



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ψηφιακή Επεξεργασία Σήματος

Εξάμηνο 6^ο (Εαρινό Εξάμηνο 2020-2021)

2^ο Εργαστηριακό Project Σπανός Νικόλαος – el18822

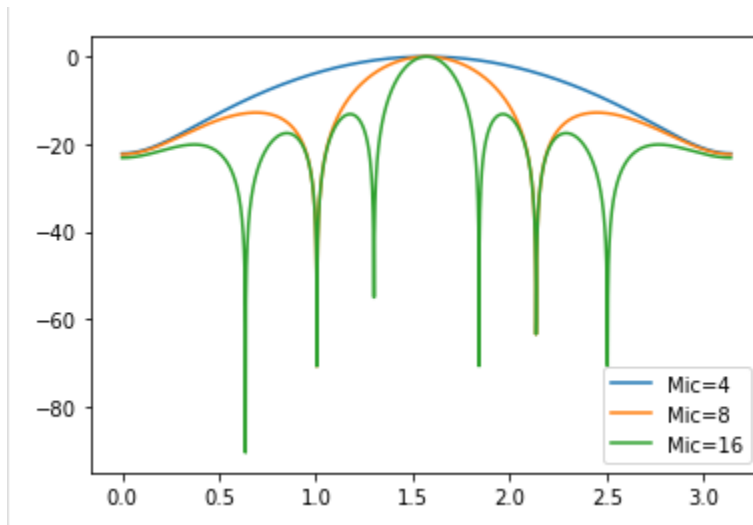
Μέρος 1^ο

1.

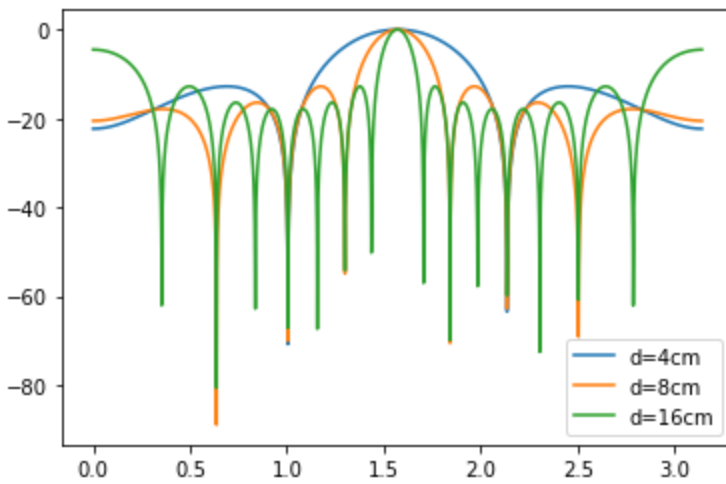
Δημιουργούμε την συνάρτηση `beam_form` που δημιουργεί για τα κατάλληλα δεδομένα το delay-and-sum beam pattern

```
def beam_form(N,theta,thetas,w,d):  
    a=np.array([])  
    a=(1/N)*(np.sin((N/2)*(w/343)*d*0.01*(np.cos(theta)-np.cos(thetas)))/np.sin((1/2)*(w/343)*d*0.01*(np.cos(theta)-np.cos(thetas)))  
    return a
```

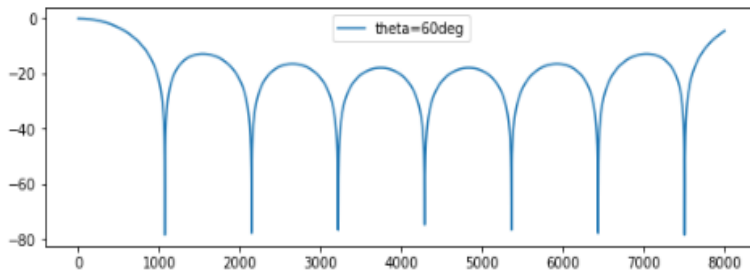
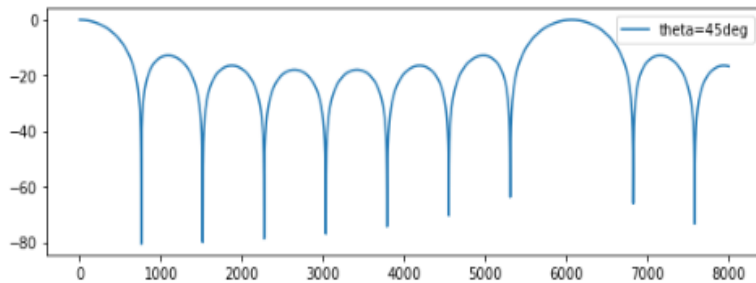
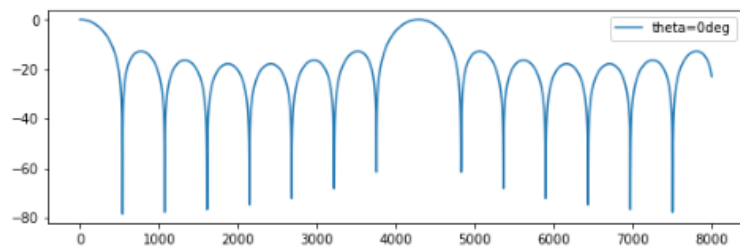
και στην συνέχεια σχεδιάζουμε για κάθε περίπτωση μικροφώνων σε κλίμακα dB τα delay-and-sum beam pattern:



και αναλόγως για τις διαφορετικές αποστάσεις μεταξύ μικροφώνων:



Για διαφορετικές γωνίες άφιξης παίρνουμε το εξής αποτέλεσμα:



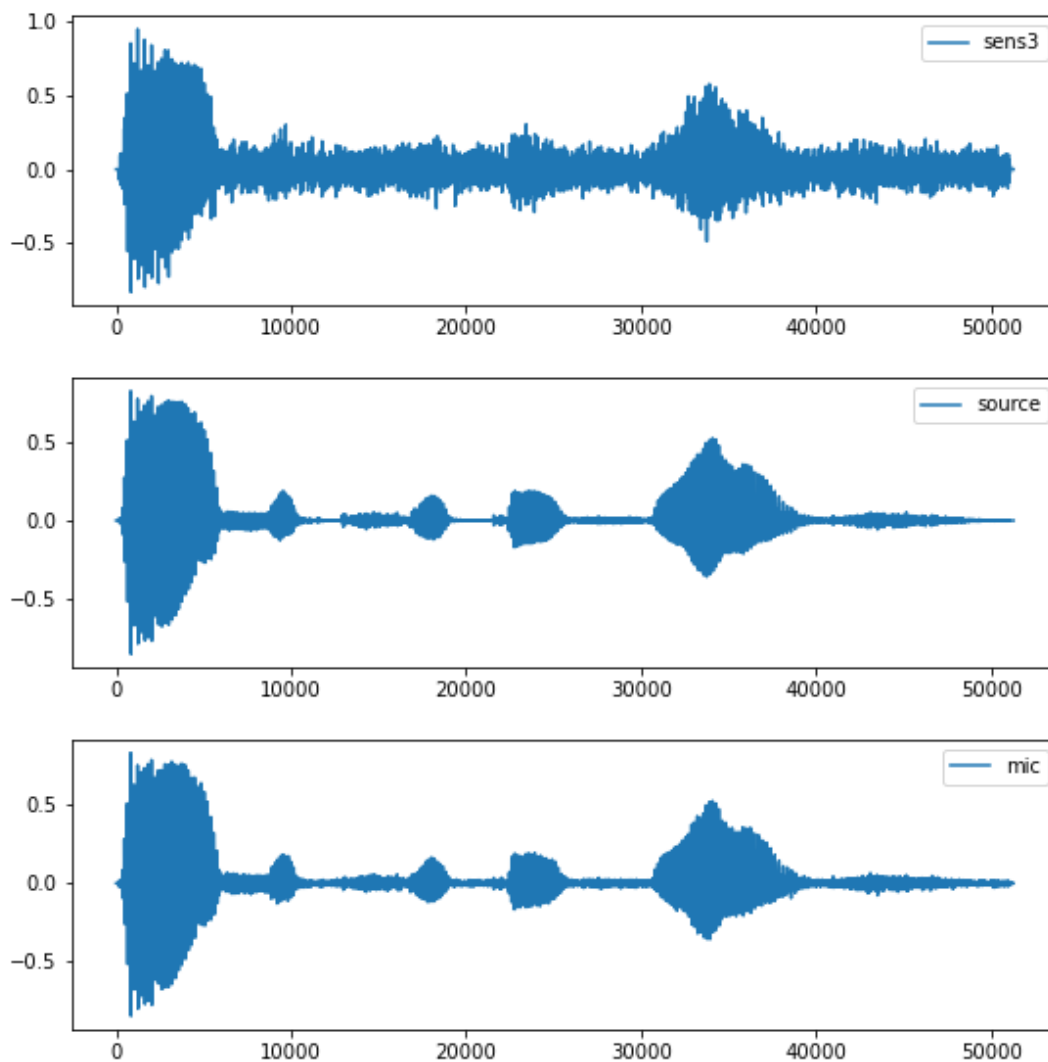
Παρατηρούμε ότι για διαφορετικές γωνίες, ο αριθμός των λοβών μειώνεται και παρουσιάζεται peak σε διαφορετικές συχνότητες. Σε γωνίες διάφορες του θ_s , ο delay-and-sum beam pattern πρακτικά θα

