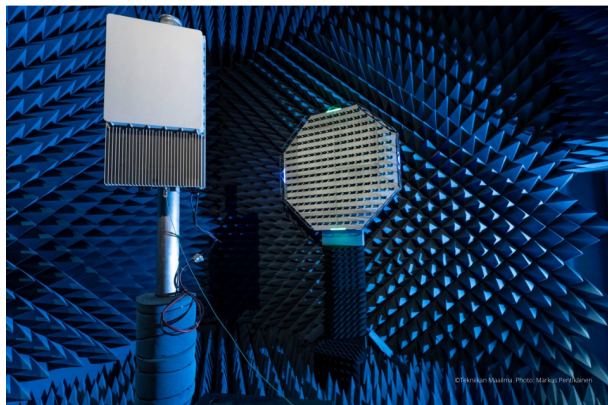


TRAITEMENT DU SON AVANCÉE

BEAMFORMING APPROACHES



Equipe :

ZAKARIA BOUHAFS

N°28612466

BRANIS GHOUL

N°28612738

Encadrants :

P.SYLVAIN ARGENTIERI

Promotion 2022 - 2023

December 2, 2022

Nous avons caractérisé et analysé la propagation du son dans la pratique précédente. Nous allons maintenant exploiter ces propriétés pour déduire la position d'une source sonore par rapport à un réseau linéaire de microphones composé de $N = 8$ microphones MEMS omnidirectionnels. Le système que nous allons utiliser est le même que précédemment.

Le Beamforming consiste à appliquer un filtre sur les signaux de chaque microphone et à additionner leurs sorties pour former le signal du beamformer, (voire fig 1). Comme expliqué pendant le cours, on utilisera des filtres dont la réponse en fréquence $W_n(f)$ est donnée par :

$$W_n(f) = TF[w_n(t)] = e^{-j2\pi \frac{f}{c} \cos \theta_0}$$

où x_n est la position du n^{ime} microphone le long de son axe, et θ_0 est la direction angulaire dans laquelle le beamformer est focalisé. .

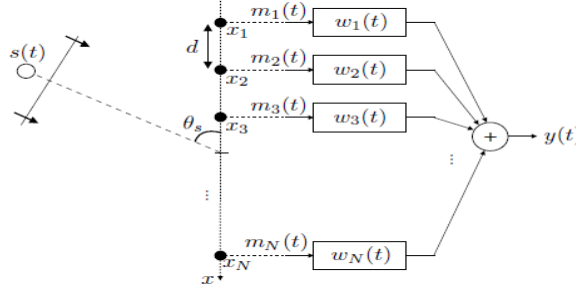


Figure 1: Experimental arrangement

Dans tout ce qui suit, nous travaillerons avec une fréquence d'échantillonnage $F_s = 20\text{kHz}$, et avec un buffer de taille $\text{BLK} = 2048$.

1. To begin, start the acquisition of the audio system, and capture one audio buffer. Plot the resulting signals as a function of time.

