



Αναφορά 2^{ης} Εργαστηριακής Άσκησης στις Ψηφιακές Επικοινωνίες Ι
«Σχεδιασμός/υλοποίηση ψηφιακών φίλτρων FIR με το MATLAB»
Μπουφίδης Ιωάννης 03120162

Μέρος 1 : Εισαγωγή

Ερώτηση 1:

Έπειτα απ'την εκτέλεση των παρακάτω εντολών στο command window

```
>> X=[-2:2]
>> fftshift(X)
>> ifftshift(X)
>> Y = fftshift(fftshift(X));
>> Z = ifftshift(fftshift(X));
>> isequal(X,Y)
>> isequal(X,Z)
```

προκύπτει ότι το διάνυσμα Z ισούται με το X, κάτι το οποίο είναι αναμενόμενο, μιας και οι συναρτήσεις `fftshift` και `ifftshift` είναι μεταξύ τους αντίστροφες (σχέση f και f^{-1})

Ερώτηση 2:

Επαναλαμβάνοντας με $X=[-1:2]$, παρατηρούμε ότι η `fftshift` και η `ifftshift` έχουν το ίδιο αποτέλεσμα. Επομένως, $Y=Z$. Ωστόσο, ισχύει και ότι $X=Y=Z$.

Ερώτηση 3:

Σε περίπτωση που στο παρακάτω παράδειγμα θέλουμε το φάσμα του βαθυπερατού σήματος X να οριστεί απευθείας όπως το αναμένει η `ifft`, πρέπει να γίνουν κάποιες τροποποιήσεις

Πρωτότυπο:

```
>> close all; clear all; clc;
>> Xb=[0 0 1 1 1 1 1 0 0] % φάσμα βαθυπερατού σήματος με άρτια συμμετρία
```

```
>> figure; subplot(2,1,1); plot([-4:4],Xb); ylabel('Xb');

>> X=ifftshift(Xb) % το φάσμα με τις αρνητικές συνιστώσες στο άνω μέρος

>> x=ifft(X) % IFFT

>> xb=fftshift(x) % πραγματικό σήμα με άρτια συμμετρία όπως αναμένεται

>> subplot(2,1,2); plot([-4:4],xb); ylabel('xb');
```

Τροποποιημένο:

```
>> close all; clear all; clc;

>> X=[1 1 1 0 0 0 1 1]

>> x=ifft(X)

>> Xb=fftshift(X)

>> figure; subplot(2,1,1); plot([-4:4],Xb); ylabel('Xb');

>> xb=fftshift(x)

>> subplot(2,1,2); plot([-4:4],xb); ylabel('xb');
```

