Sorbonne Université
Département de Sciences de l'Ingénieur
Mentions Mécanique, E3A, Automatique/Robotique

Traitement Numérique du Signal - Cahier de TP

S. Argentieri, R. Marchiano

Version 3.1

TP1 : Analyse d'un écoulement turbulent à partir de Transformées de Fourier Discrètes

Remarques préliminaires

— il n'y a pas de compte-rendu à rendre à la fin de la séance mais vous pouvez (devez) prendre des notes et poser des questions pendant la séance à distance,

— les fichiers nécessaires pour faire ce TP sont disponible sur Moodle,

— ce TP est réalisé depuis un Notebook Python fourrni sur Moodle,

— ce TP ne nécessite pas de connaissances approfondies de mécanique des fluides.

1 Présentation du problème

Un écoulement de couche de mélange est l'écoulement produit lorsque 2 écoulements de vitesses différentes se rejoignent. C'est un écoulement de base en mécanique des fluides car il permet d'observer le développement spatial des tourbillons et surtout la fusion de ces tourbillons entre eux qui est responsable du développement spatial de la couche de mélange mais surtout de source de bruit dans le cas d'écoulements à haute vitesse.

Lorsque l'on effectue une mesure d'un champ de vitesse ou de la pression dans un écoulement turbulent, le signal de vitesse obtenu met en évidence des mouvements périodiques dus à la présence des mouvements tourbillonnaires (qui passent de façon quasi-périodique) de taille différente. L'objectif de ce TP est d'utiliser les Transformées de Fourier Discrètes pour analyser les mouvements tourbillonnaires de cet écoulement (cf la vidéo sur Moodle).

Description des données utilisées à post-traiter

La figure 1 montre un champ instantané de vorticité de cet écoulement. La vorticité correspond au rotationnel du champ de vitesse et permet de mettre en évidence les zones tourbillonnaires de cet écoulement. Sur cette figure, on peut observer qu'en amont des tourbillons de petite taille se forment. Puis lors de leur développement spatial, ils fusionnent pour former des plus gros tourbillons et ainsi de suite. 3 sections sont ainsi retenues (x_1, x_2, et x_3) dans lesquelles un signal du champ de pression a été extrait en un point de l'écoulement. Vous disposez donc de 3 signaux de pression $p_1[n]$, $p_2[n]$ et $p_3[n]$ dépendant seulement du temps. Ces données sont contenues dans les fichiers `pression_x1.mat`, `pression_x2.mat` et `pression_x3.mat` respectivement.

Tous ces signaux contiennent $n_t = 6666$ points et leur fréquence d'échantillonnage est $f_e = 1333$Hz soit une durée totale d'acquisition de 5 s.

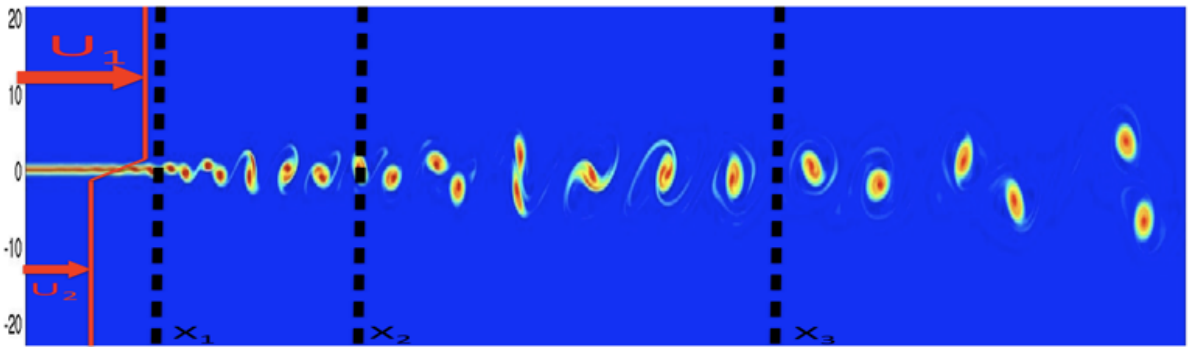

FIGURE 1 – Champ de vorticité instantané issu d'un écoulement de couche de mélange bidimensionnel.

