



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Μάθημα: Ψηφιακές Επικοινωνίες Ι

Ονοματεπώνυμο: Ειρήνη Δόντη

Α.Μ.: 03119839

2^η Σειρά Ασκήσεων

Αθήνα 2022

Περιεχόμενα

Μέρος 2: Σχεδιασμός Φίλτρων	1
Μέρος 3: Εφαρμογή Α	8

Μέρος 2: Σχεδιασμός Φίλτρων

Πειραματιστείτε:

Παρακάτω, παρατίθεται ο κώδικας που περιέχεται στο αρχείο lab2_1_19839.m:

```
clear all; close all;
% Eirini Donti - 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);
% 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); %h2=[h(middle+1:end) h(1:middle)]; isequal(h,h2) %1
h32=h(middle+1-16:middle+17);
h196=h(middle+1-98:middle+99); h64=h(middle+1-32:middle+33); %2
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,h196,h128); %wvtool(h32,h64,h128); % αποκρίσεις συχνότητας των περικομμένων h %2
% Οι πλευρικοί λοβοί είναι υψηλοί!
% Πολλαπλασιάζουμε την περικομμένη κρουστική απόκριση με κατάλληλο
% παράθυρο. Χρησιμοποιούμε την h64 και παράθυρα hamming και kaiser
wh=hamming(length(h196)); %wh=hamming(length(h64));
wk=kaiser(length(h64),5);
figure; plot(0:64,wk,'r',0:196,wh,'b'); grid; %figure; plot(0:64,wk,'r',0:64,wh,'b'); grid;
h_hamming=h196.*wh'; %h_hamming=h64.*wh';
% figure; stem([0:length(h64)-1],h_hamming); grid;
% figure; freqz(h_hamming,1);
h_kaiser=h64.*wk';
wvtool(h196,h_hamming); %wvtool(h64,h_hamming,h_kaiser); %2 rect & Hamming frequency response
% Φιλτράρουμε το σήμα μας με καθένα από τα τρία φίλτρα
y_rect=conv(s,h196);%y_rect=conv(s,h64);
figure; pwelch(y_rect,[],[],Fs);
y_hamm=conv(s,h_hamming);
figure; pwelch(y_hamm,[],[],Fs);
y_kais=conv(s,h_kaiser);
figure; pwelch(y_kais,[],[],Fs);
%
% Βαθυπερατό Parks-MacClellan
hpm=firpm(196, [0 0.1 0.15 0.5]*2, [1 1 0 0]);%hpm=firpm(196, [0 0.1 0.15 0.5]*2, [1 1 0 0]); %3) %4)
%figure; freqz(hpm,1);
t = 0:1/Fs:1-1/Fs;
s=1*sin(2*pi*500*t)+1*sin(2*pi*800*t)+1*sin(2*pi*1500*t)+1*sin(2*pi*3000*t);
figure; pwelch(s,[],[],Fs);
s_pm=conv(s,hpm);
figure; pwelch(s_pm,[],[],Fs);
%sound(20*s); % ακούμε το αρχικό σήμα, s
```

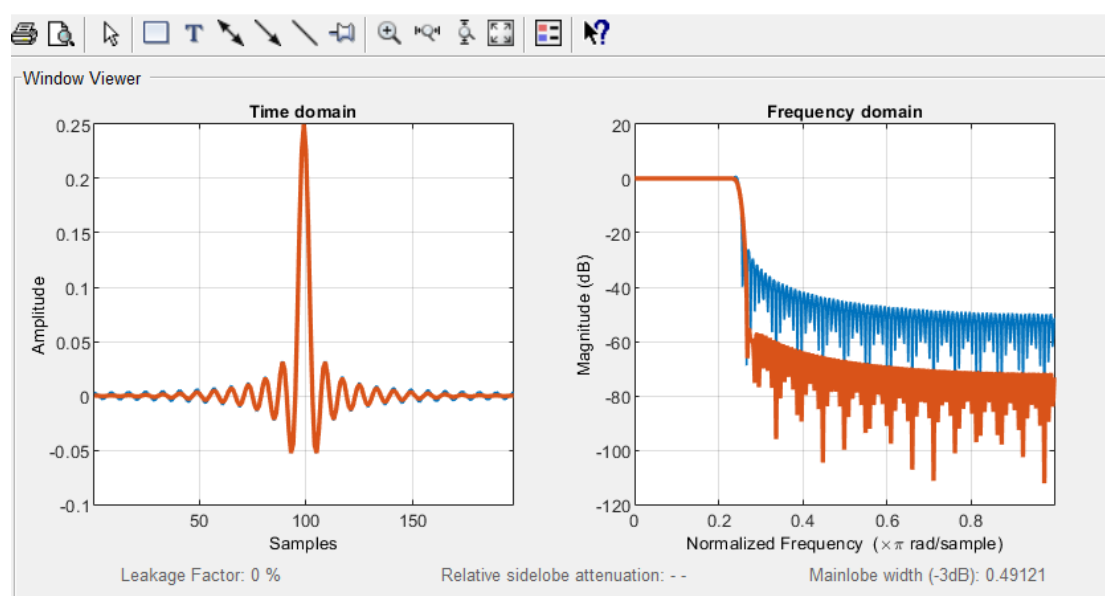
1.

