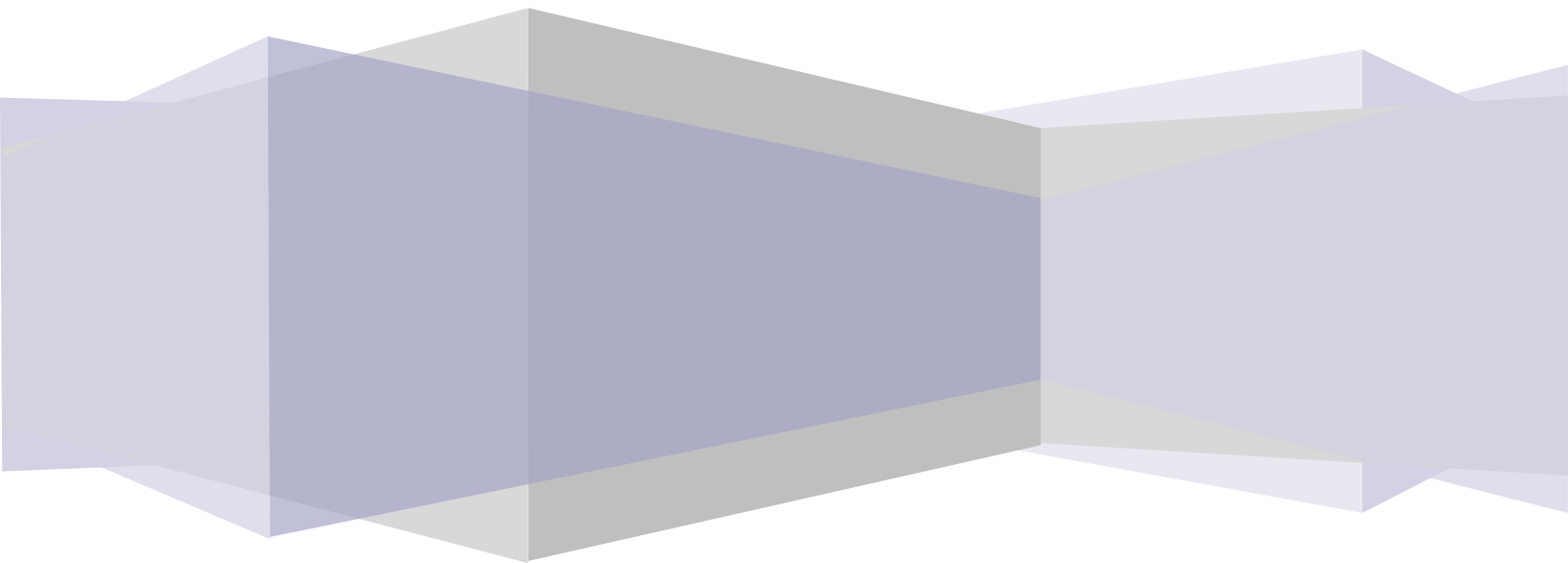


Digital Signal Processing

3η Εργαστηριακή Άσκηση: Συστοιχίες Μικροφώνων
(Microphone Arrays) και Πολυκαναλική Επεξεργασία
Σημάτων (Multichannel Signal Processing).

Διδυμιώτου Καουκάκη Κωνσταντίνα AM 03115726

Κυριάκου Αθηνά AM 03117405



Ανάγκη για ανάλυση συστοιχιών μικροφώνων

Στις μέρες μας η χρήση συστοιχιών μικροφώνων με σκοπό την καταγραφή και την επεξεργασία ακουστικών σημάτων γίνεται όλο και πιο διαδεδομένη. Πλεονεκτεί στην δυνατότητα καταγραφής και αξιοποίησης χρονικών (temporal) και χωρικών χαρακτηριστικών των ακουστικών σημάτων και αποτελεί αποτελεσματικό τρόπο αποθρομβοποίησης ενός σήματος σε διάφορες εφαρμογές. Στην παρούσα άσκηση θα μελετηθεί η αποθρομβοποίηση ενός σήματος ομιλίας (speech enhancement). Γνωρίζοντας την κατεύθυνση από την οποία καταφθάνει το επιθυμητό σήμα που επιθυμούμε να λάβουμε, μπορούμε εύκολα να το απομονώσουμε από τους ήχους που καταφθάνουν από άλλες κατευθύνσεις, σε μεγάλο τουλάχιστον ποσοστό εκμεταλλευόμενοι την τοπικότητα του φαινομένου.

Στο πρώτο μέρος εισάγουμε τύπους για πολυκαναλική επεξεργασία σημάτων υπολογίζοντας με την βοήθεια του Matlab, το beam pattern (χρησιμοποιείται για χωρική περιγραφή) του delay and sum beamformer για ομοιόμορφες γραμμικές συστοιχίες μικροφώνων για διάφορες τιμές σχετικών παραμέτρων.

Στο δεύτερο μέρος θα εφαρμόσουμε beamforming για Speech Enhancement τόσο σε προσομοιωμένα σήματα όσο και σε πραγματικά σήματα όπου ο θόρυβος δημιουργείται από διάφορες πηγές (diffuse noise field). Ακόμη θα πραγματοποιηθεί και μελέτη μονοκαναλικού Wiener φιλτραρίσματος και Post-filtering με IIR Wiener φιλτράρισμα αντίστοιχα σε προσομοιωμένα και πραγματικά σήματα με σκοπό να μετρηθούν SNR και SSNRS για να ακολουθήσει σύγκριση μεταξύ των μεθόδων.

Μέρος 1ο - Συστοιχίες Μικροφώνων και Χωρικό Φιλτράρισμα

Στην παρούσα άσκηση μελετήθηκαν μικρόφωνα σε Ομοιόμορφες Γραμμικές Συστοιχίες (ΟΓΣ), στις οποίες N μικρόφωνα είναι τοποθετημένα κατά τη διεύθυνση μια γραμμής (άξονα z) σε ισαπέχουσες (απόσταση d) μεταξύ τους αποστάσεις.

Η έξοδος του beamformer που λαμβάνουμε μέσω της εφαρμογής της τεχνικής αυτής είναι:

$$Y(\omega, \mathbf{a}) = \mathbf{H}^T(\omega) \mathbf{F}(\omega, \mathbf{a}) = \mathbf{H}^T(\omega) \mathbf{d}(\mathbf{k}) F(\omega)$$

όπου το Beam Pattern στην απλούστερη περίπτωση είναι:

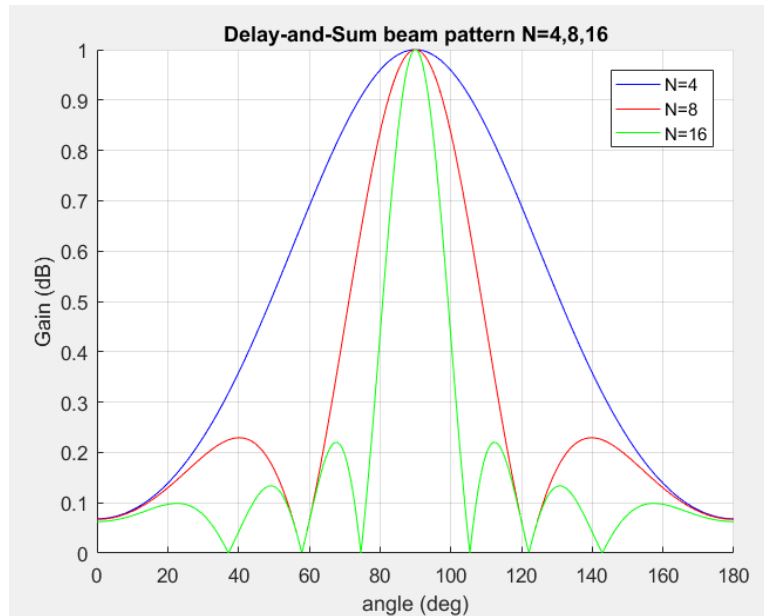
$$B(\omega, \theta) = \frac{1}{N} \frac{\sin[\frac{N}{2} \frac{\omega}{c} d (\cos \theta - \cos \theta_s)]}{\sin[\frac{1}{2} \frac{\omega}{c} d (\cos \theta - \cos \theta_s)]}$$

Ερώτημα 1.4: Μελέτη χαρακτηριστικών του delay and sum beam pattern

Θα μελετηθεί delay-and-sum beam pattern υπό διαφορετικές συνθήκες. Αρχικά διατηρούμε σταθερή την απόσταση μεταξύ των μικροφώνων στα 4cm και μεταβάλλουμε τον αριθμό των μικροφώνων που χρησιμοποιούμε (N=4,8,16)

```
%1.4.1
d=0.04; %distance cm
N=4; %number of microphones
figure('Name','Delay-and-Sum beam pattern for differents microphones') ;
title('Delay-and-Sum beam pattern N=4,8,16');
xlabel('angle (deg)')
ylabel('Gain (dB)')
hold on
B_a = abs((1/N)*sin((N/2)*(w/c)*d*(cos(theta) - cos(theta_s)))./sin((1/2)*(w/c)*d*(cos(theta) - cos(theta_s))));
semilogy((180/pi)*theta,B_a,'b')
B_a = abs((1/(2*N))*sin(((2*N)/2)*(w/c)*d*(cos(theta) - cos(theta_s)))./sin((1/2)*(w/c)*d*(cos(theta) - cos(theta_s))));
semilogy((180/pi)*theta,B_a,'r')
B_a = abs((1/(4*N))*sin(((4*N)/2)*(w/c)*d*(cos(theta) - cos(theta_s)))./sin((1/2)*(w/c)*d*(cos(theta) - cos(theta_s))));
semilogy((180/pi)*theta,B_a,'g')
legend('N=4','N=8','N=16')
grid on;
```

Με βάση τον παραπάνω κώδικα παίρνουμε το μέτρο του beam pattern σε λογαριθμική κλίμακα για διάφορο αριθμό μικροφώνων.