2 Représentations temporelles et fréquentielles des signaux p1, p2 et p3

Télécharger l'ensemble des documents disponibles sur Moodle pour le TP1, et les placer dans un même dossier à l'emplacement de votre choix. Ouvrez ensuite le notebook python `TP1_v2-Etu.ipynb` depuis Jupyter.

2.1 Représentation temporelle

1. Lecture des données : charger les données et créer un tableau de dimension n_t, noté T contenant les échantillons temporels.

Listing 1 – Chargement des données de pression depuis Python

```
x1 = mio.loadmat('pression_x1.mat')
p1 = np.array(x1['p1'])[0]
```

   Tracer le signal $p1[n]$ en fonction de $T[n]$.
2. Quelles sont les valeurs moyennes des signaux $p1[n]$, $p2[n]$ et $p3[n]$? A partir de ces valeurs, construire les 3 signaux de pression fluctuante à moyenne nulle. Ces signaux seront toujours utilisés par la suite et notées $p1m[n]$, $p2m[n]$ et $p3m[n]$.
3. Représenter sur un même graphique ces 3 signaux de pression fluctuante en fonction du temps. Limiter la durée temporelle de représentation (par exemple prendre une durée de 0.25s) pour mieux visualiser cette représentation. Qu'observez vous ?

2.2 Représentation fréquentielle

4. Construire le vecteur contenant les fréquences discrètes. Comment est défini ce vecteur ?
5. Calculer les spectres associés à ces 3 signaux de pression. On les notera $p1mw[k]$, $p2mw[k]$ et$p3mw[k]$ (on pourra consulter l'aide en ligne de la fonction `fft` pour comprendre son utilisation). Tracer sur trois figures différentes (ou superposer sur une même figure) le module des spectres de ces signaux. On pourra limiter l'axe des fréquences à $[-50 : 50]$Hz.
   Quelles différences observez-vous entre les différents spectres ?
   Comment expliquez-vous ces différences ?

3 Influence de la forme de la fenêtre d'analyse

3.1 Diminution de la durée d'analyse

En fait, les données sont issues d'une simulation numérique basée sur la résolution des équations de Navier-Stokes. Les ressources informatiques nécessaires à cette simulation sont importantes et le temps de simulation (très long) dépend du nombre d'échantillons final. En pratique, on ne dispose jamais d'autant d'échantillons que les signaux p_1, p_2 et p_3. On suppose à partir de cette question que $n_{t2} = 1998$ points.

1. Construisez le signal $p1mc[n]$ qui contient les n_{t2} premières valeurs de $p1m[n]$. Tracer ensuite sa représentation temporelle ainsi que son spectre.
2. Quelles différences observez-vous par rapport à la configuration précédente ?

3.2 Etude de trois fenêtres différentes

Pour remédier à ce problème on se propose de changer la forme de la fenêtre d'analyse. On va étudier l'influence de la forme à travers l'étude de trois fenêtres différentes :

— une fenêtre rectangulaire :

$$w_R[n] = rect_M[n] = \begin{cases} 1 & \text{si } -M/2 \leq n \leq M/2-1 \\ 0 & \text{ailleurs} \end{cases} \quad (1)$$

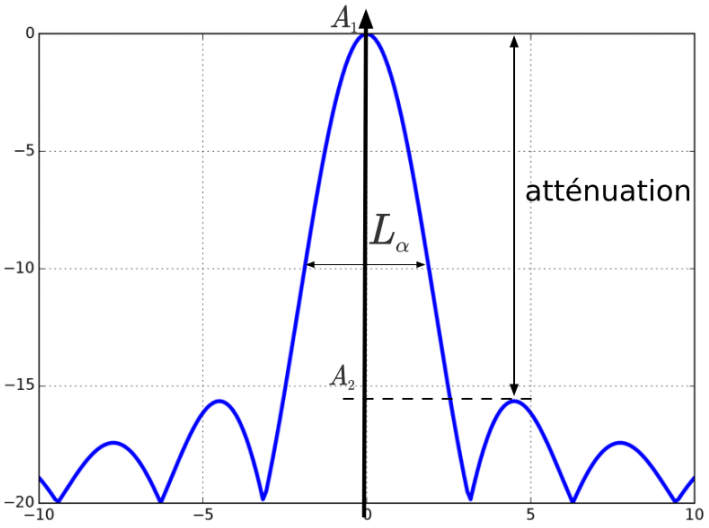

FIGURE 2 – Illustration des concept de lobe principal et lobes secondaires

— une fenêtre triangulaire :

$$w_T[n] = tri_M[n] = \begin{cases} 1 + \frac{2n}{M} & \text{si } -M/2 \leq n < 0 \\ 1 - \frac{2n}{M} & \text{si } 0 \leq n \leq M/2 - 1 \\ 0 & \text{ailleurs} \end{cases} \tag{2}$$

