Θέμα: Εισαγωγή στην Ψηφιακή Επεξεργασία Σημάτων με Matlab

Μαρία Παρέλλη *Γεώργιος-Ορέστης Χαρδούβελης*
03115155 *03115100*
6ο Εξάμηνο *6ο Εξάμηνο*

Μέρος 1ο :Χαρακτηριστικά Βραχέος Χρόνου Σημάτων Φωνής και Μουσικής (Ενέργεια και Ρυθμός Εναλλαγής Πρόσημου)

Θεωρητικό υπόβαθρο

Στα σήματα φωνής μία μέθοδος για να διαχωρίσουμε τα τμήματα σιωπής από τα τμήματα ομιλίας είναι να μελετήσουμε τα χαρακτηριστικά βραχέος χρόνου τους. Συγκεκριμένα, στα τμήματα σιωπής αναμένουμε πολύ χαμηλή ενέργεια βραχέος χρόνου(οριακά μηδενική) και υψηλό ρυθμό εναλλαγής προσήμου, ενώ στα τμήματα ομιλίας αναμένουμε υψηλότερη ενέργεια και χαμηλότερο ρυθμό εναλλαγής προσήμου αντίστοιχα. Όλα αυτά θα αποδειχθούν πειραματικά παρακάτω.

1.1

Με τις εντολές :

```
filename='speech_utterance.wav';
x=audioread(filename);
f=16000;
```

Θεωρούμε το σήμα φωνής speech_utterance.wav και αποθηκεύουμε το δειγματοληπτημένο σήμα φωνής με συχνότητα δειγματοληψίας fs=16000 Hz στον πίνακα x.
Χωρίζουμε το σήμα σε πλαίσια μήκους L=d*fs και επικάλυψης L-1 ,όπου

d=20 msec: η διάρκεια του παραθύρου
fs=η συχνότητα δειγματοληψίας

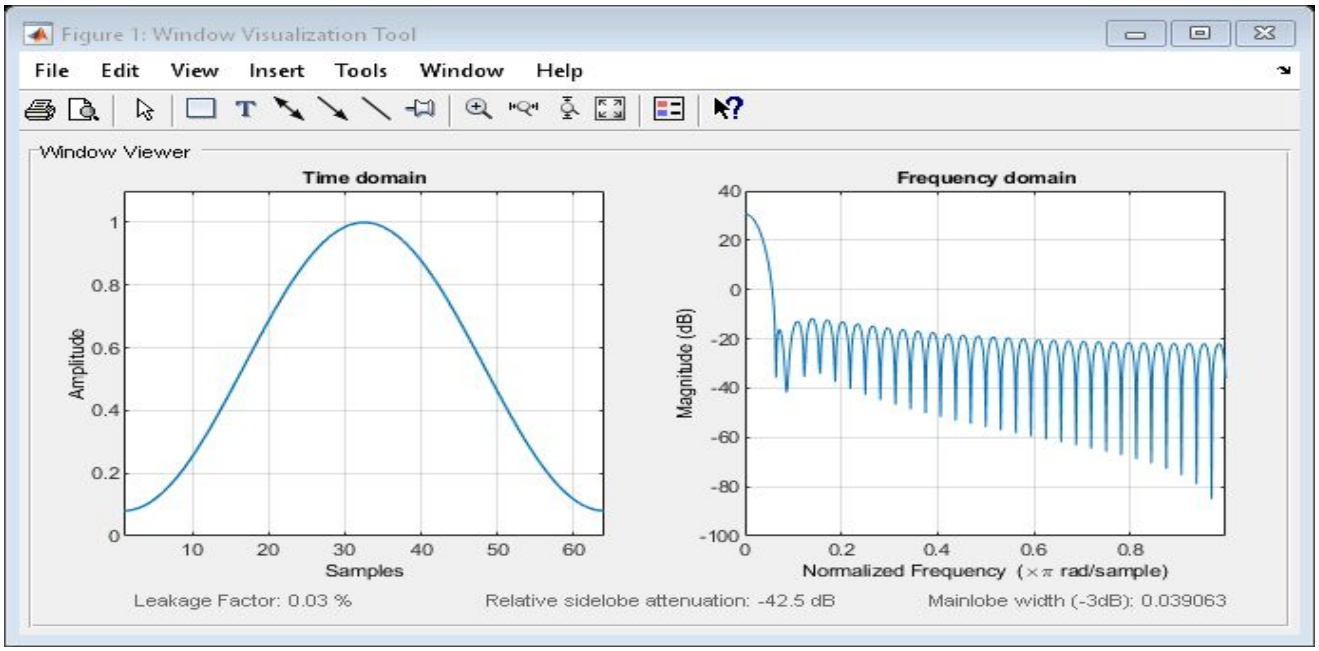
Για το παραπάνω χρησιμοποιήσαμε τη συνάρτηση **buffer** της Matlab, η οποία δέχεται ως όρισμα το σήμα, το μήκος του παραθύρου και την επικάλυψη και επιστρέφει έναν δισδιάστατο πίνακα με διαστάσεις L*Framesize,όπου Framesize:ο αριθμός των πλαισίων.

Στη συγκεκριμένη εφαρμογή χρησιμοποιούμε παράθυρα τύπου Hamming,τα οποία ορίζονται ως εξής:

$$w(n)=0.54+0.46cos\left(\frac{2\pi n}{M-1}\right) \quad 0 \le n \le M-1$$

Το παράθυρο Hamming προκύπτει χρησιμοποιώντας τη συνάρτηση hamming,η οποία δέχεται ως όρισμα το μήκος του παραθύρου:

h=hamming(L)



Εφαρμόζουμε σε κάθε πλαίσιο του σήματος το παράθυρο Hamming μετατοπισμένο κατάλληλα με τον εξής τρόπο:

```
 for i=1:n
    N(:,i)=M(:,i).*h;
end
```

Το παραπάνω αποτελεί ουσιαστικά τη συνέλιξη του σήματος φωνής με το παράθυρο hamming.

<u>Ενέργεια βραχέος χρόνου παραθυρωμένου σήματος</u>

Υπολογίζουμε την ενέργεια σε κάθε παράθυρο του σήματος ως εξής:

$$E = \sum_{i=0}^{N-1} s_i^2$$

Όπου Ν:το μήκος του παραθύρου

Η ενέργεια βραχέος χρόνου του σήματος προκύπτει ώς εξής:

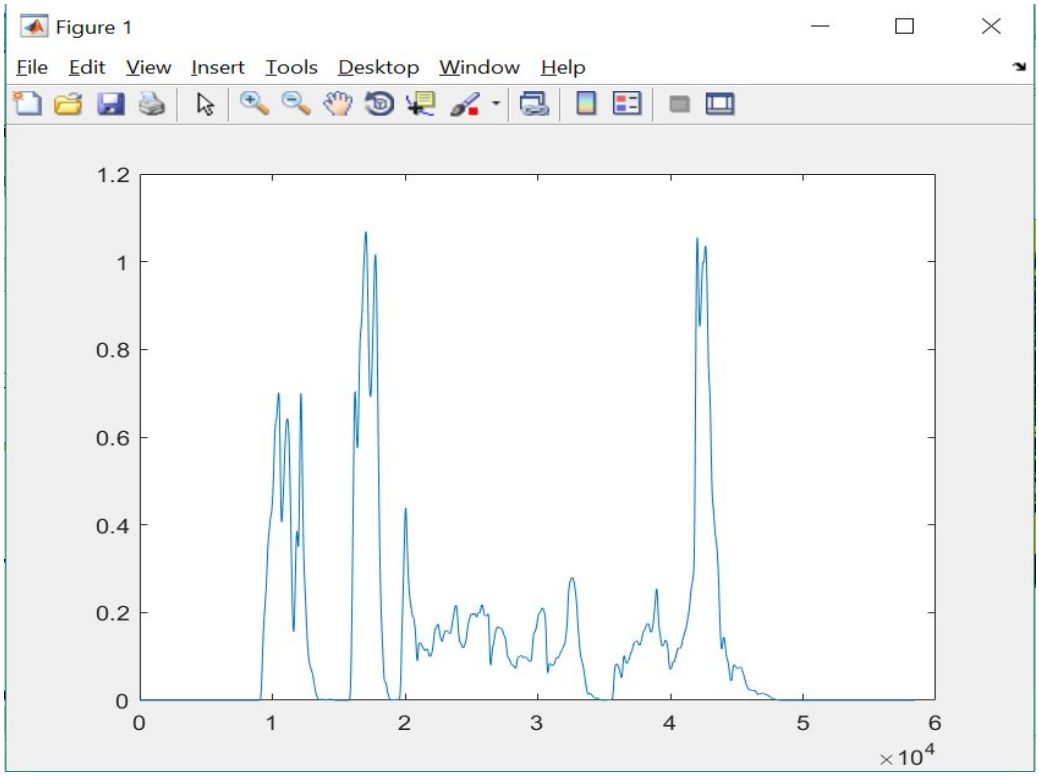
$$e(n) = \sum_{m=-\infty}^{\infty} (s(m).w(n-m))^2$$

.