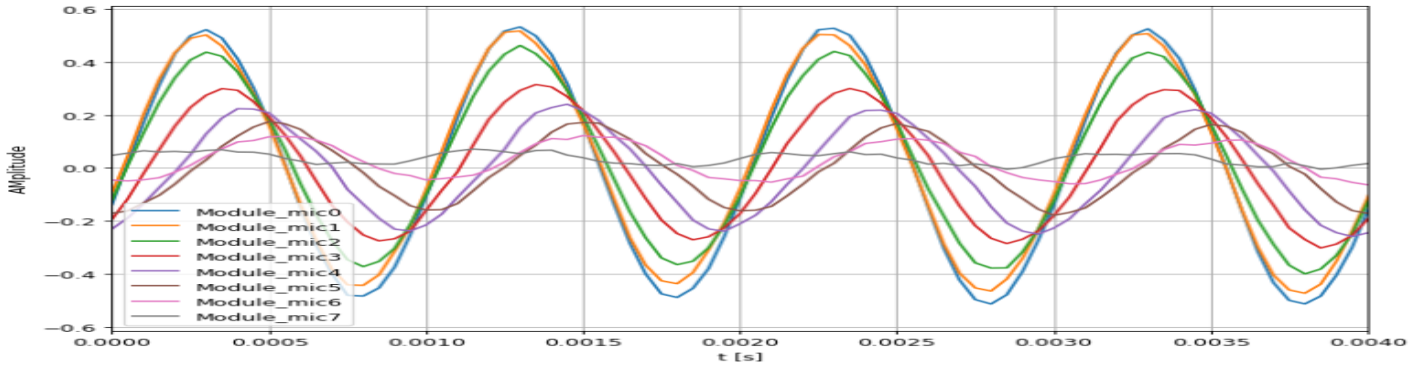


Figure 2: Représentation temporelle du signal acquis par les 8 micros

on remarque que le signal acquis est bien sinusoïdale, ce qui correspond au signal émis par notre source audio, de légères distorsions sont visible et elles sont dûes à l'environnement de travail.

2.1 Coding the beamformer filters and analyzing their properties.

- Write the position x_n as a function of s and interspace d . As a convention, the first microphone number is selected as 0, and the origin of the frame is placed at the center of the array.

$$z_n = d \times \left(n - \frac{N+1}{2}\right)$$

- Propose a function `beam_filter` returning the filter frequency response for one microphone number `mic_nb`.

```
def beam_filter(array, freq_vector, theta0=0, mic_nb: int = 0):
    """Compute the filter frequency response of a DSB beamformer for one microphone

    Args:
        array (array_server obj): array structure controlling the acquisition system.
        freq_vector (np.array): frequency vector.
        theta0 (int, optional): focusing angular direction(in degrees). Defaults to 0.
        mic_id (int, optional): microphone id. Defaults to 0.

    Returns:
        np.array: the filter frequency response. Shape is (len(freq_vector),).
    """
    N = array.mems_nb
    d = array.interspace
    # Microphone position x
    z = d*(mic_nb-((N+1)/2))
    # Filter's frequency response
    W = np.exp(-1j*2*np.pi* (freq_vector/340) *z*np.cos(theta0 * np.pi/180))
    return W
```

4. Plot the two frequency responses obtained for two filters associated to two different microphone outputs when $\theta_0 = 0^\circ$ and for frequencies between 0 and 5kHz. Explain the effect of these filters on the signals.

