

Projet modulation

Groupe 5

FRANCOIS-CHARLOT Axel

HERNANDEZ Adrien

PICCINALI Amandine

*SCHURCK Alexan**dre***

Séances du 08/01/21 et 15/01/21

Compte rendu dû pour le 25/01/21

Introduction et présentation du projet

L'objectif de ce projet est, dans un premier temps, de transmettre 4 signaux dans un même canal et en même temps en utilisant le théorème d'échantillonnage, de modulation d'amplitude et de démodulation. Dans un second temps, nous devons modifier nos programmes pour transmettre 8 signaux simultanément au lieu de 4. Le compte rendu de ce projet retrace les différentes étapes de réflexions et les productions de notre groupe afin de remplir les exigences du cahier des charges imposé dans le cadre du TP.

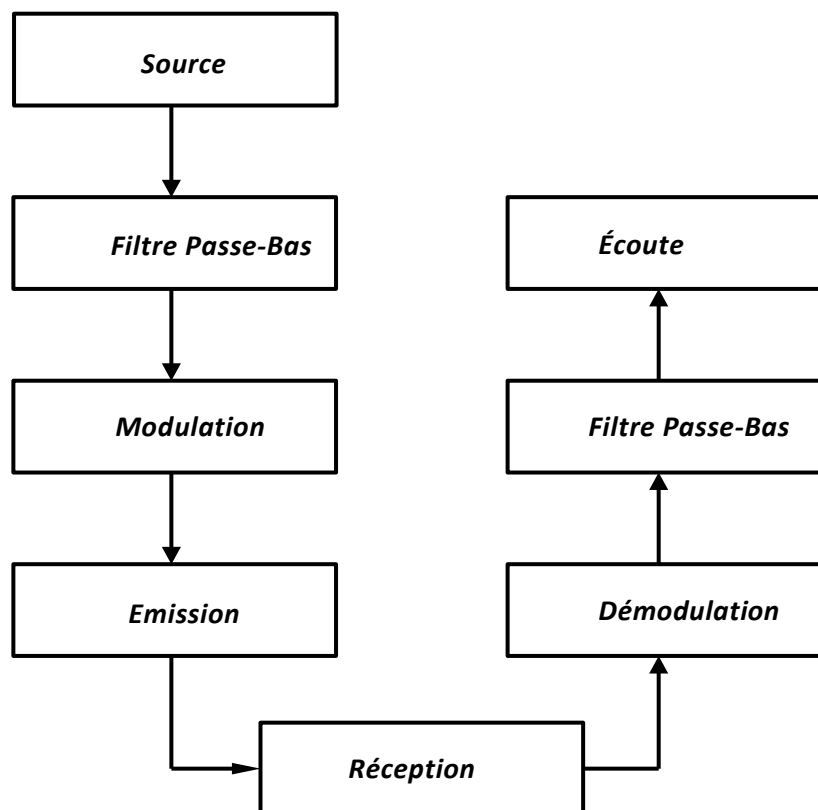
Première partie : Modulation d'Amplitude Sans Porteuse

Dans cette partie, nous devons programmer 2 codes permettant la transmission d'un message en simultané avec 3 autres messages dont s'occupent les autres équipes. La partie bureau d'étude a été faite avec toutes les équipes pour pouvoir s'accorder sur les différentes caractéristiques de nos système.

Le bureau d'étude

Lors du bureau d'étude nous avons établis les méthodes permettant d'atteindre notre objectif. Notamment, nous avons créé ce schéma résumant dans sa globalité le principe de fonctionnement du système.

Schéma du système dans sa globalité



Détermination des caractéristiques du système

Fréquence d'échantillonnage

Nous avons ensuite défini la fréquence d'échantillonnage (F_e) à 96 kHz car nous disposons d'une bande passante de 96 kHz. Ainsi, d'après le théorème de Shannon, $F_{max} = \frac{F_e}{2} = 48 \text{ kHz}$. Ainsi on restreint la bande passante à 48 kHz.

Modulation

Tout d'abord, on sait que notre signal doit être modulé à l'aide d'une porteuse qui s'exprime comme suit : $s_p(t) = \sin(2\pi\nu_0 t)$.

On multiplie donc le signal par $s_p(t)$ et on obtient alors : $x(t) = s(t)\sin(2\pi\nu_0 t)$.

En faisant la transformée de Fourier de ce signal on obtient :

$$X(\nu) = S(\nu) * \left(-\frac{1}{2i}\right)(\delta(\nu - \nu_0) - \delta(\nu + \nu_0)) = \left(-\frac{1}{2i}\right)(S(\nu - \nu_0) - S(\nu + \nu_0)).$$

Schéma du module du spectre du signal 0 avant modulation

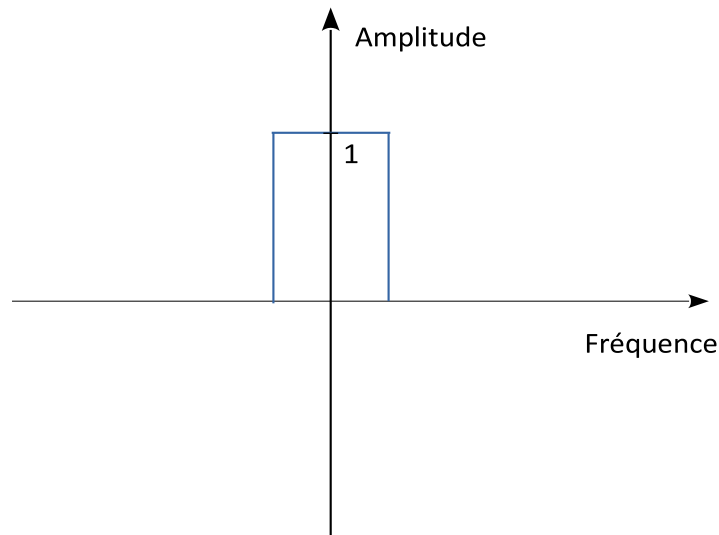
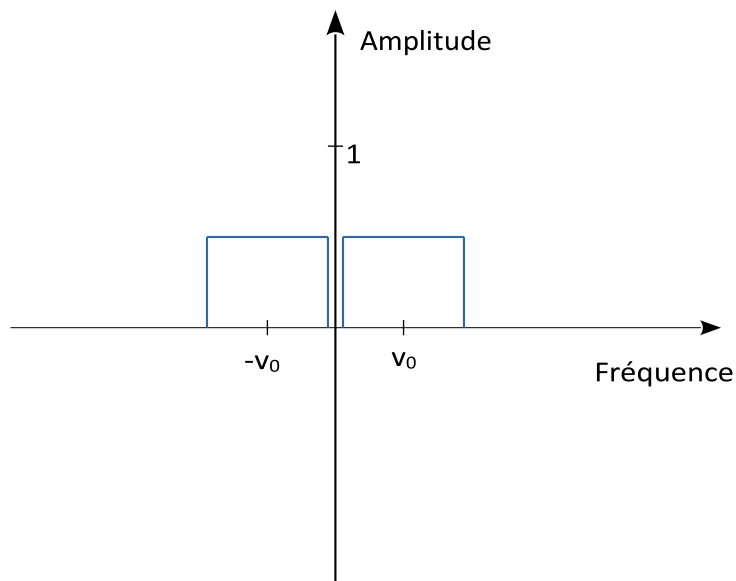


