



Département Sciences du Numérique

Traitement du signal

Projet : Projet Simulation d'une transmission au format DVB-RCS Voie retour
par satellite

Rapport

Nathan Foucher

Groupe 1-SN-C

Martin Guidez

Résumé

Ce rapport a pour but d'expliquer le sujet et les objectifs à atteindre demandés. Une synthèse des résultats obtenus sera présentée en fin de document. Nous énoncerons aussi dans ce document les questions et interrogations que notre équipe s'est posée durant la réalisation du projet ainsi que les solutions retenues pour y faire face.

Table des Matières

I. Début du projet

1. État de l'art
2. Attendus du projet

II. Travail sur le projet

1. Construction du signal MF-TDMA à décoder
 - 1.1. Étude théorique
 - 1.2. Implantation
 - 1.2.1. Modulation bande base
 - 1.2.2. Construction du signal MF-TDMA
2. Mise en place du récepteur MF-TDMA
 - 2.1. Démultiplexage des porteuses
 - 2.1.1. Synthèse du filtre passe bas
 - 2.1.2. Synthèse du filtre passe haut
 - 2.1.3. Filtrage
 - 2.2. Détection du slot utile
 - 2.3. Démodulation bande de base

III. Synthèse et conclusion

Introduction

La transmission de données par satellite est utilisée dans plusieurs situations de la vie quotidienne. On pensera notamment à la transmission des flux vidéo TV pour les abonnés à la télé satellite. Ici le flux ne va que dans un sens, de la chaîne TV vers l'abonné. Dans ce sujet on parlera de transmission pour les accès bidirectionnels, plusieurs standards permettent de le mettre en place. Nous nous concentrerons sur le standard DVB-RCS (Digital Video Broadcasting - Return Channel via Satellite).

I. Début du projet

1.1 État de l'art

Le standard DVB-RCS permet d'offrir aux utilisateurs une connexion bilatérale à l'aide d'une voie retour par satellite : lien entre leur terminal et une station d'interconnexion via une liaison satellite. La station d'interconnexion joue le rôle d'interface avec le monde de l'Internet et du réseau pour offrir des applications diverses telles que la navigation sur le WEB, la messagerie ou le transfert de données. La figure 1 schématise son fonctionnement.

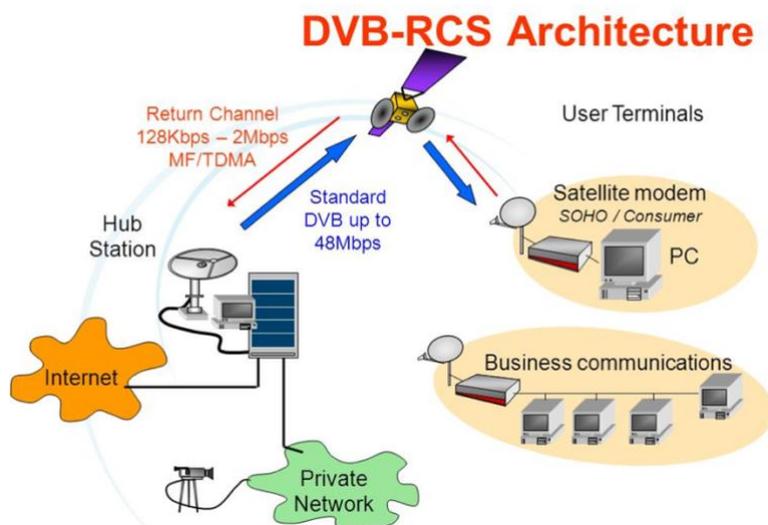


Figure 1: Fonctionnement du standards DVB-RCS

Les utilisateurs peuvent accéder à différents services en utilisant une trame MF-TDMA permettant de partager les ressources satellites entre les utilisateurs. Chaque utilisateur a une fréquence porteuse allouée qu'il utilisera sur sa trame.

1.2 Attendus du projet

On simulera dans ce projet l'utilisation de ces trames pour deux utilisateurs. Ainsi nous allons générer le signal porteur, les trames et communiquer les donner vers un récepteur aussi codé par nos soins. Nous avons deux fichiers de données utilisateurs à disposition, le but est de simuler leurs transmissions et de les décoder en bout de chaîne. Ces deux utilisateurs voudront accéder à la même station d'interconnexion. Pour tout ce TP la fréquence d'échantillonnage des données est fixée à $f_e = 120kHz$. Nous avons les fréquences porteuses suivantes :

- $f_{p_1} = 0 Hz$ Pour l'utilisateur 1 associé au slot 2
- $f_{p_2} = 46 kHz$ Pour l'utilisateur 2 associé au slot 5

Remarquons que la fréquence d'échantillonnage est bien adaptée pour respecter le critère de Shannon.