— une fenêtre de Hann :

$$w_H[n] = w_R[n] \left(0.5 + 0.5 \cos \left(\frac{2\pi n}{M} \right) \right) \tag{3}$$

Pour chacune de ces fenêtres, tracez la représentation temporelle pour $n \in [-100 : 100]$ et $M = 60$ et déterminer les propriétés de leurs spectres associés : la largeur du lobe principal (notée $L\alpha$ avec α le type de fenêtre) et l'atténuation en dB de la hauteur du lobe secondaire par rapport à la hauteur du lobe principal (ces notions sont illustrées sur la figure 2). Pour cela, tracez la quantité suivante :

$$A_{dB} = 20 \log \left(\frac{|W_\alpha(f)|}{|W_\alpha(f=0)|} \right)$$

où W_α représente le réponse en fréquences des différentes fenêtres. Cela permet de comparer toutes les figures avec la même référence.

1. Fenêtre rectangulaire. Tracer $w_R[n]$ et le module de la TFD de $w_R[n]$. Déterminer la largeur du lobe principal et l'atténuation entre le lobe principal et le lobe secondaire.
2. Fenêtre rectangulaire. Tracer $w_T[n]$ et le module de la TFD de $w_T[n]$. Déterminer la largeur du lobe principal et l'atténuation entre le lobe principal et le lobe secondaire.
3. Fenêtre de Hann. Tracer $w_H[n]$ et le module de la TFD de $w_H[n]$. Déterminer la largeur du lobe principal et l'atténuation entre le lobe principal et le lobe secondaire.

3.3 Application des fenêtres triangulaires et de Hann à l'analyse des signaux de couche de mélange

Dans cette partie, le nombre de points des fenêtres w_R, w_T et w_H est égal à $M = n_{t2}$.

1. Appliquer la fenêtre rectangulaire au signal $p1mc[n]$, on notera $p1R = p1mc[n].w_R[n]$ le résultat. Tracer la représentation temporelle et fréquentielle. Y a t-il une différence avec $p1mc[n]$?

2. Appliquer la fenêtre triangulaire au signal $p1mc[n]$, on notera $p1T = p1mc[n].w_T[n]$ le résultat. Tracer la représentation temporelle et fréquentielle.
3. Appliquer la fenêtre de Hann au signal $p1mc[n]$, on notera $p1H = p1mc[n].w_H[n]$ le résultat. Tracer la représentation temporelle et fréquentielle.
4. Quelles différences y a t il entre les trois représentations fréquentielles obtenues dans cette partie et la représentation temporelle de $p1m$? Expliquer ces différences.

Chargement des librairies python nécessaires :

```
import numpy as np
import pylab as plt
```

Exemple de fonctions PYTHON utilisées dans ce TP1 (et TP2 !) :

`plt.figure`	faire un graphe
`plt.plot`	effectuer une représentation graphique
`plt.xlabel`	mettre une légende (axe des abscisses)
`plt.ylabel`	mettre une légende (axe des ordonnées)
`plt.axis`	fixer les bornes du domaine (abscisses et ordonnées)
`plt.title`	mettre un titre à un graphe
`np.fft.fft`	effectuer une TFD rapide
`np.fft.fftfreq(NT, 1/Fe)`	crée un vecteur de fréquence sur NT points entre 0 et Fe
`np.fft.fftshift`	décale un vecteur de fréquence compris entre 0 et Fe entre -Fe/2 et Fe/2
`np.arange(a,b,c)`	générer un vecteur de a à b avec un pas c
`abs`	calculer le module d'un nombre
`np.mean`	permettre de calculer une valeur moyenne

TP 2
Démultiplexage stéréophonique

Introduction et principe

Aujourd'hui, l'ensemble des données sonores rendues disponibles par la radio ou par la télévision est de nature stéréophonique. Néanmoins, historiquement, cela n'a pas toujours été le cas. A l'origine, les transmissions étaient en effet *monophoniques*, simplement restituées à partir d'un seul haut-parleur. Ainsi, lors du passage à la stéréo, il a fallu veiller à ce que l'information audio stéréophonique soit représentée de sorte que les anciens équipements monophoniques restent compatibles. De plus, de nombreux postes radio basiques (radio-réveils, etc.) sont aujourd'hui toujours en vente, et diffusent un son uniquement monophonique.