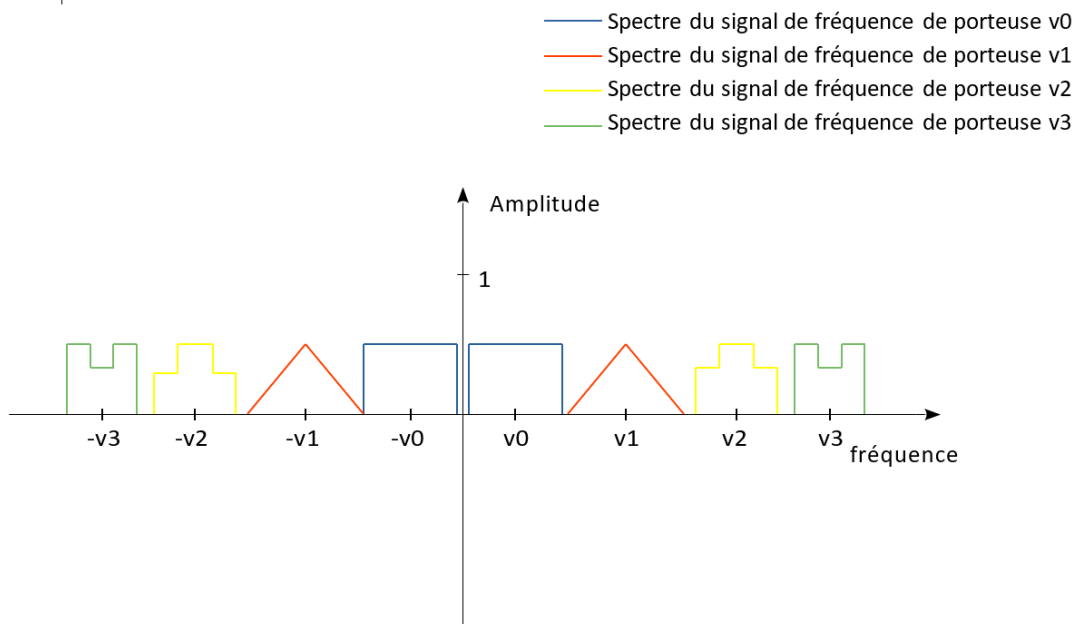
Schéma du module du spectre du signal 0 après modulation



Finalement le spectre du signal est déplacé sur les fréquences v_0 et $-v_0$ (et a une amplitude 2 fois inférieure), on peut désormais transmettre les signaux dans le canal.

On doit aussi prendre en compte le fait que le montage est sensé transmettre 4 messages simultanément. Ainsi donc, il nous faut répartir les valeurs des porteuses et des bandes passantes allouée à chaque groupe afin de permettre la transmission sans perturbation des 4 signaux de manière synchrone. Lors de notre étude, nous avons expliqué cette problématique avec le schéma suivant.

Schéma du module des spectres de 4 différents signaux multipliés par leur porteuse (signal général)



Il nous faut donc déterminer quelles sont les valeurs des fréquences des porteuses v_0 à v_3 , sachant que tous ces signaux doivent être contenus dans la bande de fréquence $[-48\text{kHz}, 48\text{kHz}]$. Cependant, bien qu'une réponse intuitive donnerait une bande passante de 12 kHz par groupe et les porteuses respectives 6, 18, 30 et 42 kHz, ce résultat est faux et nécessite de prendre en compte le

fonctionnement de la démodulation. Néanmoins, le principe de fonctionnement reste identique, nous y reviendrons juste après.

Démodulation

Une fois le signal modulé récupéré il faut le démoduler afin de pouvoir l'écouter. Pour cela on multiplie le signal par la porteuse c'est-à-dire : $y(t) = s(t) \sin(2\pi\nu_0 t) = s(t)/2 - \frac{s(t)\sin(2\pi(2\nu_0)t)}{2}$.

Dans le domaine spectral on obtient : $Y(\nu) = \frac{S(\nu)}{2} - S(\nu) * \frac{(\frac{1}{2})(\delta(\nu - 2\nu_0) + \delta(\nu + 2\nu_0))}{2}$.

On a donc : $Y(\nu) = \frac{S(\nu)}{2} + (\frac{1}{4}) S(\nu - 2\nu_0) - S(\nu + 2\nu_0)$.

Schématiquement, voici ce qu'il se passe dans l'espace de Fourier :

