits non-cohérents, c'est à dire qu'ils ne nécessitent pas d'estimer la phase du signal au préalable pour fonctionner.

La chaîne de communication est étudiée numériquement à l'aide du logiciel Matlab. Afin de rendre les choses plus concrètes, la séquence binaire est générée à partir d'une image et/ou d'un texte et une communication audio est mise en oeuvre. Le haut-parleur et le micro-phone intégrés sur vos ordinateurs peuvent être utilisés comme transmetteur/récepteur de signal audio.

2 Objectifs

2.1 Objectifs généraux

- Concevoir, analyser et mettre en oeuvre une chaîne de communication analogique FM (frequency modulation) et une chaîne de communication numérique FSK (frequency shift keying);
- Communiquer une image et un texte simultanément en modulant en fréquence deux porteuses audio.

2.2 Objectifs en termes de connaissances

- Réaliser la simulation d'une première chaîne de télécommunications avec le logiciel Matlab;
- Analyser les performances des communications analogiques et numériques, et comprendre l'intérêt de ces dernières;
- Se familiariser avec les notions fondamentales des systèmes de télécommunications, telles que la densité spectrale de puissance, la bande passante, la représentation des signaux en bande de base, le bruit blanc additif Gaussien, le récepteur par corrélation...

3 Etapes du projet

La réalisation du projet requière approximativement 6 séances de 4 heures. Afin que tous les groupes avancent à un rythme équivalent, nous proposons de baliser le projet selon six étapes principales.

3.1 Implémentation de l'émetteur FM

L'étape consiste principalement en la génération de l'enveloppe complexe du signal et en l'implémentation du passage bande de base/bande passante pour former le signal radio fréquence (RF) à émettre. Une analyse détaillée des signaux dans le domaine des fréquences est réalisée afin de déterminer leur occupation spectrale.

Pour réaliser cette étape :

- Générez une séquence binaire aléatoire à transmettre. Formez le message, l'enveloppe complexe du signal et le signal RF correspondants.

Le message analogique $m(t)$ est formé en fonction de la séquence binaire : pour transmettre un 0, le message envoyé est égal à -1 pendant T secondes ; pour transmettre un 1, le message envoyé est égal à $+1$ pendant T secondes (T est la durée d'un symbole, $F = 1/T$ est la fréquence symbole).

L'enveloppe complexe du signal émis (aussi appelée signal émis en bande de base) est fonction du message $m(t)$:

$$e_s(t) = \exp(j2\pi k_f \int_0^t m(\tau) d\tau)$$

où k_f est la sélectivité fréquentielle qui fixe la variation en fréquence par rapport à la fréquence porteuse en fonction du message.

Le signal RF (aussi appelé signal en bande passante) est généré à partir de l'enveloppe complexe en portant le signal à la fréquence f_c :

$$s(t) = \Re(e_s(t)) \cos(2\pi f_c t) - \Im(e_s(t)) \sin(2\pi f_c t)$$

- Évaluez la densité spectrale de puissance des différents signaux (message, enveloppe complexe, signal RF) avec la méthode de Welch détaillée en annexe. La densité spectrale de puissance est alors illustrée en échelle logarithmique pour pouvoir mettre les larges variations de puissance en fonction de la fréquence en évidence. Indiquez correctement les quantités mesurées sur les deux axes de vos figures afin qu'elles soient lisibles.
- Déduisez la bande passante du message et celle du signal transmis (comment la bande passante est-elle définie?).
- Répétez ces étapes en faisant varier les paramètres importants du système (en particulier F et k_f) et expliquez vos résultats.

Les paramètres du système donnés à la Table 1 sont proposés comme point de départ. Afin de réaliser la simulation à l'aide du logiciel Matlab, tous les signaux sont échantillonnés à la même cadence F_s choisie suffisamment grande pour éviter le repli spectral (la valeur proposée est-elle judicieuse?). La fonction Matlab permettant d'implémenter la méthode de Welch est donnée en annexe.