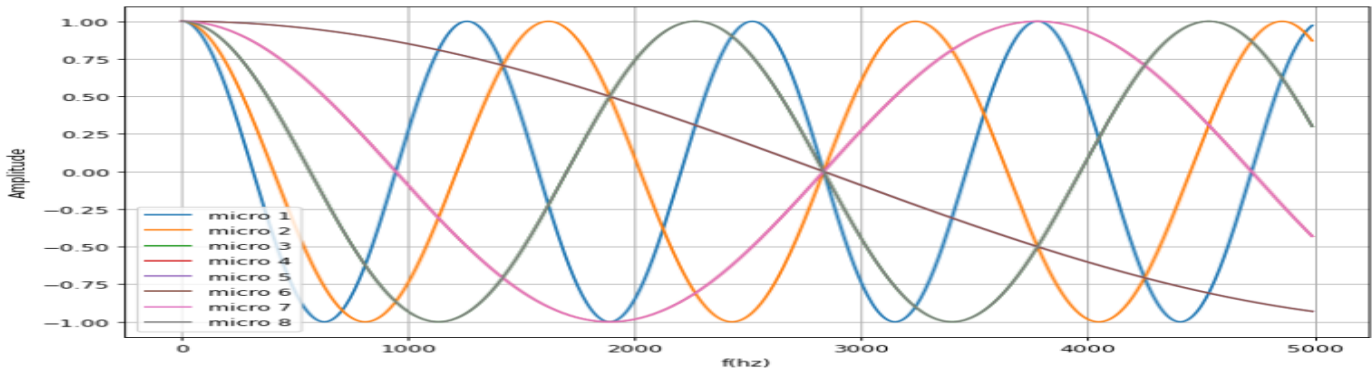


Figure 3: Filtre sur chaque micros pour $\theta_0 = 0^\circ$

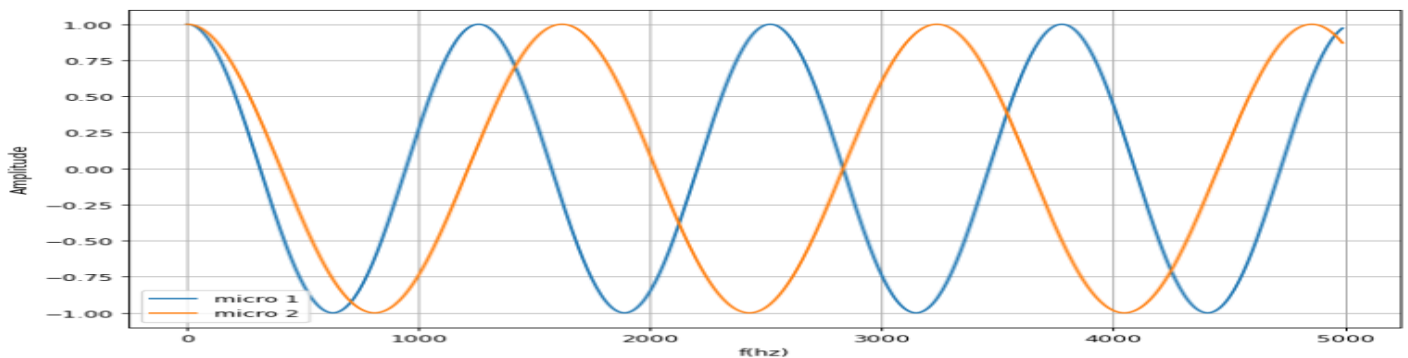


Figure 4: Filtre sur deux micros pour $\theta_0 = 0^\circ$

- On remarque que pour $f_{1_1} = f_{1_2} = 0$ Hz, le premier pics des deux signaux est le même, or que pour le deuxième pic il sera pour le microphone N°1 à une fréquence f_{2_1} égale à environ 1245 Hz et pour le microphone N°2 à une fréquence f_{2_2} environ 1615 Hz. Même remarque troisième pic, $f_{3_1} = 2515$ Hz et $f_{3_2} = 3245$ Hz.
- Quand on applique les filtres à nos 8 microphones on ajoute un retard en fonction de θ_0 , en fonction de ce dernier et lors de la sommation on aura des ondes destructives ce qui nous permettra d'avoir une bande passante de theta pour la quelle notre array laissera passer le son.

5. Compare again the filters obtained when $\theta_0 = 90^\circ$. Explain the differences.

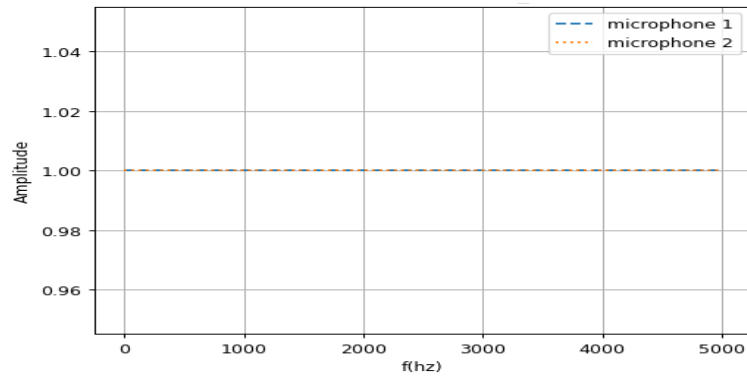


Figure 5: Filtre sur deux micros pour $\theta_0 = 90^\circ$

- Pour $\theta_0 = 90^\circ$ on aura $\cos 90^\circ = 0$, dans ce cas l'exponentielle vaut 1. Ainsi chaque filtre sera égal à 1 et n'auront aucun effet sur les micros.
- Dans le domaine fréquentielle l'effet du beamforming vu en cours sera constater lors de la sommation de chaque microphones, on va choisir uniquement les angles à 90° avec plus au moins quelques degrés.

2.2 Using the filters : coding of the beamforming.

6. **Step (a) and (b) :** After acquiring an audio buffer, compute its FFT in an array `M_fft`. Plot the result of this analysis as a function of the frequency when emitting a pure sine tone with a frequency $F_0 = 1\text{kHz}$.