Μέρος 2 : Σχεδιασμός φίλτρων

Ο κώδικας MATLAB του 2^{ου} μέρους της άσκησης είναι ο εξής:

```
%% clearing and sima.mat
clear all; close all;
% Το αρχείο "sima.mat" περιέχει το σήμα s και τη συχνότητα
% δειγματοληψίας Fs. Το φάσμα του σήματος εκτείνεται σχεδόν σε όλη την
% περιοχή συχνοτήτων μέχρι 4 KHz. Πάνω από 1 KHz, όμως, είναι θόρυβος
% και πρέπει να φιλτραριστεί.
load sima;
figure; pwelch(s,[],[],[],Fs);
pause

%% rectangular LPF
% Ορίζεται η ιδανική βαθυπερατή συνάρτηση H, με συχνότ. αποκοπ. Fs/8
H=[ones(1,Fs/8) zeros(1,Fs-Fs/4) ones(1,Fs/8)];
% Υπολογίζεται η κρουστική απόκριση με αντίστροφο μετασχ. Fourier
% Εναλλακτικά, μπορεί να χρησιμοποιηθεί η αναλυτική σχέση Sa(x)
h=ifft(H,'symmetric');
middle=length(h)/2;
h=fftshift(h);
h32=h(middle+1-16:middle+17);
h64=h(middle+1-32:middle+33);
h128=h(middle+1-64:middle+65);
% figure; stem([0:length(h64)-1],h64); grid;
% figure; freqz(h64,1); % σχεδιάζουμε την απόκριση συχνότητας της h64
wvtool(h32,h64,h128); % αποκρίσεις συχνότητας των περικυμμένων h
pause
```

```

%% hamming windows
% Οι πλευρικοί λοβοί είναι υψηλοί!
% Πολλαπλασιάζουμε την περικομμένη κρουστική απόκριση με κατάλληλο
% παράθυρο. Χρησιμοποιούμε την h64 και παράθυρο hamming
wh=hamming(length(h64));
figure; plot(0:64,wh,'b'); grid;
pause
h_hamming=h64.*wh';
% figure; stem([0:length(h64)-1],h_hamming); grid;
% figure; freqz(h_hamming,1);
wvtool(h64,h_hamming);
pause

%% filtering
% Φιλτράρουμε το σήμα μας με καθένα από τα δύο φίλτρα
y_rect=conv(s,h64);
figure; pwelch(y_rect,[],[],[],Fs); pause
y_hamm=conv(s,h_hamming);
figure; pwelch(y_hamm,[],[],[],Fs); pause

%% LPF Parks-MacClellan
hpm=firpm(64, [0 0.10 0.15 0.5]*2, [1 1 0 0]);
% figure; freqz(hpm,1);
s_pm=conv(s,hpm);
figure; pwelch(s_pm,[],[],[],Fs);
pause
sound(20*s); pause % ακούμε το αρχικό σήμα, s
sound(20*s_pm); % ακούμε το φιλτραρισμένο σήμα, s_lp

%% Πειραματισθείτε #2
h160=h(middle+1-80:middle+81);
wh160=hamming(length(h160));
h160_hamming=h160.*wh160';
wvtool(h160,h160_hamming);
pause

%% Πειραματισθείτε #3
hpm_3=firpm(160,[0 0.1 0.15 0.5]*2, [1 1 0 0]);
wvtool(hpm,hpm_3);
pause

%% Πειραματισθείτε #4
hpm_4=firpm(160,[0 0.11 0.12 0.5]*2, [1 1 0 0]);
wvtool(hpm_3,hpm_4);
pause

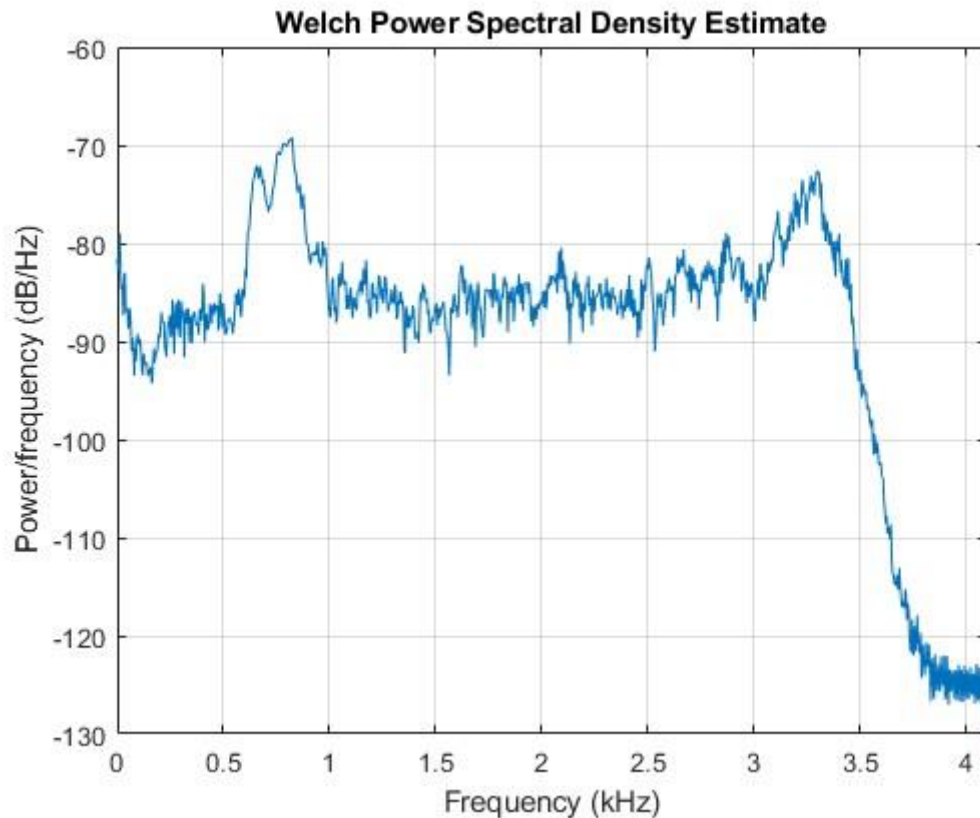
%% Πειραματισθείτε #5
Ts=1/Fs;
T=0.1; %διάρκεια σήματος
t=0:Ts:T-Ts; %χρονικές στιγμές δειγματοληψίας
x=sin(2*pi*700*t)+...
    sin(2*pi*900*t)+...
    sin(2*pi*1400*t)+...
    sin(2*pi*2500*t); %σήμα
figure; pwelch(x,[],[],[],Fs); pause
x_pm3=conv(x,hpm_3);
x_pm4=conv(x,hpm_4);
figure; pwelch(x_pm3,[],[],[],Fs); pause
figure; pwelch(x_pm4,[],[],[],Fs); pause

```

Αναλυτικότερα, κατά την εκτέλεση του section:

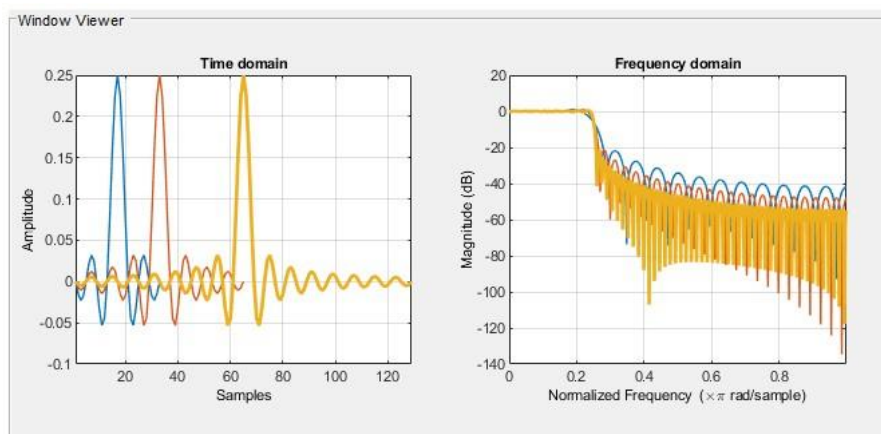
- `clearing and sima.mat`

Εμφανίζεται με την χρήση της `pwelch` το παρακάτω διάγραμμα, το οποίο αποτελεί το φάσμα του δεδομένου σήματος `s` (εκτείνεται σε όλη την περιοχή μέχρι 4kHz, αλλά για πάνω από 1kHz είναι θόρυβος που θέλουμε να αποκόψουμε)



- `rectangular LPF`

Ορίζεται μια ιδανική βαθυπερατή συνάρτηση με συχνότητας αποκοπής το 1kHz, ενώ στην συνέχεια η κρουστική της απόκριση περικόπτεται σε 3 διαφορετικά μήκη (32, 64, 128), τα οποία εμφανίζονται στο παρακάτω σχήμα τόσο στο πεδίο του χρόνου, όσο και στο πεδίο της συχνότητας.



