



Rapport de Projet Traitement du Signal

Arthur Couturier
Pierre-Jean Coquard

Département Sciences du Numérique - Première année
Janvier 2021

Table des matières

1	Introduction	3
2	Génération du signal modulé en fréquence	4
3	Modem par filtrage	6
3.1	Introduction du modem par filtrage	6
3.2	Résultats expérimentaux obtenus	7
3.3	Densités spectrales de puissance	8
3.4	Détection d'énergie	9
4	Modem de fréquence V21, démodulateur FSK	10
4.1	Le démodulateur FSK	10
4.2	Le démodulateur FSK avec gestion d'une erreur de synchronisation de phase . . .	10
5	Annexes	12
5.1	Mode d'emploi du code Matlab	12
5.2	Preuve de la formule (3)	13

Table des figures

1	Signal modulé en fréquence	3
2	Densité spectrale de puissance du signal d'origine	4
3	Modulation/Démodulation par filtrage	6
4	Résultats de la modulation/démodulation par filtrage	7
5	Densités spectrales de puissance du signal modulé en fréquence et des filtres	8
6	Reconstitution de l'image donnée dans les 6 signaux binaires différents pour $SNR_{dB} = 20dB$	9
7	Démodulateur FSK avec synchronisation supposée idéale	10
8	Démodulateur FSK avec gestion d'une erreur de phase sur la porteuse	11

1 Introduction

Objectifs : Nos objectifs de projet étaient d'implanter sous matlab un modulateur/démodulateur (Modem) de fréquence dans le but de transmettre un signal sous forme de bits. Le modem à implanter devait suivre la recommandation V21 de l'UIT (Union Internationale des Télécommunication), modem avec support de ligne téléphonique avec un débit maximal de 300 bits par seconde.

Norme V21 : La modulation de fréquence de la recommandation V21 est dite à deux états (2-FSK pour 2-Frequency Shift Keying). Alors les bits 0 et 1 reçus sont transformés en cosinus et sinus de fréquences différentes afin d'obtenir la signal à émettre
Fréquence moyenne nominale utilisée :

$$F_c = 1080 \text{ Hz}$$

Excursion de fréquence utilisée :

$$\pm \Delta f = \pm 100 \text{ Hz}$$

On aura alors $F_0 = F_c + \Delta f$ correspondant à l'état binaire 0 et $F_1 = F_c - \Delta f$ correspondant à l'état binaire 1.

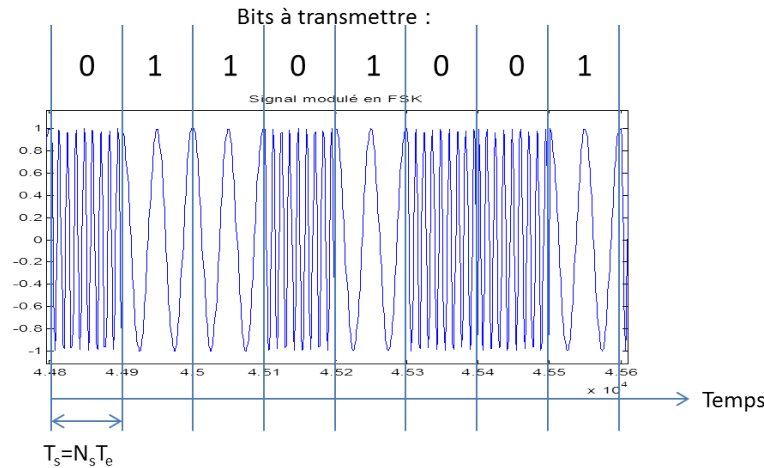


FIGURE 1: Signal modulé en fréquence

La *Figure 1* schématise la manière dont les bits sont retranscrit dans le signal modulé en fréquence. On remarque qu'on a bien $F_0 < F_1$. Les variables T_s , N_s et T_e seront explicitées dans la suite du rapport.