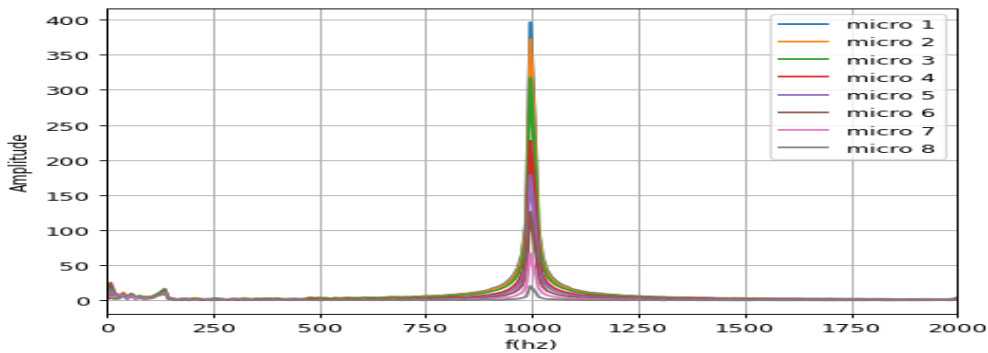


Figure 6: Module de la transformée de fourier d'un signal à 1000Hz

On commence notre analyse sur l'axe des fréquences, on remarque bien des pics au alentour des 1000 Hz qui est tout à fait logique car c'est notre fréquence d'intérêt, néanmoins un léger bruit est présent entre 0 et 200 Hz. Ce bruit est justifié par plusieurs facteur : le bruit du néon (entre 50 et 70 Hz) , le bruit des peronnes qui parlent durant l'enregistrement (entre 75 à 140 Hz pour les hommes et 170 à 250 Hz pour les femmes).

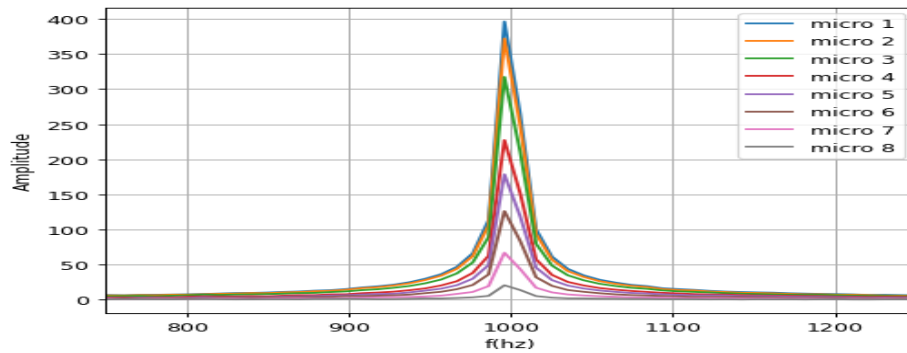


Figure 7: Module de la transformée de fourier d'un signal à 1000Hz

- On poursuit notre analyse sur l'axe d'amplitude, on remarque que l'amplitude du microphone N°1 est 16 fois plus importante que celle du microphones N°8 ce qui tout à fait logique car L'enregistrement a été effectué avec un angle theta.
- étant donné que la source est proche (pas vraiment à l'infini), le téléphone étant très orienté vers le micro 1 et 2, leur amplitude est plus grande.
- Contrairement au domaine temporel, dans le domaine fréquentiel on ne remarque pas de retard.

7. **Step (c) and (d)** : Among all the frequencies you obtained from the FFT, select the one corresponding to the source frequency. Give its exact value and index k_0 in the frequency array, and collect the corresponding FFT values of each microphone outputs in one vector M of length N .

```
k0 = (np.argmax(np.abs(Freq - F0)))
print("k0:", k0)
print("Frequence_a_k0:", Freq[k0])
M = []
Mft = np.fft.fft(m)
M = Mft[:, k0]
print("Valeurs_de_M:")
print(M)
```

On a déterminé l'indice k_0 pour le quel on observe le pics à notre fréquence d'intret qui est de 1000 Hz, ensuite on calcule la FFT et on extrait la réponse de chaque microphones (c'est pour ca que la matrice est composée de 8 valeurs complex).