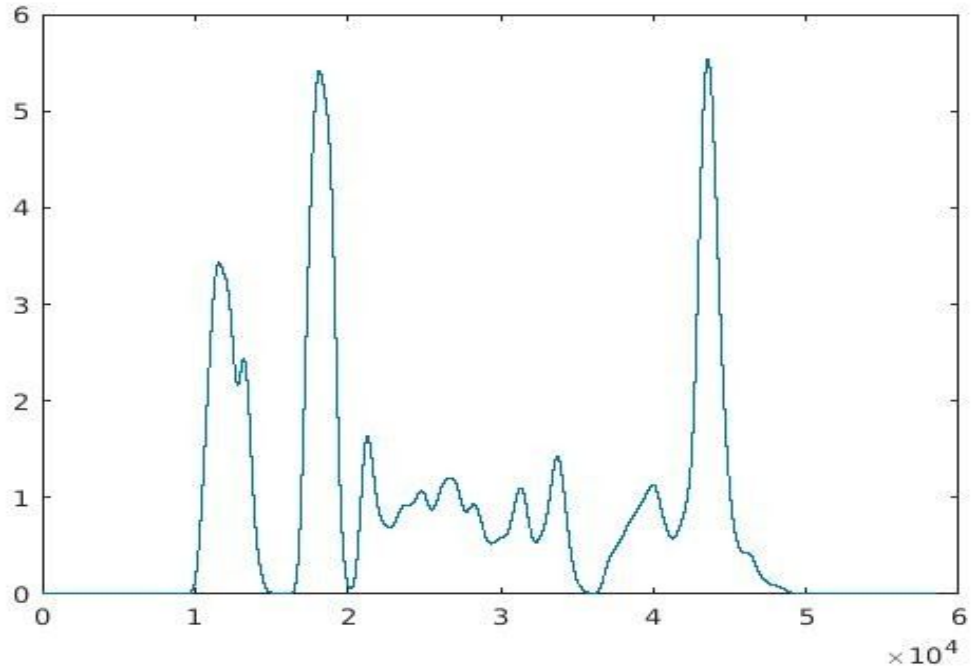
Επομένως, υπολογίζουμε αρχικά το τετράγωνο του κάθε στοιχείου από τον ήδη προκύπτον πίνακα ύστερα από την εφαρμογή του Hamming παραθύρου (*Na= N.^2;*) και ύστερα σχηματίζουμε τον μονοδιάστατο πίνακα (*En=sum(Na,1);)* ο οποίος σε κάθε στήλη/στοιχείο-i περιέχει το άθροισμα των στοιχείων της προϋπάρχουσας i-οστής στήλης του Na, δηλαδή ακριβώς το $[s(m) \cdot w(n-m)]^2$ **.**

Η ενέργεια βραχέος χρόνου προκύπτει:

- Για παράθυρο hamming μήκους 20ms:



- Για παράθυρο hamming μήκους 120ms:



Παρατήρηση1: Καθώς αυξάνεται το μήκος του παραθύρου τόσο μειώνεται και η ευκρίνεια στο πεδίο του χρόνου. Παρατηρούμε ότι οι καμπύλες της ενέργειας γίνονται πιο λείες (smooth).

Παρατήρηση 2: Σε αυτό το σημείο εύκολα διαχωρίζουμε **φωνή** από **σιωπή**, καθώς στα σημεία που είχαμε σιωπή στην γραφική παράσταση της ενέργειας έχουν τεταγμένη που τείνει στο μηδέν.

Παρατήρηση 3:Όσον αφορά στους **έμφωνους** και **άφωνους ήχους**, οι έμφωνοι ήχοι παρατηρούνται στα σημεία της γραφικής παράστασης ενέργειας που έχουμε peaks (π.χ. /α/, /ο/) , ενώ σε εκείνα με χαμηλότερες τιμές σε σχέση με το υπόλοιπο σήμα παρατηρούνται άφωνοι ήχοι (π.χ. /τ/). Αυτό συμβαίνει διότι οι έμφωνοι ήχοι έχουν μεγαλύτερη περιοδικότητα και άρα μεγαλύτερη ενέργεια. Αντίθετα για την παραγωγή των άφωνων ήχων οι φωνητικές χορδές μένουν ανοιχτές και άρα στη παράσταση λειτουργούν ως θόρυβος.

Ρυθμός εναλλαγής προσήμου

Συνεχίζοντας, παρόμοια διαδικασία ακολουθήθηκε για τον ρυθμό εναλλαγής προσήμου Zn, ο οποίος πρακτικά υποδηλώνει πόσες φορές περνάει το πλάτος του σήματος από τον οριζόντιο άξονα.

Ο ρυθμός εναλλαγής προσήμου υπολογίζεται ωχ εξής:

$$Zn = \sum_{m=-\infty}^{\infty} |sgn[x[m]] - sgn[x[m-1] \cdot w[n-m]|$$

Με τις εντολές λοιπόν:

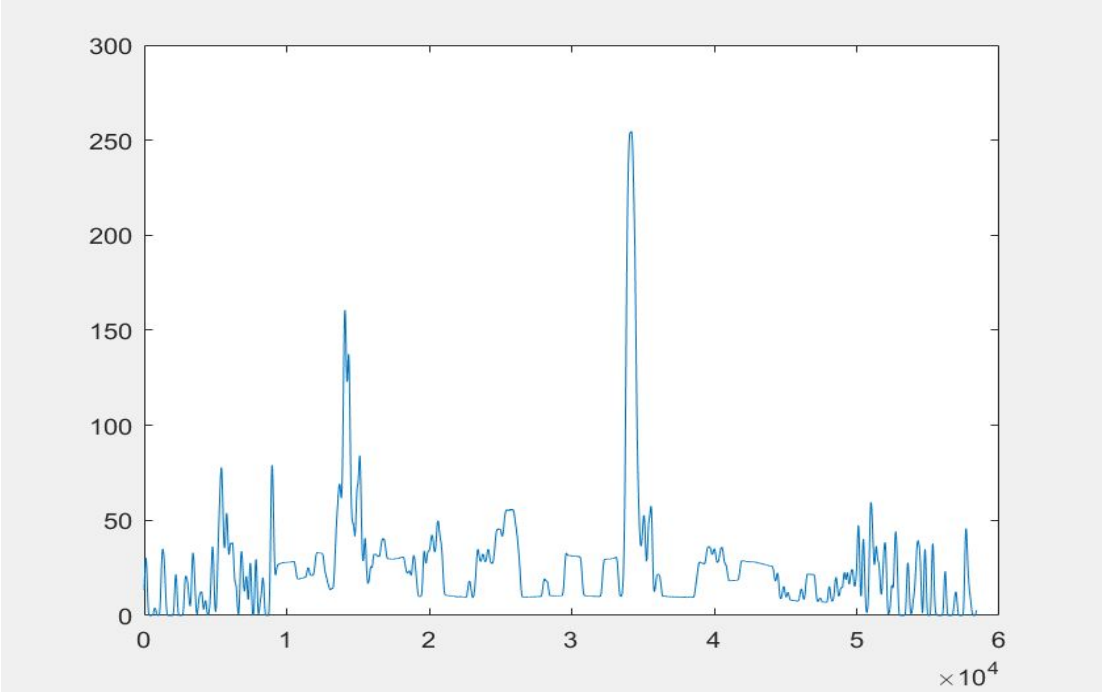
```
xn(1)=x(1);
for i=2:length(x)
   xn(i)=abs(sign(x(i))-sign(x(i-1)));
end
```

Δημιουργούμε τον πίνακα xn που αποτελεί το πρώτο μέρος του αθροίσματος και ύστερα το πολλαπλασιάζουμε όπως παραπάνω με το hamming παράθυρο.

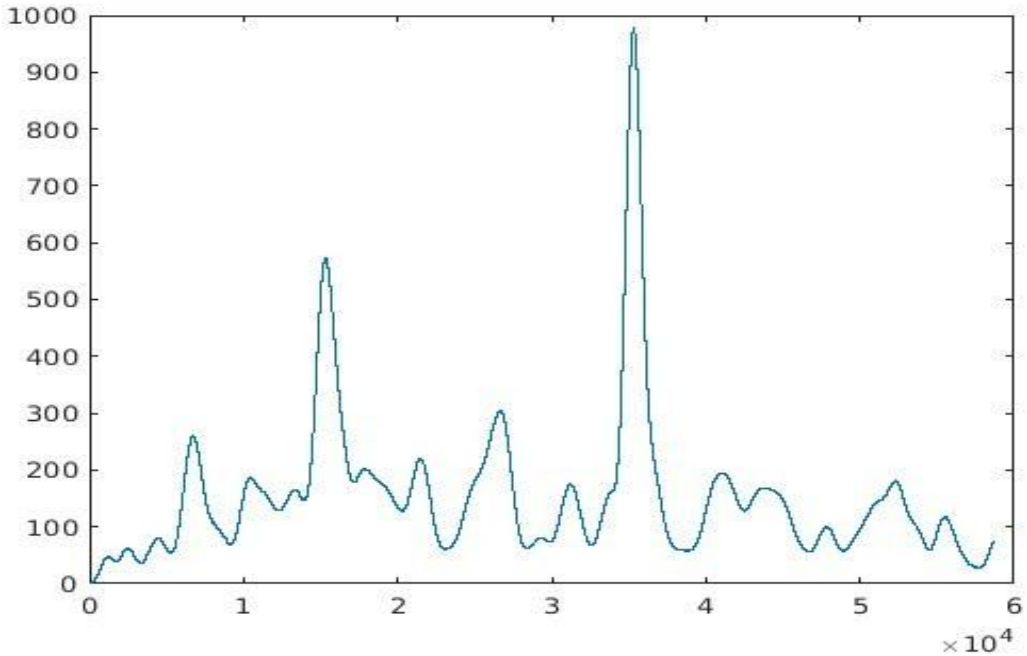
Τέλος, με την εντολή *Zn=sum(N,1)* δημιουργούμε τον μονοδιάστατο πίνακα Zn που το i-οστό του στοιχείο είναι το άθροισμα των στοιχείων της προϋπάρχουσας i-οστής στήλης στον πίνακα N.

Έτσι προκύπτει η γραφική παράσταση του ρυθμού εναλλαγής προσήμου:

- Για παράθυρο hamming μήκους 20ms:



- Για παράθυρο hamming μήκους 120 ms



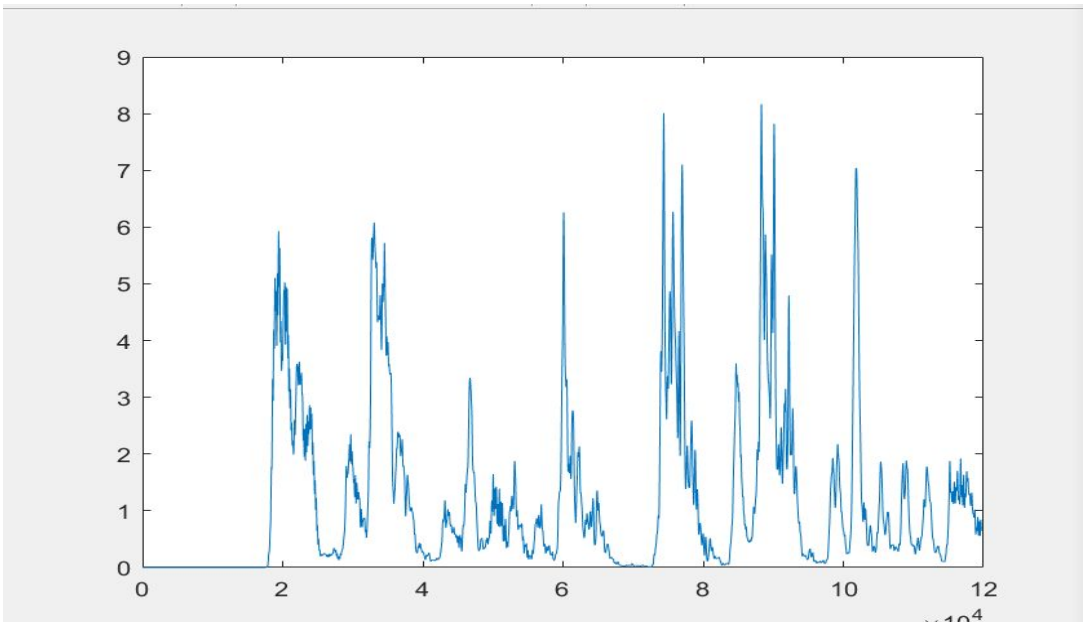
<u>Παρατήρηση1</u>: Καθώς αυξάνεται το μήκος του παραθύρου τόσο μειώνεται και η ευκρίνεια στο πεδίο του χρόνου. Παρατηρούμε ότι οι καμπύλες του ρυθμού εναλλαγής προσήμου γίνονται πιο λείες(smooth).