Les signaux de l'utilisateur 1 (respectivement de l'utilisateur 2) seront donc modulé par le signal binaire des informations transmises ($m_1(t)$ et $m_2(t)$)

Les données seront transmises sur une trame de 5 slots de 40ms chacun (voir figure 2). Pour s'approcher d'un cas d'application réel nous ajouterons un bruit gaussien sur le signal transmis.

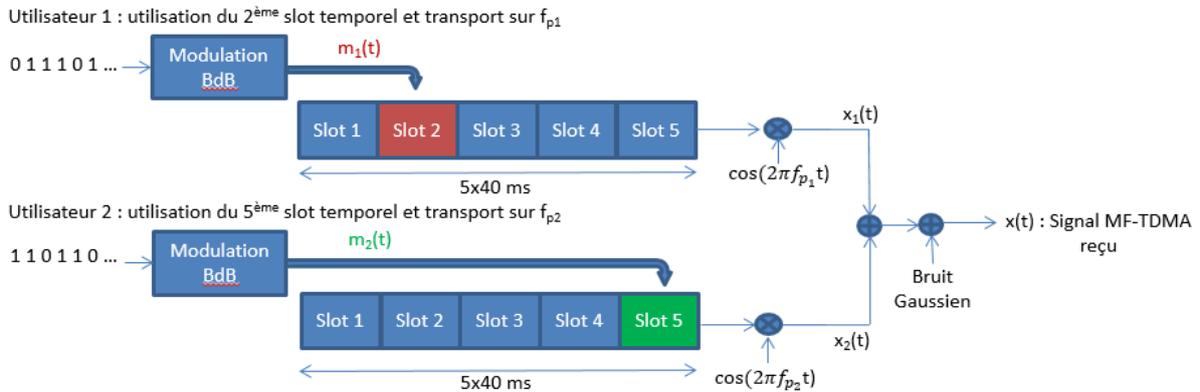


Figure 2 Schéma de la structure du signal

II. Travail sur le projet

2.1. Construction du signal MF-TDMA à décoder

2.1.1 Étude théorique

Le signal avant ajout du bruit est donc de la forme suivante :

$$x(t) = m_1(t) \cos(2\pi f_{p1} t + \varphi_1) + m_2(t) \cos(2\pi f_{p2} t + \varphi_2)$$

Comme $f_{p1} = 0$ Hz, on obtient :

$$x(t) = m_1(t) + m_2(t) \cos(2\pi f_{p2} t + \varphi_2)$$

On calcul la densité spectrale des signaux de deux façons différentes, en considérant m_1 et m_2 comme signaux aléatoires (a) puis comme déterministes (b), on obtient :

-(a) On a :

$$S_x(f) = TF[R_x(T)]$$

$$\text{Or } R_x(T) = E[x(t) \times x^*(t - T)] = R_{m1}(T) + R_{m2}(T) \times \frac{1}{2} \cos(2\pi f_{p2}T)$$

$$\text{Puis } TF[R_x(T)] = S_{m1}(f) + \frac{1}{4}S_{m2}(f - f_{p2}) + \frac{1}{4}S_{m2}(f + f_{p2}) = S_x(f)$$

-(b) On a :

$$\bullet S_x(f) = |X(f)|^2 = |M_1(f) + TF[m_2 \cos(2\pi f_{p2}t)]|^2 = \left| M_1(f) + \frac{1}{2}(M_2(f - f_{p2}) + M_2(f + f_{p2})) \right|^2 = S_{m1}(f) + \frac{1}{4}|M_2(f - f_{p2})|^2 + \frac{1}{4}|M_2(f + f_{p2})|^2 + \frac{1}{2}|M_2(f - f_{p2})M_2(f + f_{p2})| + |M_1(f)M_2(f - f_{p2})| + |M_1(f)M_2(f + f_{p2})|$$

En simplifiant les 3 dernières intégrales (on considère que les signaux occupent la même bande de fréquence $[-b, b]$ autour de 0, avec $b \ll f_{p1}$ et $b \ll f_{p2}$) ; on obtient :

$$S_x(f) = S_{m1}(f) + \frac{1}{4}|M_2(f - f_{p2})|^2 + \frac{1}{4}|M_2(f + f_{p2})|^2 = S_{m1}(f) + \frac{1}{4}S_{m2}(f - f_{p2}) + \frac{1}{4}S_{m2}(f + f_{p2})$$

On retombe bien sur la même expression que précédemment.

Aussi on calcul la puissance du bruit Gaussien à rajouter au signal en fonction du rapport signal sur bruit voulu :

$$(S/N)_{dB} = \frac{P_s}{P_N} \Leftrightarrow P_N = \frac{P_s}{10^{(S/N)_{dB}/10}}$$

2.1.2 Implantation

2.1.2.1 Modulation bande base

Afin de pouvoir transmettre un fichier binaire par modulation il faut préalablement transposer le fichier dans le domaine temporel pour obtenir $m_1(t)$ et $m_2(t)$. On appelle ces signaux des messages NRZ.