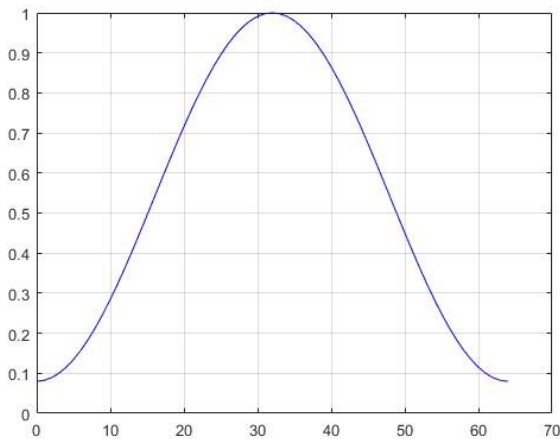
Παρατηρούμε ότι με την αύξηση του μήκους της κρουστικής απόκρισης, περιορίζεται η επίδραση των πλευρικών λοβών κατά το φιλτράρισμα, καθώς μειώνεται η στάθμη τους στο frequency domain. Επομένως, προκύπτει το συμπέρασμα ότι όσο πιο μεγάλο είναι το μήκος της κρουστικής απόκρισης, τόσο πιο αποδοτικό θα είναι το φίλτρο μας.

- **hamming windows**

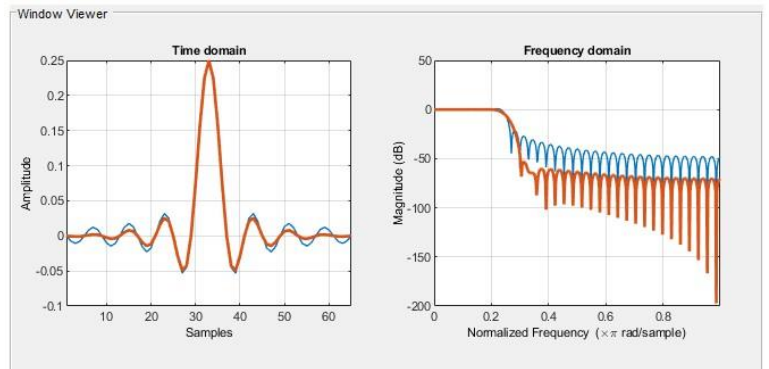
Πολλαπλασιάζουμε την περικομμένη κρουστική απόκριση με κατάλληλο παράθυρο.

Χρησιμοποιούμε την h64 και παράθυρο hamming, το οποίο φαίνεται στο Σχήμα 2.1 .

Στο Σχήμα 2.2 παρουσιάζεται η κρουστική απόκριση μήκους 64+1 **πρωτού** και **αφότου** έχει πολλαπλασιαστεί με το παράθυρο Hamming.



Σχήμα 2.1

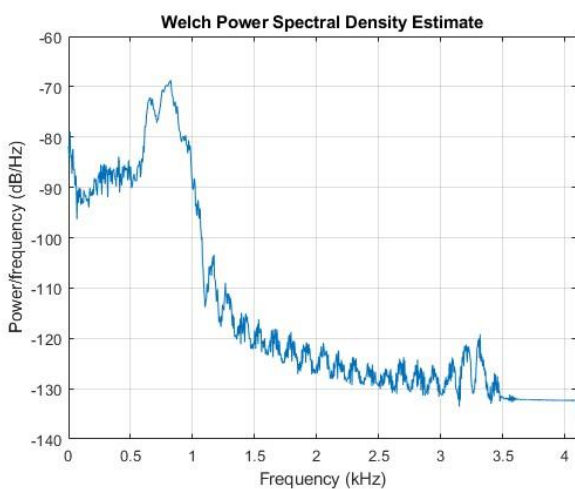


Σχήμα 2.2

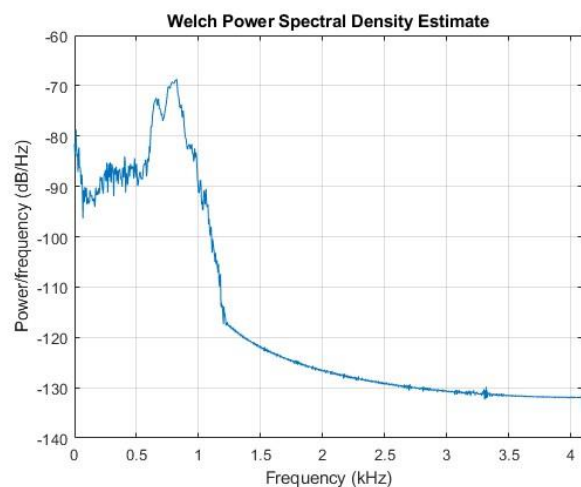
Από το Σχήμα 2.2, μπορούμε να παρατηρήσουμε ότι με τον πολλαπλασιασμό της κρουστικής με το παράθυρο, μειώνεται η στάθμη των πλευρικών λοβών τόσο στο πεδίο της συχνότητας, όσο και του χρόνου, πράγμα που κάνει το φίλτρο πιο αποδοτικό.

- **filtering**

Φιλτράρουμε το σήμα μας με καθένα από τα δύο φίλτρα, δηλαδή με το h64 πριν (Σχήμα 2.3) και αφού (Σχήμα 2.4) το πολλαπλασιάσαμε με το παράθυρο Hamming. Στα παρακάτω σχήματα παρουσιάζεται το φάσμα του φιλτραρισμένου σήματος με την βοήθεια της pwelch.