Schéma du module du spectre du signal 0 avant démodulation

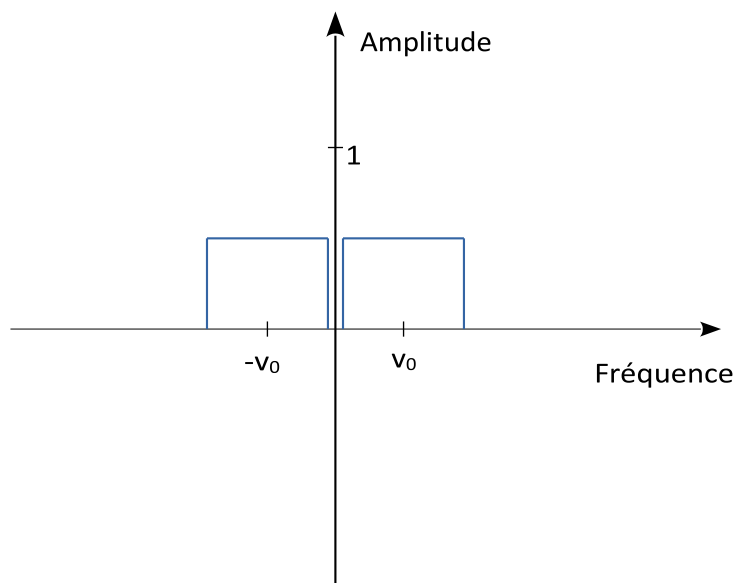
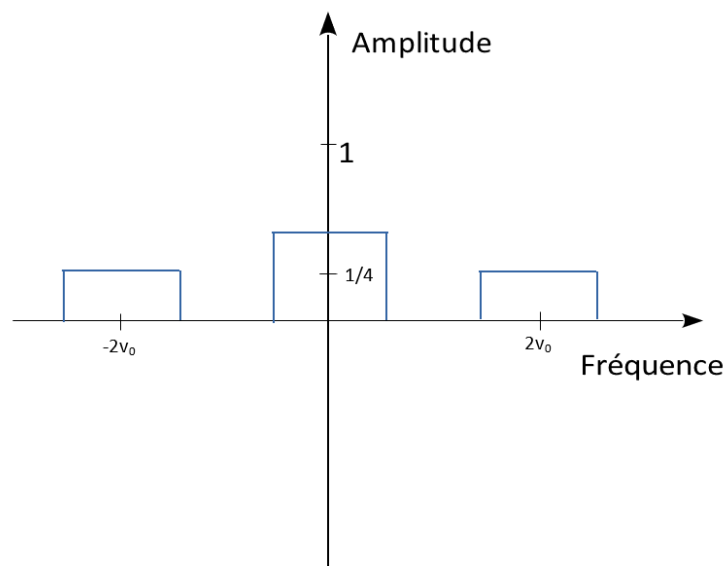


Schéma du module du spectre du signal 0 après démodulation



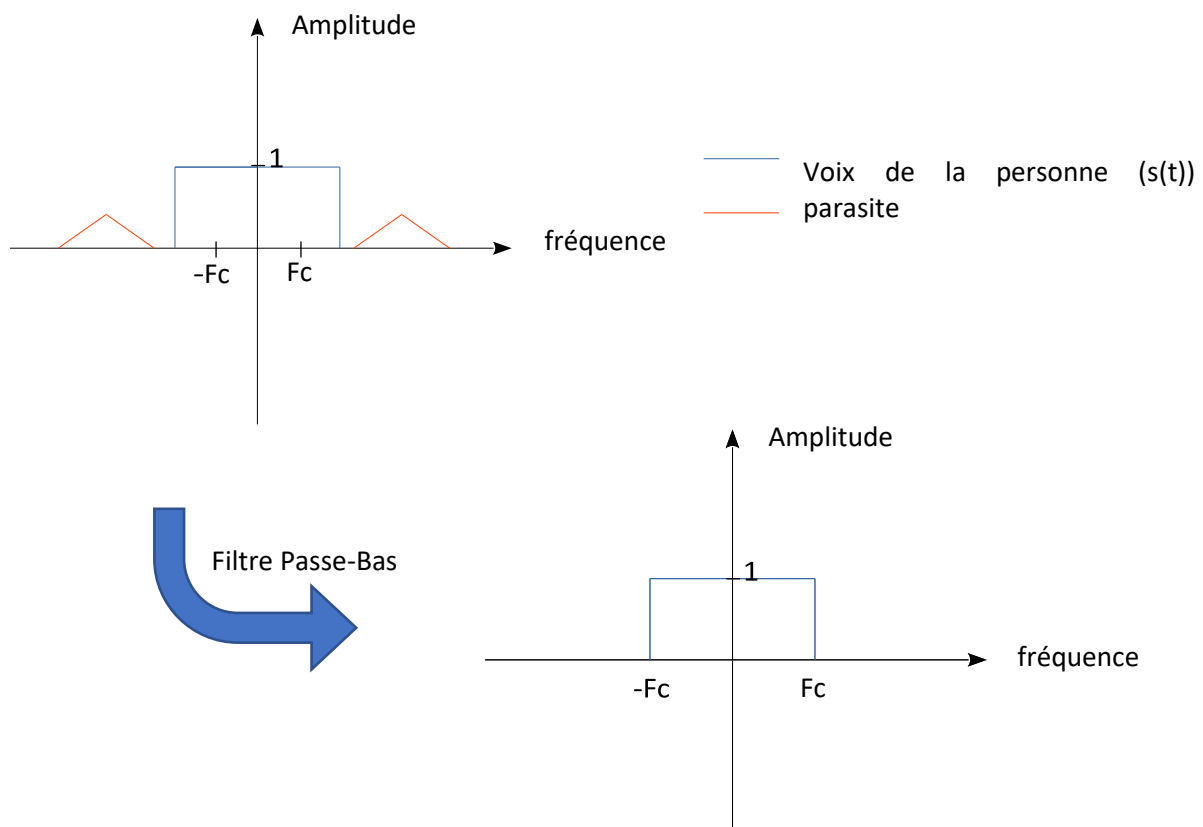
La démodulation cause alors un problème : on constate des résidus centrés sur $-2v_0$ et $2v_0$ qui risquent d'être en dehors de notre gamme de fréquence de notre bande passante, et ainsi de ne plus respecter le critère de Shannon ce qui rendra le signal non reconstituable et donc causera la perte de l'information du signal. Pour éviter cela, nous sommes forcés de diminuer à nouveau la bande passante, et comme la fréquence maximale du spectre centré sur $2v_3$ (v_3 = porteuse de plus haute fréquence) doit être 48Khz, cela signifie que notre fréquence maximale est de $48/2 = 24$ kHz, c'est-à-dire que la bande de fréquence de travail est de $[-24 \text{ kHz}, 24 \text{ kHz}]$.

A présent que la bande de fréquence de travail a été définie, on peut enfin déterminer les gammes de fréquence de travail de chaque groupe et les fréquences des porteuses. Ainsi, chaque groupe disposera de $\frac{24}{4} = 6 \text{ kHz}$ de bande passante et les porteuses seront donc respectivement 3 kHz, 9 kHz, 15 kHz et 21 kHz. À la suite de la répartition des porteuses avec les autres groupes du projet, nous nous sommes attribué la porteuse de fréquence 3 kHz. Ainsi, notre gamme de fréquence sera entre 0 et 6 khz.

Filtre passe-bas (de modulation)

A présent que les éléments précédents ont été établis, nous devons nous assurer que notre signal soit uniquement dans sa bande de fréquence pour ne pas perturber la transmission des autres messages. Ainsi, nous devons filtrer ce signal à l'aide d'un filtre passe-bas, dont la fréquence de coupure sera de 3000 Hz. Ce processus permet aussi d'éliminer les éventuels bruits parasites.