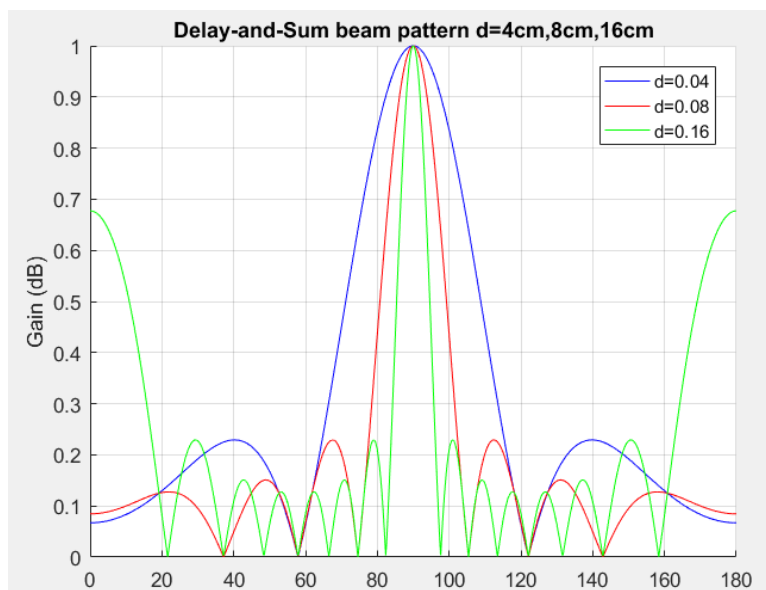
Σχήμα 1.4α: Delay and sum beam pattern N=4,8,16

- Επιθυμούμε να ενισχύσουμε τον ήχο που προέρχεται από πηγή με μια συγκεκριμένη γωνία, συνεπώς θέλουμε μικρό κύριο λοβό ώστε να ενισχύουμε κάθε φορά μόνο τον επιθυμητό ήχο συγκεκριμένης γωνίας που μας ενδιαφέρει. Η ύπαρξη παράπλευρων λοβών μαζί με τον κύριο λοβό, φανερώνει ότι εκτός από το σήμα επιθυμητής γωνίας, ενισχύονται ηχητικά σήματα που φτάνουν στην συστοιχία από άλλες γωνίες. Αυξάνοντας τον αριθμό των μικροφώνων στενεύουμε τον κύριο λοβό και τους παράπλευρους άρα πετυχαίνουμε λιγότερο θόρυβο αφού αποκόπτονται περισσότερο οι ήχοι από τις υπόλοιπες γωνίες. Μπορούμε δηλαδή να απομονώσουμε έτσι τον επιθυμητό ήχο.

Στην συνέχεια κρατώντας σταθερό τον αριθμό μικροφώνων ($N=8$) μεταβάλλουμε την απόσταση μεταξύ των μικροφώνων. ($d=4\text{cm}, 8\text{cm}, 16\text{cm}$)

```
%1.4.2
N=8;
d=0.04;
figure('Name','Delay-and-Sum beam pattern for differents distances of microphones') ;
title('Delay-and-Sum beam pattern d=4cm,8cm,16cm');
xlabel('Angle (deg)')
ylabel('Gain (dB)')
hold on
B_a = abs((1/N)*sin((N/2)*(w/c)*d*(cos(theta) - cos(theta_s)))/sin((1/2)*(w/c)*d*(cos(theta) - cos(theta_s))));
semilogy((180/pi)*theta,B_a,'b')
B_a = abs((1/N)*sin((N/2)*(w/c)*(2*d)*(cos(theta) - cos(theta_s)))/sin((1/2)*(w/c)*(2*d)*(cos(theta) - cos(theta_s))));
semilogy((180/pi)*theta,B_a,'r')
B_a = abs((1/N)*sin((N/2)*(w/c)*(4*d)*(cos(theta) - cos(theta_s)))/sin((1/2)*(w/c)*(4*d)*(cos(theta) - cos(theta_s))));
semilogy((180/pi)*theta,B_a,'g')
legend('d=0.04','d=0.08','d=0.16')
grid on;
```

Και παίρνουμε τις αντίστοιχες γραφικές:



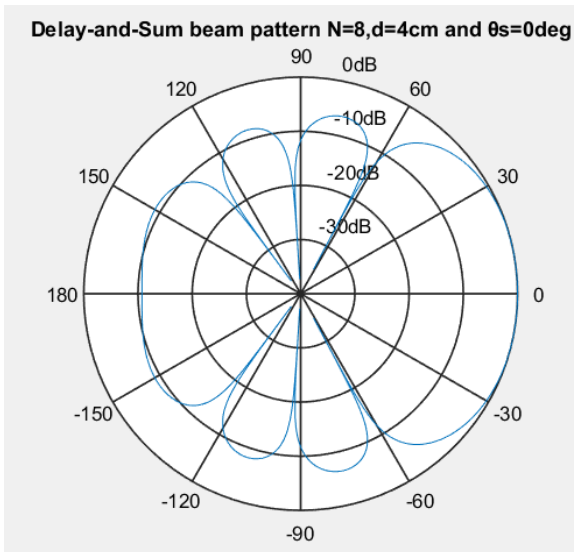
Σχήμα 1.48: Delay and Sum beam pattern ($d=4\text{cm}, 8\text{cm}, 16\text{cm}$)

- Αυξάνοντας την απόσταση μεταξύ των μικροφώνων παρατηρούμε ότι στενεύει ο κύριος λοβός και άρα έχουμε καλύτερη έξοδο όπως εξηγήσαμε προηγουμένως. Ωστόσο λαμβάνουμε υπόψιν και το φαινόμενο του spatial aliasing δηλαδή όταν η απόσταση ξεπεραάσει την τιμή $\lambda/2$ εμφανίζονται πλευρικοί λοβοί με μοναδιαίο πλάτος (0dB) και σε διαφορετικές γωνίες από τις επιθυμητές (πχ Σχήμα 1.4.β για $N=16$).

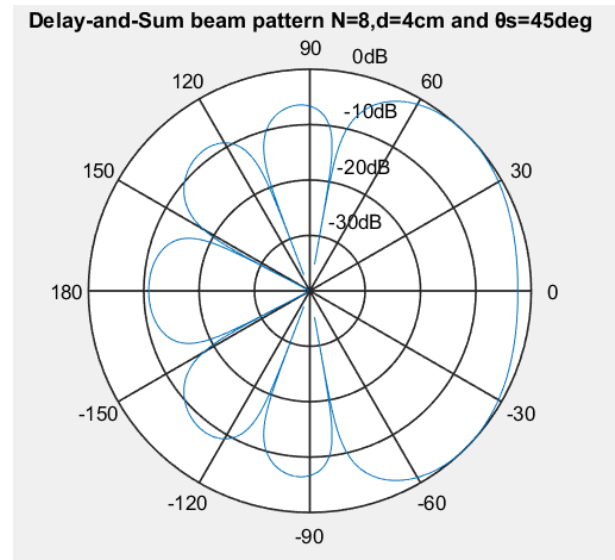
Τέλος διατηρούμε σταθερό τον αριθμό των μικροφώνων όσο και τη μεταξύ τους απόσταση και προβάλλουμε το delay-and-sum beam pattern σε λογαριθμική κλίμακα σε πολικό διάγραμμα.

```
N=8;
d=0.04;
theta_new=linspace(-pi,pi,30000);
for i= 0:0.5:1
    B_a = abs((1/N)*sin((N/2)*(w/c)*d*(cos(theta_new) - cos(i*theta_s)))/sin((1/2)*(w/c)*d*(cos(theta_new) - cos(i*theta_s))));
    figure;
    semilogr_polar(theta_new,B_a)
    title(['Delay-and-Sum beam pattern N=8,d=4cm and theta_s=' num2str(i*90) 'deg'])
end
```