Concrètement, la création, la transmission et la réception d'un signal stéréophonique s'opèrent de la façon suivante (cf. figure 1) :

— Tout d'abord, *un seul et unique signal* $x_{\text{stéréo}}[n]$, contenant l'information stéréophonique gauche et droite, est obtenu à partir d'un *multiplexeur stéréophonique* ;

— Ensuite, cet unique signal est transmis (par exemple, à l'aide d'une modulation de fréquence) ;

— Enfin, l'appareil de réception doit démoduler le signal reçu afin d'obtenir le signal $\hat{x}_{\text{stéréo}}[n]$, signal stéréophonique reçu qu'on espère quasi-identique au signal $x_{\text{stéréo}}[n]$ émis.

La contrainte principale pour une telle transmission, comme mentionné auparavant, est que le signal doit tout de même pouvoir être reçu par des récepteurs non équipés pour la stéréophonie. Cette compatibilité avec les postes monophoniques est obtenue en combinant les signaux des voies droite (D) et gauche (G) et en leur attribuant une occupation spectrale bien identifiée (et normalisée). Ainsi, l'allure du contenu fréquentiel (TFSC) du signal stéréophonique $x_{\text{stéréo}}(t)$ est représenté sur la figure 2.

L'objectif de ce TP est de proposer une solution, à base de filtrage numérique, permettant d'extraire du signal $x_{\text{stéréo}}[n]$ les deux signaux d'origine G et D. Idéalement, ces deux signaux doivent être parfaitement séparés, de façon à garantir une reproduction stéréophonique de qualité. Nous verrons que cela n'est possible que sous certaines conditions que nous allons mettre en évidence dans la suite.

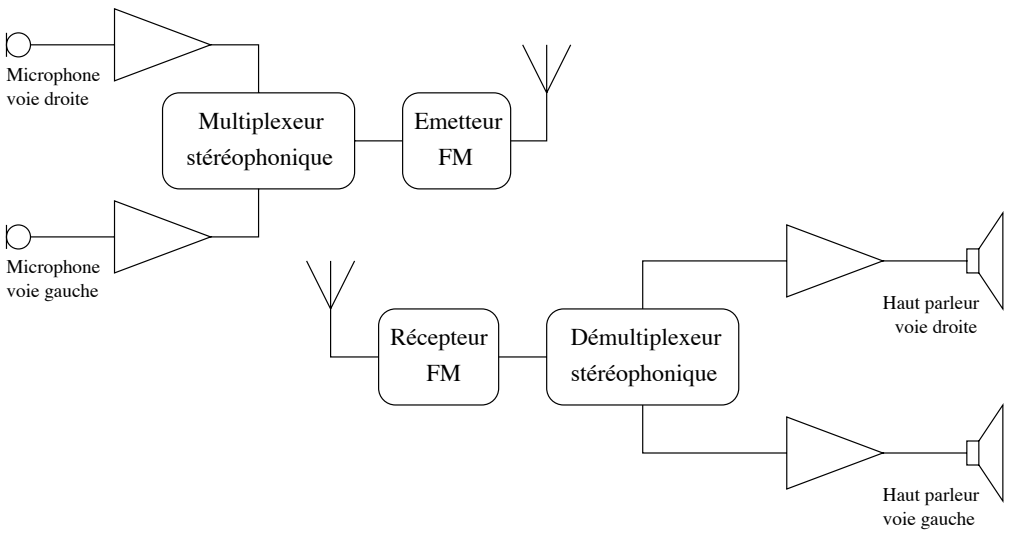

FIGURE 1 – Principe d'une transmission stéréophonique.