Σχήμα 2.3

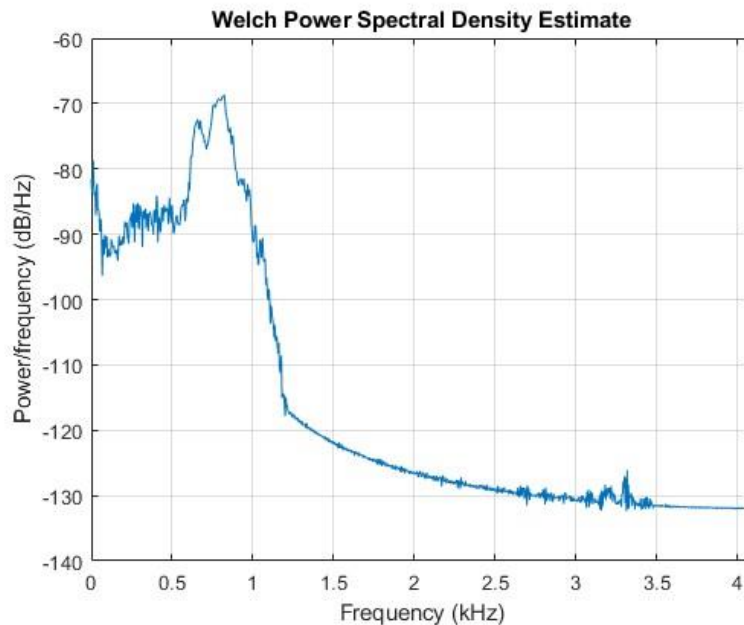


Σχήμα 2.4

Εύκολα βλέπει κανείς ότι ο θόρυβος, που υπάρχει στο Σχήμα 2.3 για τις συχνότητες πέρα από το 1 kHz, εξαφανίζεται σχεδόν τελείως στο Σχήμα 2.4, γεγονός που επιβεβαιώνει την παρατήρηση που έγινε στο προηγούμενο section, δηλαδή ότι το φίλτρο είναι πιο αποδοτικό αν «περασθεί» από κάποιο παράθυρο (εδώ Hamming).

- **LPF Parks-MacClellan**

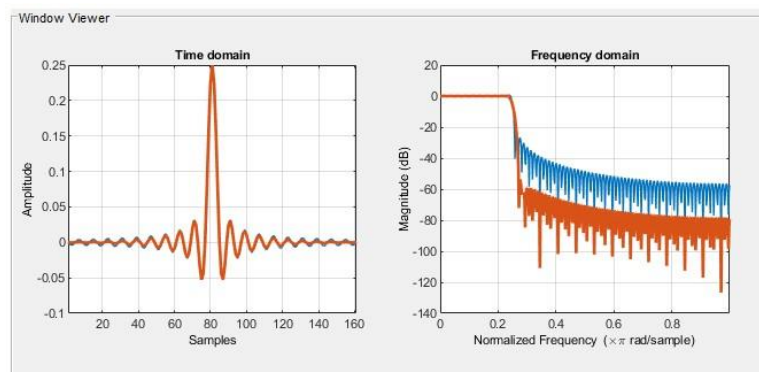
Δημιουργείται με την μέθοδο των ισοϋψών κυματώσεων ένα βαθυπερατό φίλτρο μήκους 64+1 με τα ίδια χαρακτηριστικά, όπως παραπάνω. Στην συνέχεια, το δεδομένο σήμα s «περνάει» μέσα από αυτό το φίλτρο, με το αποτέλεσμα του φιλτραρίσματος να φαίνεται στο παρακάτω σχήμα.



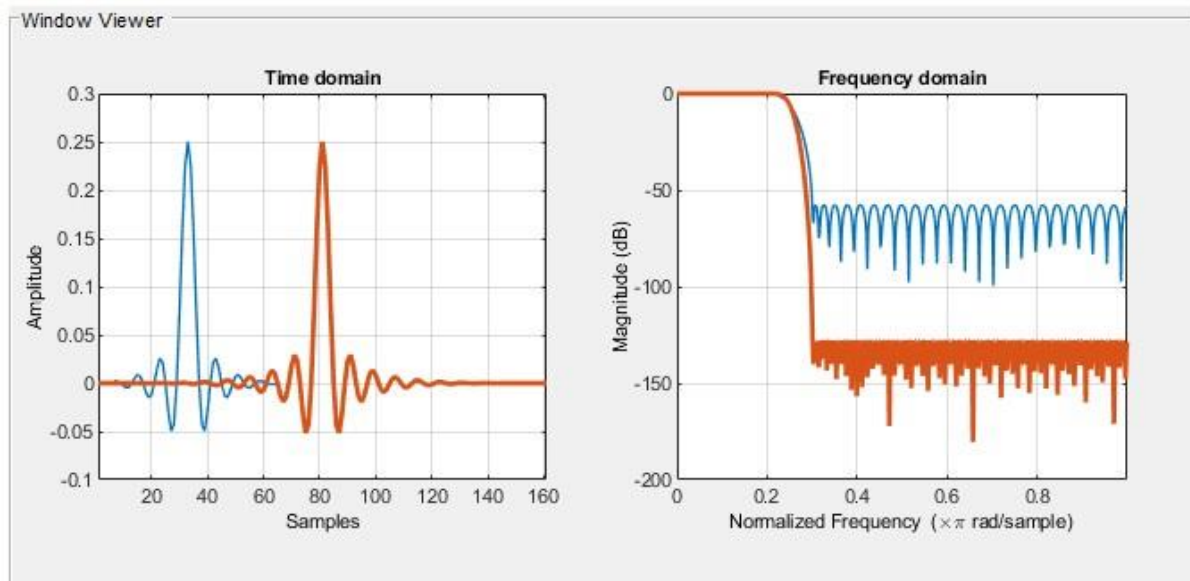
Παρατηρούμε ότι τα επίπεδα θορύβου για τις συχνότητες μετά την συχνότητα αποκοπής είναι εξίσου χαμηλά με αυτά του Σχήματος 2.4