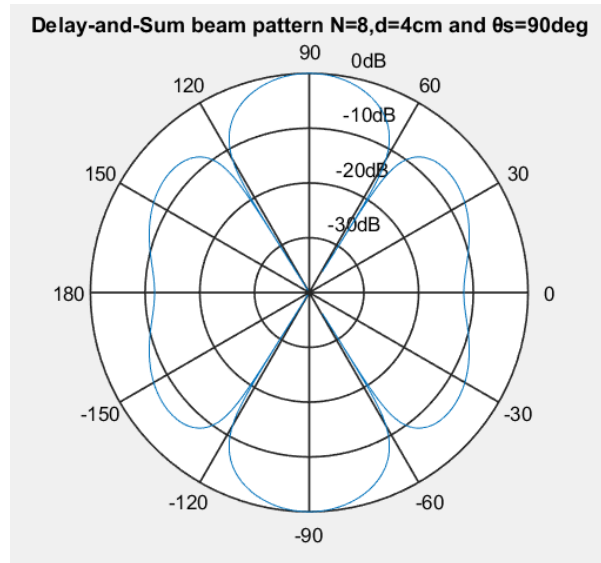
Λαμβάνουμε τα εξής:



Σχήμα 1.4 γ: Delay and sum beam pattern $\theta_s=0$



Σχήμα 1.4 δ: Delay and sum beam pattern $\theta_s=45$



Σχήμα 1.4 ε: Delay and sum beam pattern $\theta_s=90$

- Από τα παραπάνω πολικά διαγράμματα παρατηρούμε ότι μοναδιαίο κέρδος (0dB) (δηλαδή εκεί όπου το beam pattern έχει μέτρο 1 steering direction) λαμβάνουμε μόνο στις επιθυμητές τιμές 0,45 και 90 (εκεί που τέμνεται ο κύκλος 0dB) . Οι ήχοι που καταφθάνουν υπό διαφορετικές γωνίες θα εξουδετερώνονται.

Μέρος 2^ο - Εφαρμογή Beamforming για Speech Enhancement

Ερώτημα 2.1.A Delay and sum beamforming (Simulated Signals):

Στην εφαρμογή αυτή έχουμε μια γραμμική συστοιχία N=7 μικροφώνων τοποθετημένα σε απόσταση d=4cm η οποία καταγράφει σήματα που παράγονται από σημειακές πηγές. Η μια παράγει φωνή (theta_source=45) και η άλλη παράγει ζωνοπερατό θόρυβο περιορισμένο στην ζώνη 500 έως 2500 Hz.

```
for i=1:N
    dft_mic(i,:)=fft(mic(i,:));
end

%calculate weights w
w = -pi:((2*pi)/length(mic(1,:))):pi-((2*pi)/length(mic(1,:)));

%calculate d(ks)
for i=1:N
    d_ks(i,:) = exp(1i*(N-1)*w*fs*d*(cos(theta_source))/(2*c)).*exp(-1i*(i-1)*w*fs*d*(cos(theta_source))/c);
end

%delay filter
for i=1:N
    Y(i,:)=(d_ks(i,:)).*(dft_mic(i,:)); %frequency
    y(i,:)=ifft(Y(i,:)); %time
    real_y(i,:)=real(y(i,:)); %take only the real part of the signal
end

%final signal
y_output=zeros;
for i=1:N
    y_output=y_output + real_y(i,:);
end
y_output=y_output/N; %average
y_output = y_output';
noise = y_output - source;
audiowrite('sim_ds.wav',y_output,fs);
```

Αρχικά πρέπει να ευθυγραμμίσουμε χρονικά τα σήματα τα οποία εξαιτίας της $\theta_s=45$ περιέχουν το θορυβώδες σήμα μετατοπισμένο κατά τ_n .

$$\tau_n = \frac{\mathbf{a}^T \mathbf{p}_n}{c} = -\frac{(n - \frac{N-1}{2})d\cos\theta}{c}$$

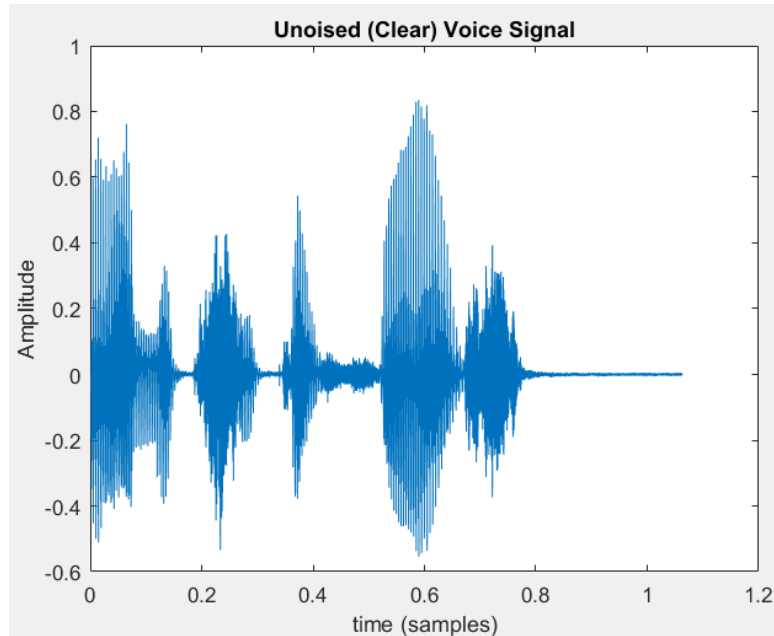
Επίσης δημιουργούμε το διάνυσμα $\mathbf{d}(\mathbf{k})$

$$\mathbf{d}(\mathbf{k}) = e^{-j\frac{N-1}{2}\frac{\omega}{c}d\cos\theta} [1, e^{j\frac{\omega}{c}d\cos\theta}, \dots, e^{j(N-1)\frac{\omega}{c}d\cos\theta}]^T$$

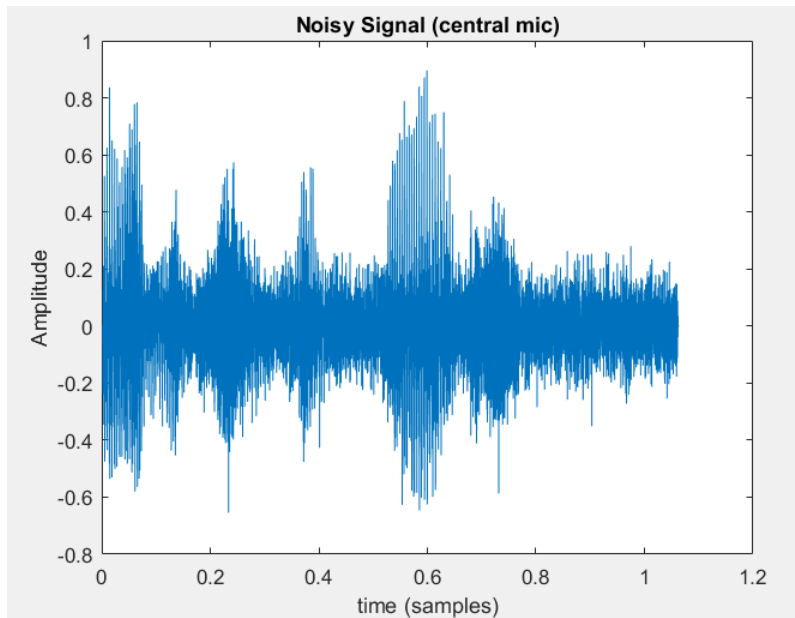
Στην συνέχεια πολλαπλασιάζουμε το Fourier κάθε σήματος μικρόφωνου με το $\mathbf{d}(\mathbf{k})$ ώστε να ευθυγραμμιστούν και να βρούμε την έξοδο κάθε μικροφώνου στο πεδίο της συχνότητας. Στην συνέχεια βρίσκουμε για το κάθε σήμα τον αντίστροφο Fourier (real part only). Αφού πραγματοποιήσουμε την ευθυγράμμιση, αθροίζουμε τα σήματα και διαιρούμε με τον αριθμό αυτών. Ο θόρυβος σχεδόν απαλείφεται. Το αποτέλεσμα της παραπάνω διαδικασίας αποθηκεύεται ως **sim_ds.wav**. Ακούγοντας το παρατηρούμε ότι αποτελεί μια πολύ καλή προσέγγιση του καθαρού σήματος φωνής.