8. **Step(e)** : In a loop among all microphones, compute each filters for the position θ_0 and the frequency value you obtained in the previous step. Apply then these filters to the array M defined before.

```
W= []
for i in range(N):
    W.append(beam_filter(antenne, Freq[k0], theta0 = 0, mic_nb=i))
W = np.array(W)
```

```
YW = W*M
print(YW)
```

on applique un filtrage pour chaque sortie de microphones.

9. **Step(e)** : Combine then the filters outputs to form the beamformer output $Y_{\theta_0}[k_0]$. Compute then the corresponding power $P(\theta_0)$ at k_0 of the beamformer output.

```
Y_tet0 = np.sum(YW)
P = np.sum(np.abs(YW))**2
print("r_ponse_en_sortie_Y_pour_k0:", Y_tet0)
print("Puissance_pour_k0:", P)
```

On a maintenant la réponse en sortie Y pour k_0 , cette réponse est la somme des sorties des 8 microphones d'où le `np.sum`, la puissance est calculée suivant sa formule classique.

10. For a direction θ_0 of your choice, compute $P(\theta_0)$ for (i) a source emitting from a direction close to θ_0 , or (ii) far from it. Compare the two values.

pour $\theta_0 = 90^\circ$

```
# For theta0 = 90
W = []
for i in range(N):
    W.append(beam_filter(antenne, Freq[k0], theta0 = 90, mic_nb=i))
YW = M * W
Y_tet90 = np.sum(YW)
P90 = np.abs(Y_tet90)**2
print(f"Y_tet90[k0] vaut:", Y_tet90)
print("La puissance P vaut:", P90)
```

pour $\theta_0 = 0^\circ$

```
# For theta0 = 0
W = []
for i in range(N):
    W.append(beam_filter(antenne, Freq[k0], theta0 = 0, mic_nb=i))
YW = M * W
Y_tet0 = np.sum(YW)
P0 = np.abs(Y_tet0)**2
print(f"Y_tet0[k0] vaut:", Y_tet0)
print("La puissance P vaut:", P0)
```

on remarque que la puissance pour $\theta_0 = 90^\circ$ est plus grande que la puissance pour $\theta_0 = 0^\circ$, en effet la source a été placé à environ $\theta_0 = 100^\circ$.

11. **Step(g)** : Repeat now the previous code in a loop for θ_0 values ranging from 0 to 180° . You should then obtain an array P where each value corresponds to the power of the beamformer output at F_0 for each angular polarization. Plot the array P as a function of the angle θ_0 .

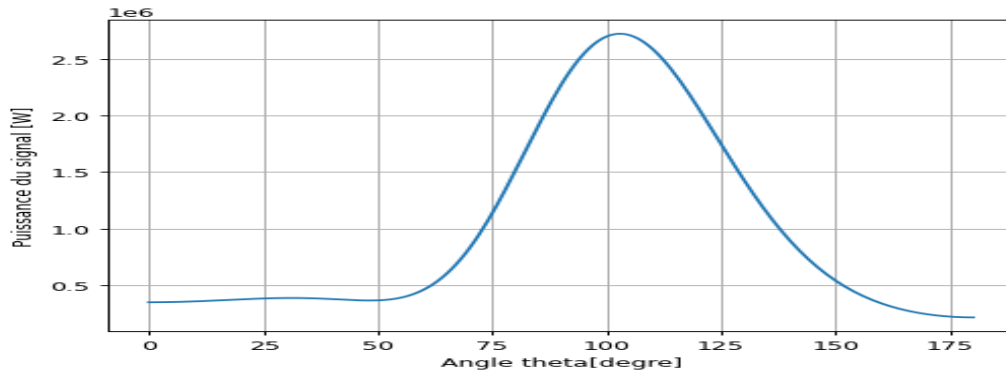


Figure 8: Puissance en fonction de l'angle

Le beamforming nous retourne donc bien un maximum aux alentours de 100 degré.