<u>Παρατήρηση 2</u>: Σε αυτό το σημείο εύκολα διαχωρίζουμε **φωνή** από **σιωπή**, καθώς στα σημεία που έχουμε σιωπή ,στην γραφική παράσταση έχουν πολύ υψήλό ρυθμό εναλλαγής προσήμου.

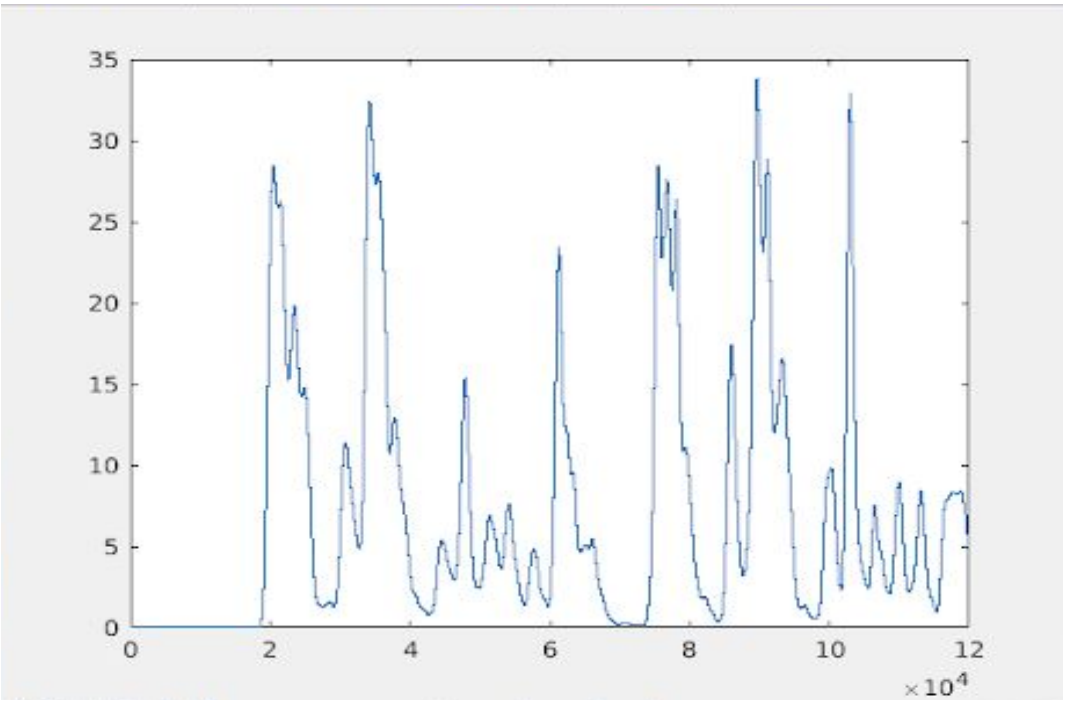
<u>1.2</u>

Επαναλαμβάνουμε τώρα ακριβώς την ίδια διαδικασία για το σήμα μουσικής “music.wav”. Η **ενέργεια βραχέους χρόνου** που προκύπτει με παράθυρο hamming είναι:

- παράθυρο hamming μήκους 20ms:

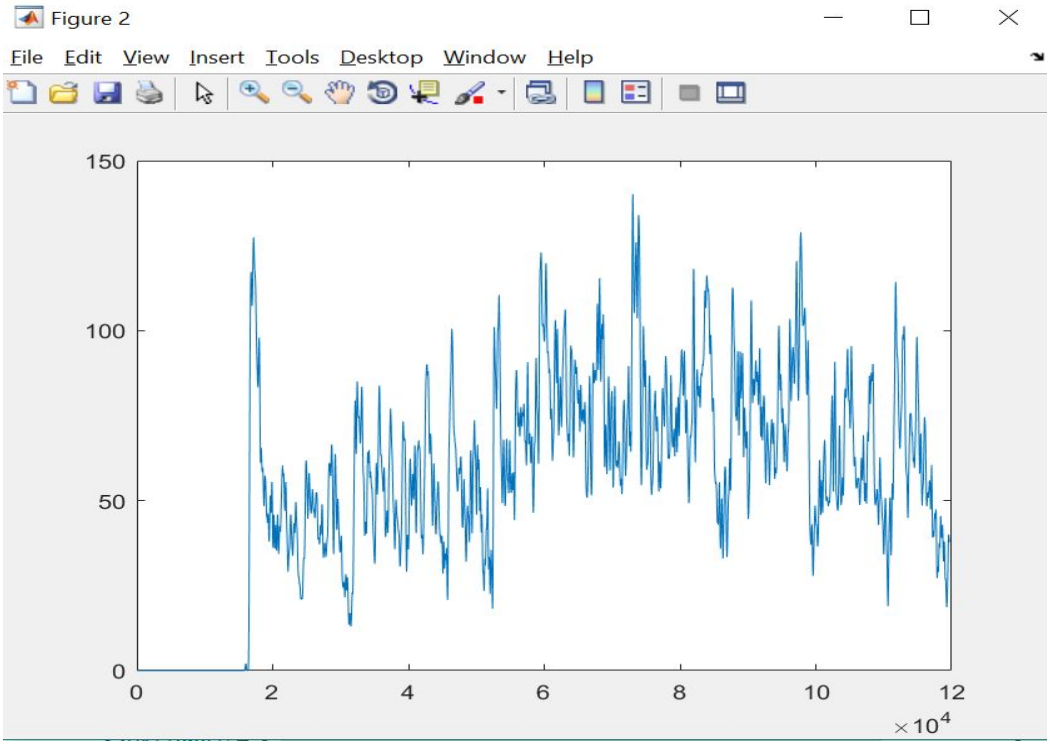


- παράθυρο hamming μήκους 120ms:

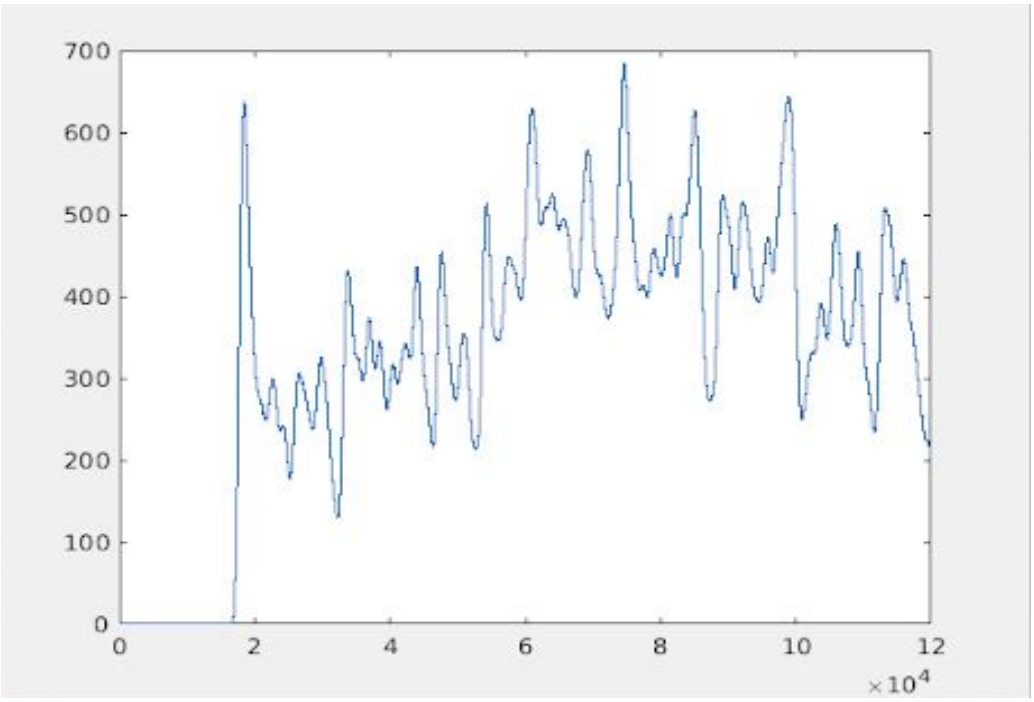


Αντίστοιχα, για τον ρυθμό εναλλαγής προσήμου έχουμε για:

- παράθυρο hamming μήκους 20ms:



- παράθυρο hamming μήκους 120ms:



Παρατηρούμε πως τα συμπεράσματα είναι όμοια με εκείνα του ερωτήματος ***1.1***,με τις εξής προσθήκες:

1. Στην γραφική παράσταση της ενέργειας έχουμε περισσότερες κορυφές,δηλαδή περιοχές υψηλής ενέργειας,το οποίο είναι λογικό εφόσον έχουμε σήμα μουσικής.
2. Ο ρυθμός εναλλαγής προσήμου είναι κατά μέσο όρο υψηλότερος από ότι στο σήμα φωνής.

Μέρος 2ο - Ανάλυση και Σύνθεση Σήματος με τον Μετ/σμό Fourier Βραχέος Χρόνου (STFT)

Θεωρητικό υπόβαθρο

Ο μετασχηματισμός Fourier είναι ο πιο διαδεδομένος μετασχηματισμός όσον αφορά την ανάλυση σημάτων. Παρόλα αυτά, όταν πρόκειται για χρονικά μεταβαλλόμενα σήματα (non-stationary signals) που επιθυμούμε μια τρισδιάστατη ανάλυση σε σχέση τόσο με τον χρόνο

όσο και την συχνότητα δεν μας καλύπτει ο FT. Έτσι χρησιμοποιούμε τον STFT (Μετ/σμός Fourier Βραχέος Χρόνου) ,που μας δίνει αυτή την δυνατότητα.
Επιπλέον, με τον μετασχηματισμό STFT, όπως θα δούμε και παρακάτω, μπορούμε εύκολα να ανασυνθέσουμε το αρχικό σήμα δεδομένου ότι τηρούνται κάποιες συνθήκες μεταξύ του μήκους και της επικάλυψης για το συγκεκριμένο είδος παραθύρου.