Déroulement de l'étude : L'étude débutera avec la formation du signal à émettre d'après un fichier d'informations binaires. Puis ce signal est abimé par un canal à bruit additif blanc et Gaussien. Par la suite nous simuleront une réception de ce signal abimé avec un modem de réception ayant pour objectif de retrouver l'information émise. Tout d'abord nous utiliserons un modem qui traitera le signal reçu par filtrage, puis une seconde version capable de travailler avec des fréquences proches car la première version nécessite de travailler avec des fréquences suffisamment éloignées pour ne pas être utilisables dans cadre de la norme V21.

2 Génération du signal modulé en fréquence

Comme expliqué en introduction, la première étape est de former un signal modulé en fréquence décrivant le signal binaire.

Considérons N_s le nombre d'échantillons distants de $T_e = 1/F_e$ la période d'échantillonnage. Alors chaque bit sera codé par un morceau de cosinus sur une période de $T_s = N_s.T_e$ secondes. Afin d'adapter le signal au premier récepteur que nous allons implanter, nous avons choisi $F_0 = 6000\text{Hz}$ pour le codage des bits 0 et $F_1 = 2000\text{Hz}$ pour le codage des bits 1. Par ailleurs, nous fixerons la fréquence d'échantillonnage $F_e = 48\text{kHz}$.

Le signal généré modulé en fréquence suivra donc l'équation suivante :

$$x(t) = (1 - NRZ(t)) \times \cos(2\pi F_0 t + \phi_0) + NRZ(t) \times \cos(2\pi F_1 t + \phi_1) \quad (1)$$

Avec $NRZ(t)$ un signal de type NRZ polaire, ϕ_0 et ϕ_1 des variables aléatoires indépendantes uniformément réparties sur $[0; 2\pi]$.

Nous avons tout d'abord généré NRZ d'après les bits à transmettre en configurant N_s tel que le débit de bits transmis respecte la norme V21 et soit donc bien de 300 bits par seconde. Nous avons ensuite déduit sa densité spectrale d'après la formule suivante :

$$S_{NRZ}(f) = \frac{1}{4}T_s \text{sinc}^2(\pi f T_s) + \frac{1}{4}\delta(f) \quad (2)$$

Nous avons ensuite généré ϕ_0 et ϕ_1 aléatoirement et avons alors construit $x(t)$ d'après la formule (1). On peut ensuite en déduire de l'expression la densité spectrale de x (détail du calcul en annexe) :

$$S_x(f) = (\delta(f) - S_{NRZ}(f)) * \frac{1}{4}(\delta(f - F_0) + \delta(f + F_0)) + S_{NRZ}(f) * \frac{1}{4}(\delta(f - F_1) + \delta(f + F_1)) \quad (3)$$

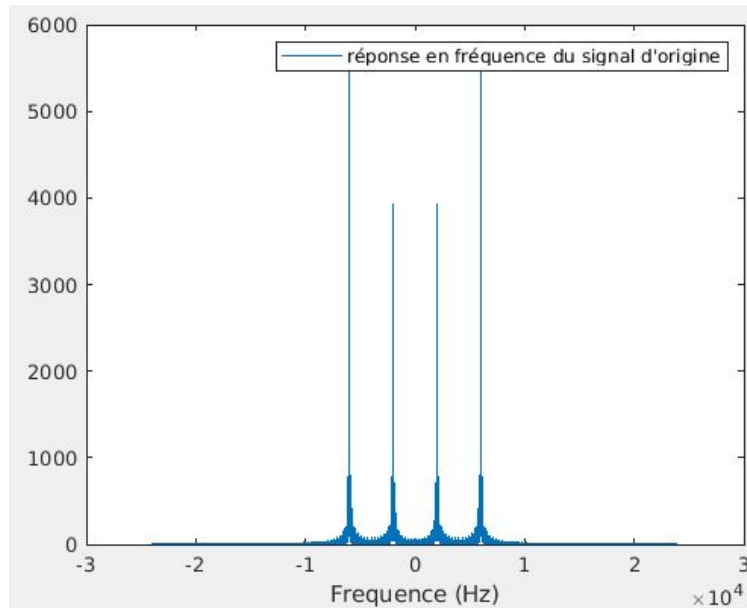


FIGURE 2: Densité spectrale de puissance du signal d'origine

La Figure 2 nous permet de constater des pics de fréquence à 2kHz et 6kHz. Ces pic correspondent à la somme de deux cosinus de fréquence F_0 et F_1 .