Paramètre	Notation	Valeur
Nombre de symboles émis	N	1000
Fréquence symbole	F	500 [Hz]
Sélectivité fréquentielle	k_f	500 [Hz]
Fréquence porteuse	f_c	6 [kHz]
Fréquence d'échantillonnage Matlab	F_s	20 [kHz]

TABLE 1 – Paramètres

3.2 Implémentation du récepteur FM

L'étape consiste principalement en l'implémentation du passage bande passante/bande de base et en la démodulation FM du signal. Dans cette étape, nous considérons que le bruit additif est négligeable de telle sorte que le signal reçu est exactement égal au signal envoyé. La séquence binaire est finalement récupérée pour valider la chaîne.

Pour réaliser cette étape :

- Générez la réponse impulsionnelle du filtre passe bas (FPB) utilisé dans la conversion du signal en bande de base. Par simplicité, considérez le filtre passe-bas idéal avec une fréquence de coupure variable et tronquez la réponse impulsionnelle de manière à ne perdre qu'une partie faible de l'énergie dans cette approximation. Discutez du choix de la fréquence de coupure du filtre.
- Implémentez la conversion du signal reçu en bande de base : multiplication par cosinus/sinus et convolution par le filtre passe-bas. Illustrez la densité spectrale de puissance du signal obtenu avant et après filtrage passe-bas et montrez que les répliques du signal à deux fois la fréquence porteuse sont éliminées par le filtre (d'où proviennent ces répliques?).
- Implémentez le discriminateur de fréquence présenté au cours (double circuit à pente) et vérifiez son fonctionnement en comparant le message reçu au message envoyé. Le plus simple est d'implémenter les équations temporelles du discriminateur de fréquence. Expliquez intuitivement pourquoi un circuit à pente permet de récupérer la fréquence instantanée du signal.
- Le message estimé au récepteur doit être interprété pour retrouver l'information binaire envoyée. A cette fin, le message est moyenné sur une durée symbole et l'information binaire est choisie en fonction du signe du résultat obtenu (pourquoi doit-on moyenner la signal sur la durée d'un symbole?). Retrouvez la séquence binaire de départ.

3.3 Simulation d'un canal idéal bruité

Un bruit blanc additif Gaussien (BBAG) corrompt le signal envoyé. En pratique, il est causé par l'ensemble des signaux perturbateurs captés par le récepteur et par les imperfections de fonctionnement des éléments électroniques constituant le récepteur. Dans cette étape, le bruit est ajouté à la chaîne Matlab. La figure de mérite est évaluée pour quantifier la capacité du démodulateur FM à gérer le bruit additionnel. L'impact du bruit sur la qualité de l'estimation de la séquence binaire peut être illustré.

Pour réaliser cette étape :

- Estimez d'abord la puissance du signal reçu à l'entrée du récepteur dans votre chaîne de simulation en moyennant le signal élevé au carré. L'énergie d'un bit E_b en est déduite en multipliant la puissance reçue par la durée du bit.
- Générez ensuite un vecteur de bruit blanc Gaussien à ajouter au signal reçu pour simuler les dégradations introduites par le système. Il est important d'adapter la puissance du bruit en fonction du rapport énergie d'un bit E_b sur densité spectrale unilatérale de puissance N_0 que vous souhaitez simuler. Un bon point de départ est $E_b/N_0 = 10$ [dB]. Dans votre simulation, la puissance du bruit est égale à la densité spectrale de puissance $N_0/2$ multipliée par la bande de fréquences sur laquelle le bruit est généré, ici la fréquence d'échantillonnage F_s (pourquoi ?). Déduisez finalement le rapport puissance du signal sur puissance de bruit (signal-to-noise power ratio, SNR) du canal.
- Évaluez le rapport puissance de signal sur puissance de bruit à l'entrée du démodulateur FM (SNR in) et à sa sortie (SNR out). Attention, le passage bande passante/bande de base est une opération linéaire tandis que le démodulateur lui-même est une opération non-linéaire. Pour calculer les puissances du signal et du bruit en sortie de démodulateur, il n'est donc pas correct d'introduire séparément le signal et le bruit dans le démodulateur pour en observer la sortie ! Expliquez votre démarche.
- Comparez le SNR en entrée de démodulateur au SNR du canal calculé précédemment. Expliquez le gain ou la perte de SNR.
- Comparez le SNR en entrée et en sortie de démodulateur FM. Calculez pour ce faire la figure de mérite (figure-of-merit, FoM) du récepteur définie comme le rapport entre les deux :

$$\text{FoM} = \frac{\text{SNR out}}{\text{SNR in}}.$$

Une figure de mérite plus grande signifie que le récepteur est plus à même de réduire l'impact du bruit sur la communication.