2.1

Το παρόν ερώτημα παραπέμπει για αρχή στην προηγούμενη άσκηση καθώς πραγματοποιείται όμοια διαδικασία. Παρόλα αυτά δεν πραγματοποιείται άμεσα στο πρόγραμμα αλλά μέσω της συνάρτησης mySTFT, ή οποία παίρνει ως μεταβλητές το σήμα, την συχνότητα δειγματοληψίας, το μήκος παραθύρου και τέλος το βήμα ανάλυσης.

Μέσα στην συνάρτηση, αρχικά, μέσω της συνάρτησης buffer που επιστρέφει έναν δισδιάστατο πίνακα με διαστάσεις L*Framesize,όπου Framesize:ο αριθμός των πλαισίων.

Ύστερα, εφαρμόζουμε σε κάθε πλαίσιο του πίνακα το παράθυρο hamming και τέλος, από τον προκύπτον πίνακα N, υπολογίζουμε τον Short Time Fourier Transform - STFT.

Γνωρίζουμε πως ισχύει:

$$STFT(\tau, \omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot w[n-\tau] \cdot e^{-j\omega n}$$

Οπότε για τον μετασχηματισμό του πίνακα οφείλουμε να μετασχηματίσουμε κάθε στοιχείο του πίνακα.

Εδώ είναι θεμιτό να αναφέρουμε ότι στην MATLAB χρησιμοποιείται ο μετασχηματισμός DFT έναντι του DTFT με $\omega_k = 2\pi k/N$ *με N=nfft* $\geq$ *L.Χρησιμοποιείται ο αλγόριθμος fft.*

Για να υπολογίσουμε τον STFT του πίνακα N καλούμε την εντολή fft ως εξής:

STFT=fft(N,nfft);

Η συνάρτηση πέρα από τον μετ/σμό STFT επιστρέφει και άλλα μεγέθη όπως,τους άξονες του χρόνου και της συχνότητας,το μήκος του DTFT,το μήκος L του κάθε πλαισίου, τον αριθμό των στηλών n του παραθυροποιημένου μετασχηματισμένου σήματος, καθώς και τον πίνακα N πριν υποστεί τον μετ/σμό Fourier, γιατί θα χρησιμοποιηθούν σε παρακάτω ερωτήματα.

Επομένως, καλώντας την εντολή *[STFT, f, t, nfft, L, n, N]= mySTFT(x, fs, d, 0.02); ,* έχουμε στην μεταβλητή STFT τον αντίστοιχο μετασχηματισμό STFT του σήματος.

2.2

Για να γίνει η απεικόνιση του πλάτους του STFT έπρεπε να θέσουμε τις κατάλληλες τιμές στους άξονες του χρόνου και τη συχνότητας με αντίστοιχους πίνακες,f,t,τους οποίους επιστρέφει η συνάρτηση mySTFT.
Οι συχνότητες είναι οι συχνότητες συνεχούς χρόνου με :
$f\kappa = \omega\kappa \cdot fs/2\pi$

Αυτό γίνεται διαδοχικά με τις εντολές:
*t=(L/2:hopsize:L/2+(c-1)*hopsize)/fs;*

και

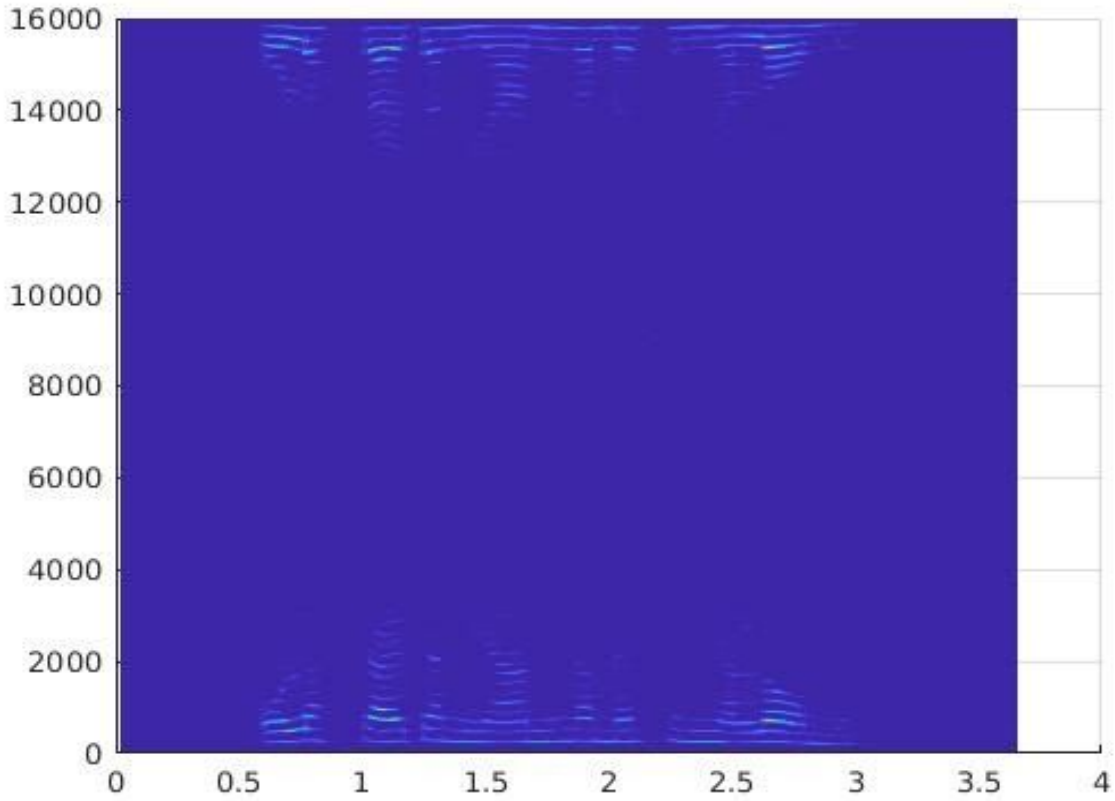
```
for i=1:L
    y(i)=fs*i/L;
end
```

΄Υστερα έχουμε την τρισδιάστατη απεικόνιση του πίνακα |STFT|, που αποτελεί συνάρτηση 2 μεταβλητών, μέσω της συνάρτησης surf.
Αυτό γίνεται ως εξής:

```
figure();
surf(t,y,abs(STFT), 'edgecolor','none');
view(0,90);
xlabel('Time');
ylabel('Frequency');
```

Το κομμάτι "*'edgecolor','none'*" είναι σημαντικό ώστε να μην έχει περίγραμμα, δεδομένου πως έχουμε επικαλύψεις σε τέτοιο βαθμό ώστε το περίγραμμα να καλύπτει την απεικόνιση μας (ή μέρος της).

Η απεικόνιση φαίνεται παρακάτω:



Στη συνέχεια, συνεχίζουμε απομονώνοντάς τα χρονικά τμήματα του STFT που αντιστοιχούν στα φωνήεντα /α/ και /ο/ (δύο για το καθένα).

Ακούγοντας προσεκτικά το σήμα παρατηρούμε πως για πρώτη φορά το φωνήεν ‘α’ ακούγεται στο χρονικό διάστημα [0.748243, 0.797224] ενώ το ‘ο’ στο [0.570688, 0.662527]. Πολλαπλασιάζοντας με την συχνότητα δειγματοληψίας βλέπουμε πως αυτά τα φωνήεντα αντιστοιχούν το ‘α’ στο εύρος δειγμάτων [1198, 1276] ενώ το ‘ο’ στο [913, 10601].

Στη συνέχεια, πηγαίνουμε στο παραθυροποιημένο σήμα μας αποθηκευμένο στον πίνακα N και εντοπίζουμε που βρίσκονται αυτά τα δείγματα, σπάζοντας στα σε δύο μέρη για κάθε φωνήεν.

Εύκολα προκύπτει πως δύο τμήματα για το /α/ είναι:
-στην 2η στήλη από την γραμμή 593 μέχρι τέλους (640)
-στην 3η στήλη από την γραμμή 1 μέχρι την γραμμή (420)

Αντίστοιχα για το /ο/ έχουμε:
-στην 3η στήλη από την γραμμή 559 μέχρι την γραμμή 600
-στην 3η στήλη από την γραμμή 601 μέχρι την γραμμή 636

Έτσι, απομονώνουμε τους αντίστοιχους πίνακες, ενώ για να τους απεικονίσουμε αργότερα συναρτήσει της μεταβλητής y, κάνουμε το μέγεθος κάθε πίνακα ίσο με L προσθέτοντας μηδενικά. Όλα τα παραπάνω γίνονται με τις εξής εντολές:

```
alpha1=N(593:end,2);
alpha1=padarray(alpha1,(L-length(alpha1))/2);

alpha2=N(1:420,3);
alpha2=padarray(alpha2,(L-length(alpha2))/2);

omikron1=N(559:600,3);
omikron1=padarray(omikron1,(L-length(omikron1))/2);

omikron2=N(601:636,3);
omikron2=padarray(omikron2,(L-length(omikron2))/2);
```

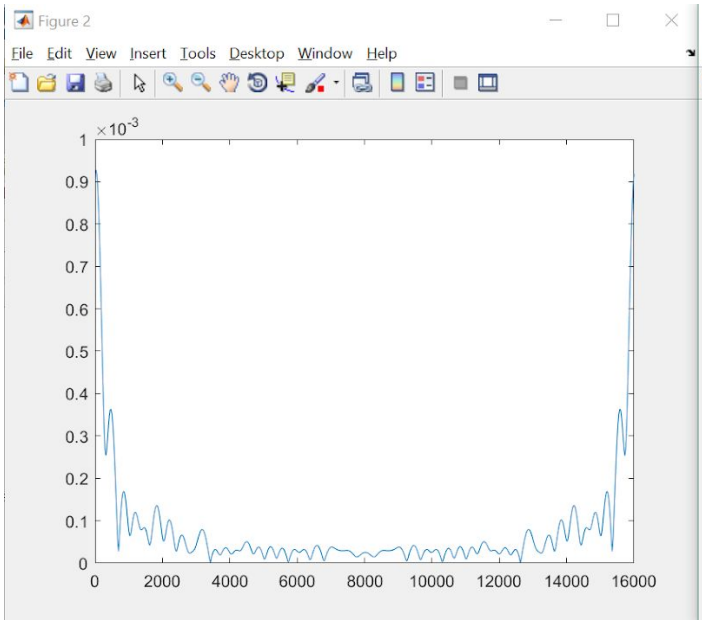
Εν συνεχεία, για κάθε ένα από τους παραπάνω πίνακες υπολογίζουμε το μέτρο του μετ/σμού Fourier του (δηλαδή το μέτρο του μετ/σμού Fourier του κάθε στοιχείου). Τέλος, απεικονίζουμε τα αποτελέσματα συναρτήσει της συχνότητας. Τα παραπάνω υλοποιούνται ως εξής:

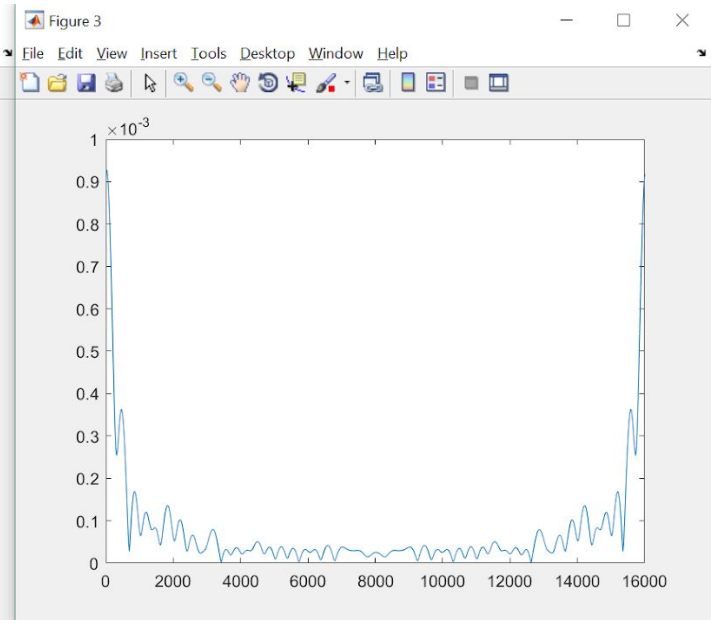
```
a_f1=abs(fft(alpha1));
a_f2=abs(fft(alpha1));
o_f1=abs(fft(omikron1));
o_f2=abs(fft(omikron2));

figure();
plot(y,a_f1);
figure();
plot(y,a_f2);
figure();
plot(y,o_f1);
figure();
plot(y,o_f2);
```

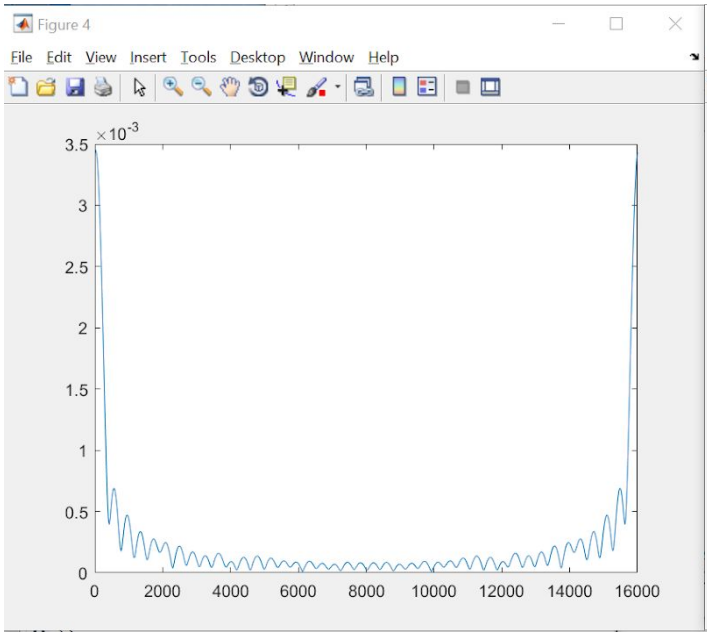
Τα αποτελέσματα που προκύπτουν είναι:

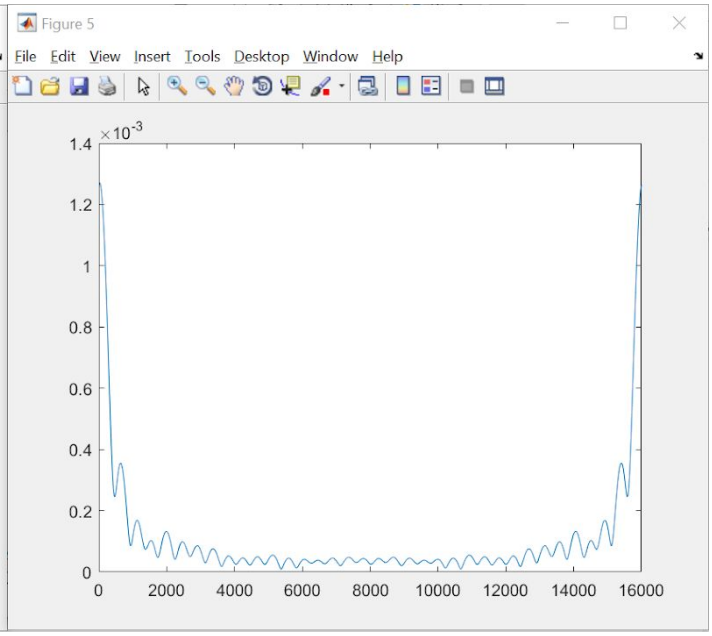
- Για τους 2 πρώτους πίνακες που αντιπροσωπεύουν το φωνήεν /α/:





- Για τους 2 τελευταίους πίνακες που αντιπροσωπεύουν το φωνήεν /ο/:





Παρατηρούμε τόσο από τις παραπάνω παραστάσεις όσο και από το φασματογράφημα,ότι το φωνήεν /α/ έχει περισσότερες υψηλότερες αρμονικές από ότι το /ο/.

2.3

Για να ανακατασκευάσουμε το αρχικό σήμα από τον STFT εφαρμόζουμε την αντίστροφη διαδικασία. Ακολουθούμε τα εξής βήματα.

1. Για την ανακατασκευή του κάθε παραθύρου εφαρμόζουμε αντίστροφο IDFT μετσχηματισμό.Για το σκοπό αυτό χρησιμοποιούμε τη συνάρτηση **ifft** του Matlab,η οποία δέχεται ώς ορίσματα τον STFT,που επέστρεψε η συνάρτηση mySTFT και το μήκος του DTFT(nfft).

2. Για την ανακατασκευή του αρχικού σήματος από τα επιμέρους ανακατασκευασμένα γειτονικά παράθυρα χρησιμοποιούμε την μέθοδο **Overlap-Add** και συνεπώς το τελικό σήμα προκύπτει με την κατάλληλη τοποθέτηση των πλαισίων στο χρόνο και πρόσθεση των επικαλυπτόμενων πλαισίων. Αυτό υλοποιήθηκε ως εξής:

```
[Framesize,nFrames]=size(IFFT);
shift=Framesize/2;
output=zeros((nFrames-1)*shift+Framesize,1);

for k=1:nFrames
  start=(k-1)*shift+1;
  finish=Framesize+start-1;
  output(start:finish)=IFFT(:,k)+output(start:finish);
end
```

Με τη συνάρτηση **sound** ακούμε το ανακατασκευασμένο σήμα και παρατηρούμε ότι είναι όμοιο με το αρχικό.

2.4

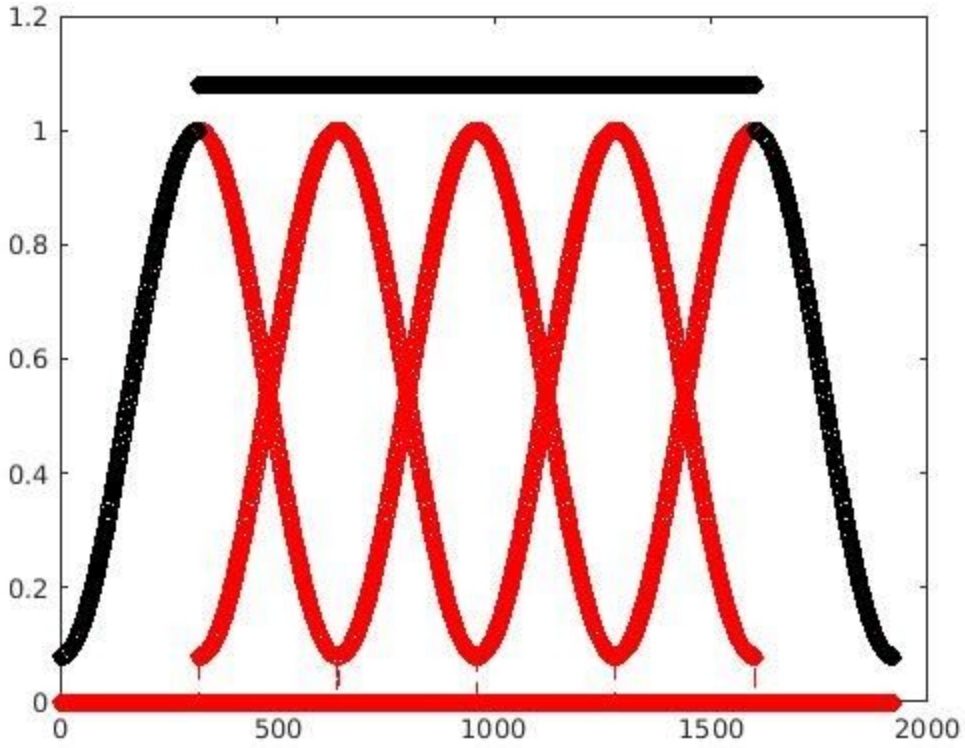
Για να είναι δυνατή η ανακατασκευή ενός σήματος από τα επικαλυπτόμενα παράθυρά του με απλή πρόσθεση τους στις αρχικές χρονικές τους θέσεις ,δηλαδή

$$x[n] = \sum_{m=-\infty}^{\infty} x_m[n] = x[n] * \sum_{m=-\infty}^{\infty} w[n - m * R]$$

θα πρέπει να ισχύει η εξής συνθήκη :