Τροποποιούμε τον κώδικα στη γραμμή 14 ώστε να επιτύχουμε το ίδιο αποτέλεσμα με τη χρήση της συνάρτησης `fftshift()`. Με άλλα λόγια, αλλάζουμε την εντολή `h2=[h(middle+1:end) h(1:middle)]`; σε `h = fftshift(h)`; . Επαληθεύουμε την επιλογή μας με την εντολή `isequal(h,h2)`, η οποία επιστρέφει την τιμή 1 και συνεπώς η αντικατάσταση της εντολής είναι σωστή, γιατί επιστρέφεται η ίδια κρουστική συνάρτηση.

2.

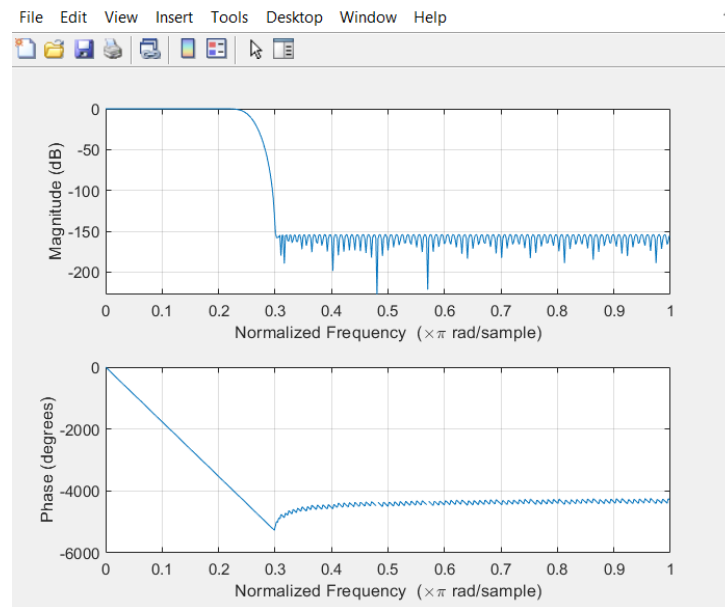
Τροποποιούμε τον κώδικα ώστε να χρησιμοποιηθεί βαθυπερατό φίλτρο μήκους 196+1 αντί 64+1, με τη βοήθεια της εντολής για τη δημιουργία της κρουστικής συνάρτησης $h_{196}=h(\text{middle}+1-98:\text{middle}+99)$; Σχεδιάζουμε την απόκριση συχνότητας του φίλτρου με σχεδιασμό παραθύρου για τις περιπτώσεις ορθογωνίου και Hamming, με τη βοήθεια των εντολών που φαίνονται στον παραπάνω κώδικα.