Spectre du module du signal non filtré puis filtré (filtre passe-bas)



Filtre passe-bas (de démodulation)

Une fois le signal transmis, le signal général doit être filtré afin d'obtenir le signal voulu. Un moyen d'atteindre cet objectif serait d'utiliser un filtre passe bas dont la fréquence de coupure serait de $[-F_c, F_c]$ pour récupérer le signal démodulé à la partie démodulation précédente.

Schéma du module du spectre du signal 0 après démodulation et avant filtrage

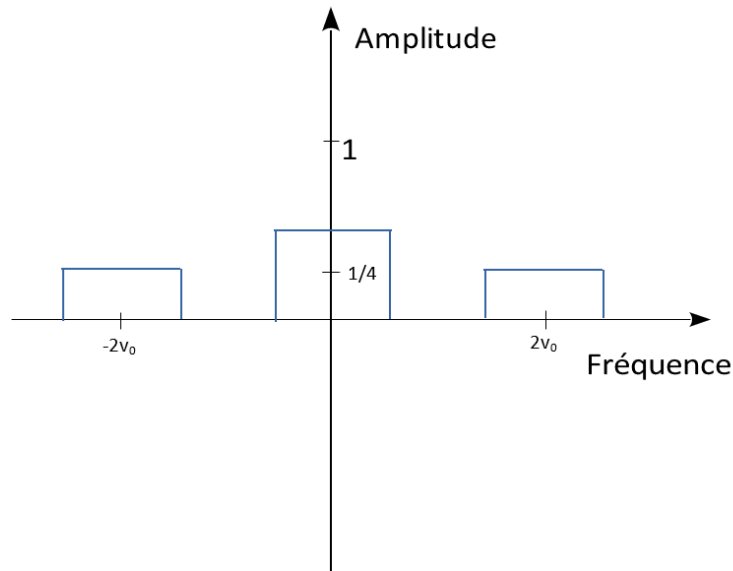
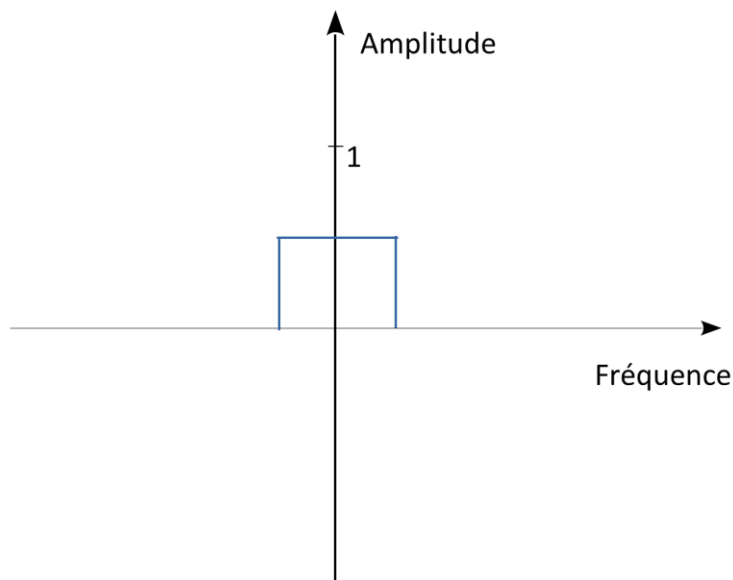


Schéma du module du spectre du signal 0 après démodulation et après filtrage



Ainsi, à la sortie de ce filtre passe bande, on a bien récupéré notre signal d'entrée $s(t)$, identique en tout point, à la seule différence que son amplitude est divisée par 2 (mais cela est simplement corrigeable).

Conclusion du bureau d'étude

Par soucis de clarté, voici l'ensemble des valeurs déterminées lors du bureau d'étude :

- Fréquence d'échantillonnage : $F_e = 96 \text{ kHz}$
- Fréquence des porteuses : $v_0 = 3 \text{ kHz}$, $v_1 = 9 \text{ kHz}$, $v_2 = 15 \text{ kHz}$, $v_3 = 21 \text{ kHz}$
- Fréquence de coupure des filtres passe-bas : $F_c = 3 \text{ kHz}$
- Fréquence de porteuse de notre groupe : $v_0 = 3 \text{ kHz}$

Blocs fonctionnels et algorithmes développés

Afin d'obtenir le résultat souhaité, nous avons programmé 2 fichiers, Modulation et Démodulation. De plus, nous avons utilisé les programmes fournis lors du TP. Nous allons décrire ici leur utilité et leur fonctionnement schématique.

Modulation.m

Ce fichier gère la partie modulation du programme. Son fonctionnement sous forme de bloc fonctionnels est le suivant :



De manière plus explicite, voici le fonctionnement détaillé :

- Le fichier appelle la fonction **RecordModulation** pour enregistrer un message vocal énoncé par un des élèves. Nous avons choisi d'enregistrer un message de 3 secondes. Nous avons, en bonus, programmé la possibilité d'importer le message depuis un fichier audio au format « .wav » et de fréquence d'échantillonnage 96 kHz.
- Le message enregistré est filtré grâce à la fonction **PasseBas** avec une fréquence de coupure de 3 kHz. Filtrer le message permet d'éviter qu'il dépasse de notre bande de fréquence attribuée.
- Le signal est ensuite à proprement dit modulé (modulation d'amplitude sans porteuse) simplement en multipliant le signal par la porteuse $\sin(2\pi v_0 t)$, où v_0 est la fréquence de notre porteuse, c'est-à-dire 3 kHz.
- Le signal modulé est ensuite lu en boucle par Matlab grâce à la fonction **sound** et enregistré sous la forme d'un fichier audio (utile pour déboguer).

Demodulation.m

Ce fichier gère la partie démodulation du programme. Son fonctionnement sous forme de bloc fonctionnels est le suivant :



De manière plus explicite, voici le fonctionnement détaillé :

- Le fichier appelle la fonction **RecordModulation** pour enregistrer le message modulé qui est lui est fourni par **Modulation.m**. Nous avons aussi programmé la possibilité d'importer le message modulé depuis un fichier audio au format « .wav » et de fréquence d'échantillonnage 96kHz (utile pour déboguer).