απορρίπτει το σήμα, εκτελώντας χωρικό φιλτράρισμα. Έτσι, θα λαμβάνεται μόνο το σήμα το οποίο προέρχεται από την γωνία στην οποία έχουμε κατεθύνει τα μικρόφωνα.

2.1(A)

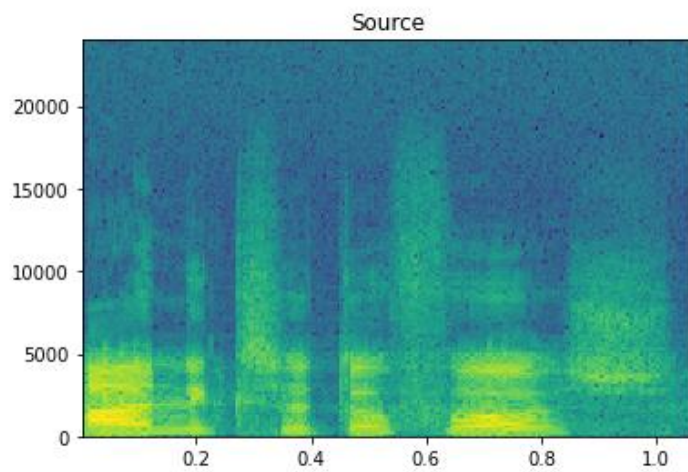
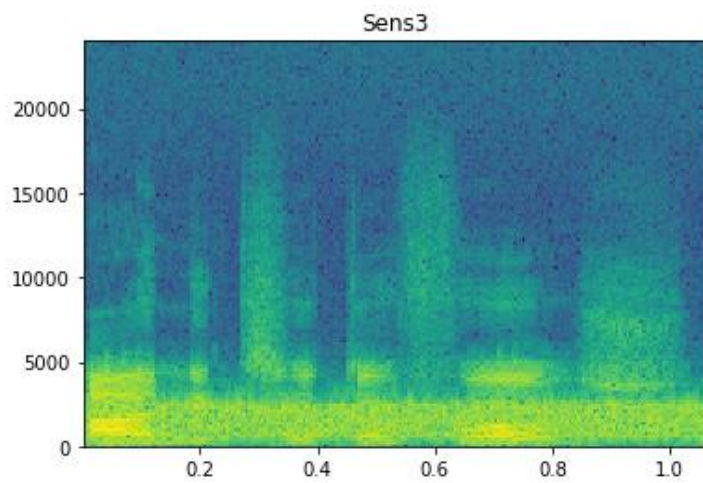
2.

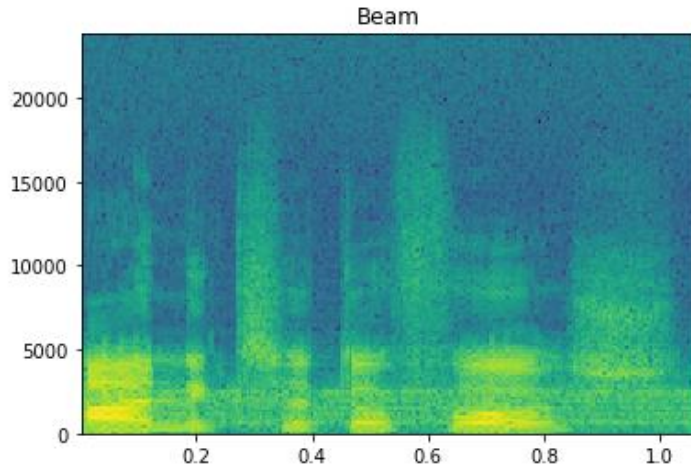
Σχεδιάζουμε τις κυματομορφές των σημάτων:



Παρατηρούμε πως στο τρίτο γράφημα, που αποτελεί την έξοδο του beamformer ότι έχει γίνει σχετικά καλή αποθυροβοποίηση, παρατηρώντας την διαφορά του με το 3^ο μικρόφωνο και την ομοιότητα του με το καθαρό σήμα.