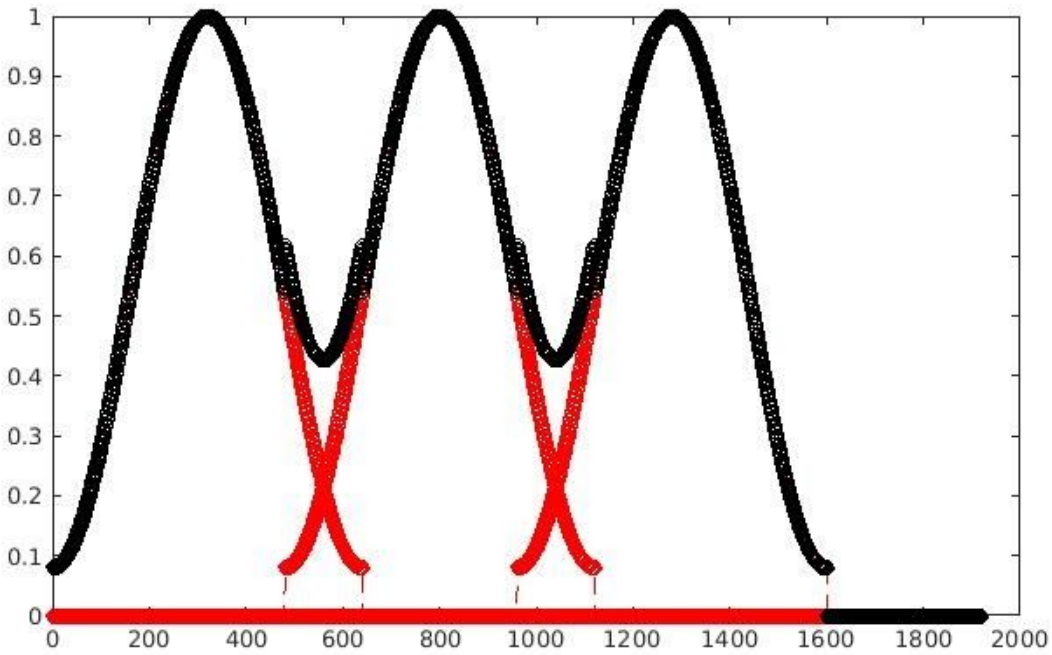
$$\sum_{m=-\infty}^{\infty} w[n - m \cdot R] = 1 \quad \forall n \in Z$$

Καλώντας τη συνάρτηση,που βρίσκεται στο υλικό ola.m με ορίσματα το μήκος παραθύρου (L), το βήμα ανάλυσης (L/2) και το winoverlap-add span λαμβάνουμε την εξής εικόνα:



Παρατηρούμε,από τη σταθερή μαύρη γραμμή ότι η συνθήκη **OLA** για τις παραμέτρους που χρησιμοποιήσαμε ικανοποιείται.

Αν χρησιμοποιήσουμε Hamming παράθυρο μήκους 40 ms και βήμα ανάλυσης 30 msec,παρατηρούμε ότι η επικάλυψη είναι L/4, δηλαδή υποδιπλάσια από το προηγούμενο ερώτημα. Σε αυτήν την περίπτωση,η συνάρτηση ola μας δίνει την παρακάτω εικόνα:



Παρατηρούμε ότι η συνθήκη OLA δεν ικανοποιείται για τις παραμέτρους αυτές και συνεπώς δεν είναι δυνατή η πλήρης ανακατασκευή του σήματος φωνής. Το γεγονός αυτό επιβεβαιώνεται και από το άκουσμα του ανακατασκευασμένου σήματος.

Μέρος 3ο - Φασματική Ανάλυση Ημιτονοειδών και Ανίχνευση Απότομων Μεταβάσεων με τον Μετ/σμό Fourier Βραχέος Χρόνου (STFT) και τον Μετ/σμό Wavelets (διακριτοποιημένο DT-CWT)

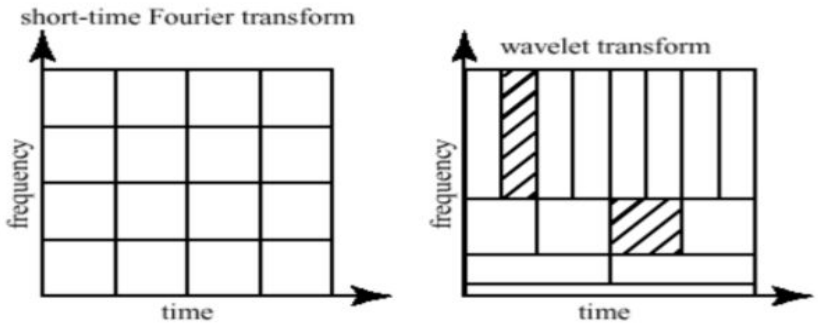
Θεωρητικό Υπόβαθρο

Ο μετασχηματισμός STFT που αναλύθηκε παραπάνω σε συγκεκριμένες περιπτώσεις παρουσιάζει όπως είδαμε ορισμένα μειονεκτήματα. Αναλυτικότερα, όταν έχουμε μικρό μήκος παραθύρου έχουμε αρκετά χειρότερη διακριτική ικανότητα στην συχνότητα, ενώ με μεγάλο μήκος παραθύρου χειρότερη διακριτική ικανότητα στον χρόνο. Στον μετασχηματισμό STFT το μήκος του παραθύρου είναι σταθερό και επιλέγεται εξαρχής.
Ο μετασχηματισμός Wavelet λειτουργεί όμοια με τον STFT με την έννοια ότι παράγουν ένα δισδιάστατο φάσμα για ανάλυση στο πεδίο του χρόνου και της συχνότητας. Όμως παρουσιάζουν ορισμένες σημαντικές διαφορές. Η κύρια διαφορά τους είναι ότι ο μετασχηματισμός wavelet χρησιμοποιεί παράθυρο μεταβλητού μεγέθους. Η μητρική συνάρτηση ψ μπορεί να μεγεθυνθεί η να σμικρυνθεί κατά έναν παράγοντα s (s>1 ή s<1), αν η περιοχή έχει υψηλή ή χαμηλή συχνότητα αντίστοιχα.
Διαπιστώνεται ότι σε υψηλές συχνότητες στο μετ/σμό wavelet παρατηρείται καλή ανάλυση στο χρονικό πεδίο και κακή ανάλυση στο συχνοτικό πεδίο, ενώ σε χαμηλές συχνότητες ισχύει το αντίστροφο.
Για τα σήματα συνεχούς χρόνου ο Μετ/σμός Wavelet ορίζεται ως εξής:

$$CWT(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{\infty}^{\infty} x(t) \cdot \psi^{*}\left(\frac{t-\tau}{s}\right) dt$$

όπου ψ(t): η βασική μητρική συνάρτηση

Στα διακριτά σήματα εφαρμόζεται με την διακριτοποιημένη του μορφή DT-CWT και στην Matlab υλοποιείται με την συνάρτηση cwtft.
Εδώ φαίνεται μια σύγκριση των δύο μετασχηματισμών (STFT vs Wavelets):



3.1

Έχουμε το σήμα:

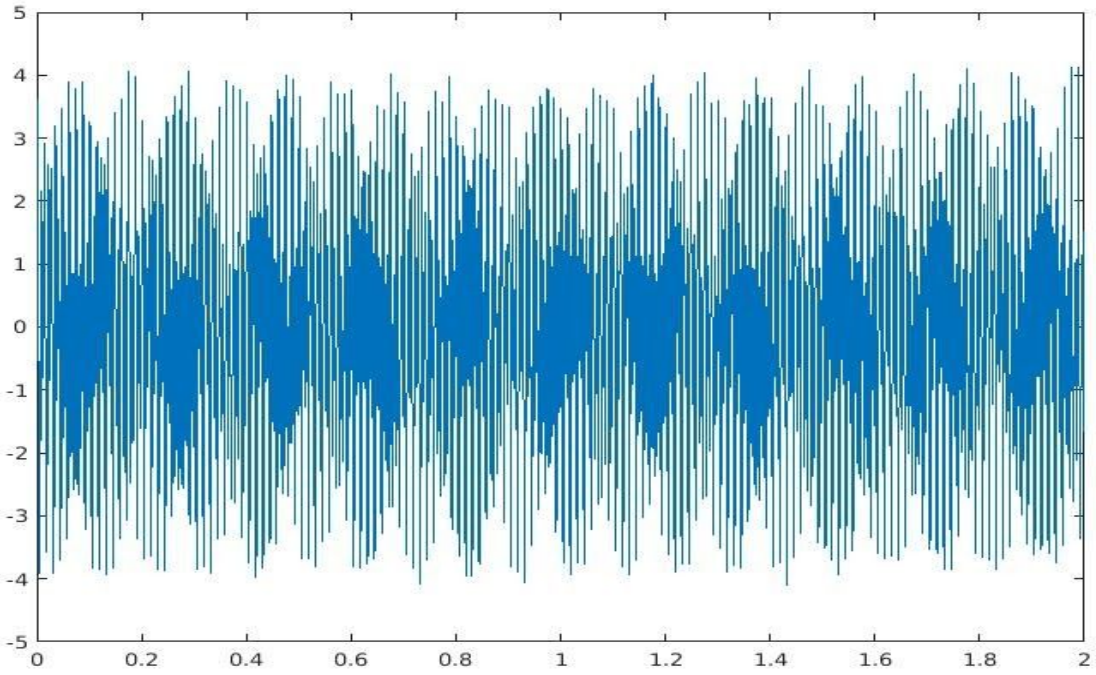
$$x(t) = 1.5 * cos(2\pi 80t) + 2.5 * sin(2\pi 150t) + 0.15 * u(t)$$

Όπου u(t): λευκός Gaussian θόρυβος μηδενικής μέσης τιμής.