Πειραματισθείτε

1. Η γραμμή 14 του κώδικα μπορεί να τροποποιηθεί ως εξής προκειμένου να έχουμε το ίδιο αποτέλεσμα με τη χρήση μιας εκ των συναρτήσεων `ifftshift` ή `fftshift`:
`h=fftshift(h);`
2. Έπειτα από τροποποίηση του κώδικα ώστε να χρησιμοποιηθεί βαθυπερατό φίλτρο μήκους 160+1 αντί του 64+1 και σχεδιασμό (παρακάτω σχήμα) της απόκρισης συχνότητας του φίλτρου με σχεδιασμό παραθυρού (ορθογωνικού και **Hamming**), παρατηρεί κανείς ότι ανεξάρτητα του μήκους κρουστικής το φίλτρο είναι πάντοτε πιο αποδοτικό (πιο χαμηλοί-περιορισμένοι πλευρικοί λοβοί) όταν πολλαπλασιαστεί με ένα παράθυρο Hamming, σε σχέση με πριν πολλαπλασιαστεί

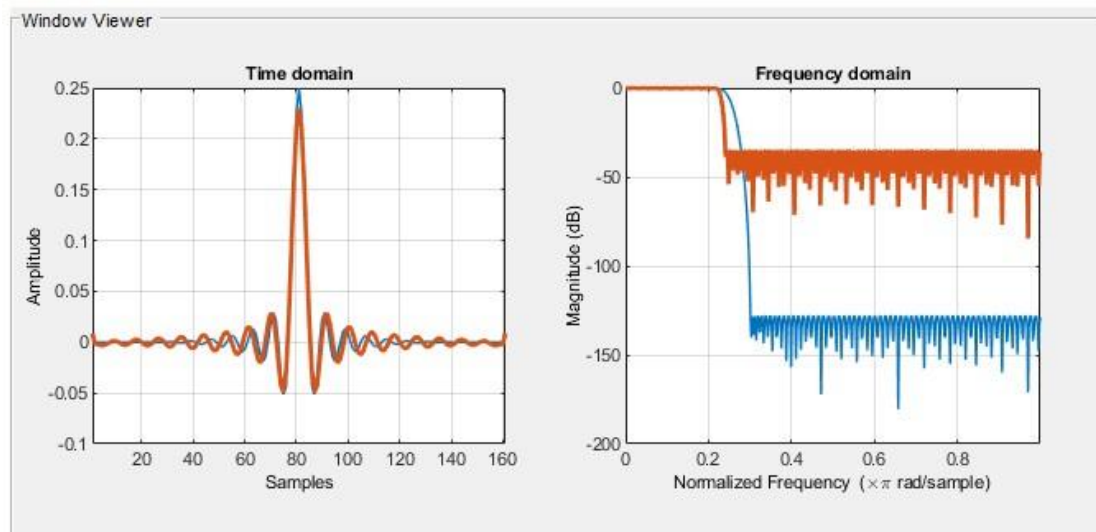


3. Έπειτα από την δημιουργία φίλτρου Parks-McClellan μήκους 160+1, με τις ίδιες οριακές συχνότητες, όπως στο παράδειγμα (0.1, 0.15), σχεδιάζουμε (παρακάτω σχήμα) την απόκριση συχνότητας τόσο αυτού, όσο και του φίλτρου μήκους 64+1

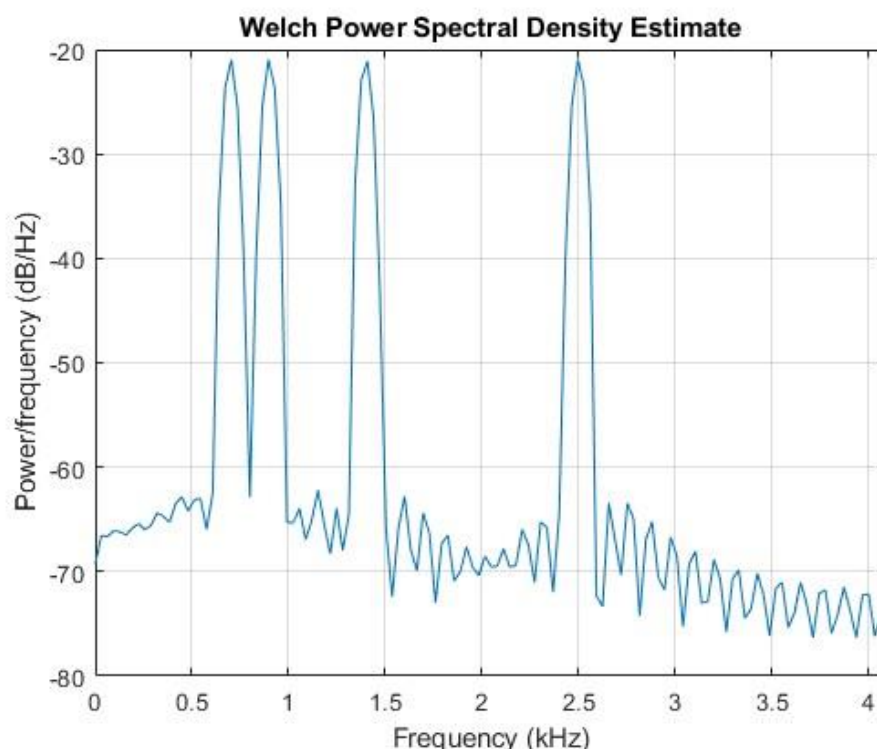


Όπως και στα φίλτρα που δημιουργήσαμε με την μέθοδο των παραθύρων, έτσι και εδώ όσο πιο μεγάλο είναι το μήκος της κρουστικής, τόσο πιο αποδοτικό είναι το φίλτρο καθώς περιορίζεται η επίδραση των πλευρικών λοβών («πέφτει» η στάθμη τους)