Σχεδιάζουμε και τα σπεκτρογραφήματα:





Τα σπεκτρογραφήματα επιβεβαιώνουν το προηγούμενο συμπέρασμα , καθώς παρατηρείται συγκρίτικα λιγότερος θόρυβος στο Beam από ότι στο 3^ο μικρόφωνο.

3.

Υπολογίζουμε το SNR χρησιμοποιώντας τον τύπο των RMS τιμών και προβάλλοντας το σε κλίμακα dB:

```
: noise1=sens3-source
  noise2=mic-source
  rms_noise1 = np.sqrt(np.mean(noise1**2))
  rms_noise2 = np.sqrt(np.mean(noise2**2))
  rms_beam = np.sqrt(np.mean(mic**2))
  rms_mic = np.sqrt(np.mean(sens3**2))
  SNR1=20*np.log10(rms_beam/rms_noise2)
  SNR2=20*np.log10(rms_mic/rms_noise1)
  print(np.abs(SNR1))
  print(np.abs(SNR2))
```

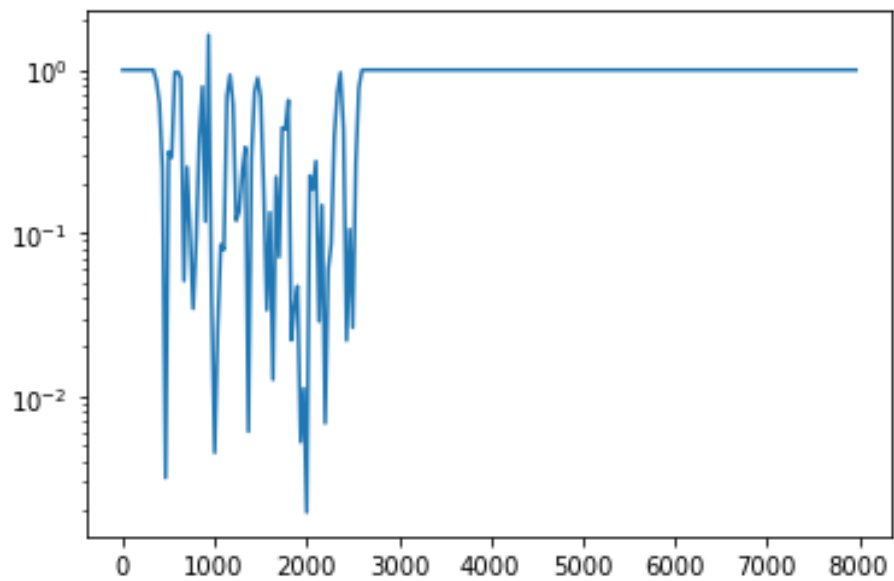
```
20.463404432053935
4.821568131446838
```

Συγκρίνοντας το σήμα με τον αντίστοιχο θόρυβο σε κάθε περίπτωση, παρατηρούμε ότι το Signal to Noise Ratio στο delay-and-sum beam pattern είναι πολύ μεγαλύτερο , με βελτίωση κατά 5 φορές, σε σχέση με του 3^{ου} μικροφώνου, αποδεικνύοντας ότι κάνει αρκετά καλή δουλειά στην αποθρονοποίηση κατεθυνόμενου θορύβου.

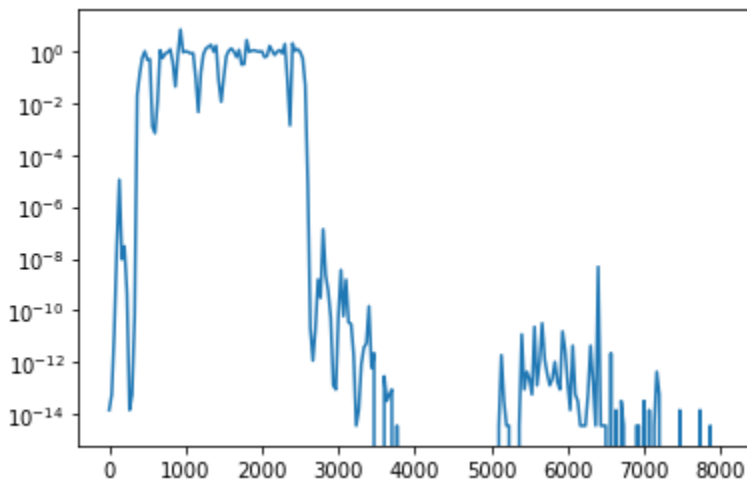
2.2.(B)

1.