Σχεδιάζουμε, με τη βοήθεια της εντολής `wntool(h196, h_hamming)`;, την απόκριση συχνότητας του φίλτρου με σχεδιασμό παραθύρου Hamming (πορτοκαλί διάγραμμα) και ορθογωνικού παραθύρου (μπλε διάγραμμα), όπως απεικονίζεται παρακάτω:

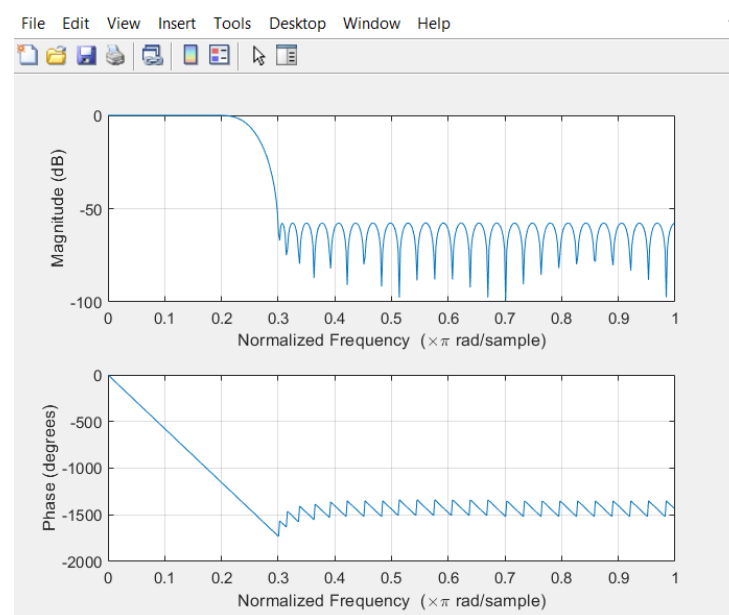


3.

- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους 196+1 με οριακές συχνότητες τις (0.1 0.15). Σχεδιάζουμε, όπως φαίνεται παρακάτω, την απόκριση συχνότητας για το παραπάνω φίλτρο:



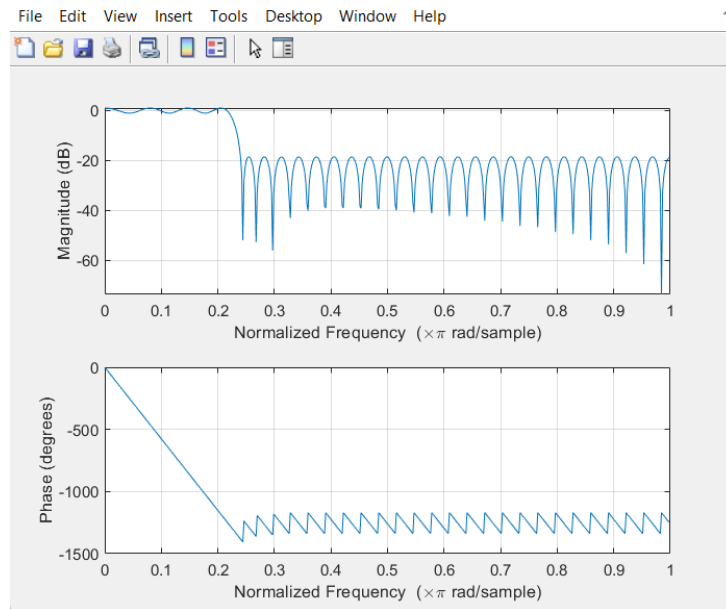
- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους 64+1 με οριακές συχνότητες τις (0.1 0.15). Σχεδιάζουμε, όπως φαίνεται παρακάτω, την απόκριση συχνότητας για το παραπάνω φίλτρο:



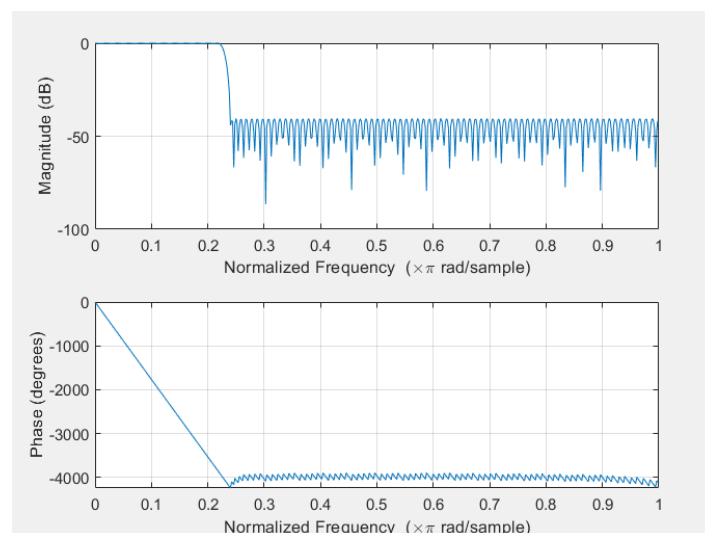
Τα συμπεράσματά μας τα αναλύσαμε, όπως ζητήθηκε, στο αρχείο κειμένου lab2_19839.txt.