Παρακάτω έχουμε τις κυματομορφές και τα σπекτρογραφήματα του καθαρού σήματος, του θορυβώδους σήματος καθώς και του σήματος μετά στην έξοδο του delay and sum beamformer:

ΚΥΜΑΤΟΜΟΡΦΕΣ

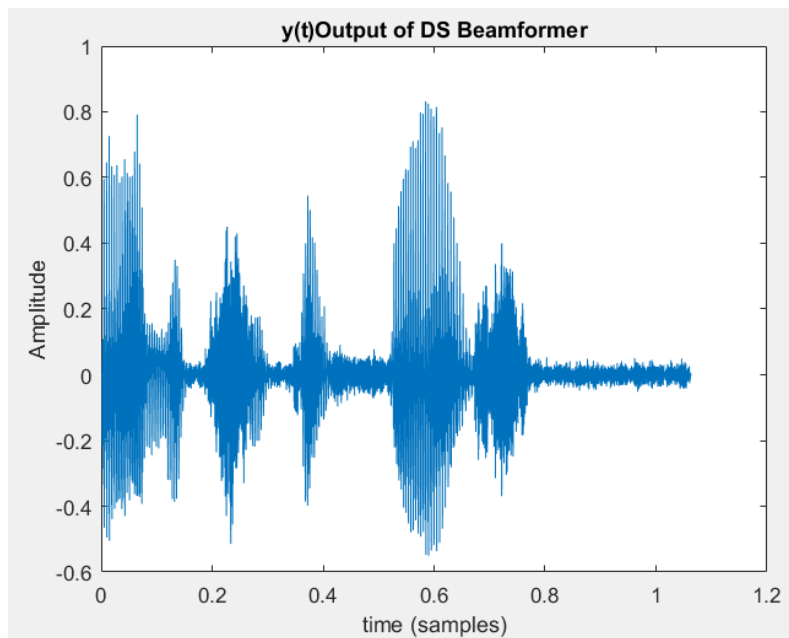


Σχήμα 2.1.Αα: Κυματομορφή καθαρού σήματος



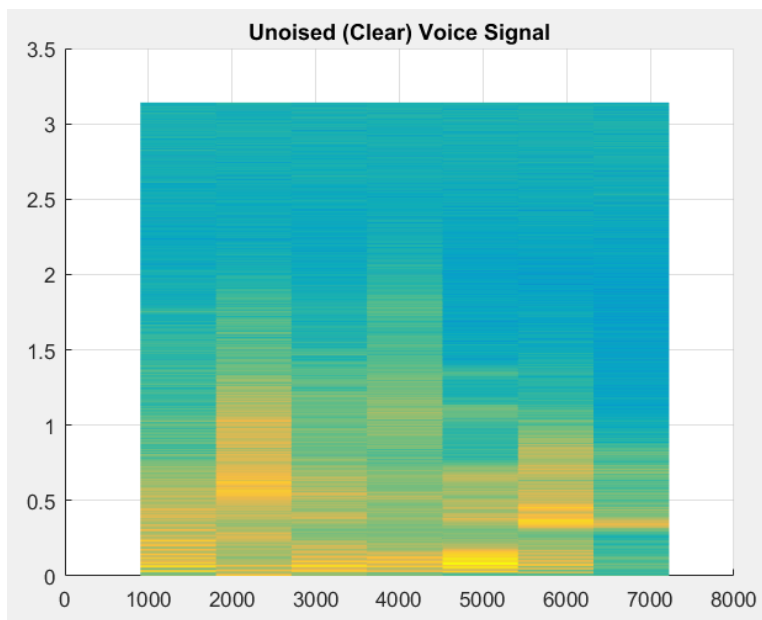
Σχήμα 2.1.Αβ: Κυματομορφή θορυβώδους σήματος

κεντρικού μικροφώνου

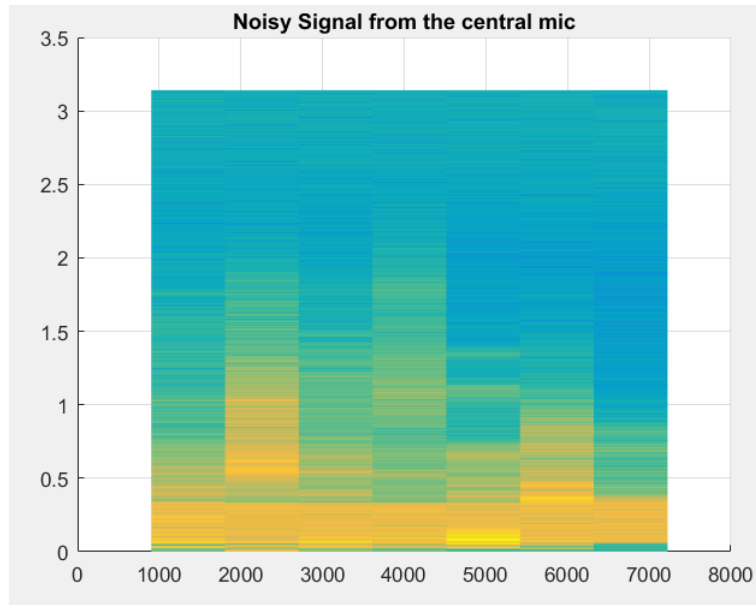


Σχήμα 2.1.Αγ: Κυματομορφή εξόδου delay and sum beamformer

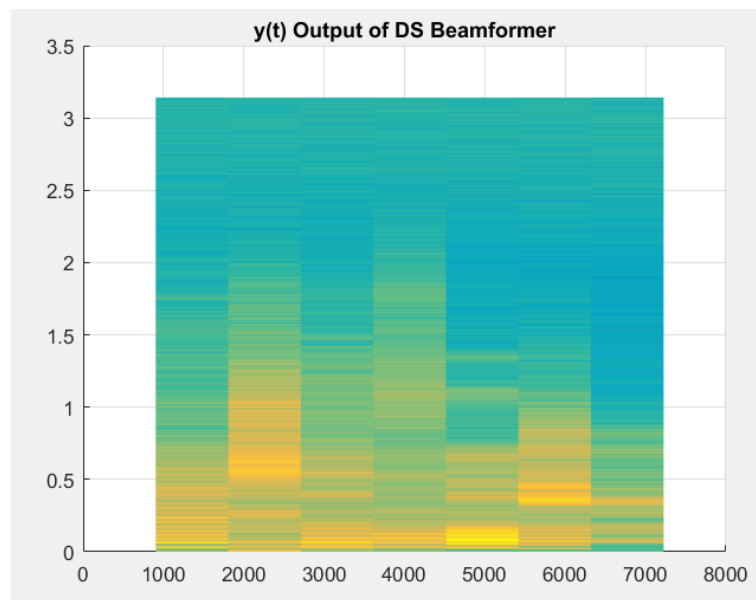
ΣΠΕΚΤΡΟΓΡΑΦΗΜΑΤΑ



Σχήμα 2.1δ.Α: Σπεκτρογράφημα καθαρού σήματος



*Σχήμα 2.1.Α: Σπεκτρογράφημα θορυβώδους σήματος
κεντρικού μικροφώνου*



Σχήμα 2.1.Αζ: Σπεκτρογράφημα εξόδου delay and sum beamformer

- Στο θορυβώδες σήμα (Σχ.2.1ε) παρατηρούμε ότι σε σχέση με το καθαρό σήμα(Σχ.2.1δ), προστέθηκαν συχνοτικές συνιστώσες στην περιοχή [500 Hz,2500 kHz] που αντιστοιχούν στις κανονικοποιημένες συχνότητες 0.0208 έως 0.1042 . Οι συνιστώσες αυτές όπως αναμενόταν παρατηρούνται καθ' όλη τη διάρκεια του

σήματος. Μετά την επεξεργασία του σήματος δηλαδή στην έξοδο του beamformer, παρατηρούμε ότι το μεγαλύτερο μέρος του θορύβου αυτού έχει παραληφθεί (Σχ. 2.1ζ). Επίσης τα (Σχ. 2.1δ) και (Σχ.2.1ζ) είναι παρόμοια μεταξύ τους και φαίνεται ο θόρυβος στην ζώνη [500,2500] Hz να έχει σχεδόν εξαλειφθεί άρα είχαμε επιτυχημένη αποθορυβοποίηση.