Χρησιμοποιώντας την συνάρτηση welch για να υπολογίσουμε τα φάσματα ισχύος θορύβου και σήματος, σχεδιάζουμε το σήμα μέχρι τις συχνότητες 8000 , που έχουμε βρει ότι βρίσκονται κοντά στην sampling frequency 240 που επιστρέφει η συνάρτηση welch:



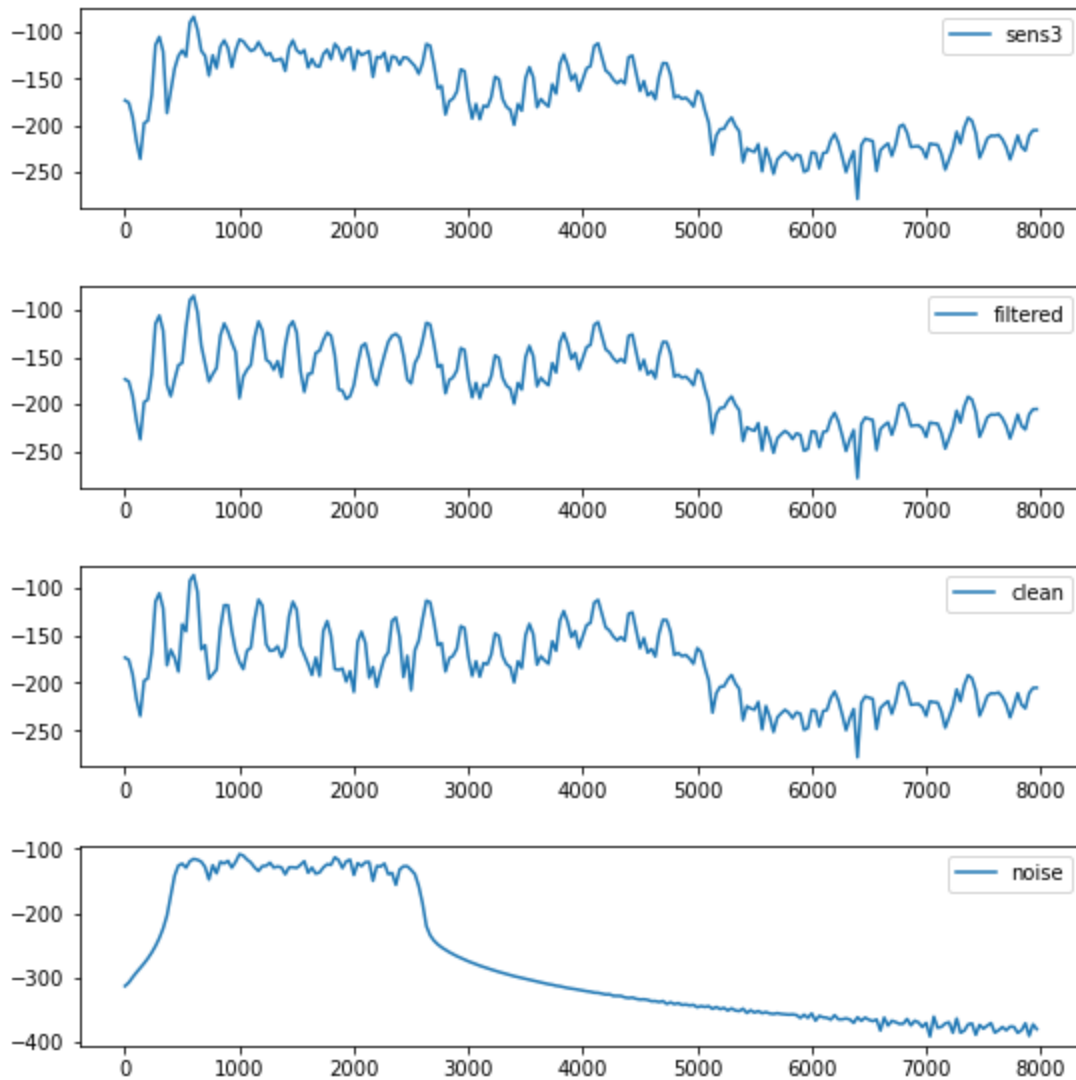
2. Υπολογίζοντας το noise distortion , σχεδιάζουμε με την ίδια μέθοδο:



Παρατηρούμε ότι στις συχνότητες στις οποίες είναι συγκεντρωμένη η ενέργεια του θορύβου , δηλαδή στις συχνότητες [500,2500] Hz το noise_distortion είναι αρκετά μεγάλο , ενώ ύστερα πέφτει.