4. Αλλάζοντας τις οριακές συχνότητες του φίλτρου του ερωτήματος 3 σε (0.11, 0.12) και συγκρίνοντας τις αποκρίσεις συχνότητας, όπως φαίνονται παρακάτω, των δύο φίλτρων (του νέου και αυτού του ερωτήματος 3), παρατηρεί κανείς το εξής:
Όσο πιο μικρή είναι η διαφορά μεταξύ των οριακών συχνοτήτων, τόσο πιο απότομη είναι η «πλαγιά» (slope) στην απόκριση συχνότητας του φίλτρου. Βέβαια σε αυτή την περίπτωση η στάθμη των πλευρικών λοβών φαίνεται να μεγαλώνει, οπότε τελικά το φίλτρο με την μικρότερη διαφορά μεταξύ των οριακών συχνοτήτων μπορεί να είναι λιγότερο αποδοτικό.

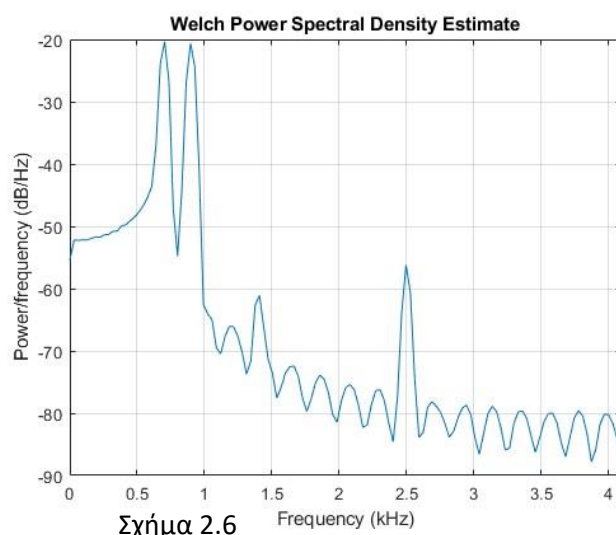
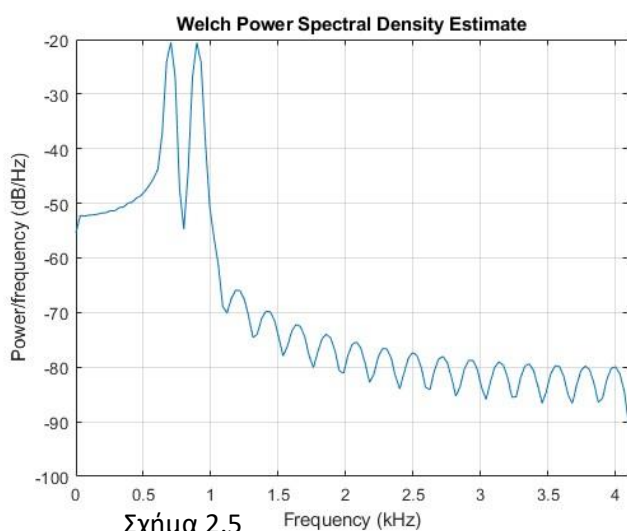


5. Αντικαθιστούμε το σήμα s με άθροισμα τεσσάρων ημιτονικών συναρτήσεων μοναδιαίου πλάτους, συχνότητας 700, 900, 1400 και 2500 Hz και διάρκειας 1.0 sec, του οποίου η φασματική πυκνότητα (μέσω pwelch) φαίνεται στο παρακάτω σχήμα.



Το φάσμα του σήματος αποτελείται από peaks («κορυφές») στις συχνότητες των επιμέρους τεσσάρων ημιτονικών συναρτήσεων (όπως αναμενόταν) και θόρυβο.

Αν φιλτράρουμε αυτό το σήμα με τα δύο φίλτρα Parks-McClellan των ερωτημάτων 3 και 4, τα σήματα που θα λάβουμε είναι αυτά του Σχήματος 2.5 και 2.6 αντίστοιχα



Πράγματι, η αρχική υπόθεση ότι τελικά το αρχικό φίλτρο του ερωτήματος 3 μπορεί να είναι πιο αποδοτικό επιβεβαιώνεται, καθώς στο σχήμα 2.6 υπάρχει αισθητά περισσότερος θόρυβος μετά το 1 kHz. Επομένως, μπορούμε να βγάλουμε το συμπέρασμα ότι σε ένα φίλτρο είναι προτιμότερο να έχουμε χαμηλότερη στάθμη πλευρικών λοβών, παρά πιο απότομο slope.