Αυτό διαπιστώνεται και από τα SNR που υπολογίσαμε. SNR του θορυβώδους σήματος από το κεντρικό μικρόφωνο της συστοιχίας και του σήματος εξόδου από το σύστημα delay and sum beamformer.

```
%SNR of noisy signal from central mic and of y(t)
sx=mean(mic(4,:).^2)-mean(mic(4,:))^2;
noise_mic=(mic(4,:))'-source;
ss=mean(noise_mic.^2)-mean(noise_mic)^2;
SNR_centralmic=10*log10(sx/ss);
display(SNR_centralmic);

sx=mean(y_output.^2)-mean(y_output)^2;
noise_beamformer=y_output-source;
ss=mean(noise_beamformer.^2)-mean(noise_beamformer)^2;
SNR_beamformer=10*log10(sx/ss);
display(SNR_beamformer);

SNR_total=SNR_beamformer/SNR_centralmic;
display(SNR_total);
```

Σύμφωνα με τα παραπάνω βρίσκουμε ότι **SNR_centralmic=4.8491 dB** και **SNR_beamformer=17.5836 dB**.

- Ο σηματοθορυβικός λόγος δείχνει την σχέση μεταξύ των ενεργειών του σήματος και του θορύβου που αυτό περιέχει. Επομένως παρατηρούμε ότι η έξοδος του συστήματος του delay and sum beamformer που σχεδιάσαμε έχει πολύ λιγότερο θόρυβο από το κεντρικό μικρόφωνο της συστοιχίας και είναι πιο κοντά στο καθαρό σήμα φωνής όπως φαίνεται και στα παραπάνω σχήματα.

Ερώτημα 2.1.Β: Μονοκαναλικό Wiener Φιλτράρισμα (Simulated Signals)

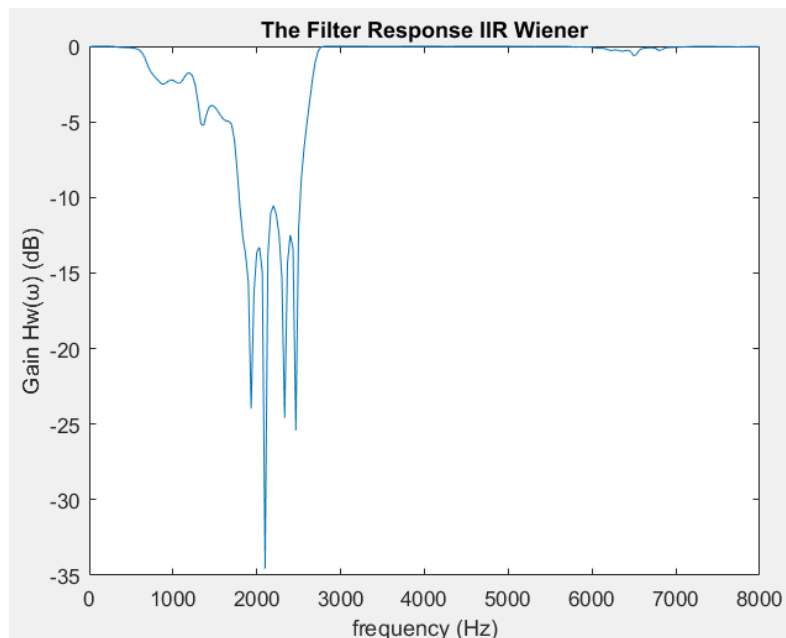
Στην συνέχεια θα απομονώσουμε ένα συγκεκριμένο πλαίσιο του σήματος του κεντρικού μικροφώνου της συστοιχίας (0.36sec - 0.38sec) και θα το φιλτράρουμε με IIR Wiener που κατασκευάσαμε.

```
%frame parameters
t_start=0.36;
t_stop=0.39;
x=mic(4, :);
y=x';
noise=y-source;
st=source(t_start*fs:t_stop*fs);
ut=noise(t_start*fs:t_stop*fs);
xt=y(t_start*fs:t_stop*fs);
[Pu,f]=pwelch(ut,[],[],length(ut),fs,'twosided');
[Px,f]=pwelch(xt,[],[],length(xt),fs,'twosided');
[Ps,f]=pwelch(st,[],[],length(st),fs,'twosided');
H_w=1-(Pu./Px); %the response
```

Υπολογίζουμε τα φάσματα ισχύος του σήματος και του θορύβου με εκτίμηση Welch και στην συνέχεια υπολογίζουμε την απόκριση συχνότητας του φίλτρου με σύμφωνα με τον τύπο:

$$H_W(\omega) = \frac{P_s(\omega)}{P_x(\omega)} = 1 - \frac{P_v(\omega)}{P_x(\omega)},$$

Η απόκριση που παίρνουμε είναι

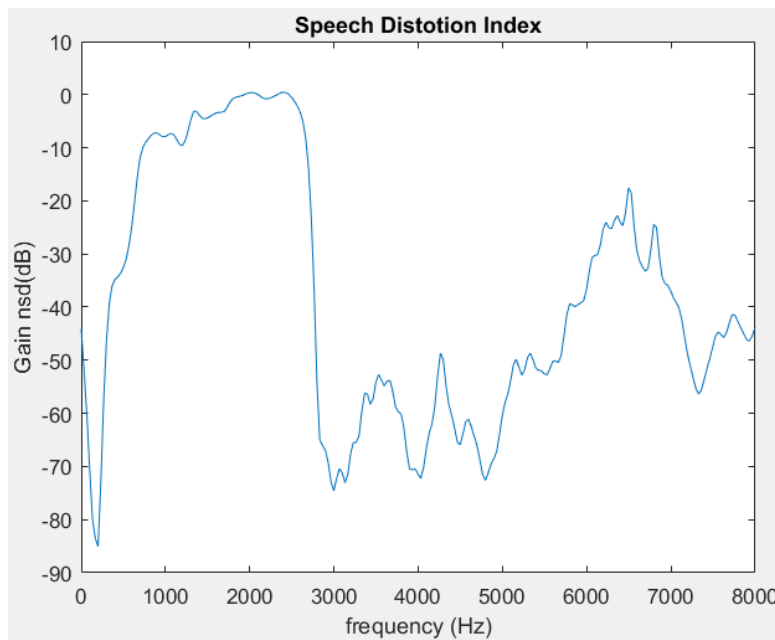


Σχήμα 2.1.Βα: Απόκριση του IIR Wiener φίλτρου

Βασιζόμενοι στην απόκριση συχνότητας του φίλτρου Wiener υπολογίζουμε το speech distortion index για να μετρήσουμε την παραμόρφωση που εισάγει το φίλτρο στο σήμα φωνής.

```
%calculate the speech distortion index  
nsd = (abs(1-H_w)).^2;
```

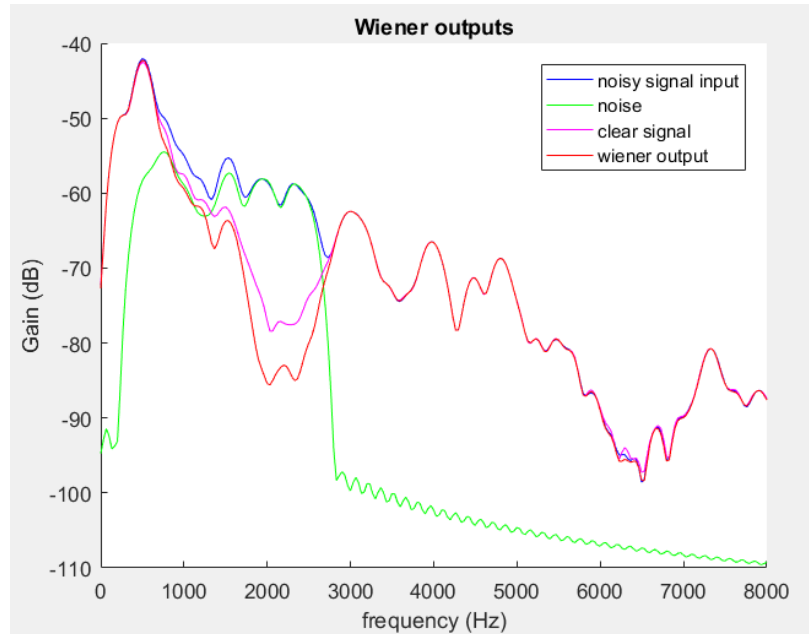
Σε λογαριθμική κλίμακα και σε συχνότητες [0,8]kHz



Σχήμα 2.1.B6: Παραμόρφωση ομιλίας

• Παρατηρούμε ότι στις συχνότητες θορύβου [500-2500 Hz] έχουμε μεγάλη παραμόρφωση, ενώ έχουμε καλή απόκριση στις συχνότητες της ανθρώπινης ομιλίας [20-250 Hz].