Nb : il est possible de visualiser les courbes $x(t)$ et $NRZ(t)$, $S_x(f)$ et $S_{NRZ}(f)$

Nous avons par la suite considéré que le canal de transmission ajoutait un bruit blanc Gaussien au signal émis afin de modéliser les perturbations. La puissance de ce bruit Gaussien était telle que le rapport signal/bruit vallait 50dB. Alors :

$$SNR_{dB} = 10 \log \left(\frac{P_x}{P_b} \right) = 50 \text{ dB} \iff P_b = P_x \times (-SNR_{dB}/10) \quad (4)$$

Avec P_x la puissance du signal modulé et P_b celle du bruit blanc Gaussien ajouté.

Par ailleurs nous avons tenté de faire varier SNR_{dB} et nos observations ont montré que lorsque $SNR_{dB} \geq 30 \text{ dB}$, alors le bruit était moindre et même quasi nul lorsque ce rapport dépassait les 40 dB.

3 Modem par filtrage

3.1 Introduction du modem par filtrage

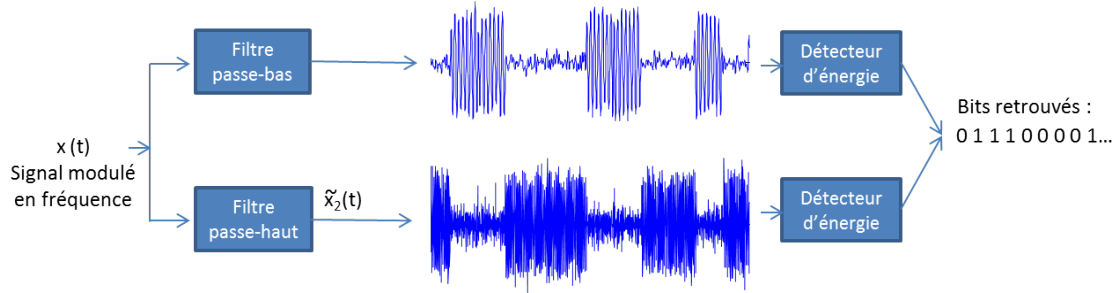


FIGURE 3: Modulation/Démodulation par filtrage

La *Figure 3* permet de schématiser ce que nous avons réalisé lors de la partie de modulation et démodulation par filtrage. Tout d'abord nous allons introduire deux filtres passe-haut et passe-bas afin de filtrer respectivement les morceaux de cosinus de fréquences $F_1 = 2000$ Hz et $F_0 = 6000$ Hz puis réaliser une détection d'énergie toutes les T_s secondes afin de retrouver les bits précédemment envoyés.

Nb1 : Il n'est pas nécessaire de faire l'introduction des deux filtres, mais nous avons souhaité montré que cette méthode n'est pas dépendante d'un certain type de filtre employé.

Nb2 : Nous noterons par la suite h_b et H_b le filtre passe-bas et sa transformé de Fourier ainsi que h_h et H_h le filtre passe-haut et sa transformé de Fourier. Par ailleurs, nous avons fixé leur ordre à 601