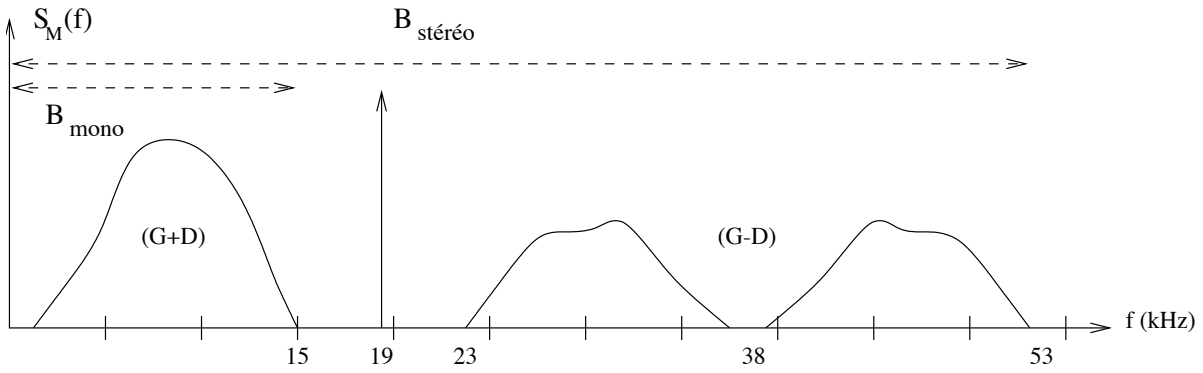

FIGURE 2 – Allure du spectre d'un signal stéréo

1 Analyse du signal stéréophonique

Nous allons dans un premier temps analyser le signal stéréophonique $x_{\text{stéréo}}[n]$, et vérifier qu'il possède les caractéristiques attendues.

1.1 On suppose que le signal stéréophonique est échantillonné à une fréquence $f_e = 132300$ Hz. Justifier le choix de cette fréquence d'échantillonnage.

1.2 Charger le signal stéréophonique et tracer son allure temporelle. Commenter le résultat obtenu.

Listing 2 – Charger un fichier Matlab sous Python

```
from scipy.io.matlab import mio
x = mio.loadmat('stereo.mat')
stereo = np.array(x['stereo'])[0]
Fe = 132300
```

1.3 Analyser en fréquence le signal stéréophonique. Montrer que l'on retrouve un contenu fréquentiel proche de celui attendu (représenté figure 2). Préciser alors la bande passante du signal.

2 Démultiplexage stéréophonique

Maintenant que nous avons étudié les caractéristiques du signal stéréphonique, nous allons chercher à en extraire les deux signaux gauche (G) et droite (D). Pour cela, l'architecture suivante est utilisée :

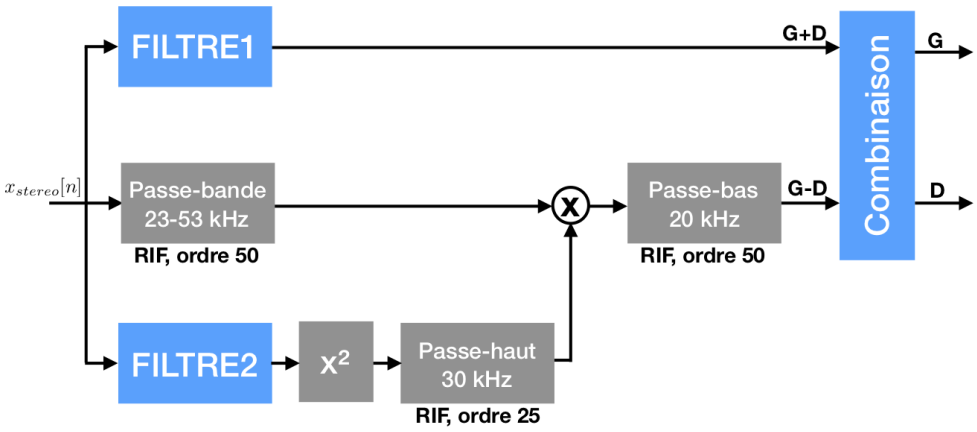

FIGURE 3 – Architecture du démultiplexeur. 2 filtres sont à synthétiser.

2.1 Extraction du signal monophonique : synthèse du filtre `FILTRE1`

Une première étape du traitement visant à extraire les signaux G et D seuls consiste à extraire tout d'abord le signal G+D. Celui-ci correspond au signal monophonique qui sera émis par tous les postes non compatibles avec la stéréo. Cette opération est effectuée par un filtre numérique **FILTRE1** appliqué au signal stéréo $x_{stereo}(t)$.