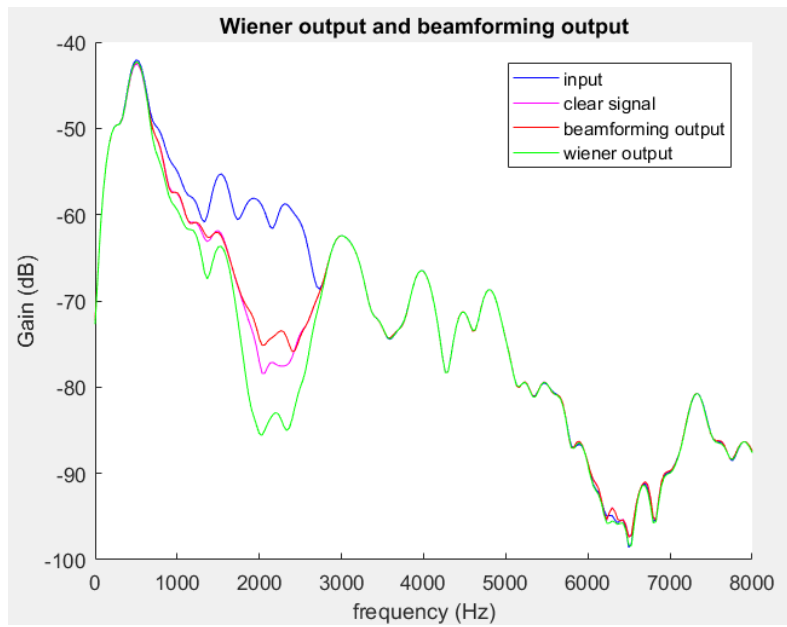
Παρακάτω σχεδιάζουμε τα φάσματα ισχύος του καθαρού σήματος $s(t)$, του θορυβώδους σήματος στην είσοδο $x(t)$, του θορύβου $u(t)=x(t)-s(t)$ και την έξοδο από το φιλτράρισμα Wiener αντίστοιχα.



Σχήμα 2.1.βγ: Έξοδος του Wiener φίλτρου για είσοδο θορυβώδες σήματος και σύγκριση με το καθαρό

Υπολογίζουμε πάλι τα SNR εισόδου του φίλτρου Wiener x_t και το SNR της εξόδου του και βρίσκουμε **SNR_{xt}=7.9415 dB** και **SNR_{out_t}=9.9849 dB**

Βρίσκουμε την βελτίωση της πολυκαναλικής μεθόδου (**SNR_{beamformer}=17.5836 dB**) έναντι του μονοκαναλικού φιλτραρίσματος ίση με **beltiwh_SNR=76.1016 %** . Επομένως συμπεραίνουμε ότι με την πολυκαναλική μέθοδο delay and sum beamformer επιτυγχάνουμε μείωση του θορύβου σε πολύ μεγαλύτερο βαθμό από ότι με το μονοκαναλικό Wiener φιλτράρισμα.



Σχήμα 2.1.Bδ: Σύγκριση μεταξύ Beamforming output και Wiener output

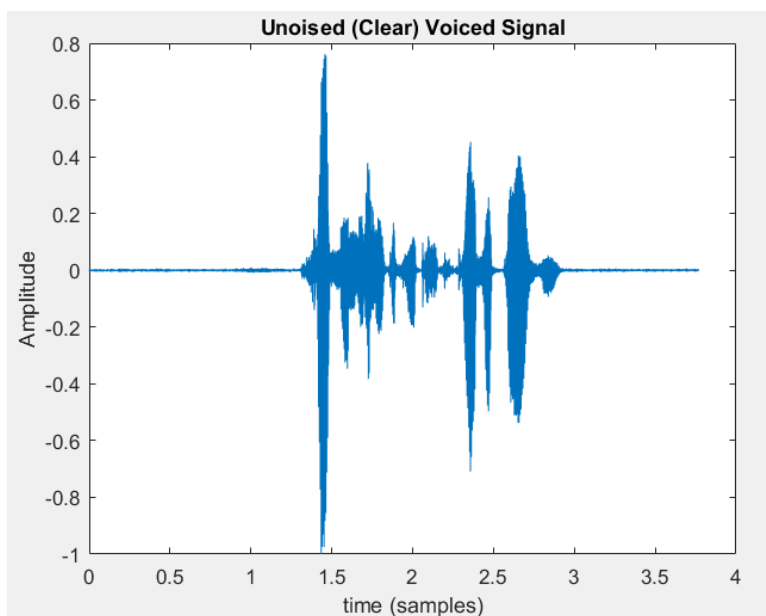
Από το παραπάνω σχήμα παρατηρούμε ότι πράγματι το σήμα που λαμβάνουμε στην έξοδο του συστήματος beamforming (κόκκινο χρώμα) προσεγγίζει καλύτερα το καθαρό σήμα φωνής $s(t)$ (ροζ χρώμα) από ότι η έξοδος από το φίλτρο Wiener (πράσινο χρώμα).

Ερώτημα 2.2.A Delay and sum beamforming (Real Signals):

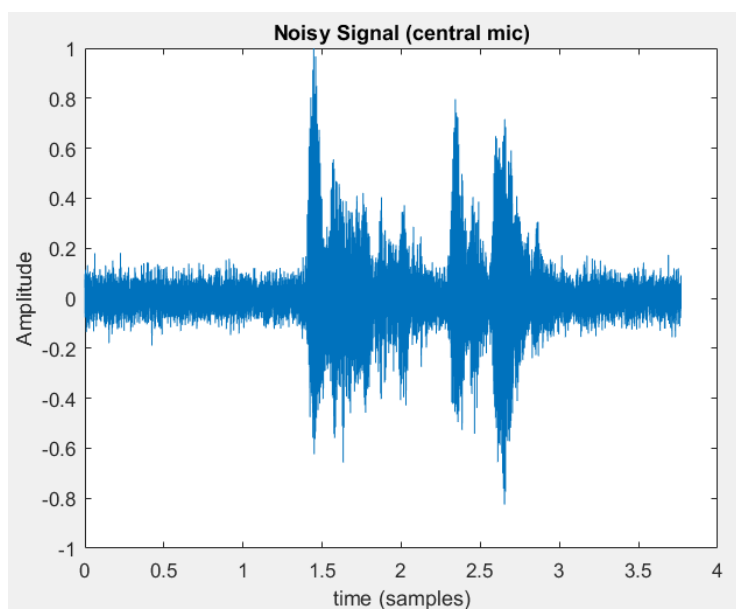
Εδώ έχουμε το ίδιο σενάριο με προηγουμένως μόνο που θόρυβο από διάφορες πηγές όπως ανεμιστήρες κ.α οι οποίες δημιουργούν ένα ισοτροπικό και ομογενές πεδίο θορύβου (diffuse noise field).

Ακολουθώντας την ίδια διαδικασία με το ερώτημα 2.1.A έχουμε τις εξής κυματομορφές και σπεκτρογραφήματα.