4.

- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους 64+1 με οριακές συχνότητες τις (0.11 0.12). Σχεδιάζουμε, όπως φαίνεται παρακάτω, την απόκριση συχνότητας για το παραπάνω φίλτρο:



- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους 196+1 με οριακές συχνότητες τις (0.11 0.12). Σχεδιάζουμε, όπως φαίνεται παρακάτω, την απόκριση συχνότητας για το παραπάνω φίλτρο:



Τα συμπεράσματά μας τα αναλύσαμε, όπως ζητήθηκε, στο αρχείο κειμένου lab2_19839.txt.

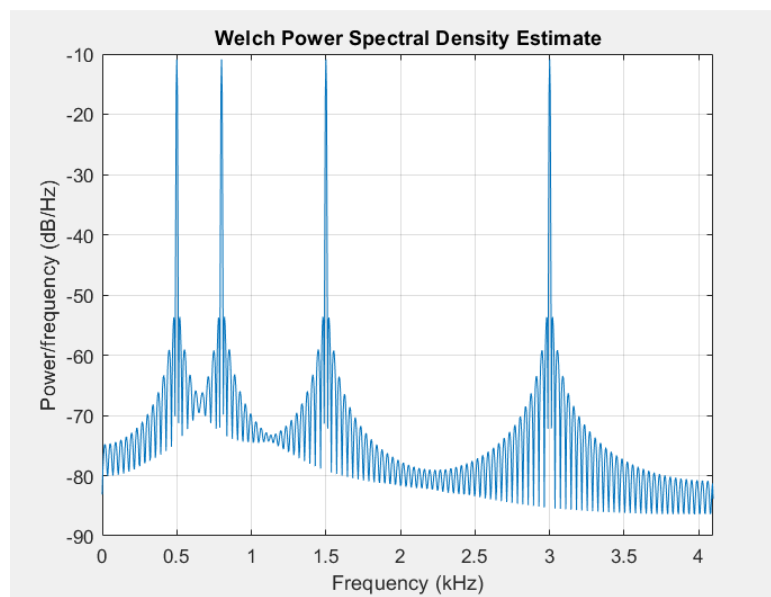
5.

Αντικαθιστούμε το σήμα s με άθροισμα τεσσάρων ημιτονικών συναρτήσεων μοναδιαίου πλάτους, συχνότητας 500,800,1500 και 3000 Hz και διάρκειας 1.0 sec.

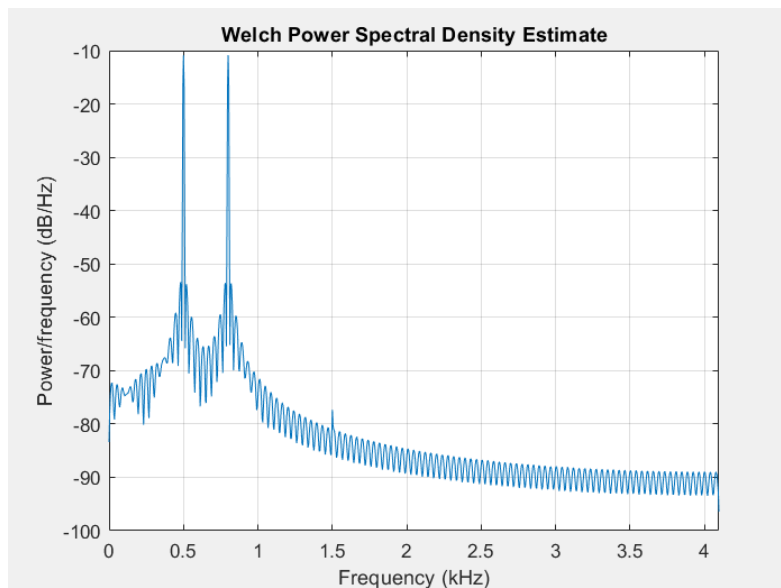
Χρησιμοποιήσαμε τις παρακάτω εντολές στον κώδικα Matlab:

```
t = 0:1/Fs:1-1/Fs;  
s=1*sin(2*pi*500*t)+1*sin(2*pi*800*t)+1*sin(2*pi*1500*t)+1*sin(2*pi*3000*t);
```