2.1.1 Sur quelle bande de fréquence se situe le signal G+D ? En déduire la nature (passe-haut, passe-bas, passe-bande) du filtre permettant d'extraire G+D du signal stéréo.

2.1.2 Proposer alors le gabarit du filtre à synthétiser.

2.1.3 Synthétiser un filtre RIF et RII satisfaisant le gabarit précent à l'aide de la commande Python `filtre()` fourni.

Listing 3 – Synthétiser un filtre depuis Jupyter

```
FILTRE = filtre()
fig, ax = plt.subplots(1,3,figsize=(10,3))
FILTRE.display(ax, fig)
```

2.1.4 Comparer les réponses en fréquence (en gain et phase !) des deux filtres obtenus. En quoi ces réponses sont-elles différentes ? Quel est, selon vous, le "meilleur" filtre ?

2.1.5 Visualiser les réponses impulsionnelles des deux filtres RIF et RII obtenus. Commenter et comparer leur allure.

Listing 4 – Calcul de la réponse impulsionnelle

```
n, rep_imp = FILTRE.impulse_response()
```

2.1.6 Utiliser les filtres synthétisés pour filtrer le signal stéréophonique $x_{stereo}[n]$ et récupérer ainsi le signal monophonique $x_{mono}[n]$.

Listing 5 – Appliquer un filtre FILTRE d'entrée x et sortie y

```
n, y = FILTRE.filter(x)
```

2.1.7 Déterminer et afficher le spectre du signal monophonique. Vérifier que le filtre précédent a bien supprimé les composantes fréquentielles souhaitées.

2.1.8 Ecouter le signal monophonique et conclure.

Listing 6 – Ecouter un signal x de fréquence d'échnatillonnage Fe

```
display(Audio(x_mono_rif, rate=Fe))
```

2.2 Extraction de la porteuse

Nous venons de déterminer un filtre **FILTRE1** permettant d'extraire du signal stéréophonique l'information monophonique G+D. Il s'agit maintenant de déterminer le filtre numérique **FILTRE2** permettant d'extraire la porteuse à 19kHz située au sein du signal stéréophonique. Cette porteuse va permettre, grâce à l'utilisation d'autres filtres numériques (en gris sur le schéma de la figure 3), d'extraire *in fine* le signal G-D.

2.2.1 Compte tenu du spectre du signal stéréophonique représenté figure 2, préciser la nature (passe-haut, passe-bas, passe-bande) du filtre **FILTRE2** permettant d'extraire la porteuse du signal stéréo. Préciser alors son gabarit en fréquence.

2.2.2 Synthétiser le filtre souhaité. Tracer sa réponse en fréquence.

2.2.3 Filtrer le signal stéréphonique avec le filtre **FILTRE2** obtenu. Vérifier son bon fonctionnement en traçant l'allure temporelle et le contenu fréquentiel de son signal de sortie. Conclure.

2.3 Mise en œuvre du démultiplexeur

Nous disposons maintenant des 2 filtres **FILTRE1** et **FILTRE2** nécessaire à la mise en œuvre du multiplexeur complet. Il s'agit maintenenant de récupérer les 2 signaux **G+D** et **G-D** grâce à cette architecture. Pour cela, vous disposez de la fonction `demultiplex(stereo, FILTRE1, FILTRE2)` qui implémente pour vous l'architecture complète de la figure 3 et prend en paramètre les 2 filtres que vous avez synthétisé.

2.3.1 Utiliser la fonction `demultiplex()` pour extraire les 2 signaux G+D (monophonique) et G-D du signal stéréophonique.

Listing 7 – Utilisation de la fonction demultiplex() au signal x

```
n, g_plus_d, g_moins_d = demultiplex(x, FILTRE1, FILTRE2)
```

2.3.2 Combiner les deux signaux G+D et G-D de façon à obtenir les signaux gauche G et droite D seuls. Tracer leurs allure temporelles.

2.3.3 Ecouter les signaux obtenus. A-t'on réussi à séparer les signaux gauche G et droite D ? Expliquer.

2.3.4 Modifier en conséquence les filtres **FILTRE1** et **FILTRE2** pour que le démultiplexage fonctionne correctement. Tracer l'allure des signaux obtenus, et vérifier en écoutant les signaux que les modifications apportées fonctionnent. Conclure