3.2 Résultats expérimentaux obtenus

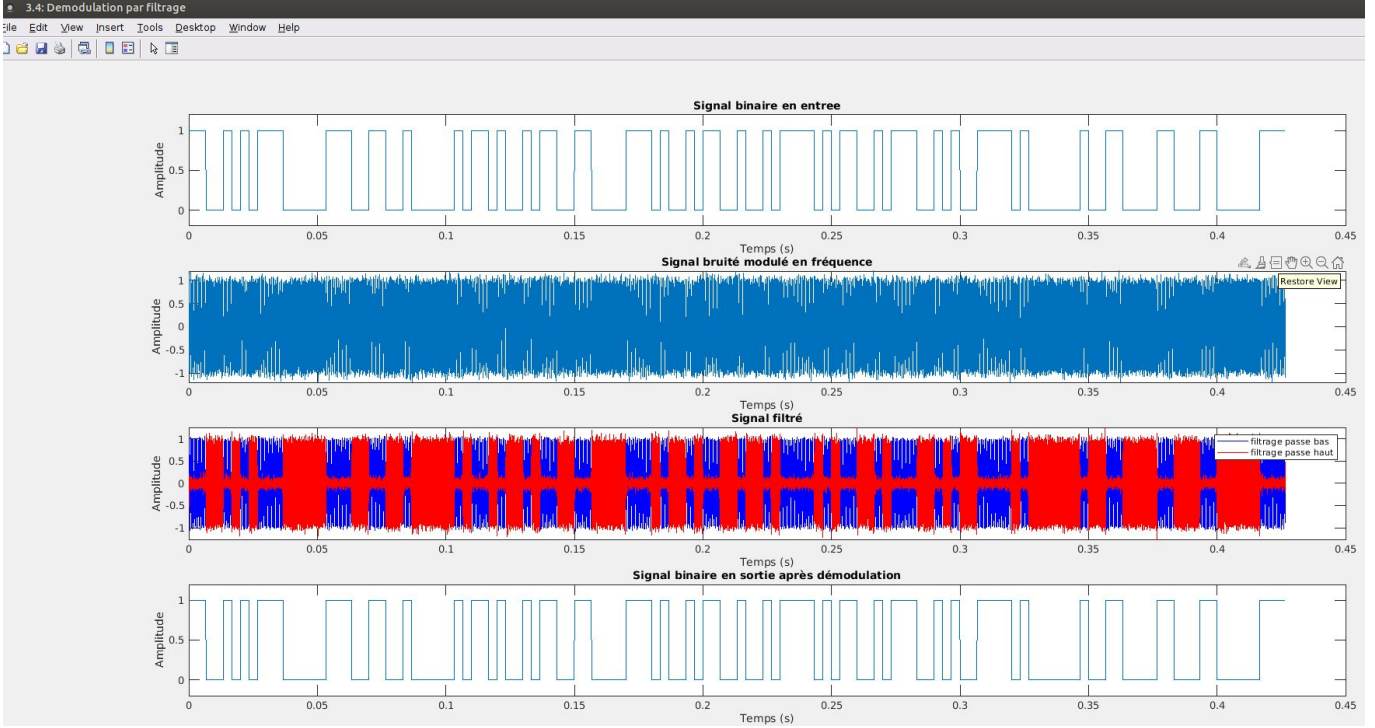


FIGURE 4: Résultats de la modulation/démodulation par filtrage

La *Figure 4* nous permet de mettre en avant les différentes étapes précédemment expliquées. Sur le premier graphique, on peut appercevoir un exemple de signal de 64 bits à envoyer.

Le second graphique nous permet de visualiser le signal modulé en fréquence auquel nous avons ajouté un bruit avec un rapport $SNR_{dB} = 20$ dB. Ce signal est celui effectivement transmis et que nous traitons par la suite afin de retrouver les bits émis.

La troisième graphique nous permet de mettre en avant la complémentarité des filtrages par filtre passe-bas en bleu et passe-haut en rouge. On construit d'abord la fonction de transfert du filtre passe bas. La transformée de Fourier d'un filtre passe bas idéal étant une porte de taille $fc = 4000Hz$ dans notre cas, la fréquence de coupure. Par transformée inverse on peut donc en déduire pour un filtre passe bas d'ordre quelconque :

$$\forall k \in \llbracket -(ordre - 1)/2, (ordre - 1)/2 \rrbracket h_b(k) = 2 \times fc \times Te \times sinc(2 \times fc \times Te \times k) \quad (5)$$

Pour le passe haut, on a :

$$H_h = 1 - H_b \iff h_h = \delta - h_b \quad (6)$$

On filtre ensuite le signal par un de ces filtres avec la fonction *filter*. On a choisis dans notre cas le passe haut pour la suite de la demodulation, mais on pourrait facielement effectuer la même démarche avec le filtre passe haut.