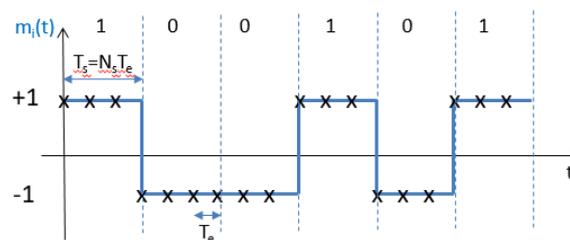


Figure 3 Message type NRZ

Sachant que chacun des signaux est transmis sur une trame de 40ms il faut définir T_s la période temporelle que va prendre un bit du fichier (voir figure 3). Comme

nous travaillons avec des valeurs discrètes on a la relation $T_s = N_s \times T_e$ avec N_s un nombre d'échantillons. On sait que chaque fichier à transmettre contient 480 bits. Soit T la durée d'un slot :

$$T_s = \frac{T}{480} =$$

Puis :

$$N_s = \frac{T_s}{T_e} = 40e - 3 \times \frac{f_e}{480} = 10$$

On construit les signaux $m_1(t)$ et $m_2(t)$ de la manière suivante :

```
m1 = [bits_utilisateur1*2-1];
m2 = [bits_utilisateur2*2-1];
m1 = repelem(m1,Ns);
m2 = repelem(m2,Ns);
```

Les deux premières lignes permettent de créer un signal oscillant entre -1 et +1. Ensuite le module repelem répète N_s fois chaque bit.

Figure 4 Capture d'écran de la syntaxe sur matlab

On obtient :

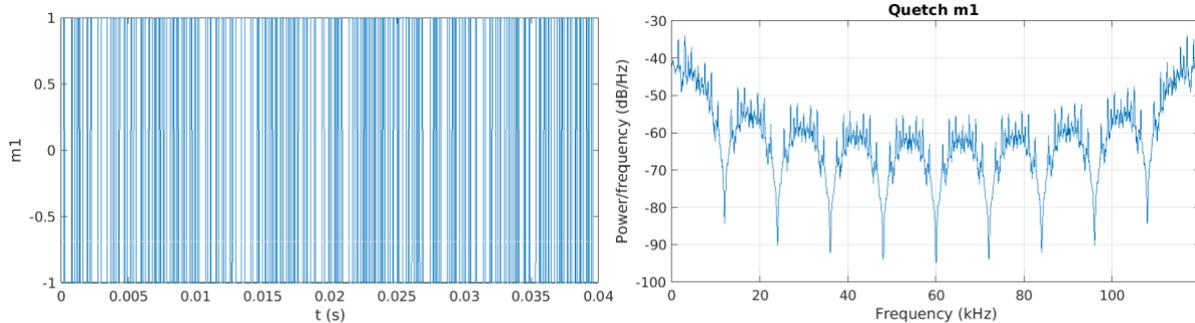


Figure 5 tracé de $m_1(t)$ ainsi que sa DSP

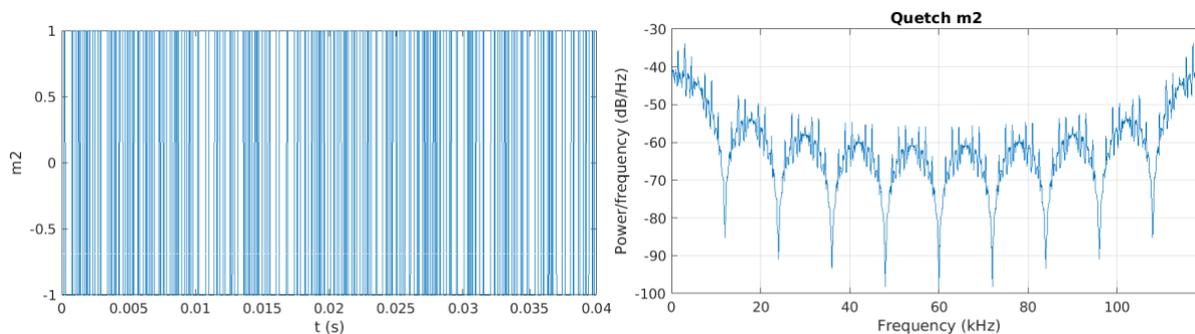


Figure 6 tracé de $m_2(t)$ ainsi que sa DSP

On retrouve bien sur les DSP une répartition de la puissance du signal tous les intervalles de $\frac{1}{T_s} = 12kHz$.