Δειγματοληπτούμε το παραπάνω σήμα με συχνότητα Fs=1000 Hz

α) Υπολογίζουμε το σήμα x[n] με βάση την δειγματολειπτούμενη συχνότητα ενώ ο θόρυβος που προστίθεται δημιουργείται με την βοήθεια της εντολής *u=randn([1,length(t)]); .*

Η γραφική παράσταση του σήματος x[n] που προκύπτει σε σχέση με τον χρόνο έχει ως εξής:

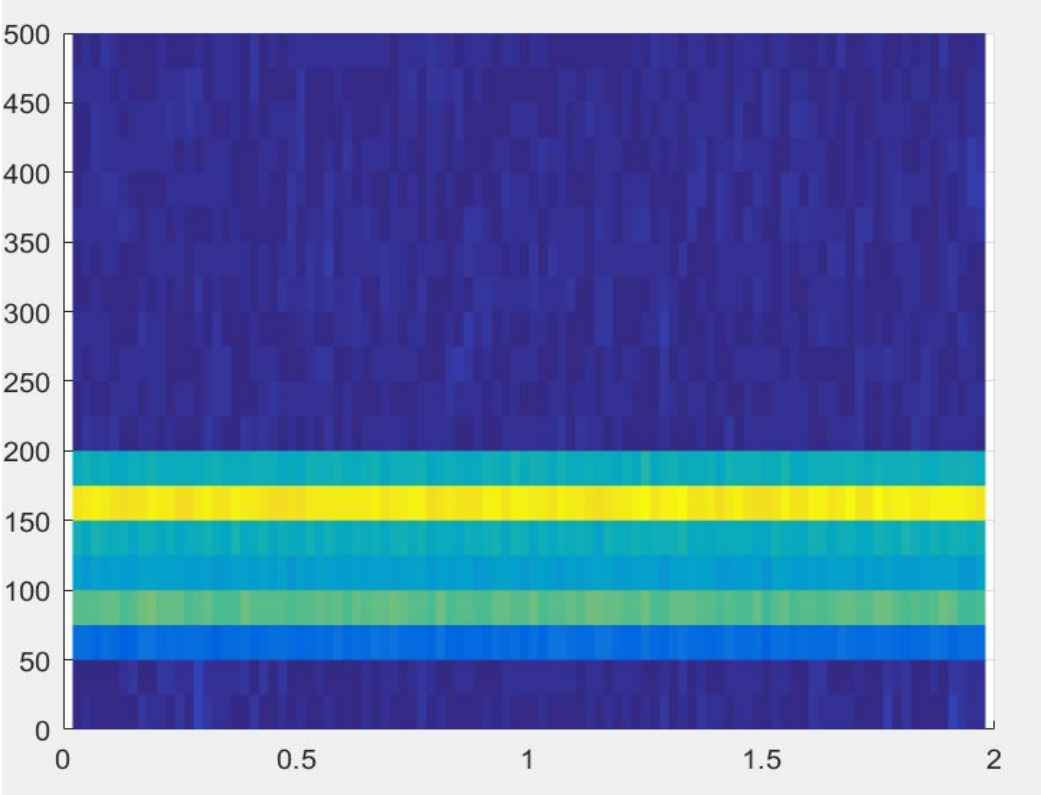


β) Για τον υπολογισμό του STFT(Short Time Fourier Transform), χρησιμοποιήσαμε τη ρουτίνα **spectogram**, η οποία δέχεται ώς ορίσματα το σήμα, το μήκος παραθύρου, την επικάλυψη, το μήκος του DTFT και τη συχνότητα δειγματοληψίας και επιστρέφει το μετασχηματισμένο κατά

STFT σήμα, τον πίνακα κυκλικών συχνοτήτων, καθώς και τον πίνακα χρονικών στιγμών στις οποίες έχει υπολογιστέι το spectogram.

[s,f,t] = spectrogram(x,window,noverlap,nfft,fs)

Στη συνέχεια,με χρήση της ρουτίνας **surf** αναπαριστούμε το πλάτος |STFT(τ,f)| σε συνάρτηση των μεγεθών του χρόνου και της συχνότητας που έχουν επιστραφεί από την συνάρτηση spectrogram.



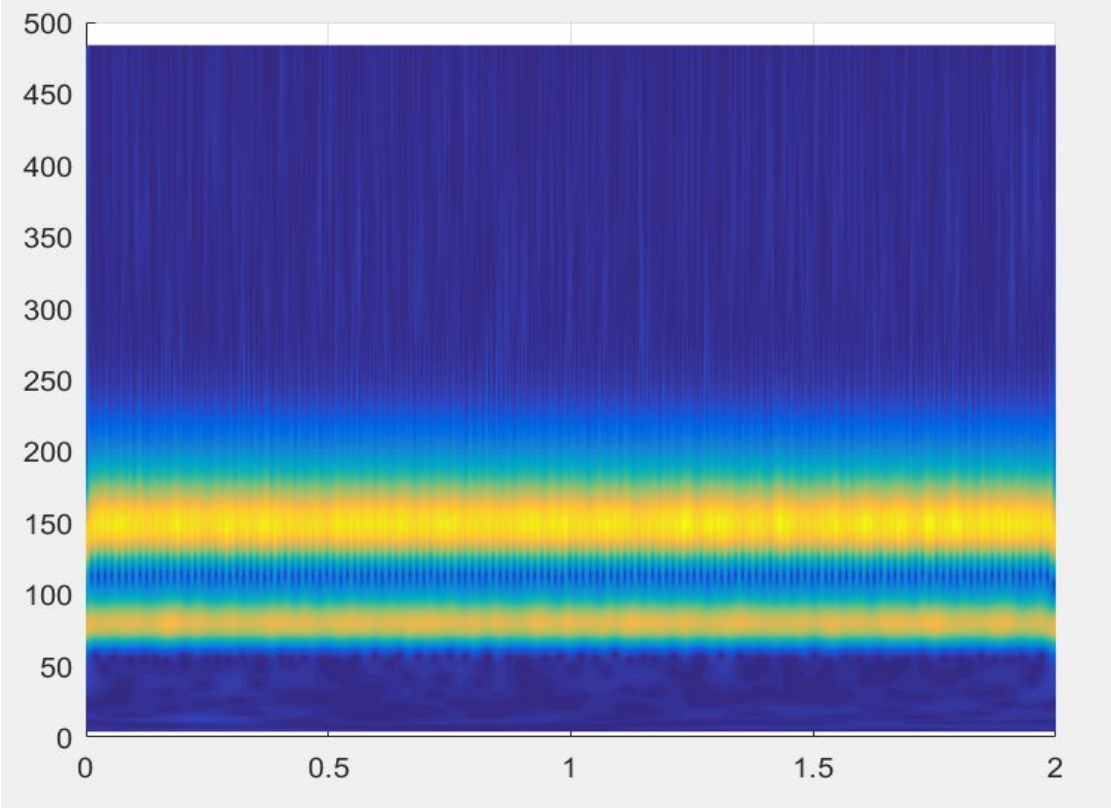
γ) Υπολογίζουμε στη συνέχεια τον DT-CWT μετασχηματισμό με την βοήθεια της cwtft συνάρτησης, επιλέγοντας Morlet Wavelet. Στην Matlab με την κωδική λέξη ‘morl’ επιλέγεται αυτόματα το analytic Morlet Wavelet για το οποίο στο πεδίο Fourier ισχύει:

$$\hat{\Psi}(s\omega) = \pi^{-1/4} e^{-(s\omega-\omega_0)^2/2} U(s\omega)$$.

Ακόμη, για τον προσδιορισμό των κλιμάκων s και συχνοτήτων f χρησιμοποιείται η συνάρτηση *wavescales* που μας δώθηκε, η οποία δέχεται σαν ορίσματα τον τύπο του wavelet (μόνο το Morlet είναι αποδεκτό) και την συχνότητα δειγματοληψίας.

Επομένως, υπολογίζουμε τον επιθυμητό μετασχηματισμό μέσω της εντολής:

cwt = cwtft({X,1/fs}, 'scales',s,'wavelet','morl');
και ύστερα, αναπαριστώντας το πλάτος του σε συνάρτηση των κατάλληλων τιμών χρόνου και συχνότητας έχουμε το εξής:

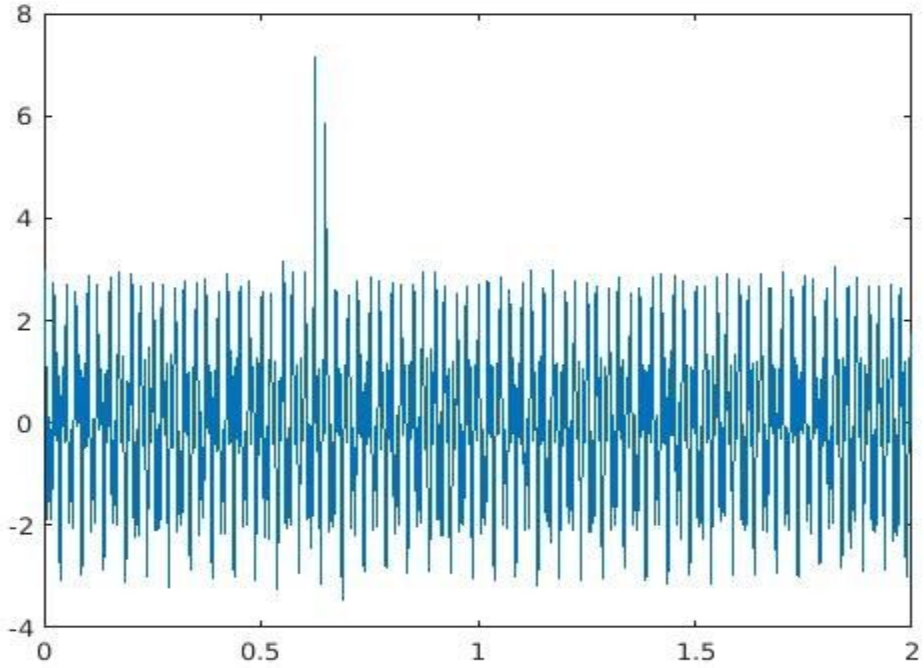


δ) Όπως ήταν αναμενόμενο λόγω του ημιτόνου και του συνημιτόνου, παρατηρούμε ότι στις περιοχές συχνότητας γύρω από τα 80 και 150 Hz το χρώμα είναι έντονο κίτρινο. Το γεγονός αυτό υποδηλώνει ότι η ενέργεια/πλάτος των περιοχών συχνότητας αυτών είναι μεγαλύτερη. Συνεπώς μπορέσαμε να διακρίνουμε με ακρίβεια τις βασικές συχνότητες του σήματος.

Ακόμη, παρατηρούμε, όπως άλλωστε ήταν αναμενόμενο, πως στον μετ/σμό wavelet για χαμηλές συχνότητες έχουμε κακή ανάλυση στο χρονικό πεδίο και καλή ανάλυση στο συχνοτικό πεδίο, ενώ όσο μεγαλώνει η συχνότητα γίνεται το αντίθετο. Συγκεκριμένα, η ανάλυση στο χρονικό πεδίο καταλήγει αρκετά ευδιάκριτη.