3.3 Densités spectrales de puissance

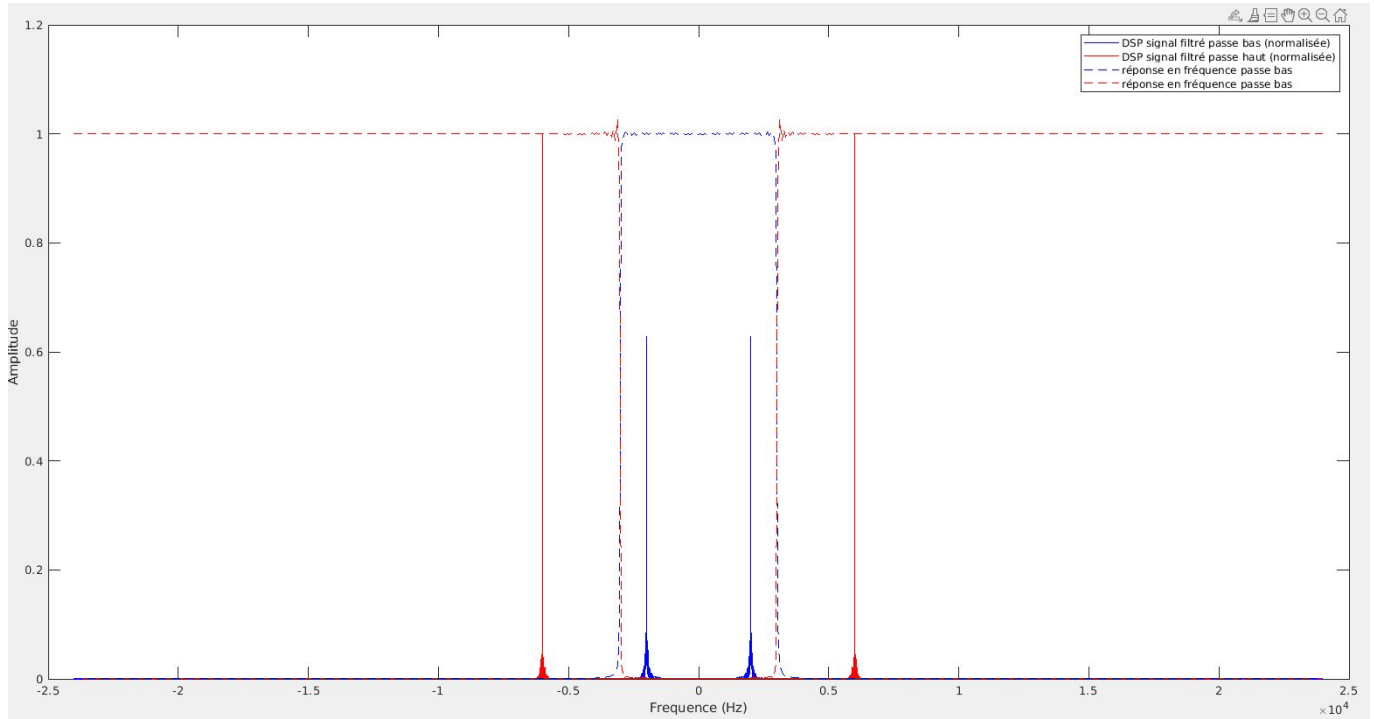


FIGURE 5: Densités spectrales de puissance du signal modulé en fréquence et des filtres

La *Figure 5* permet de visualiser sur un même graphique les densités spectrales de puissance du signal modulé (traits pleins) en fréquence par un filtre (bleu pour le passe-bas et rouge pour le passe-haut) ainsi que les réponses en fréquence de ces filtres. On remarque ici que le filtre pas-bas isole bien les morceaux de cosinus de fréquence 2kHz (bits 0) et que le filtre passe-haut isole quant à lui bien ceux de fréquence 6kHz (bits1).