3.

Σχεδιάζοντας τα φάσματα ισχύος ξανά με την χρήση της welch παίρνουμε τα εξής αποτελέσματα:



Παρατηρούμε ότι το κεντρικό μικρόφωνο περιέχει τον θόρυβο στις αντίστοιχες συχνότητες , ενώ το φιλτραρισμένο σήμα πλησιάζει σε αρκετά μεγάλο βαθμό το πραγματικό σήμα, καθώς ο θόρυβος έχει φιλτραριστεί σε μεγάλο βαθμό.

4. Υπολογίζοντας τα SNR όπως πριν παίρνουμε τα εξής αποτελέσματα:

```
rms_filt = np.sqrt(np.mean(filtered**2))
rms_sig = np.sqrt(np.mean(sig**2))
rms_noise1 = np.sqrt(np.mean((sig-clean)**2))
rms_noise2 = np.sqrt(np.mean((filtered-clean)**2))
SNR_filt=20*np.log10(rms_filt/rms_noise2)
SNR_sig=20*np.log10(rms_sig/rms_noise1)
print(SNR_filt)
print(SNR_sig)
```

```
9.61128545747215
5.270766019821167
```

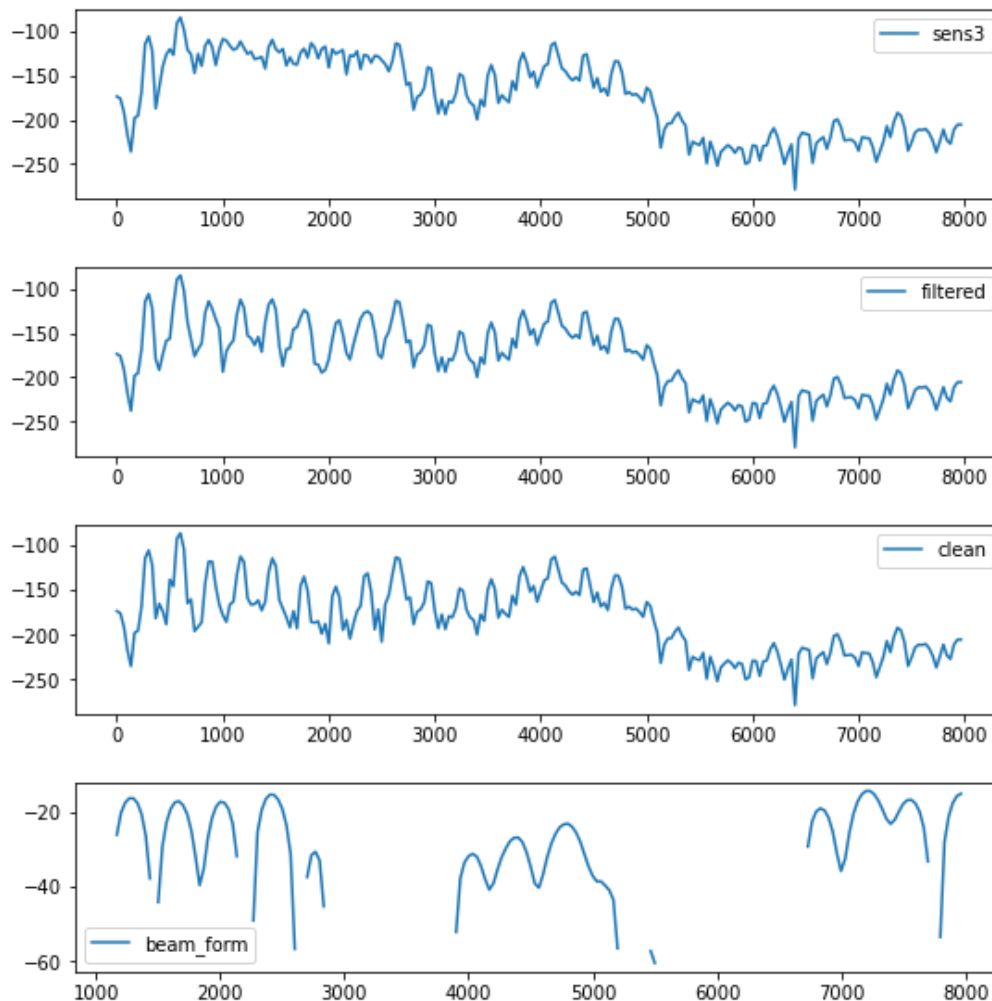
Παρατηρούμε ότι το SNR βελτιώθηκε κατα σχεδόν 2 φορές, ενώ στην πολυκαναλική μέθοδο , όπως βλέπουμε παρακάτω:

```
mic_seg=mic[22560:24000]
rms_mic_seg = np.sqrt(np.mean(mic_seg**2))
rms_noise3 = np.sqrt(np.mean((mic_seg-clean)**2))
SNR_mic_seg= 20*np.log10(rms_mic_seg/rms_noise3)
print(SNR_mic_seg)
```