- Le signal est ensuite démodulé en multipliant le signal par la porteuse $\sin(2\pi\nu_0 t)$, où ν_0 est la fréquence de notre porteuse, c'est-à-dire 3kHz.
- Le message enregistré est filtré grâce à la fonction **PasseBas** avec une fréquence de coupure de 3 kHz. Filtrer le message permet d'éliminer les composantes de fréquence $-2\nu_0$ et $2\nu_0$ (cf. partie démodulation du bureau d'étude)
- Le signal démodulé est ensuite lu par Matlab grâce à la fonction **sound** et enregistré sous la forme d'un fichier audio.

PasseBas.m (fourni)

La fonction **PasseBas** permet d'obtenir un signal filtré par un passe bas à une fréquence de coupure choisie.

RecordModulation.m (fourni)

La fonction **RecordModulation** permet d'enregistrer un signal audio grâce à un microphone dans matlab.

TransFourier.m (fourni)

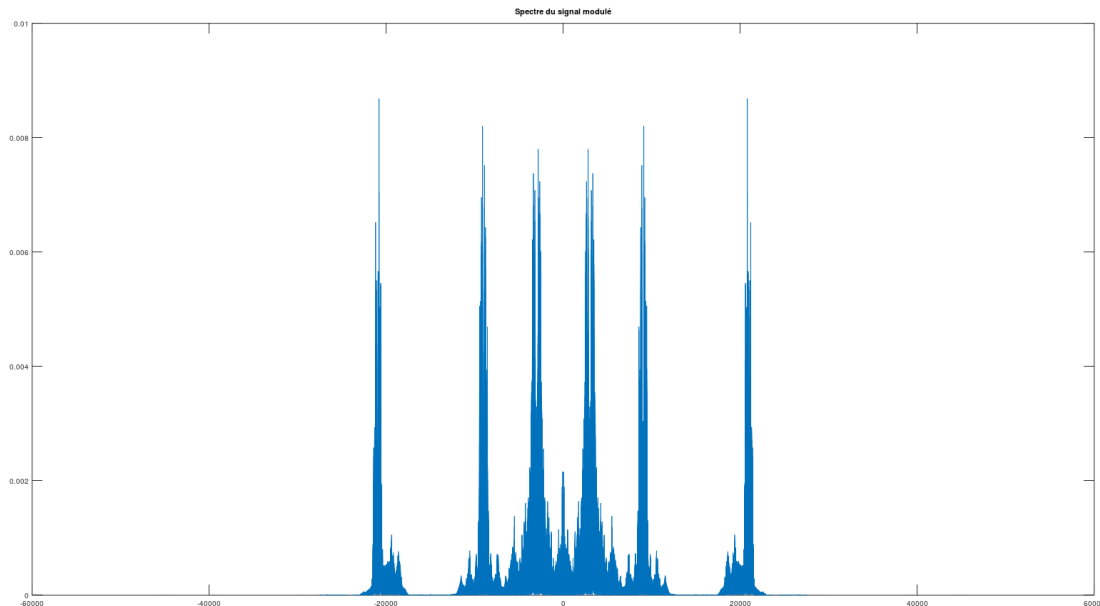
La fonction **TransFourier** permet d'effectuer la transformée de Fourier d'un signal. Cette fonction n'est pas indispensable pour l'exécution du programme est n'est utile qu'à des fins de compréhension pour afficher des courbes.

Codes Matlab

Les codes Matlab que le groupe a programmé et commenté sont fournis dans le fichier d'annexe.

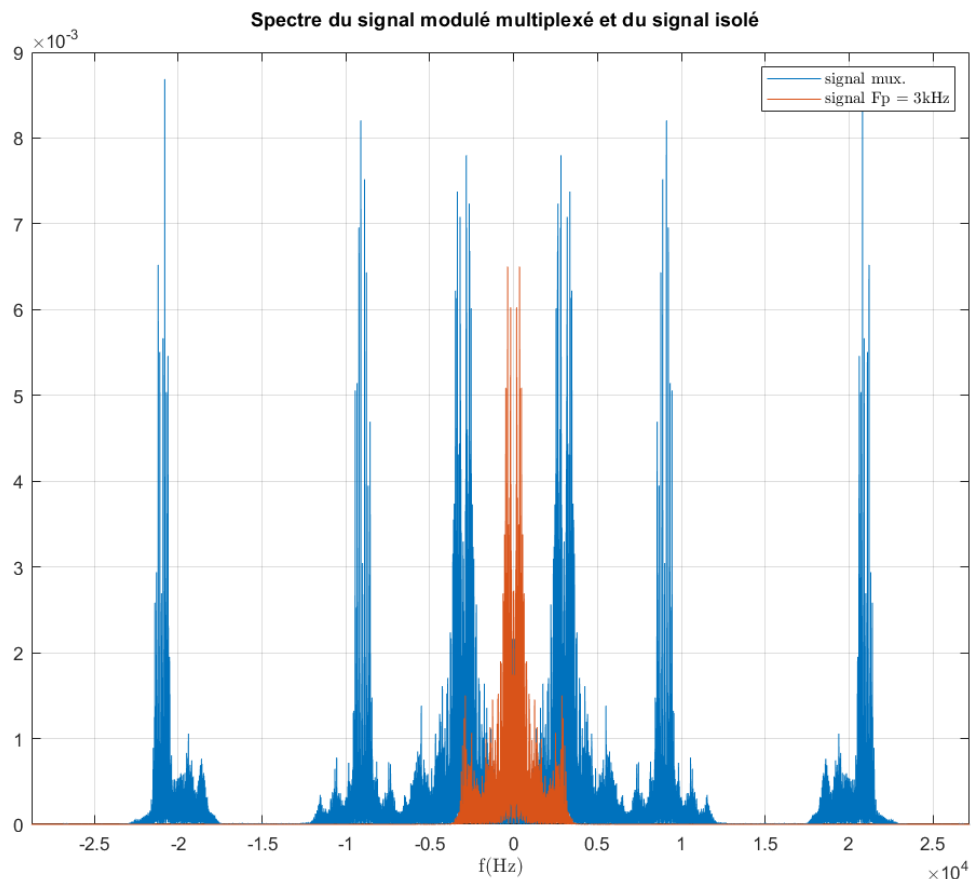
Figures obtenues lors du TP

Une fois que tous les groupes ont validé le bon fonctionnement individuel de ses programmes de modulation et démodulation, nous sommes passés à l'étape de mise en commun du moyen de transmission avec un multiplexeur. Nous avons enregistré un message puis l'avons transmis en même temps pour tous les groupes. Ainsi nous avons enregistrés le signal émis par tous les groupes dans le programme de démodulation, et voici le spectre de Fourier du signal reçu :