2.1.2.2 Construction du signal MF-TDMA

On génère ensuite un signal de durée $T = 5 \times 40e - 3 ms$ contenant les 5 slots. On place ensuite les signaux générés précédemment dans les slots correspondants (figure 7) :

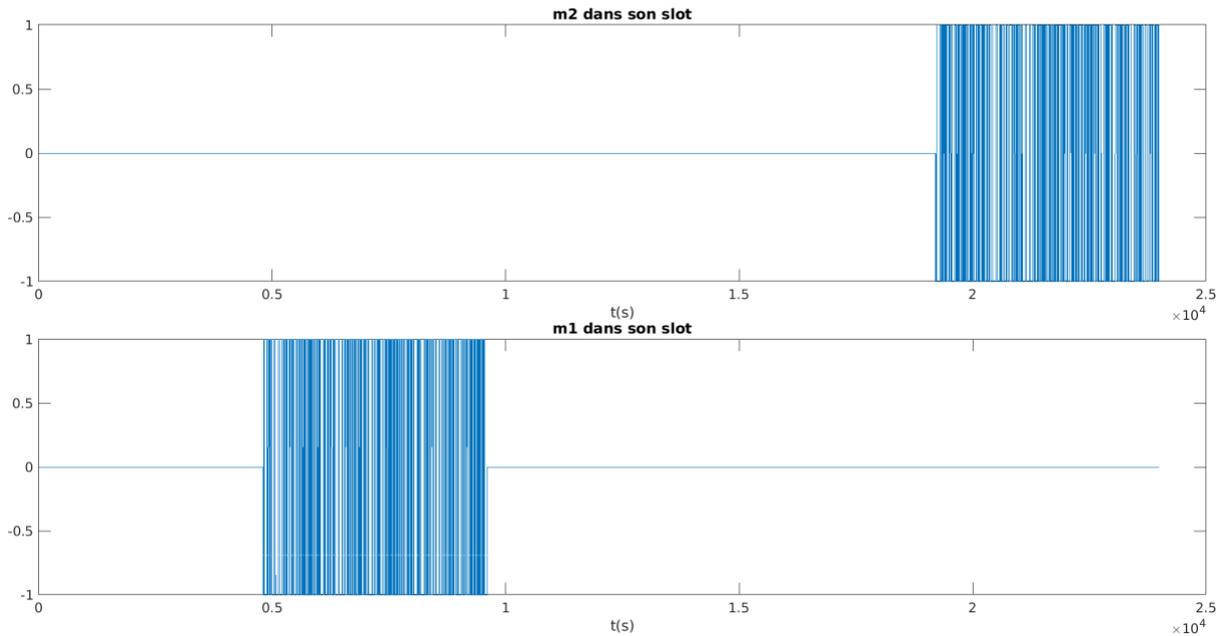


Figure 7 m1 et m2 dans leurs slots respectifs

On crée alors les signaux porteurs :

$$p1(t) = \cos(2 \times \pi \times fp1 \times t) \text{ et } p2(t) = \cos(2 \times \pi \times fp2 \times t)$$

Par modulation d'amplitude on effectue l'opération suivante :

$$x(t) = m1(t) \times p1(t) + m2(t) \times p2(t)$$

On obtient alors le signal transmis, il reste seulement à ajouter le bruit (on prend $(S/N)_{dB} = 100dB$ pour obtenir le signal simulé (figure 9) :

$$x_{transmis}(t) = x(t) + P_{bruit} \times randn(1, taille(x))$$

avec $randn(1, taille(x))$ taille(x) nombres oscillants entre -1 et 1.

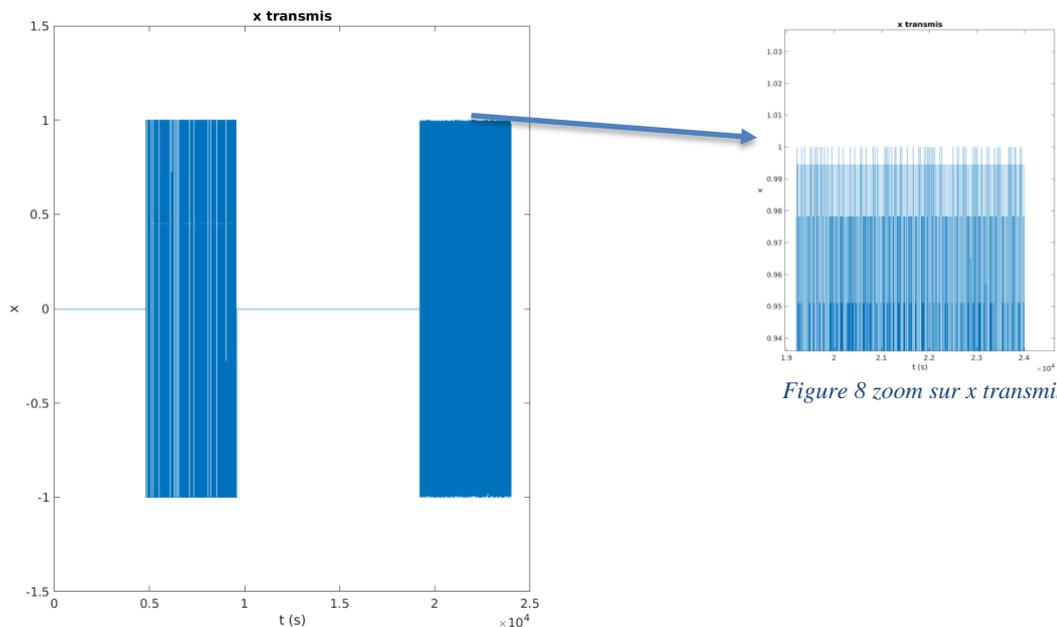


Figure 8 zoom sur x transmis

Figure 9 x transmis

On remarque que les signaux ont bien été modulés et altérés à cause du bruit. La DSP est présentée figure 10.