12. **Step(h)** : Find the θ_0 value corresponding to position of the maximum in P and compare it with the actual (but approximate) position of the sound source.

```
theta_o=np.argmax(Power)
print("max_de_la_puissance_pour_theta:",theta_o )
print("max_puissance",Power[theta_o])
```

Pendant l'acquisition nous avons une réalité terrain de $\theta_0 = 100^\circ \pm 5^\circ$ (nous n'avons pas calculé avec exactitude l'angle). L'angle pour le quel la puissance est maximale est de 103° qui est presque équivalent à la réalité terrain en prenant compte des erreurs de positionnement.

2.3 Analyzing the beamformer performances.

From now on, We can use our own code:

```
def beamforming(m, thetas , F0 , Fs):
    N, BLK = np.shape(m)
    Mfft = np.fft.fft(m)
    Freq = np.arange(0, Fs, Fs/BLK)
    k0 = (np.argmin(np.abs(Freq - F0)))
    #On selectionne pour chaque micro la fr quence d'interet k0
    M = Mfft[:, k0]
    Power = []
    for theta in thetas:
        W = []
        for i in range(N):
            W.append(beam_filter(antenne, Freq[k0], theta0 = theta, mic_r
        YW = M*W
        P = np.abs(np.sum(YW))**2
        Power.append(P)
    return Power
```


13. Plot the energy maps you obtain when using source frequencies $F_0 = 400\text{Hz}$, $F_0 = 1\text{kHz}$, $F_0 = 2\text{kHz}$ and $F_0 = 4\text{kHz}$ emitting from a fixed arbitrary position. Comment and explain carefully the differences between these curves.

On commence par faire des acquisitions pour différentes fréquences (on restarte le kernel à chaque fois pour éviter des problèmes de buffer), ensuite nous calculons le beamforming pour ces différentes fréquences et ensuite on trace le diagramme polaire pour bien représenter le côté spatial du filtrage.

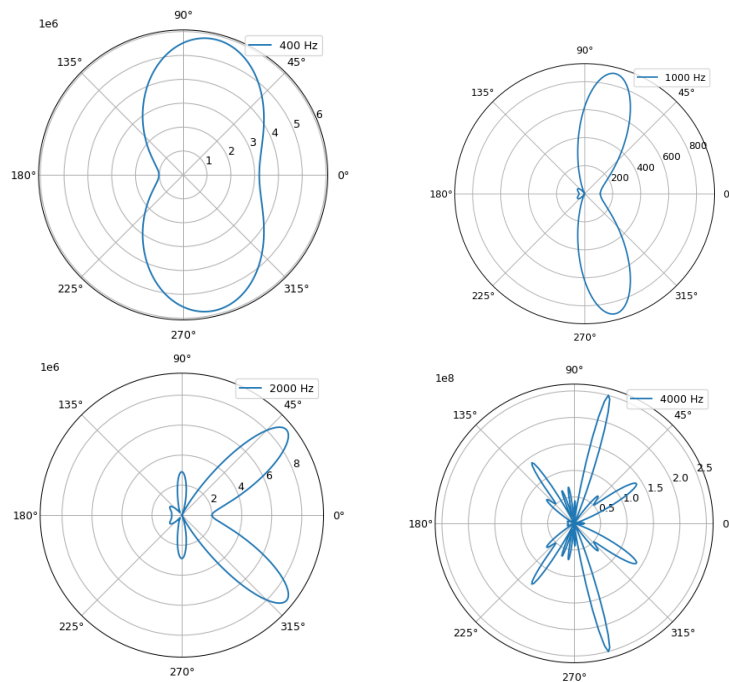


Figure 9: Représentation polaire de la puissance en fonction de l'angle

- D'après la figure 9, le diagramme de directivité d'une antenne varie avec la fréquence d'intérêt. Ainsi, une antenne est plus directive lorsque la fréquence d'intérêt augmente. Cependant, si la valeur maximale de la fréquence d'intérêt est dépassée (non respectant du théorème de Shannon) : $d < d_{max} = \min * c/2f_{max}$, le diagramme de directivité laisse passer des ondes acoustiques provenant de directions différentes non souhaitées, ceci est décrit par un repliement spectral dans le domaine fréquentiel, on note le diagramme est symétrique (0° à 180° et de 180° à 360°), dans notre analyse on va s'intéresser à la première partie seulement.