Μέρος 3 : Εφαρμογή Α

Ο κώδικας MATLAB του 3^{ου} μέρους της άσκησης είναι ο εξής:

```
%% clearing and sima.mat

clear all; close all;
% Το αρχείο "sima.mat" περιέχει το σήμα s και τη συχνότητα
% δειγματοληψίας Fs. Το φάσμα του σήματος εκτείνεται σχεδόν σε όλη την
% περιοχή συχνοτήτων μέχρι 4 KHz. Πάνω από 1 KHz, όμως, είναι θόρυβος
% και πρέπει να φιλτραριστεί.
load sima;

%% rectangular BPF
% Ορίζεται η ιδανική ζωνοπερατή συνάρτηση H, με ζώνη διέλευσης(0.7kHz,
% 1.5kHz)
H=[zeros(1,700) ones(1,800) zeros(1,5192) ones(1,800) zeros(1,700)];
% Υπολογίζεται η κρουστική απόκριση με αντίστροφο μετασχ. Fourier
% Εναλλακτικά, μπορεί να χρησιμοποιηθεί η αναλυτική σχέση Sa(x)
h=ifft(H,'symmetric');
middle=length(h)/2;
h=fftshift(h);
h64=h(middle+1-32:middle+33);
% figure; stem([0:length(h64)-1],h64); grid;
% figure; freqz(h64,1); % σχεδιάζουμε την απόκριση συχνότητας της h64
wvtool(h64); % αποκρίσεις συχνότητας των περικομμένων h
pause

%% hamming window
% Οι πλευρικοί λοβοί είναι υψηλοί!
% Πολλαπλασιάζουμε την περικομμένη κρουστική απόκριση με κατάλληλο
% παράθυρο. Χρησιμοποιούμε την h64 και παράθυρο hamming
wh=hamming(length(h64));
figure; plot(0:64,wh,'b');
title('Hamming Window ΜΠΟΥΦΙΔΗΣ ΙΩΑΝΝΗΣ'); grid;
pause
h_hamming=h64.*wh';
% figure; stem([0:length(h64)-1],h_hamming); grid;
% figure; freqz(h_hamming,1);
wvtool(h_hamming);
pause

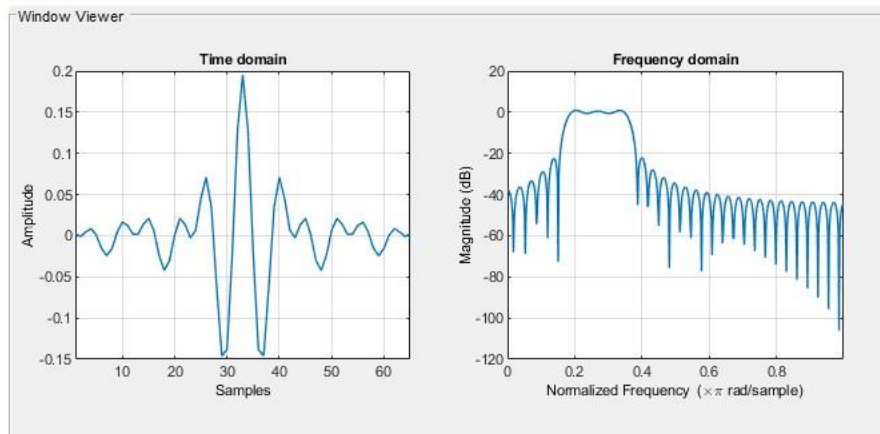
%% filtering
% Φιλτράρουμε το σήμα μας με το φίλτρο
s_hamm=conv(s,h_hamming);
figure; pwelch(s_hamm,[],[],[],Fs); pause

%% BPF Parks-MacClellan
hpm=firpm(64, [0 0.13 0.17 0.35 0.39 1], [0 0 1 1 0 0]);
% figure; freqz(hpm,1);
s_pm=conv(s,hpm);
figure; pwelch(s_pm,[],[],[],Fs);
pause
sound(20*s); pause % ακούμε το αρχικό σήμα, s
sound(20*s_pm); % ακούμε το φιλτραρισμένο σήμα, s_1
```

Αναλυτικότερα, κατά την εκτέλεση του section:

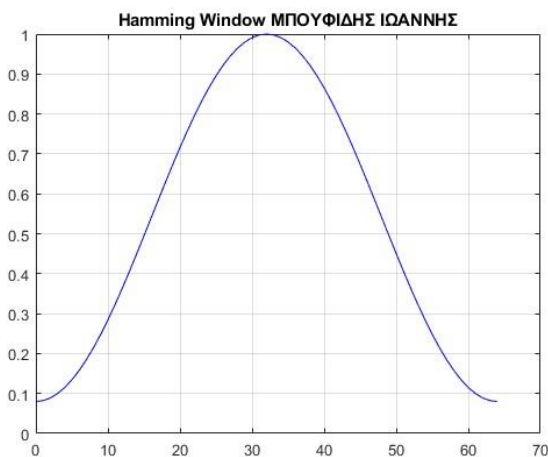
- **rectangular BPF**

Ορίζεται μια ιδανική ζωνοπερατή συνάρτηση με ζώνη διέλευσης (0.7kHz, 1.5kHz), ενώ στην συνέχεια η κρουστική της απόκριση περικόπτεται σε μήκος 64, το οποίο εμφανίζεται στο παρακάτω σχήμα τόσο στο time, όσο και στο frequency domain με την βοήθεια του wntool.

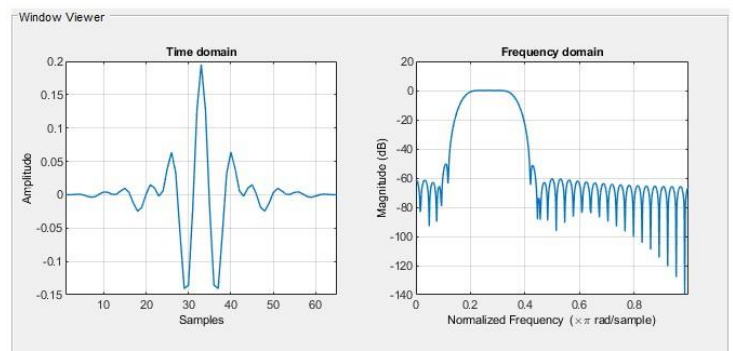


- **hamming window**

Πολλαπλασιάζουμε την περικομμένη κρουστική απόκριση με κατάλληλο παράθυρο. Χρησιμοποιούμε την h64 και παράθυρο hamming, το οποίο φαίνεται στο Σχήμα 3.1. Στο Σχήμα 3.2 παρουσιάζεται η κρουστική απόκριση μήκους 64+1 αφού έχει πολλαπλασιαστεί με το παράθυρο Hamming.