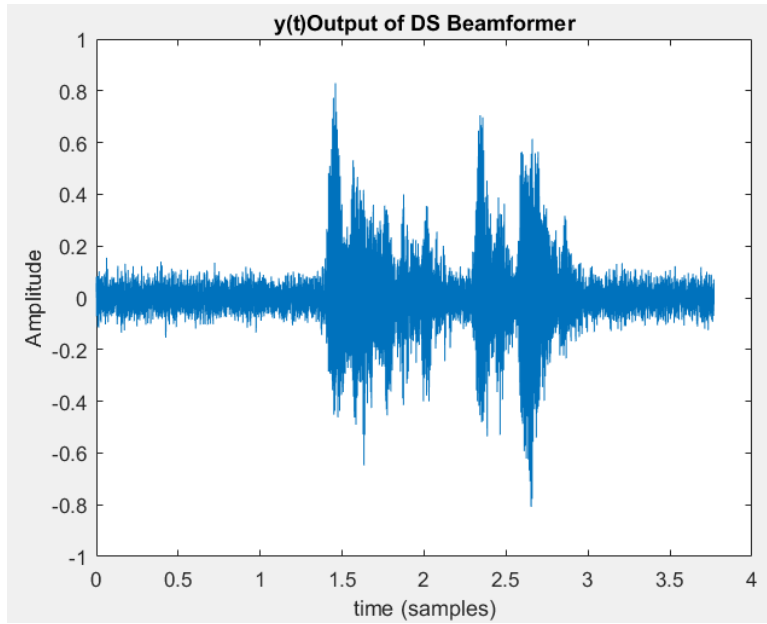
ΚΥΜΑΤΟΜΟΡΦΕΣ



Σχήμα 2.2.Αα: Κυματομορφή καθαρού σήματος

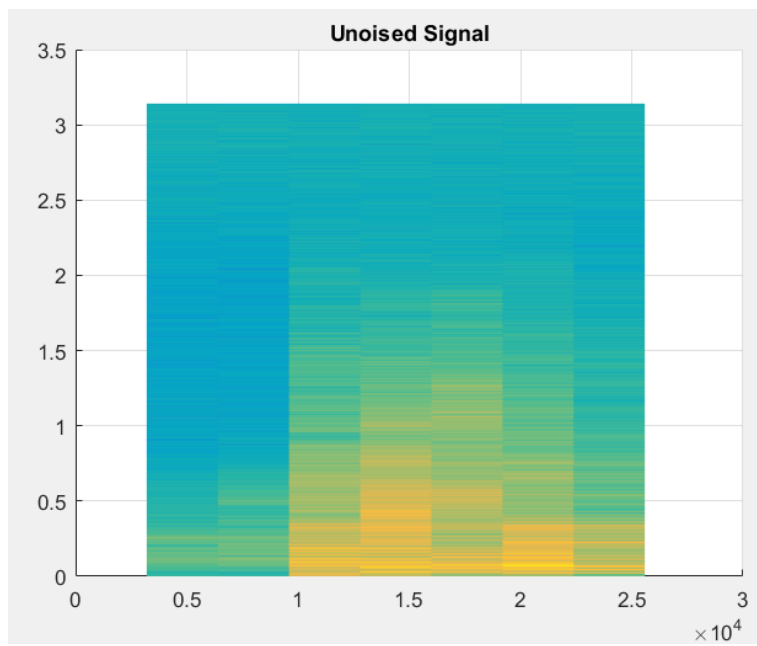


*Σχήμα 2.2.Αβ: Κυματομορφή θορυβώδους σήματος
κεντρικού μικροφώνου*

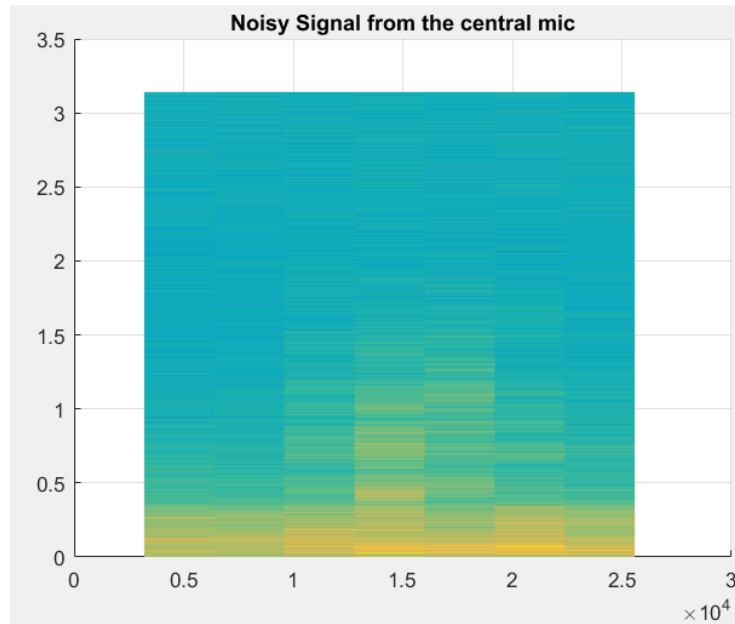


Σχήμα 2.2.Αγ: Κυματομορφή εξόδου delay and sum beamformer

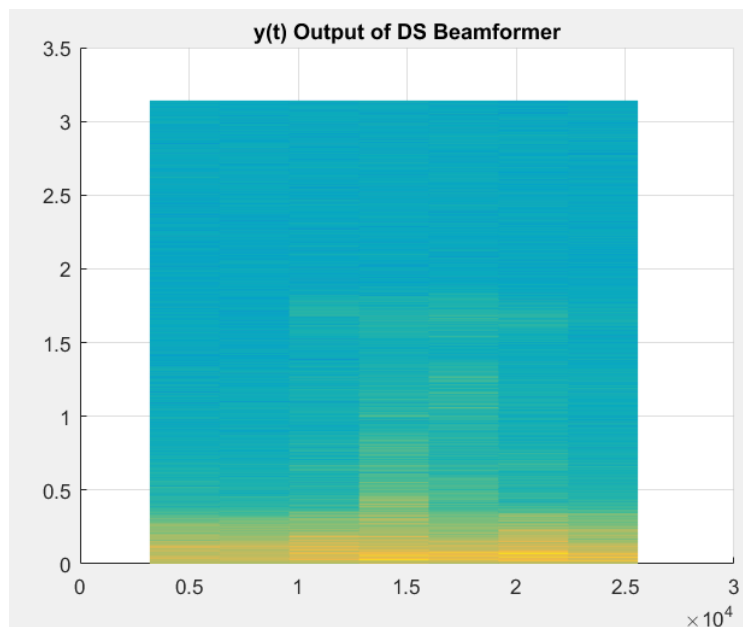
ΣΠΕΚΤΡΟΓΡΑΦΗΜΑΤΑ



Σχήμα 2.2.Αδ: Σπεκτρογράφημα καθαρού σήματος



Σχήμα 2.2.Αε: Σπεκτρογράφημα θορυβώδους σήματος
κεντρικού μικροφώνου



Σχήμα 2.2.Αζ: Σπεκτρογράφημα εξόδου delay and sum beamformer

Παρατηρώντας τα παραπάνω σχήματα βλέπουμε ότι η έξοδος από το beamformer είναι πιο κοντά στο θορυβώδες σήμα παρά στο καθαρό σήμα που σημαίνει ότι στα πραγματικά σήματα δεν επιτυγχάνεται καλή αποθορυβοποίηση.

Για τα σήματα φωνής έχει βρεθεί ότι το ολικό SNR ως μετρική ποιότητας δεν συμβαδίζει με την υποκειμενική αντίληψη που έχει ο άνθρωπος για το σήμα. Έτσι μετράμε το segmental SNR (SSNR) το οποίο ορίζεται ως το μέσο SNR των πλαισίων βραχέος χρόνου του σήματος φωνής:

$$SSNR = \frac{10}{M} \sum_{m=0}^{M-1} \log_{10} \frac{\sum_{n=Lm}^{Lm+L-1} s^2(n)}{\sum_{n=Lm}^{Lm+L-1} v^2(n)},$$

```
x=mic(4,:);
mic_noise=x(1:15000); %only noise from the central mic
%sound(mic_noise,fs);
t = 0.025; %seconds
win_len= t*fs; %window length (samples)
M=0;
sum=0;
P_mic_noise=mean(mic_noise.^2);
mic_framed=buffer(x',win_len);
[n,m]=size(mic_framed);
for i=1:m
    Px=mean(mic_framed(:,i).^2);
    Ps=abs(Px-P_mic_noise);
    SNR=10*log10(Ps/P_mic_noise);
    if SNR>0
        M=M+1;
        if SNR>35
            SNR=35;
        end
        sum=sum+SNR;
    end
end
SSNR_centralmic=sum/M;
display(SSNR_centralmic);

beamformer_noise=y_output(1:15000);
%sound(beamformer_noise,fs);
M=0;
sum=0;
P_beamformer_noise=mean(beamformer_noise.^2);
beamformer_framed=buffer(y_output,win_len);
[n,m]=size(beamformer_framed);
for i=1:m
    Px=mean(beamformer_framed(:,i).^2);
    Ps=abs(Px-P_beamformer_noise);
    SNR=10*log10(Ps/P_beamformer_noise);
    if SNR>0
        M=M+1;
        if SNR>35
            SNR=35;
        end
        sum=sum+SNR;
    end
end
SSNR_beamformer=sum/M;
display(SSNR_beamformer);
```

Με βάση τον παραπάνω κώδικα βρίσκουμε **SSNR_centralmic=9.7610 dB** και **SSNR_beamformer=10.2738 dB**

Το αποτέλεσμα της παραπάνω διαδικασίας αποθηκεύεται ως **real_ds.wav**. Ακούγοντας το παρατηρούμε ότι δεν αποτελεί πολύ καλή προσέγγιση του καθαρού σήματος χρησιμοποιώντας μόνο την μέθοδο delay and sum beamforming καθώς υπάρχει ακόμη αρκετός θόρυβος.