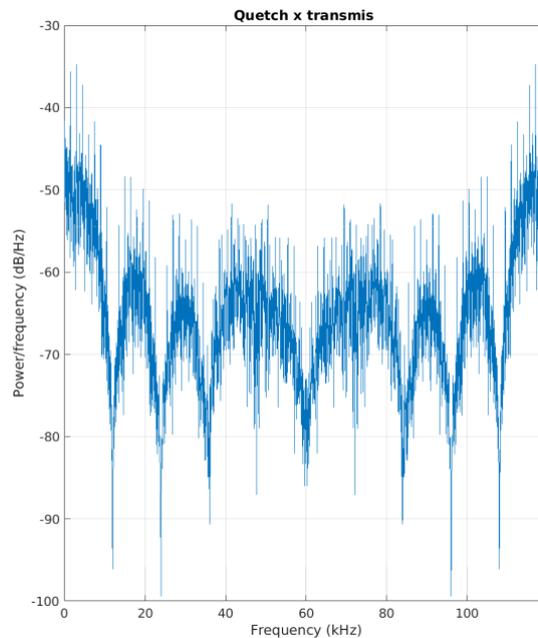


Figure 10 DSP x transmis

2.2. Mise en place du récepteur MF-TDMA

Cette partie du TP consiste à simuler le fonctionnement du récepteur (figure 11). Nous allons premièrement appliquer des filtres adaptés au signal pour réobtenir les signaux notés x_1 et x_2 . On choisira d'avoir des filtres d'ordre 61 afin d'avoir une petite fenêtre de transition, et donc de s'approcher d'un filtrage parfait, tout en ne monopolisant pas trop de puissance de calcul. Ensuite nous réaliserons diverses opérations (modulation puis re-filtrage) pour retourner dans la bande de base et détecter les slots utiles où sont transmises les données des utilisateurs.

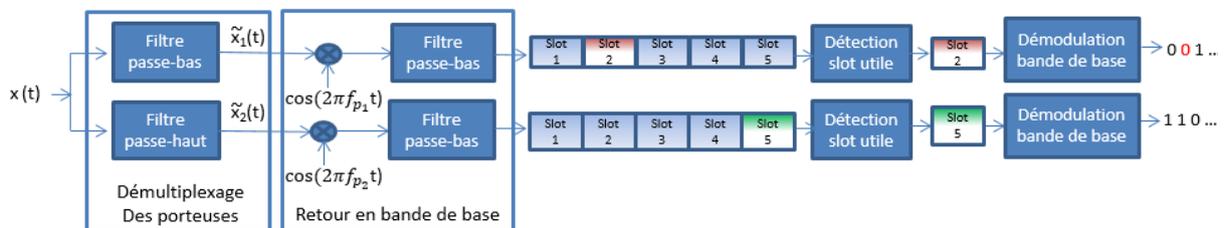


Figure 11 Schéma du fonctionnement du récepteur

2.2.1 Démultiplexage des porteuses

On va réaliser cette opération par l'utilisation de filtres. L'objectif sera de récupérer seulement les signaux portés par le cosinus à la fréquence f_{p1} d'un côté

(données de l'utilisateur 1, noté x_1) et f_{p2} de l'autre (données de l'utilisateur 2, noté x_2). On rappelle que $f_{p2} \gg f_{p1}$ c'est pourquoi il semble évident d'implémenter un filtre passe-bas RIF pour retrouver x_1 et à l'inverse un filtre passe-haut pour retrouver x_2 .

2.2.1.1 Synthèse du filtre passe-bas

On testera plusieurs fréquences de coupures pour le filtre passe bas. Celle que notre équipe a retenue sera $f_c = 23 \text{ kHz}$. On remarque bien sur la réponse fréquentielle (figure 12) que le filtre est un passe-bas de fréquence de coupure égale à 23kHz.