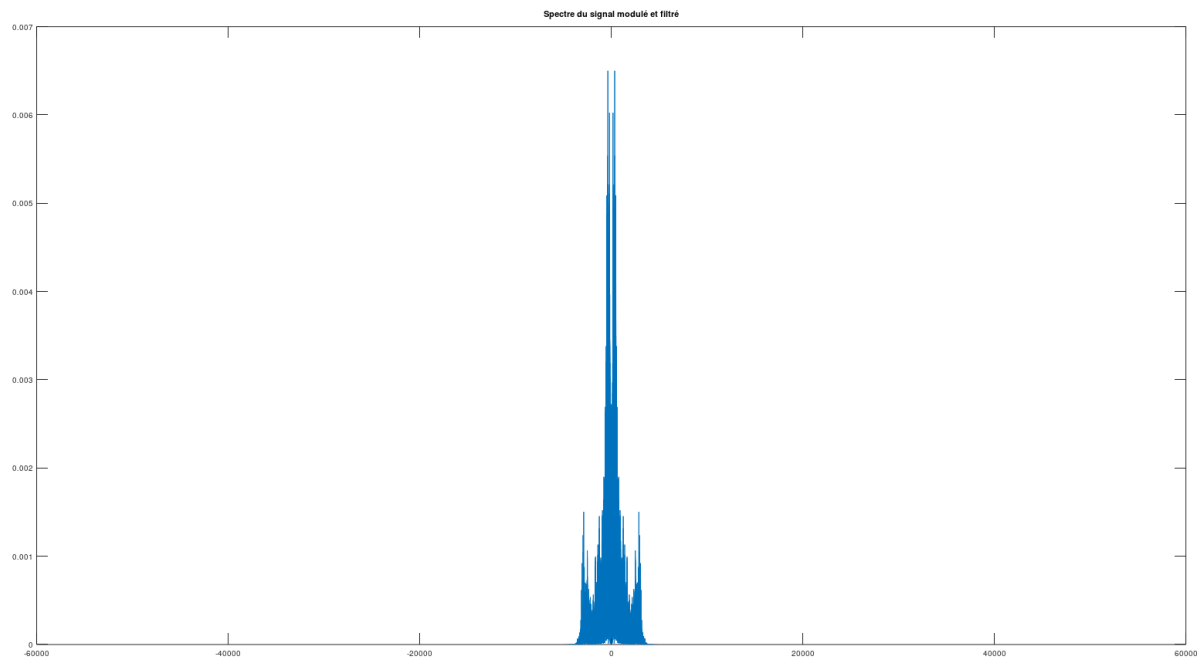
On remarque qu'on a 3 signaux (nous n'étions que 3 groupes au lieu de 4), respectivement centrés sur les fréquences 3 kHz, 9 kHz et 21 kHz. Ces valeurs correspondent bien aux valeurs des porteuses que chaque groupe s'étaient attribuées.

Une fois le signal démodulé mais non encore filtré, le spectre du signal est le suivant :



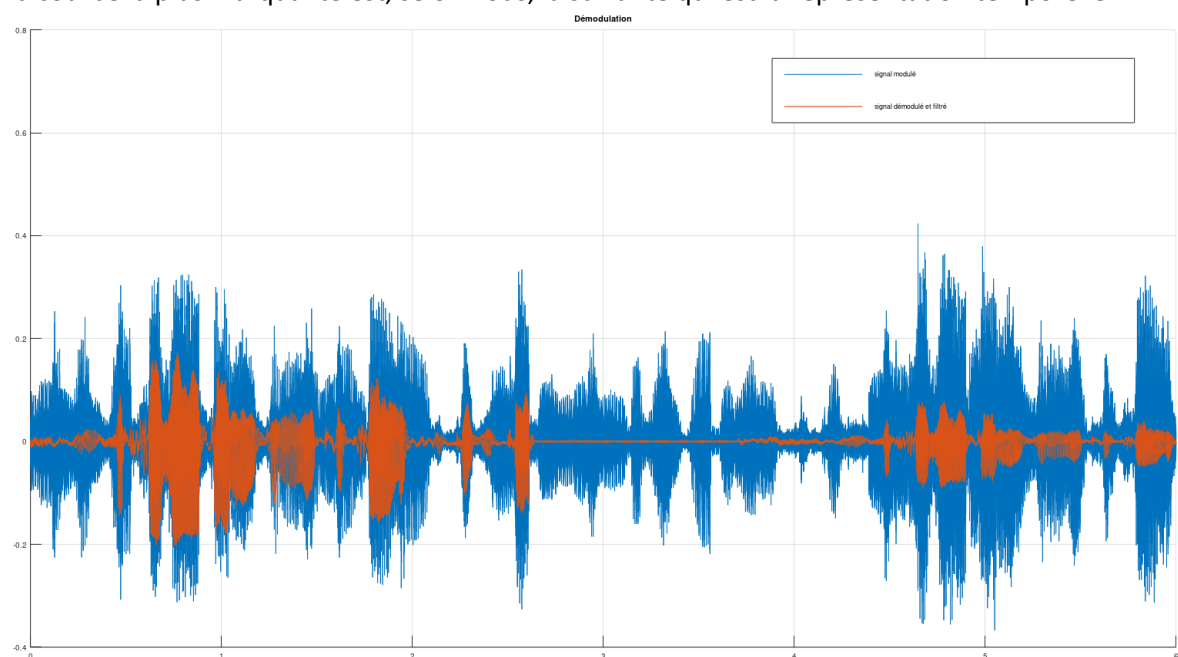
On retrouve bien la forme du spectre de notre signal, qui était auparavant centré sur -3 kHz et 3 kHz, à présent centré sur 0 kHz. De plus, on remarque que, conformément à l'étude théorique, l'amplitude des pics est 2 fois plus petite.

Une fois le signal filtré, on a alors le spectre suivant :



Le filtre passe bas a bien isolé notre signal des autres.

La courbe la plus marquante est, selon nous, la suivante qui est la représentation temporelle :



On constate que notre signal est en réalité une partie minime du signal reçu, et que finalement une opération mathématique « simple » et un filtrage permet d'isoler d'un brouhaha de plusieurs messages transmis simultanément grâce à la modulation en un seul message compréhensible.

Interprétations et analyses critiques des résultats

Le système que nous avons proposé permet bien de remplir le cahier des charges : on entend bien notre message sans perturber les autres qui sont diffusés en parallèle.