-Si la source émet un signal de fréquence importante, on constate que le lobe principale semble être plus fin, on retrouve des lobe secondaires en plus grand nombre mais d'amplitudes décroissante (étonnant).

- Fréquences faibles, on récupère des diagrammes d'antenne et tellement plus large jusqu'à ne pas avoir des lobe secondaires. Donc, quand la fréquence diminue, le diagramme d'antenne est de moins en moins sélectif, hors ça peut paraître étonnant, mais finalement les fréquences sont associées à des longueurs d'ondes et en plus on a affaire à un microphone linéaire et équidistant, on est face à un système de formation de voie qui est entrain d'effectuer un échantillonnage spécial de l'information acoustique,

cette échantillonnage et effectuer par un réseau de microphone placer régulièrement dans l'espace et cette échantillonnage est plus au moins adapter à la fréquence de l'onde sonore que avec laquelle nous travaillons, visiblement pour des fréquences faibles associer a de grande longueurs d'onde on a un échantillonnage spécial qui s'opère sur 42 cm et pour qui est associer à une longueur d'onde de 400hz visiblement cette longueur n'ai pas très importante par rapport à la longueur d'onde à échantillonner et donc on récupère une information/sélectivité spatial extrêmement faible.

- Le dernier phénomène, il est extrêmement important, car à présent on ce dit que à chaque fois l'on augmente la fréquence le diagramme d'antenne est sélectif, mais le problème lorsque nos sources émet des fréquences trop hautes, on récupère toujours un premier lobe dans le diagramme d'antenne qui correspond au fréquence d'intérêt (d'habitude un lobe extrêmement fin), des lobe secondaires d'amplitudes qui vont on diminuant, mais aussi d'autres lobes secondaires avec une grande amplitudes! enfaite en est dans un cas analogue de ce que on appelle le repliement spectrale qui directement lier au fait que nous avons pas respecter le théorème de Shannon spatial, qui a pour conséquences visiblement que l'on récupère des onde sonore dans d'autres directions non souhaitées., et concrètement cela veut dire on est incapable de faire la différence entre les source qui émettent des ondes sonores dans les différentes positions et donc incapable de retrouver la position spatial de la source sonore souhaité.

- Pour la fréquence 2000 Hz, la directivité est entre 0° est 45° au lieu être situé entre 45° et 90° comme pour les autres fréquences. On a surement toucher au téléphone(recevoir un appel ou bien la batterie était faible) et sur ce fait le position de la source est différente des autres fréquences(400 Hz, 1000 Hz, 4000 HZ)

14. For a frequency $F_0 = 1\text{kHz}$ and a source moving around the array, plot the estimated position as a function of time. Comment the effectiveness of the approach and its limits.

La taille du Buffer est de 2048 (BLK=2048) avec $F_s = 20000\text{Hz}$ ce qui nous fait 0.1s, l'objectif est de récupérer à chaque pas de temps les signaux reçu par les 8 microphones, on applique la fonction beamforming, ensuite on estime l'angle en fonction du temps grace au max de la puissance. Notre audio contient environs 11s , on va utiliser que les 6 premiere second (le reste n'est pas utile) , donc 6s/0.1s on aura besoin de 60 itérations, on prends 70 par sécurité.