Σχήμα 3.1

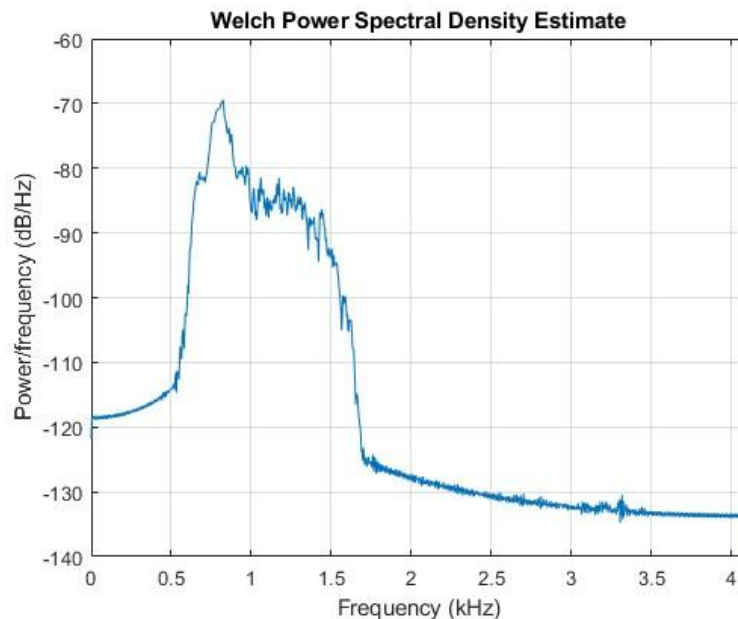


Σχήμα 3.2

Αν συγκρίνουμε το Σχήμα 3.2 και το σχήμα που προέκυψε από την εκτέλεση του προηγούμενου section, μπορούμε να παρατηρήσουμε ότι με τον πολλαπλασιασμό της κρουστικής με το παράθυρο, μειώνεται σημαντικά η στάθμη των πλευρικών λοβών τόσο στο πεδίο της συχνότητας, όσο και του χρόνου, πράγμα που κάνει το φίλτρο πιο αποδοτικό.

- **filtering**

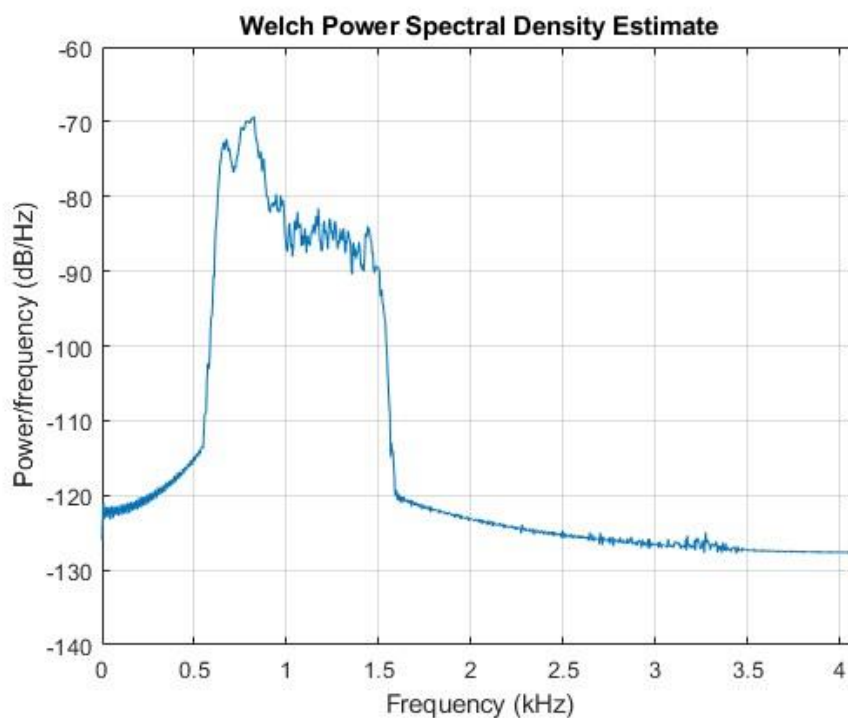
Φιλτράρουμε το δεδομένο σήμα μας s με το φίλτρο του παραθύρου Hamming. Στο παρακάτω σχήμα παρουσιάζεται το φάσμα του φιλτραρισμένου σήματος με την βοήθεια της `pwelch`.



Το σήμα έχει περάσει μόνο ανάμεσα στις συχνότητες που ανήκουν στην ζώνη διέλευσης (0.7kHz, 1.5kHz), όπως και αναμενόταν.

- **BPF Parks-MacClellan**

Δημιουργείται με την μέθοδο των ισοϋψών κυματώσεων ένα ζωνοπερατό φίλτρο μήκους 64+1 με τα ίδια χαρακτηριστικά, όπως παραπάνω. Στην συνέχεια, το δεδομένο σήμα s «περνάει» μέσα από αυτό το φίλτρο, με το αποτέλεσμα του φιλτραρίσματος να φαίνεται στο παρακάτω σχήμα.



Παρατηρούμε ότι τα επίπεδα θορύβου για τις συχνότητες εκτός της ζώνης διέλευσης είναι εξίσου χαμηλά με αυτά του σήματος που προέκυψε όταν περάσαμε (προηγούμενο section) το σήμα s από το φίλτρο που δημιουργήθηκε με την μέθοδο των παραθύρων (εδώ Hamming)