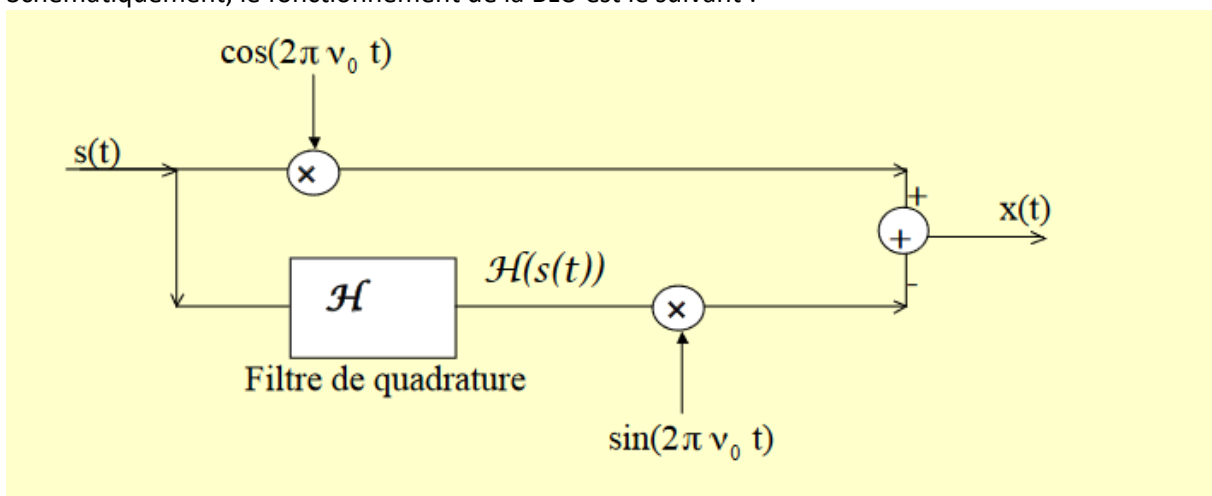
Cependant, le point négatif de notre système est la perte en qualité du signal à la sortie par rapport à l'enregistrement initial. Néanmoins, le signal final reste tout à fait compréhensible.

Seconde partie : Modulation d'Amplitude en Bande Latérale Unique

Dans cette partie, nous devons modifier nos 2 codes précédents pour permettre la transmission de 2 messages en simultan  , soit un total de 8 messages gr  ce aux autres   quipes.

Fonctionnement de la modulation d'amplitude BLU

Nous avons pr  c  demment vu que, puisque notre signal d'entr  e   tait r  el, le spectre de notre signal   tait r  el et pair. Par cons  quent, cela signifie que seule la moiti   de notre spectre est r  ellement utile pour la transmission des informations, puisque l'autre moiti   se retrouve par sym  trie. Le but de la modulation d'amplitude BLU est de supprimer la moiti   du spectre d'un premier signal pour le remplacer par celui d'un autre signal, et de finalement transmettre 2 messages en m  me temps. Sch  matiquement, le fonctionnement de la BLU est le suivant :



La modulation BLU repose sur l'utilisation de la transform  e de Hilbert. On utilise un filtre de quadrature, ce filtre donne le signal d'entr  e retard   en quadrature de phase soit avec un retard de $\pi/2$. On utilise la commande `hilbert` de matlab qui renvoie le signal analytique associ   au signal donn  e en argument. On garde la partie imaginaire de ce signal pour conserver uniquement la transform  e de Hilbert avec la commande `imag`.

On cr  e un signal $s_1(t)$ et on obtient un signal $x_1(t)$ avec le calcul suivant :

$$x_1(t) = s_1(t) \cos(2\pi v_0 t) - \mathcal{H}(s_1(t)) \sin(2\pi v_0 t)$$

On cr  e ensuite un signal $s_2(t)$ et on obtient un signal $x_2(t)$ avec le calcul suivant :

$$x_2(t) = s_2(t) \cos(2\pi v_0 t) + \mathcal{H}(s_2(t)) \sin(2\pi v_0 t)$$

Et finalement on   met un signal $x(t) = x_1(t) + x_2(t)$

Pour d  moduler le signal $x(t)$, on utilise un filtre passe bande pour r  cup  rer la moiti   du signal qui nous int  resse puis on multiplie par $\cos(2\pi v_0 t)$. Puis, on utilise un filtre passe bas comme dans la premi  re partie.

Blocs fonctionnels et algorithmes d  velopp  s

Afin d'obtenir le r  sultat souhait  , nous avons programm   2 nouveaux fichiers, **ModulationBLU** et **D  modulationBLU**.

ModulationBLU.m

Ce fichier gère la partie modulation en BLU du programme. Son fonctionnement sous forme de bloc fonctionnels est le suivant :

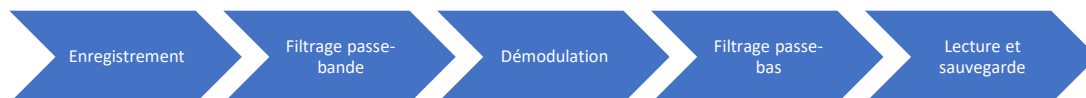


De manière plus explicite, voici le fonctionnement détaillé :

- Le fichier appelle la fonction **RecordModulation** pour enregistrer un message vocal énoncé par un des élèves. Nous avons choisi d'enregistrer un message de 3 secondes. Nous avons, en bonus, programmé la possibilité d'importer le message depuis un fichier audio au format « .wav » et de fréquence d'échantillonnage 96kHz.
- Le message enregistré est filtré grâce à la fonction **PasseBas** avec une fréquence de coupure de 3kHz. Filtrer le message permet d'éviter qu'il dépasse de notre bande de fréquence attribuée.
- Le signal est ensuite à proprement dit modulé en utilisant les équations détaillées à la partie précédente.
- Le signal modulé est ensuite lu en boucle par Matlab grâce à la fonction **sound** et enregistré sous la forme d'un fichier audio (utile pour débayer).