Nb : Les amplitudes rendent compte du rapport de présence d'un bit plutôt que l'autre dans le signal émis.

3.4 Détection d'énergie

Une fois le signal filtré, pour chaque bit on somme chaque échantillon lui correspondant puis on compare cette somme à un seuil K . On a donc pour un bit :

$$\text{Si } \sum_{i=1}^{N_s} x_i^2 > K \text{ alors bit 1 sinon bit 0} \quad (7)$$

On a déterminé expérimentalement avec la fonction *estimer_seuil_experimental* que le seuil idéal dans notre cas se situe autour de $K = 30$.

le dernier graphique de la *Figure 4* montre le signal entièrement reconstruit sous forme de signal NRZ. On a observé un taux d'erreur binaire nul même pour des SNR_{dB} très faible ($SNR_{dB} < 1$).



FIGURE 6: Reconstitution de l'image donnée dans les 6 signaux binaires différents pour $SNR_{dB} = 20dB$

La *Figure 6* montre le résultat de la démodulation des signaux binaires du sujet une fois l'image reconstruite avec la fonction *reconstitution_image*. Cette image correspond à une des peintures murale des batiments de l'Enseeiht

Mais modifiant les fréquences pour correposndre à la norme V21, soit $F_1 = 980Hz$ et $F_0 = 1180Hz$, et en adaptant les filtres avec une fréquence de coupure $f_c = 1080Hz$ même pour un signal non bruité on obtient presque toujours un taux d'erreur binaire compris en 1% et 4%. Ce démodulateur n'a donc pas un pouvoir séparateur suffisant pour dissocier à coup sûr des signaux transmis sur deux bandes de fréquence aussi proches.

4 Modem de fréquence V21, démodulateur FSK

4.1 Le démodulateur FSK

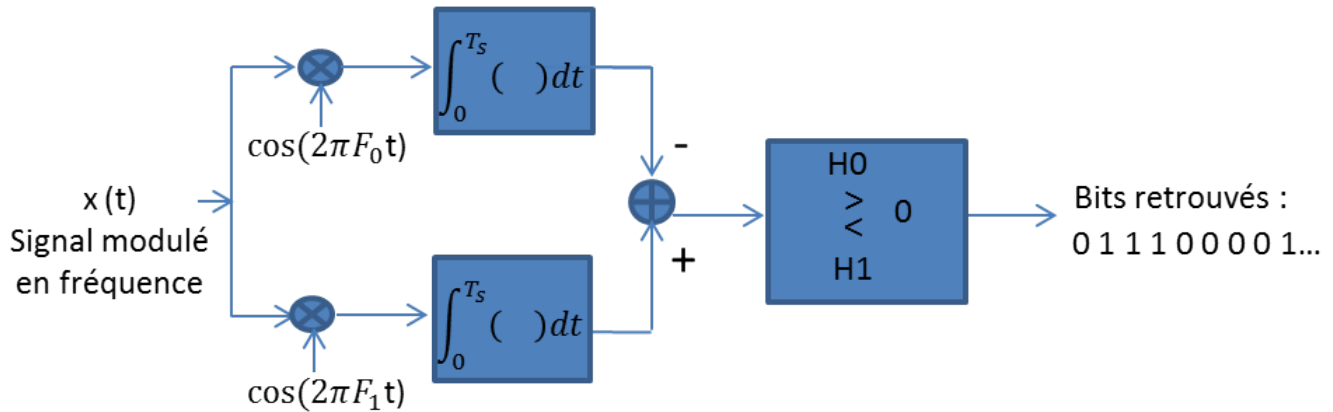


FIGURE 7: Démodulateur FSK avec synchronisation supposée idéale

La Figure 7 schématise le principe du démodulateur FSK. Ici, $x(t)$ représente le signal modulé et bruité que nous avons généré en partie 2. Ce signal est ensuite multiplié par un cosinus de fréquence F_0 ou F_1 (telles que décrites dans la norme V21) avant d'être intégré. La différence des résultats en sortie d'intégration nous permet d'obtenir un signal qui une fois normalisée permet de déduire un bit : si le résultat lors d'une estimation de bit est positif, alors le bit vaut 1. Sinon, il vaut 0. Nous avons ensuite comparé la séquence obtenue avec celle émise afin de déduire un pourcentage d'erreur. Sur tous les essais effectués, ce pourcentage est nul, ce qui montre que cette méthode est fiable.