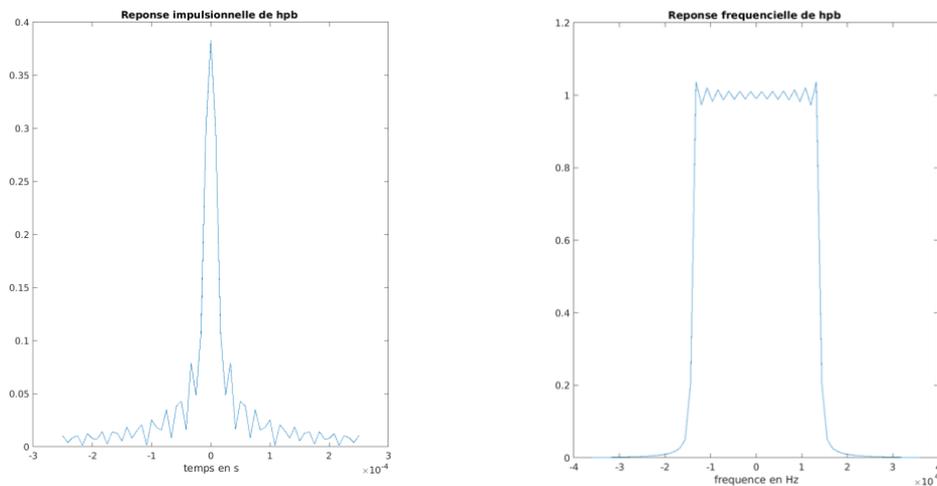


Figure 12 Réponse impulsionnelle et fréquentielle du filtre passe bas.

Pour filtrer nous utilisons la fonction conv de Matlab avec le paramètre 'same'. Cela permet de réaliser un filtrage sans décalage simplement (ne pas prendre en compte le décalage posait un problème pour récupérer le signal final).

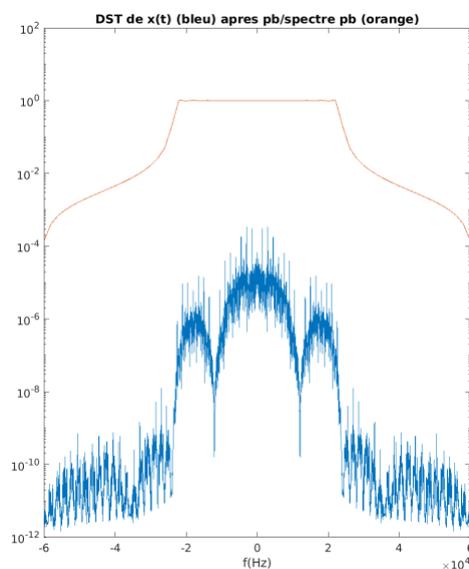


Figure 13 DST de x après passe-bas et spectre du passe-bas

On remarque bien sur la figure 14 que le filtre passe-bas permet d'obtenir une répartition de la puissance du signal plus importante entre -24kHz et $+24\text{kHz}$. C'est bien le résultat attendu.

2.2.1.2 Synthèse du filtre passe-haut

Comme le filtre passe-bas permet de récupérer le signal porté par $f_{p1} \ll f_{p2}$, on souhaite créer un filtre passe-haut permettant de récupérer le signal porté par f_{p2} . La fréquence de coupure $f_c = 23\text{kHz}$ semble ainsi être le choix le plus évident. On remarque bien sur la réponse fréquentielle de la figure 15 que l'on a bien un filtre passe haut de fréquence de coupure égale à 23kHz .

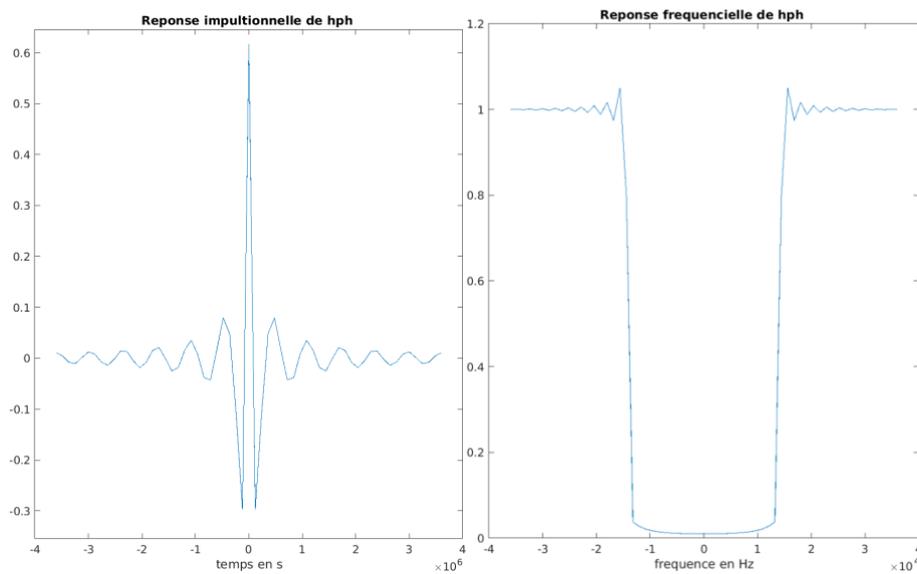


Figure 14 Réponse impulsionnelle et fréquentielle du filtre passe-haut.