- Étudiez l'impact des paramètres du système sur la FoM. Illustrez en particulier la FoM en fonction du rapport E_b/N_0 pour différentes valeurs de sélectivité fréquentielle k_f et interprétez vos résultats.
- Illustrez l'impact du bruit sur le message et sur la séquence binaire estimés pour différentes valeurs du rapport E_b/N_0 .

3.4 Modulation/démodulation d'une porteuse audio

Une image et un texte sont transmis simultanément sur deux porteuses audio de fréquences différentes. Étant donné que nous réalisons ici une chaîne de communication réelle, il est inutile de rajouter un bruit supplémentaire au récepteur dans Matlab.

Pour réaliser cette étape :

- Choisissez une image noir et blanc de taille raisonnable (200 × 200 pixels par exemple) et un texte, et convertissez les en deux séquences binaires.
- À partir de chacune des deux séquences binaires, générez l'enveloppe complexe des signaux FM. Portez les deux signaux sur deux fréquences suffisamment espacées pour ne pas qu'ils interfèrent entre eux et additionnez les résultats pour former le signal RF à transmettre. Discutez du choix des paramètres de modulation (durée symbole, sélectivité fréquentielle, fréquence porteuse).
- Transmettez le signal audio correspondant à l'aide du haut-parleur d'un premier ordinateur et enregistrez le signal audio à l'aide du micro-phonie d'un second ordinateur.

- Convertissez le signal reçu en bande de base séparément pour le signal d'image et de texte en adaptant la fréquence porteuse et la bande passante du filtre passe-bas pour sélectionner chaque signal. Avant de pouvoir démoduler les signaux, il est nécessaire de synchroniser temporellement le récepteur sur le signal reçu. Le temps d'arrivée du signal est estimé en observant finement la puissance du signal reçu, éventuellement en échelle logarithmique pour mieux percevoir les variations de puissance. Démodulez les signaux pour retrouver les deux séquences binaires.
- Reformez l'image et le texte de départ. Il est intéressant de faire varier physiquement la distance entre les ordinateurs et les conditions de bruit environnant et d'évaluer leur impact sur l'image et le texte retrouvés.

Les fonctions Matlab permettant de convertir une image ou un texte en séquence binaire, et réciproquement, sont données en annexe. Les fonctions permettant de générer et de récupérer un signal audio à partir de Matlab sont aussi détaillées.

3.5 Implémentation du récepteur FSK

Cette étape consiste en l'implémentation du récepteur numérique optimal. Ce récepteur remplace le démodulateur FM implémenté lors de la deuxième étape.

Pour réaliser cette étape :

- Implémentez la conversion du signal reçu en bande de base : multiplication par cosinus/sinus et convolution par le filtre passe-bas. Cette étape est identique à celle réalisée à la deuxième étape.
- Implémentez le récepteur à corrélation. Chaque segment du signal reçu en bande de base $e_r(t)$ correspondant à un symbole binaire émis est corrélé avec les deux hypothèses de signal relatives à l'envoi d'un bit 0 ou 1 :

$$\begin{aligned} e_{s_0}(t) &= \exp(-j2\pi k_f t) \\ e_{s_1}(t) &= \exp(+j2\pi k_f t) \end{aligned}$$

Le bit sélectionné en observant le segment temporel $[kT, (k+1)T]$ est celui correspondant à un module de corrélation maximum :

$$\begin{array}{c} 0 \\ \left| \int_{kT}^{(k+1)T} e_r(t) (e_{s_0}(t))^* dt \right| > \left| \int_{kT}^{(k+1)T} e_r(t) (e_{s_1}(t))^* dt \right| \\ 1 \end{array}$$

- Vérifiez le fonctionnement du récepteur à corrélation en comparant la séquence binaire reçue à celle envoyée en l'absence de bruit.
- Vérifiez que la performance du récepteur est indépendante de la différence relative de phase ϕ entre les fréquences porteuses de l'émetteur et du récepteur et expliquez pourquoi. Ce n'est pas le cas du récepteur proposé dans le cours théorique qui suppose que la différence de phase a été précisément compensée de sorte que $\phi = 0$; on parle de récepteur cohérent dans ce cas.