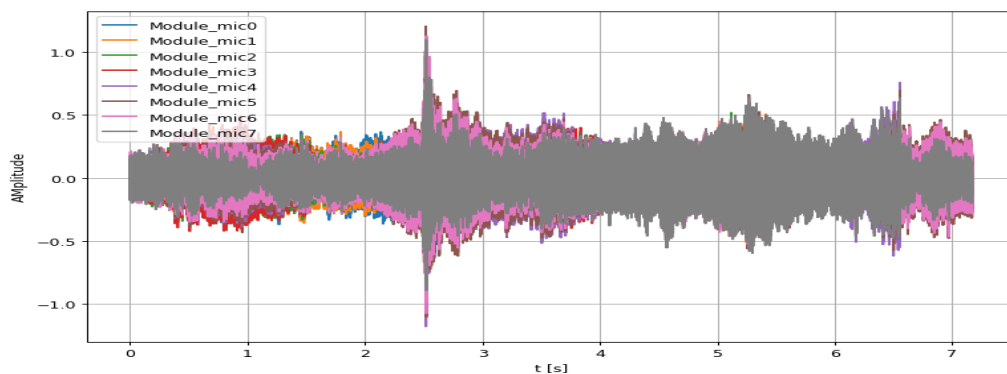


Figure 10: Représentation temporelle du signal acquis par les 8 microphones

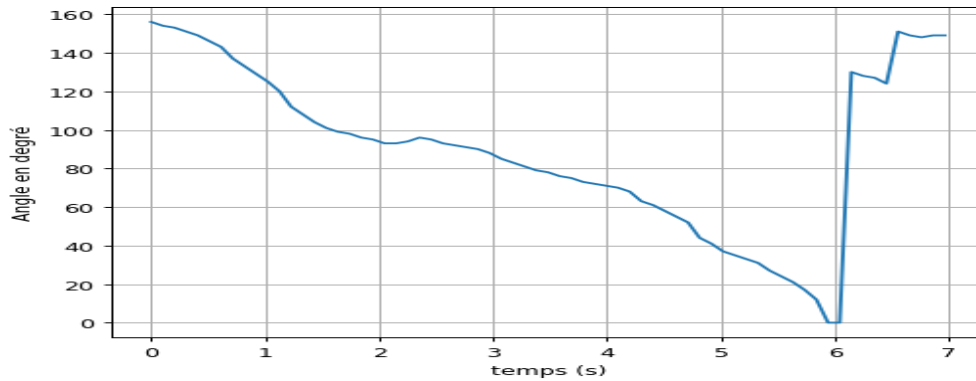


Figure 11: Angle estimé en fonction du temps

- On devrait trouver une allure de l'angle θ qui augmente ou diminue en fonction du temps et en fonction de notre trajectoire (de 0° à 180° ou de 180° à 0°). Si nous partons de 0 à 180° la fonction augmente, sinon, l'inverse.
- Dans notre cas on part de 180° à 0° , néanmoins on commence par un maximum de 160° , cela est dû à une mauvaise synchronisation entre la personne qui fait les déplacements de la source sonore et la personne devant l'ordinateur portable qui lance l'acquisition.
- On a vu que $f = 1000\text{Hz}$ n'était pas très sélectif, Plusieurs directions ont une énergie élevée ce qui peut fausser le résultat lorsqu'on recherche la puissance maximale.
- Le fait de faire un demi-cercle avec une distance constante n'a pas été respecté lors de l'acquisition ce qui justifie la forme de la courbe, à partir de $t = 6\text{s}$ on refait le sens inverse mais avec une vitesse plus rapide (d'où le 0 à 130° en une fraction de seconde), une fois de plus cette anomalie fait partie des erreurs terrain lors de l'expérimentation.

Conclusion

En conclusion, ce TP nous a permis de mettre en pratique les différentes notions vues dans le cours de Beamforming pour la localisation spatiale d'une source sonore.

La localisation acoustique est une technique permettant de mesurer l'intensité et la direction d'une source sonore. Une technique populaire pour son calcul est l'algorithme de Beamforming "Sum-and-Delay", qui génère un angle approximatif de son emplacement tout en réduisant le bruit dans le signal.

Le Beamforming repose sur le fait que différents haut-parleurs répondent au même signal d'entrée de différentes manières, par exemple en retardant légèrement le signal, en le jouant à des volumes différents ou en utilisant des effets d'annulation. Les différents réglages des haut-parleurs permettent au système de contrôler la taille, la forme et la direction de l'onde sonore. Ces opérations sont effectuées à l'aide du diagramme de directivité $D(r, k)$ qui représente le filtrage spatial et fréquentiel effectué par le réseau de microphones.

Les résultats obtenus et les phénomènes observés sont en accord avec nos attentes.

Nous tenons à remercier toute personne ayant contribué pour la réalisation de ce TP.