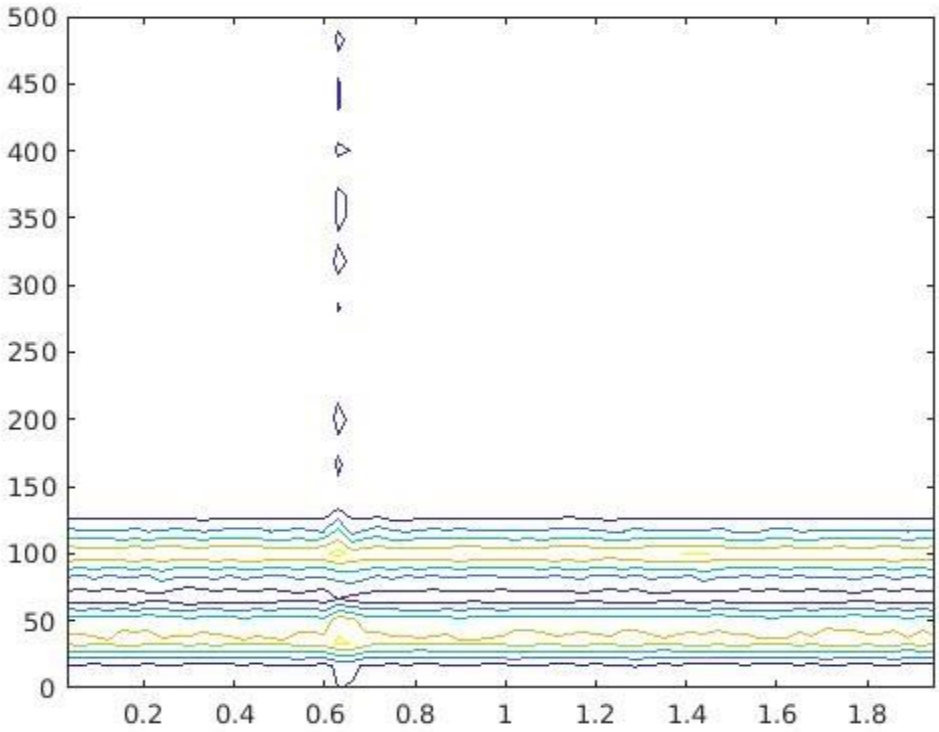
3.2

α) Ομοίως με το παραπάνω ερώτημα υπολογίζουμε το σήμα x[n] συμπεριλαμβανομένου και τον απότομων μεταβολών και έχουμε την παρακάτω γραφική παράσταση:
Γίνονται φανερές οι απότομες μεταβολές στο σήμα μας.

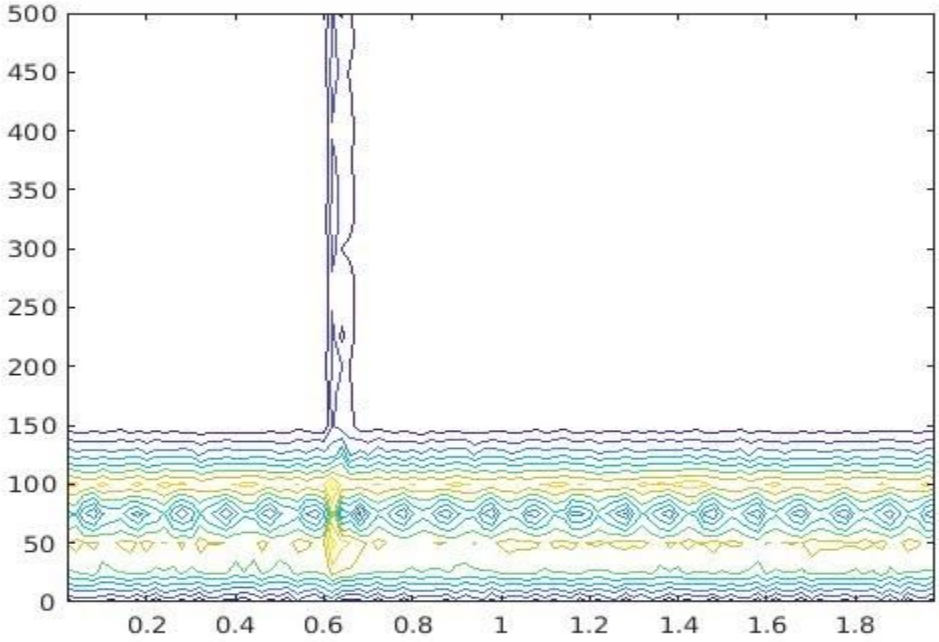


β) Εδώ, χρησιμοποιώντας της συνάρτηση spectrogram και ακριβώς όπως στο ερώτημα 3.1 υπολογίζουμε τον STFT και αναπαριστούμε το πλάτος του συναρτήσει των κατάλληλων τιμών χρόνου και συχνότητας για τρία διαφορετικά μήκη παραθύρου και επικάλυψη πάντα 50%.Για τον σκοπό αυτό χρησιμοποιούμε τη συνάρτηση **contour.**
Έχουμε λοιπόν:

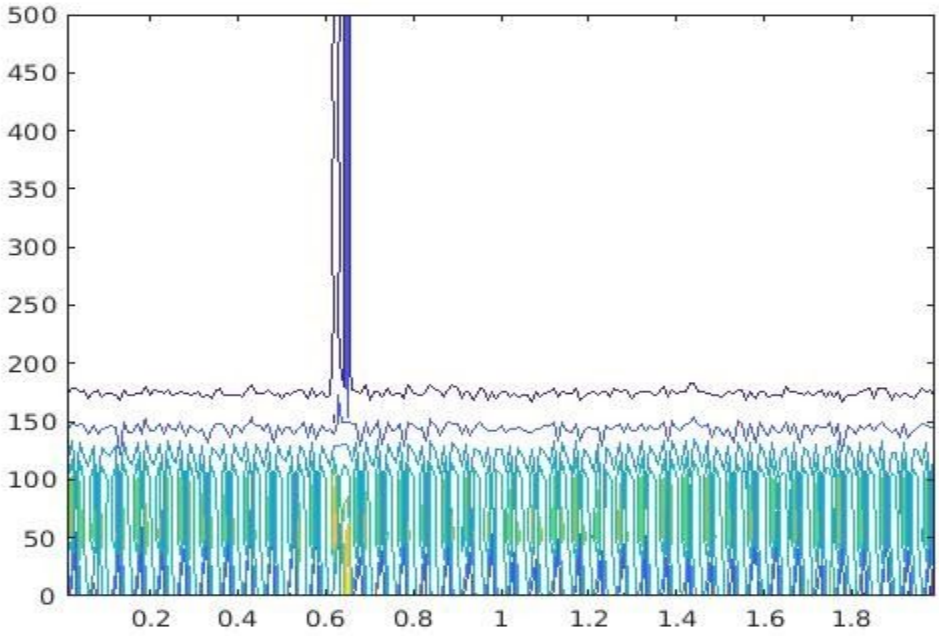
- Για μήκος παραθύρου 0.06sec



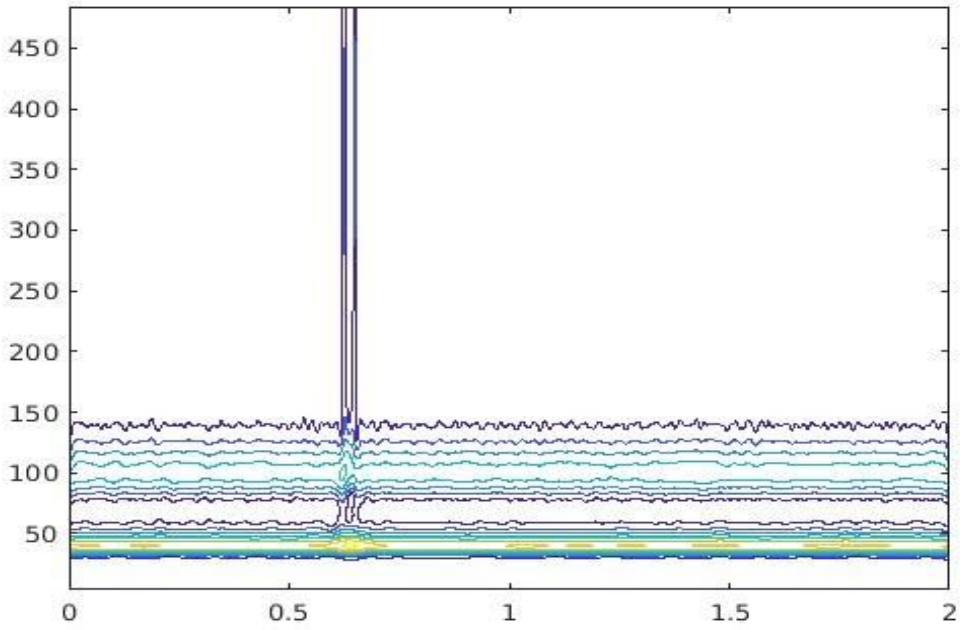
- Για μήκος παραθύρου 0.04sec



- Για μήκος παραθύρου 0.02sec



γ) Στη συνέχεια, πάλι όμοια με το ερώτημα 3.1 υπολογίζουμε τον DT-CWT μετασχηματισμό του σήματος και αναπαριστώντας το πλάτος του σε συνάρτηση με τις κατάλληλες τιμές χρόνου και συχνότητας έχουμε:



δ) Όσον αφορά τον STFT, γνωρίζουμε πως όσο μικρότερο το μέγεθος του παραθύρου τόσο μικρότερη η ανάλυση στο πεδίο των συχνοτήτων αλλά τόσο καλύτερη η ανάλυση στο πεδίο του χρόνου.
Όσο μικρότερο το μήκος του παραθύρου, οι στιγμιαίες απότομες μεταβολές γίνονται όλο και πιο ξεκάθαρες, φτάνοντας στα 0.02sec όπου εύκολα διακρίνονται, όμως οι βασικές συχνότητες του σήματος γίνονται όλο και πιο δυσδιάκριτες.

Στον DT-CWT παρατηρούμε ότι μπορούμε να πετύχουμε ταυτόχρονα καλή διακριτική ικανότητα τόσο στο πεδίο του χρόνου όσο και στο πεδίο της συχνότητας.Αυτό επιτυγχάνεται δίνοντας καλή χρονική ανάλυση και κακή συχνοτική ανάλυση στις υψηλές συχνότητες και κακή χρονική ανάλυση και καλή συχνοτική ανάλυση στις χαμηλές συχνότητες,