DemodulationBLU.m

Ce fichier gère la partie démodulation du programme. Son fonctionnement sous forme de bloc fonctionnels est le suivant :



De manière plus explicite, voici le fonctionnement détaillé :

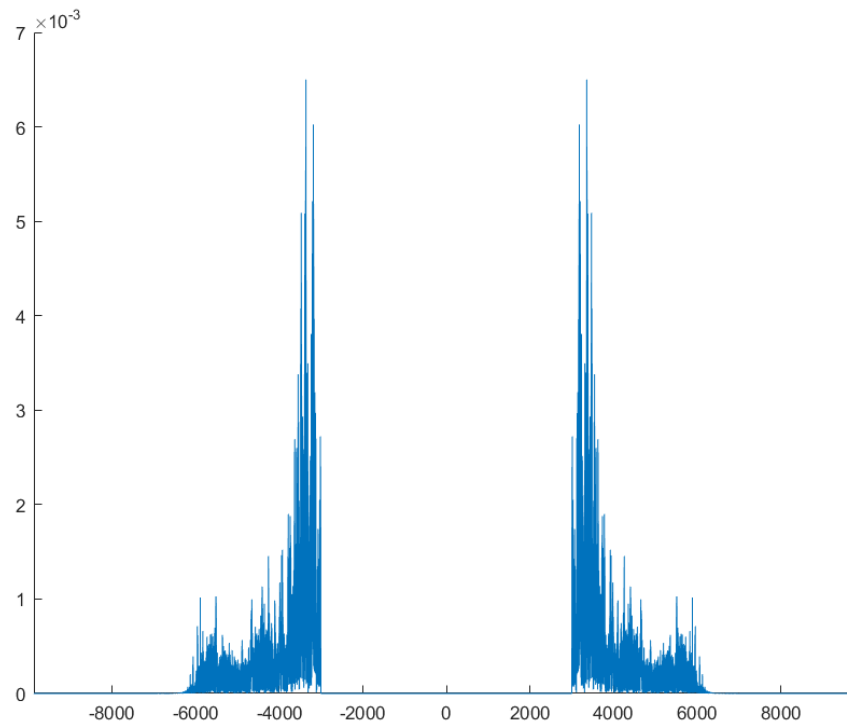
- Le fichier appelle la fonction **RecordModulation** pour enregistrer le message modulé qui est lu par **ModulationBLU**. Nous avons aussi programmé la possibilité d'importer le message modulé depuis un fichier audio au format « .wav » et de fréquence d'échantillonnage 96kHz (utile pour débayer).
- Le message enregistré est filtré grâce à la fonction **PasseBande** pour isoler chaque message. Le premier a une fréquence est défini entre 0 et 3kHz, et le second entre 3kHz et 6kHz.
- Le signal est ensuite démodulé en multipliant le signal par la porteuse $\cos(2\pi v_0 t)$, où v_0 est la fréquence de notre porteuse, c'est-à-dire 3kHz.
- Le message enregistré est filtré grâce à la fonction **PasseBas** avec une fréquence de coupure de 3kHz. Filtrer le message permet d'éliminer les composantes de fréquence $-2v_0$ et $2v_0$ (cf partie démodulation du bureau d'étude).
- Le signal démodulé est ensuite lu par Matlab grâce à la fonction **sound** et enregistré sous la forme d'un fichier audio.

PasseBande.m (fourni)

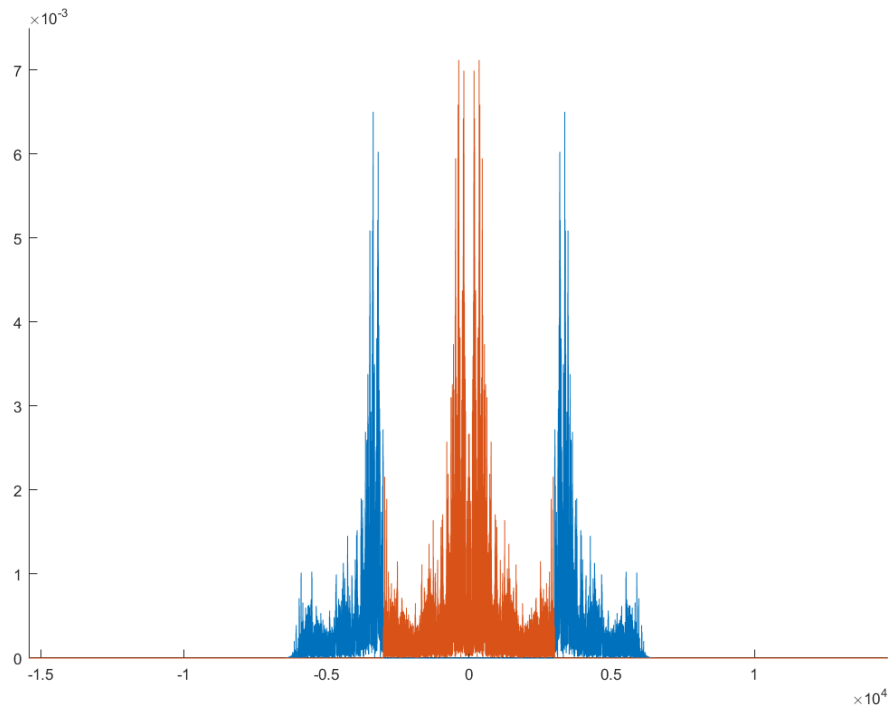
Cette fonction permet de créer un filtre passe bande entre 2 fréquences fournis. Ce programme nous a causé pas mal de soucis et nous n'avons pas réussi à le faire fonctionner correctement. A chaque fois, le signal qu'il retournait était rempli de Nan, ce qui nous a empêché de valider le programme de démodulation en intégralité.

Figures obtenues

La figure la plus intéressante de cette partie est la figure du spectre du signal émis :



On distingue vraiment bien l'effet de la modulation en BLU : le centre du signal bleu est nul. Cela permet alors d'introduire un autre signal dans l'interstice. Le spectre du signal émis est bien la somme du spectre des 2 signaux, comme on le voit sur cette figure :



Interprétations et analyses critiques des résultats

Cette seconde méthode permet effectivement de transmettre 2 messages en même temps au lieu d'un seul. C'est donc une grande augmentation de l'optimisation de l'utilisation de la bande passante. En revanche, nous ne sommes pas parvenus à extraire les 2 messages en même temps : si un seul message est émis on arrive bien à le récupérer, cependant les problèmes rencontrés avec la fonction **PasseBande** nous ont empêchés de valider cette partie du TP.

Conclusion

Lors de notre projet nous avons réussi à remplir le premier cahier des charges en transmettant et réceptionnant un message en parallèle de 3 autres.

Malheureusement, nous n'avons pas réussi à mettre au point la partie démodulation de la seconde partie du TP. Néanmoins, la partie modulation fonctionne comme prévu.

Annexes

Codes M atlab, fournis dans le fichier annexe.