20.513016937728136

Η βελτίωση του SNR ήταν σημαντικά μεγαλύτερη, άρα καταλαβαίνουμε ότι συμφέρει περισσότερο σε αυτήν την περίπτωση θορύβου η πολυκαναλική μέθοδος.

Σχεδιάζουμε τα φάσματα ισχύος:

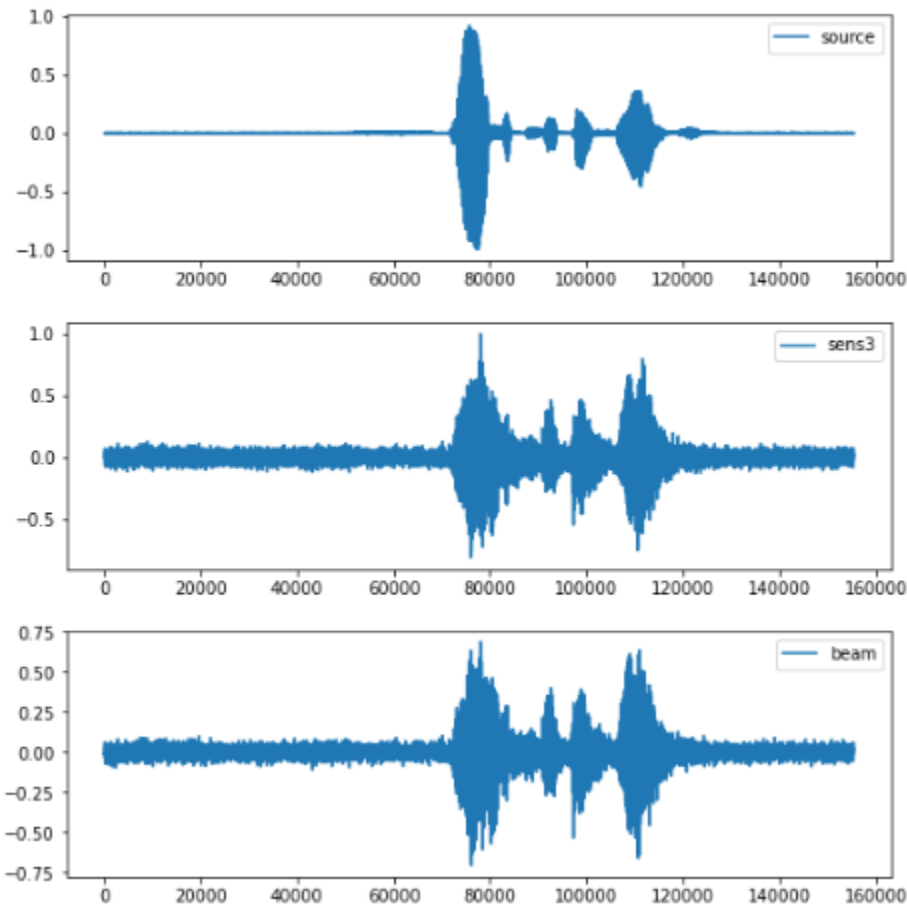


Παρατηρούμε ότι η έξοδος του delay-and-sum beam pattern παρουσιάζει σημαντικά μεγαλύτερη ισχύ από το καθαρό σήμα και σχεδόν καθόλου σε άλλες. Σε γενικές γραμμές, το μονοκαναλικό φίλτρο δίνει καλύτερα αποτελέσματα όσον αφορά το φάσμα ισχύος , καθώς προσεγγίζει σε μεγάλο βαθμό το