Nb : l'intégrale sous matlab n'étant pas possible pour un nombre fini d'échantillons, nous avons en fait fait une somme sur N_s échantillons, le tout multiplié par dt la durée entre deux échantillons pour avoir le résultat.

On remarquera cependant que cette démodulation ne prend pas en compte une erreur de synchronisation entre l'émetteur et le récepteur. Nous traitons ce cas dans une dernière partie.

4.2 Le démodulateur FSK avec gestion d'une erreur de synchronisation de phase

En effet, ce problème de synchronisation de phase n'étant pas réalisable en pratique, nous impose de considérer que les phases ϕ_0 et ϕ_1 utilisées lors de la génération du signal ne sont pas les mêmes que lors de la démodulation. Car en changeant ces phases dans l'étape de démodulation,

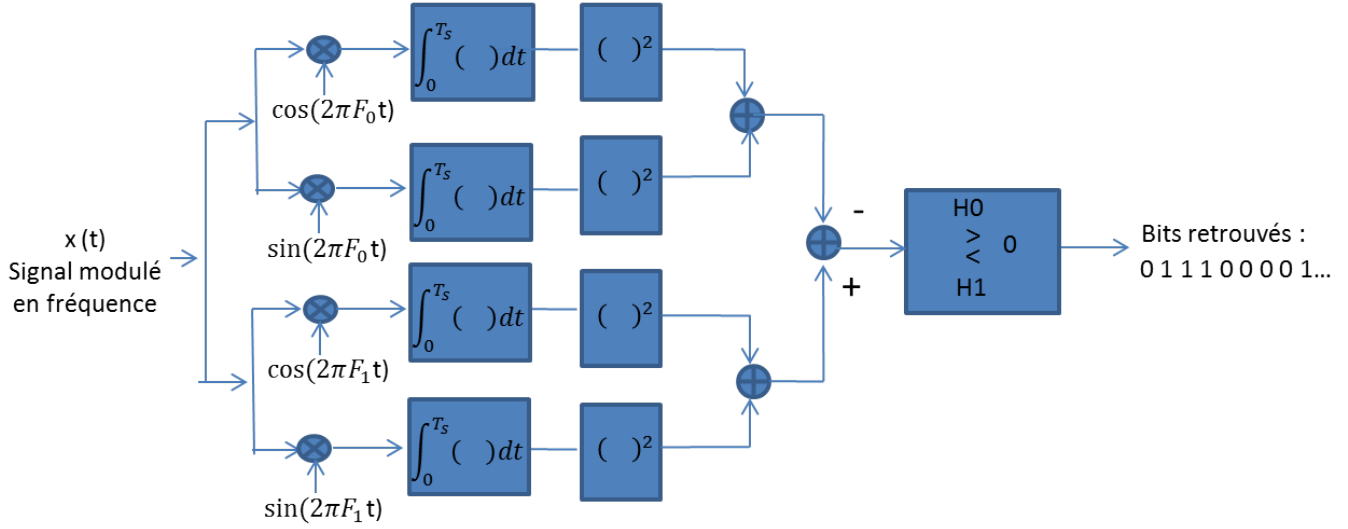


FIGURE 8: Démodulateur FSK avec gestion d'une erreur de phase sur la porteuse

La *Figure 8* schématise la démodulation FSK avec cette gestion d'erreur. En effet, précédemment, avec l'émetteur et le récepteur en phase, la multiplication avec un cosinus puis l'intégrale permettait la détection d'un bit ou l'autre.