Ερώτημα 2.2.B Delay Post Filtering με Wiener φίλτρο

Υλοποιούμε τον beamformer του ερωτήματος 2.1.A και την έξοδο του την διαιρούμε με παράθυρα Hamming μήκους 1200 δειγμάτων (25ms μήκος παραθύρου) και επιλέγουμε επικάλυψη 50%. Σε κάθε πλαίσιο του παραθύρου εφαρμόζουμε μονοκαναλικό wiener φιλτράρισμα όπως στο ερώτημα 2.1.B και στην συνέχεια υλοποιούμε την μέθοδο overlap add για την ανακατασκευή του σήματος. Το σήμα εξόδου το ονομάζουμε **real_mmse.wav**.

```

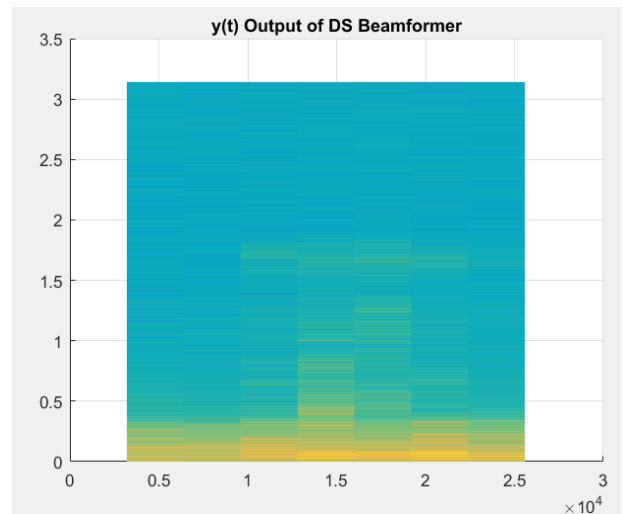
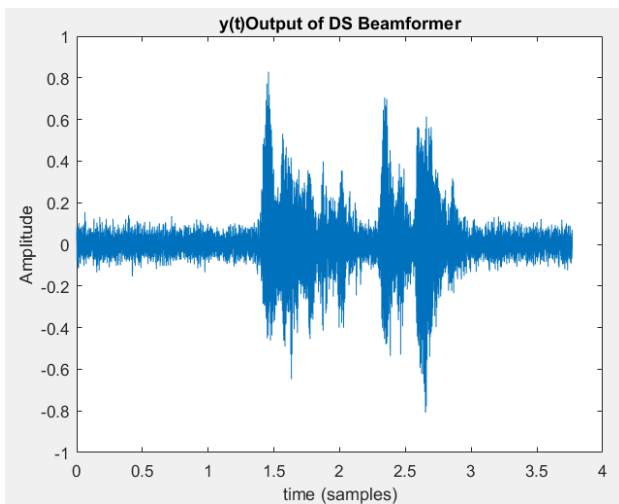
w=hamming(win_len);

u_noise=sig_framed(:,4); %only noise from the central mic
%sound(u_noise,fs);
for i=1:c
    [Px_framed,f]=pwelch(sig_framed(:,i),[],[],length(sig_framed(:,i)),'twosided');
    [Pu_noise,f]=pwelch(u_noise,[],[],length(u_noise),'twosided');
    H_w=1-(Pu_noise./Px_framed);
    y_windowed=sig_framed(:,i).*w;
    X_w=fft(y_windowed);
    filter_output(:,i)=ifft(H_w.*X_w); %filter output for each frame
end
len=length(y_output);
wiener_output=zeros(len,1);
%first 720 frames
for n=1:step
    wiener_output(n)=filter_output(n,1);
end
for i=2:(c-1)
    for n=1:win_len
        wiener_output((i-2)*step+n)=wiener_output((i-2)*step+n) + filter_output(n,i);
    end
end
for n=((c-1)*step)+1:len
    wiener_output(n)=filter_output(n-(step*(c-1)),c);
end

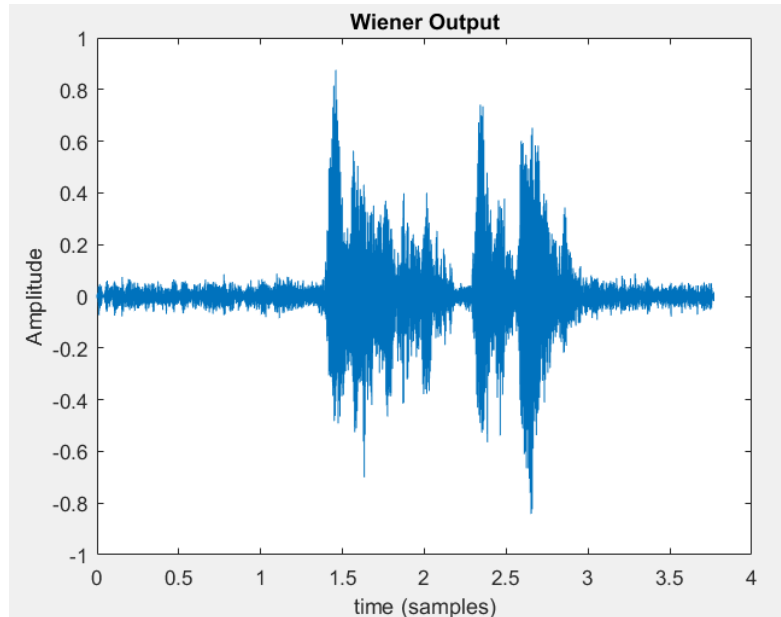
sound(wiener_output,fs);
audiowrite('real_mmse.wav',wiener_output,fs);

```

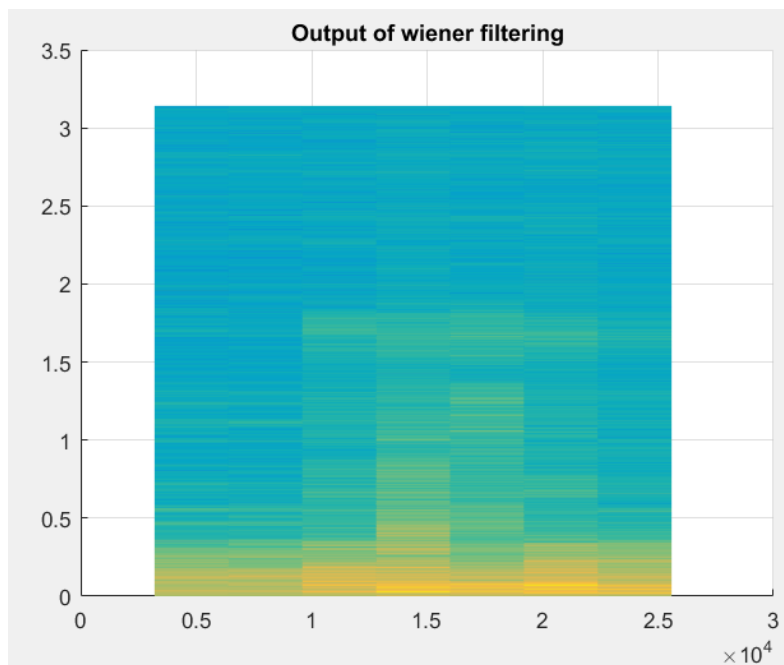
Η είσοδος του wiener φίλτρου είναι πρακτικά η έξοδος του beamformer (Σχ 2.2.Αγ και Σχ. 2.2.Αζ)



και η έξοδος του wiener φίλτρου είναι η εξής (κυματομορφή και σπεκτρογράφημα)



Σχήμα 2.2.Βα: Κυματομορφή εξόδου Wiener φίλτρου



Σχήμα 2.2.Ββ :Σπεκτρογράφημα εξόδου Wiener φίλτρου

Παρατηρούμε ότι έχουμε πετύχει επιπλέον αποθρομβοποίηση από την έξοδο του beamformer και το σήμα μας μοιάζει περισσότερο με το καθαρό μας σήμα (Σχ. 2.2.Αα και Σχ. 2.2.Αδ)

Στην συνέχεια υπολογίσαμε τα SSNR για την είσοδο και την έξοδο του Wiener φίλτρου τα οποία είναι **SSNR_beamformer=10.2738 dB** (είσοδος) και **SSNR_wiener=15.3405 dB** . Παρατηρούμε ότι έχουμε πετύχει αποθρομβοποίηση καλύτερη.

```
%Calculate SSNRs average
SSNRs_average=(SSNR_beamformer+SSNR_centralmic)/2;
display(SSNRs_average);

veltiwsh_SSNR=((SSNR_wiener-SSNRs_average)/SSNRs_average)*100;
display(veltiwsh_SSNR);
```

Ο μέσος όρος των SSNRs εισόδων του delay and sum beamformer και του wiener φίλτρου υπολογίστηκε όπως παραπάνω και είναι **SSNRs_average=10.0174 dB**

Και η βελτίωση υπολογίστηκε **veltiwsh_SSNR=53.8313 %** .Δηλαδή έχουμε πετύχει θεαματική βελτίωση σε σχέση με την μέθοδο του delay and sum beamforming μόνη της.

*Αρχεία κώδικα που συμπεριλαμβάνονται

lab3_1.m: ο κώδικας του 1ου μέρους

lab3_2_1.m: ο κώδικας του 2ου μέρους για Beamforming σε προσομοιωμένα σήματα

lab3_2_2.m: ο κώδικας του 2ου μέρους για Beamforming σε πραγματικά σήματα