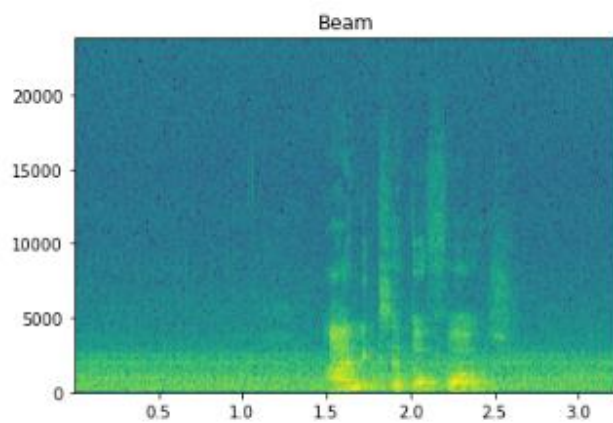
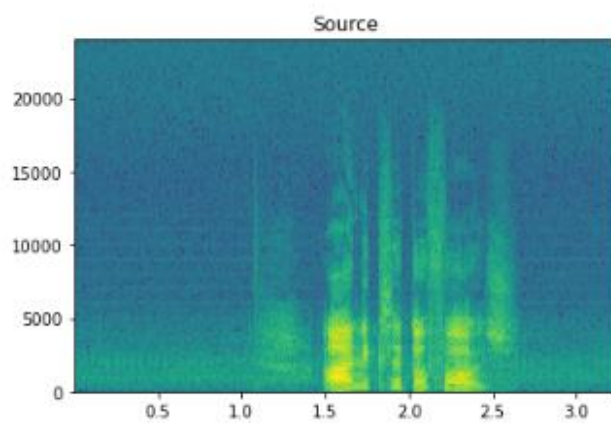
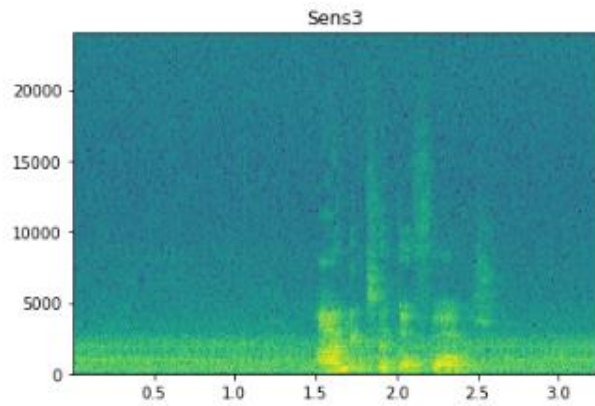
αρχικό σήμα, ενώ η πολυκαναλική μέθοδος αποδίδει παραπάνω ισχύ στην κατεύθυνση του σήματος το οποίο μας ενδιαφέρει.

2.2.(A)

Ύστερα από τον υπολογισμό του beam forming με τις κατάλληλες παραμέτρους, σχεδιάζουμε τα σήματα και τα σπεκτρογραφήματά τους:



Παρατηρούμε ότι για το συγκεκριμένο είδος θορύβου, ο delay-and-sum beam pattern δεν εκτέλεσε καλή αποθορυβοποίηση του σήματος, καθώς προέρχεται από όλες τις γωνίες γύρω από τα μικρόφωνα και δεν είναι εστιασμένος.



Το ίδιο φαινόμενο παρατηρούμε και στα σπекτρογραφήματα, καθώς ένα μεγάλο μέρος του θορύβου έχει παραμείνει στις αντίστοιχες συχνότητες.

(3)

Χρησιμοποιούμε για τον υπολογισμό των SSNR 85 παράθυρα των 1830 δειγμάτων μήκος και κατώφλι

-1. Παίρνουμε τα εξής αποτελέσματα:

```

SSNR1=0
noise=sens3[0:5000]
for i in range(85):
    a=np.sum((sens3[i*1830:i*1830+1830])**2)
    b=np.sum(noise**2)
    a=a-b
    SNR=10*np.log10(a/b)
    if(SNR>35):
        SNR=35
    if(SNR>-1):
        SSNR1=SSNR1+SNR
SSNR1=SSNR1/85

SSNR2=0
noise=mic[0:5000]
for i in range(85):
    a=np.sum(mic[i*1830:i*1830+1830]**2)
    b=np.sum(noise**2)
    a=a-b
    SNR=10*np.log10(a/b)
    if(SNR>35):
        SNR=35
    if(SNR>-1):
        SSNR2=SSNR2+SNR
SSNR2=SSNR2/85
print(SSNR1)
print(SSNR2)

```

```

1.7952835875398971
1.8074657196868362

```

Παρατηρούμε ότι το σήμα έχει βελτιωθεί αλλά ελάχιστα, σε βαθμό που δεν έχει σχεδόν αλλάξει καθόλου. Αυτό μας δείχνει ότι αυτή η διαδικασία του beamforming δεν αποδίδει καλά για diffuse noise field μορφής θορύβου, αλλά περισσότερο για θόρυβο που προέρχεται από μια συγκεκριμένη γωνία.