Dans ce modèle nous multiplions le signal modulé par un cosinus et un sinus de même fréquence. La somme du carré des intégrales va permettre de normaliser la détection car le cosinus et le sinus sont en opposition de phase. Ce qui va différer sur la somme du carré des intégrales pour l'une ou l'autre des deux branches du schéma va être la différence de fréquence dans les cosinus et sinus introduits. Ainsi nous avons pu détecter si le signal à un moment donné transmettait un bit 1 ou 0.

Nous avons enfin testé ce démodulateur en introduisant de manière volontaire des erreurs de phase afin de simuler ce décalage. Nos résultats ont montré un pourcentage d'erreur nul, tout comme lorsque nous supposons que le modulateur et le démodulateur étaient en phase.

5 Annexes

5.1 Mode d'emploi du code Matlab

Le Code matlab est composé de plusieurs modules. Les fichiers à exécuter contenant les exercices sont `Demodulation_filtrage.m` et `Demodulation_FSK.m`.

Tout d'abord le fichier `donnees.m` contient toutes les données constantes utiles au projet dans le but des les rendre plus facilement accessibles pour exécuter le code avec des paramètres différents. Il contient notamment des paramètres alternatif que l'on peut utiliser en décommentant des parties de code. Il est possible d'utiliser les signaux fournis dans le sujet à la place du signal aléatoire en décommentant les lignes 24 et 25 ainsi qu'une des lignes de 18 à 23 suivant quel signal on veut charger. De plus, les fréquences F_0 , F_1 et f_c sont définies par les valeurs données pour la démodulation par filtrage, mais on peut facilement passer à la norme V21 en décommentant les lignes 47 à 49.

Le fichier `Demodulation_filtrage.m` contient le code et les tracés des exercices jusqu'à la fin de l'exercice 3. Il utilise le fichier `donnees.m` et trace les courbes demandées. Il s'appuie sur la fonction `construire_NRZ.m` pour construire les courbes NRZ à partir d'un signal binaire. Il utilise aussi la fonction `estimer_seuil_experimental` pour donner une approximation à 10^{-2} près du seuil de détection d'énergie optimal pour avoir un taux d'erreur binaire le plus faible possible pour le signal binaire en entrée. Enfin il utilise aussi la fonction `afficher_resultat` qui affiche 4 courbes différentes : le signal NRZ en entrée, le signal bruité modulé en fréquence, le signal bruité modulé en fréquence filtré et enfin le signal NRZ. Ces trois fonctions sont définies et implémentées dans leur fichier `.m` respectif. Enfin, le taux d'erreur binaire de la démodulation est affiché dans le terminal.

Le fichier `Demodulation_FSK.m` contient la partie 4 du sujet et utilise les résultats calculés avec le fichier `Démodulation_filtrage.m`. De la même manière que pour la démodulation par filtrage, l'exécution de ce fichier affiche les courbes de résultat et les taux d'erreur binaire pour les parties 4.1 et 4.2

5.2 Preuve de la formule (3)

D'après (1) on a : $x(t) = (1 - NRZ(t)) \times \cos(2\pi F_0 t + \phi_0) + NRZ(t) \times \cos(2\pi F_1 t + \phi_1)$

$$\Leftrightarrow Rx(\tau) = (1 - R_{NRZ}(\tau)) \times F_0 \int_{\frac{1}{-2F_0}}^{\frac{1}{2F_0}} \cos(2\pi F_0 \tau + \phi_0) \cos(2\pi F_0(t - \tau) + \phi_0) dt + R_{NRZ}(\tau) \times F_1 \int_{\frac{1}{-2F_1}}^{\frac{1}{2F_1}} \cos(2\pi F_1 \tau + \phi_0) \cos(2\pi F_1(t - \tau) + \phi_0) dt$$

car les fonctions NRZ et cosinus sont indépendantes

$$\Leftrightarrow S_x(f) = TF[R_x(\tau)]$$

$$\Leftrightarrow S_x(f) = (\delta(f) - S_{NRZ}(f)) * \frac{1}{4}(\delta(f - F_0) + \delta(f + F_0)) + S_{NRZ}(f) * \frac{1}{4}(\delta(f - F_1) + \delta(f + F_1))$$