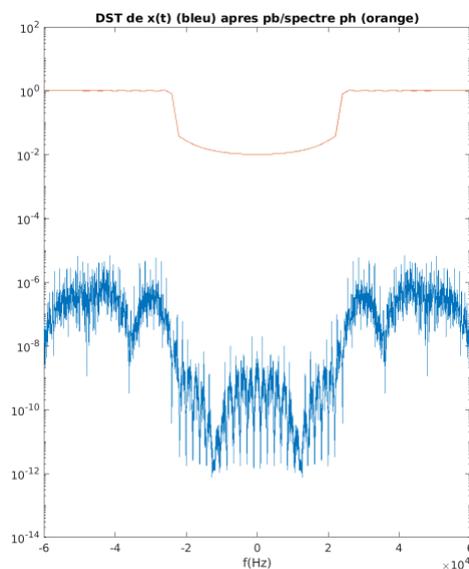


Figure 15 DST de x après passe-haut et spectre du passe-haut

On remarque bien sur la figure 16 que le filtre passe-haut permet d'obtenir une répartition de la puissance du signal plus importante hors des bornes -24kHz et $+24\text{kHz}$. C'est bien le résultat attendu.

2.2.1.3 Filtrage

En filtrant $x(t)$ grâce aux filtre passe-bas on obtient le résultat présenté figure 16. On remarque comme on s'y attendait que le message porté par f_{p2} est fortement affaibli alors que celui porté par f_{p1} semble intacte.

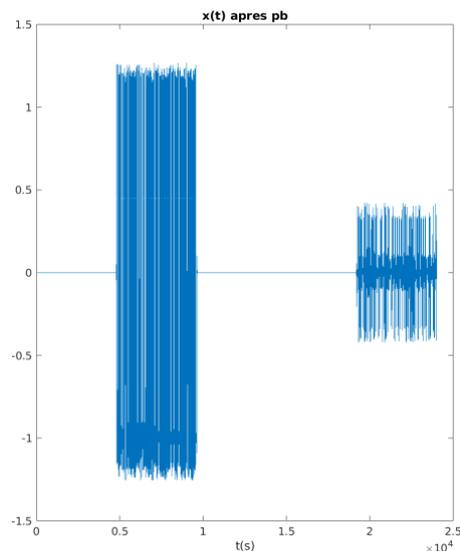


Figure 16 $x(t)$ après application du filtre passe bas

On en filtrant $x(t)$ grâce aux filtre passe-haut on obtient le résultat présenté figure 17. On remarque comme on s'y attendait que le message porté par f_{p1} est fortement affaibli alors que celui porté par f_{p2} semble intacte.

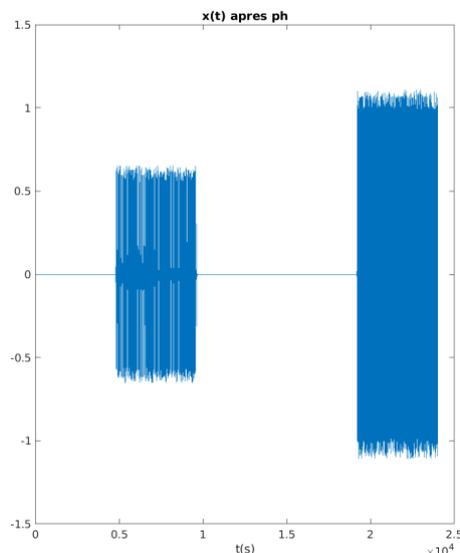


Figure 17 $x(t)$ après application du filtre passe haut

On note les deux signaux obtenus $x_1^{\sim}(t)$ et $x_2^{\sim}(t)$.

2.2.2 Retour en bande de base

On suppose nos systèmes parfaitement synchronisés. Afin de retourner en bande de base nous allons multiplier chaque signal x_1^{\sim} et x_2^{\sim} par leur signal porteur. Ainsi nous allons réaliser les opérations $x_1^{\sim} \times p_1$ et $x_2^{\sim} \times p_2$.

Nous réalisons ensuite un filtrage avec le même filtre passe-bas que précédemment. On obtient les résultats suivants (figure 18).