3.6 Comparaison de performance

Les performances des récepteurs analogiques et numériques peuvent enfin être comparées en présence de bruit. La probabilité d'erreur binaire est évaluée pour chaque récepteur en calculant le taux d'erreur binaire (bit error rate, BER), donné par le nombre d'erreurs divisé par le

nombre de bits envoyés. La probabilité d'erreur est illustrée en échelle logarithmique (pourquoi?) en fonction du rapport E_b/N_0 exprimé en décibels.

L'impact du choix des paramètres du système donnés à la Table 1 est intéressant à étudier (en particulier F et k_f). Le nombre de symboles simulés N est aussi un paramètre de simulation critique (pourquoi?).

4 Annexes

4.1 Evaluation de la densité spectrale de puissance par la méthode de Welch

La méthode d'estimation de la densité spectrale de puissance des signaux est détaillée dans l'article 'The Use of Fast Fourier Transform for the Estimation of Power Spectra : A Method Based on Time Averaging Over Short, Modified Periodograms', publié par Peter D. Welch en 1967. L'idée générale est de découper le signal en un ensemble de segments, se recouvrant partiellement, de calculer la transformée de Fourier du signal sur chaque segment, et de moyenner le module carré sur chaque fréquence en fonction des segments émis. La fonction Matlab suivante implémente cette méthode :

```
function [psd,f] = welch(x,t,L,D)

% INPUTS :
% - x : input signal
% - t : time
% - L : segment size
% - D : delay between segments
% OUTPUTS :
% - psd : power spectral density
% - f : frequency

% Gather segments in a matrix
N = floor((length(x)-L)/D + 1);
Xt = [];
for nn = 1 : N,
    Xt = [Xt , x((nn-1)*D+1 : (nn-1)*D+L).'];
end;
% Multiply with window and compute FFT
W = 1 - ([0 : L-1]-(L-1)/2).^2 * 4/(L+1)^2;
Wt = repmat(W',1,N);
Yf = fftshift(fft(Xt.*Wt,L,1),1);
% Compute PSD
psd = sum(abs(Yf).^2,2)'/N;
psd = psd/max(psd);
f = [-L/2 : L/2-1]/L * 1/(t(2)-t(1));

end
```

4.2 Conversion texte en séquence binaire

```
text_tx = 'text';  
text_tx_int = uint8(double(text_tx)); % 8 bits/character  
text_tx_bin = de2bi(text_tx_int,8); % decimal to binary  
nbchar = size(text_tx_bin,1);  
bits_tx = double(reshape(text_tx_bin',1,nbchar*8));
```

4.3 Conversion séquence binaire en texte

```
text_rx_bin = reshape(bits_rx,8,nbchar)'; % nbchar : number of characters  
text_rx_int = bi2de(text_rx_bin)'; % binary to decimal  
text_rx = char(text_rx_int); % decimal to text
```

4.4 Conversion image en séquence binaire

```
image_tx = 'file'; % file.jpg  
image_tx_int = uint8(imread(image_tx,'jpeg')); % gray, 8 bits/pixel  
image_tx_bin = (image_tx_int>130); % black and white, 1 bit/pixel  
[nbl,nbc] = size(image_tx_bin);  
bits_tx = reshape(image_tx_bin,1,nbl*nbc);
```

4.5 Conversion séquence binaire en image

```
image_rx_bin = reshape(bits_rx,nbl,nbc); % nbl : number of lines, nbc : number of columns  
image_rx = image_rx_bin*255; % white : 255, black : 0  
figure(1); colormap gray; image(image_rx); % plot result
```

4.6 Génération d'un fichier audio

```
wavwrite(samples_tx,Fs,32,'file.wav'); % Fs : sampling frequency
```

4.7 Enregistrement du signal audio

```
r = audiorecorder(Fs,16,1);  
disp('Start recording')  
record(r);  
pause(70); % 70 seconds  
stop(r);  
disp('End recording')  
samples_rx = getaudiodata(r,'double');
```