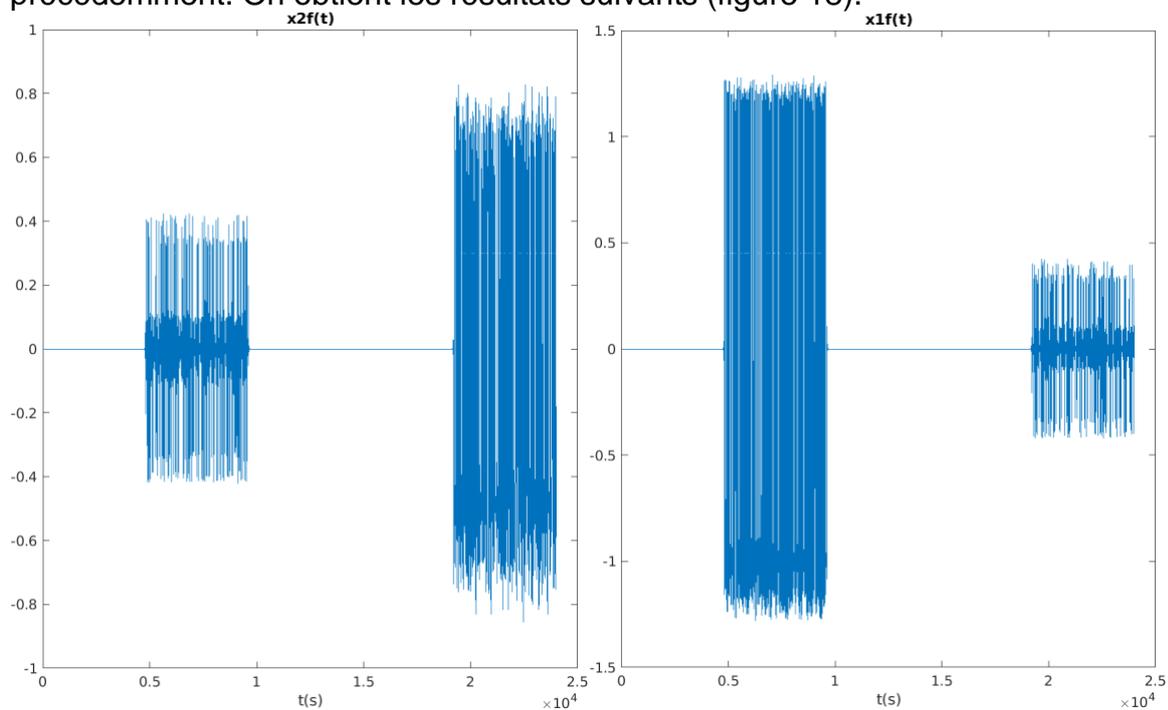


Figure 18 x_1 et x_2 après avant la détection de slot

2.2.3 Détection du slot utile

Pour détecter le slot utile on va utiliser un détecteur d'énergie, on calcul l'énergie du signal sur chaque slot et prenons celui avec la plus importante. On utilisera la fonction Matlab suivante (figure 19).

```

function slot = divideslot(x)
    slot1 = x(1:4800);
    slot2 = x(4801:4800*2);
    slot3 = x(4800*2+1:4800*3);
    slot4 = x(4800*3+1:4800*4);
    slot5 = x(4800*4+1:end);
    v = [slot1; slot2; slot3; slot4; slot5];
    [~, idx] = max([mean(slot1.^2), mean(slot2.^2), mean(slot3.^2), mean(slot4.^2), mean(slot5.^2)]);
    slot=v(idx,:);
end

```

Figure 19 fonction Matlab

Pour trouver simplement le slot avec le plus d'énergie nous avons simplement calculé la moyenne des slots au carrés qui sera proportionnel aux énergies.

La fonction renvoie bien le slot 2 pour x1f et le slot 5 pour x2f.

2.2.4 Démodulation bande de base

Enfin nous utilisons les fonctions définies dans le TP :

```
SignalFiltre=filter(ones(1,Ns),1,MessageRetrouve) ;
```

```
SignalEchantillonne=SignalFiltre(Ns :Ns :end) ;
```

```
BitsRecuperes=(sign(SignalEchantillonne)+1)/2 ;
```

On obtient les messages suivants :

- « Ma probabilité d'apparition d'appariation au semestre 5 est assez élevée » pour le message de l'utilisateur 1.

- « Mes initiales, placées à l'envers, évoquent un vert support » pour le message de l'utilisateur 2.

Après concertation on pense que M. Jean-Yves Tourneret est le professeur mystère.

III. Synthèse et conclusion

En première partie du TP nous avons créé un signal temporel à partir de données binaires. Puis nous avons utilisé la modulation d'amplitude pour transmettre ces données par les ondes. Une fois cela fait l'ajout d'un bruit a permis de simuler les perturbations subis par le signal. On remarque qu'un rapport signal/bruit de 100dB n'a pas grande influence. En revanche avec un bruit plus élevé le signal est fortement perturbé. Nous avons ensuite appliqué plusieurs filtres sur les différents signaux. Au vu des fréquences des signaux porteurs (0 Hz et 46 kHz) l'idée de choisir une fréquence de coupure entre les deux semblait évidente pour sélectionner les données. On aura choisi une fréquence $f_c = 23 \text{ kHz}$. Enfin après un retour en bande de base et une détection de slot utile grâce à une analyse de l'énergie de chaque slot, nous avons retrouvé les informations transmises.

Ce TP nous a permis de comprendre la transmission de données binaires par signal analogique, nous y avons notamment retravaillé les notions de modulation et de démodulation et plus globalement de traitement du signal.