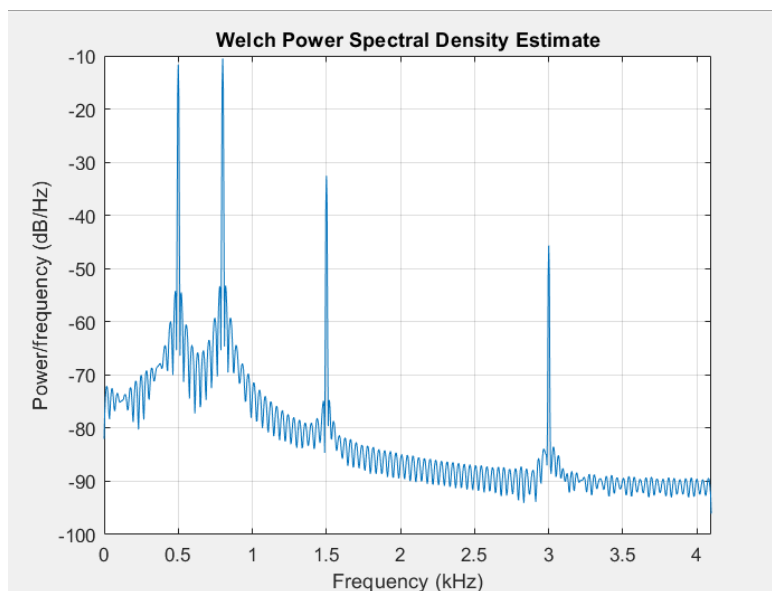
- ❖ Σχεδιάζουμε τη φασματική πυκνότητα του αρχικού σήματος με τη βοήθεια της εντολής `pwelch(s,[],[],[],Fs);`



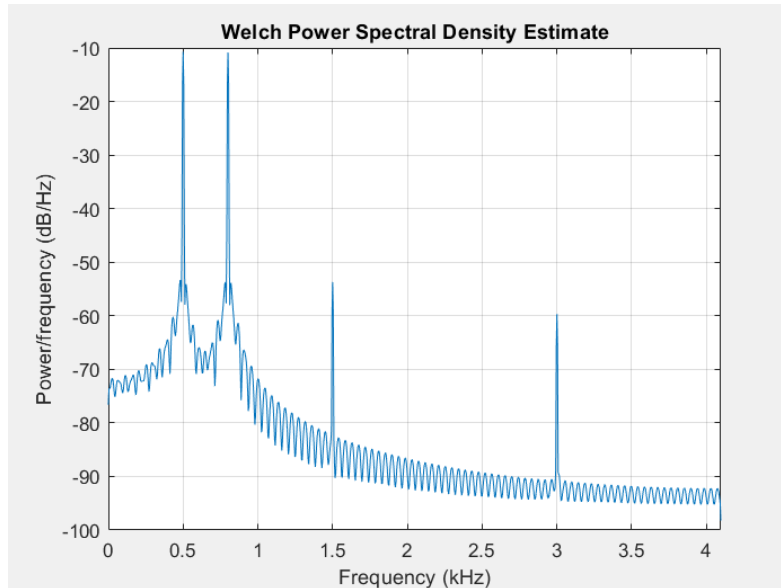
- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους $64+1$ με οριακές συχνότητες τις $(0.1 \ 0.15)$. Σχεδιάζουμε, όπως φαίνεται παρακάτω, το φιλτράρισμα για το παραπάνω φίλτρο:



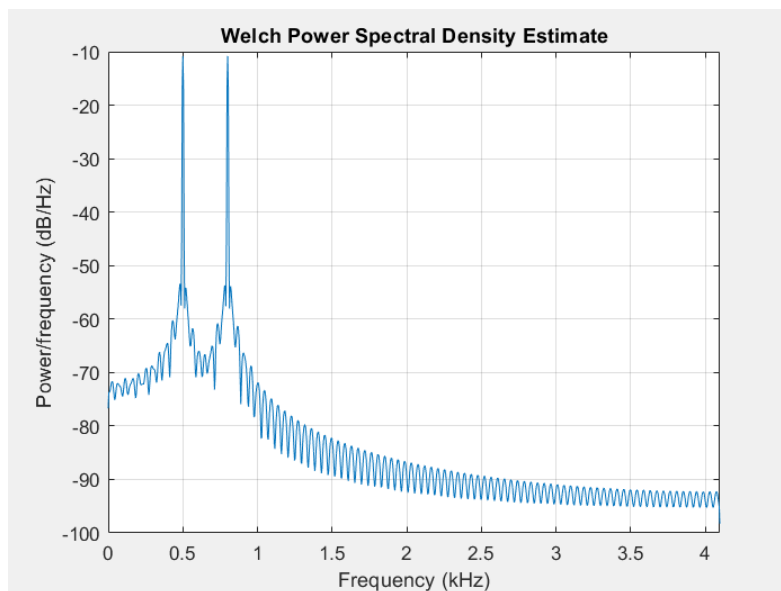
- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους $64+1$ με οριακές συχνότητες τις $(0.11 \ 0.12)$. Σχεδιάζουμε, όπως φαίνεται παρακάτω, το φιλτράρισμα για το παραπάνω φίλτρο:



- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους 196+1 με οριακές συχνότητες τις (0.11 0.12). Σχεδιάζουμε, όπως φαίνεται παρακάτω, το φιλτράρισμα για το παραπάνω φίλτρο:



- ❖ Σχεδιάζουμε το φίλτρο Parks-McClellan μήκους 196+1 με οριακές συχνότητες τις (0.1 0.15). Σχεδιάζουμε, όπως φαίνεται παρακάτω, το φιλτράρισμα για το παραπάνω φίλτρο:



Τα συμπεράσματά μας τα αναλύσαμε, όπως ζητήθηκε, στο αρχείο κειμένου lab2_19839.txt.

Μέρος 3: Εφαρμογή Α

Παρακάτω, παρατίθεται ο κώδικας που περιέχεται στο αρχείο lab2_2_19839.m:

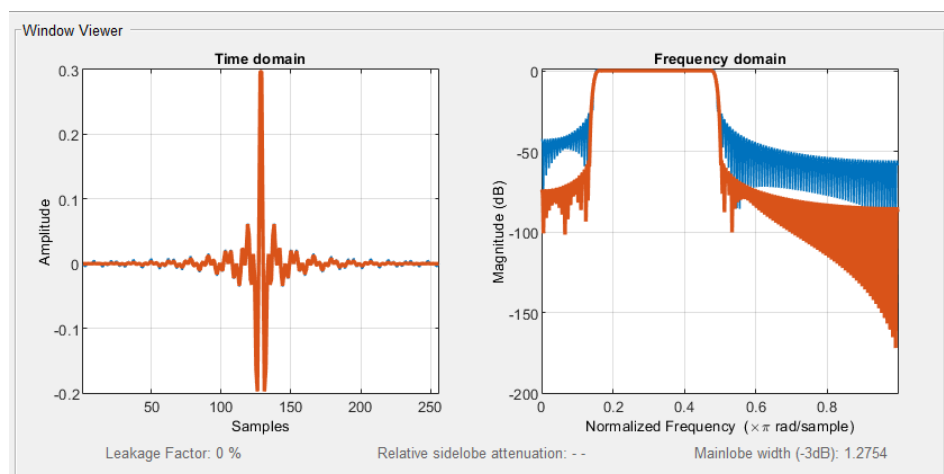
```
% Part 3 - Eirini Donti
% (α)
% Bandpass FIR filter - closed-form and windowing
clear all; close all;
load sima; % Fs = 8192;
flow=600; fhigh=2000; Ts=1/Fs;
f2m1=(fhigh-flow); f2p1=(flow+fhigh)/2; N=256;
t=[-(N-1)/2:(N-1)/2]*Ts/2;
hbp=2/Fs*cos(2*pi*f2p1*t).*sin(pi*f2m1*t)/pi./t;
hbpw=hbp.*kaiser(length(hbp),5);
wvtool(hbp,hbpw);
sima_bp=conv(s,hbp);
figure; pwelch(sima_bp,[],[],[],Fs); hold on; legend("(before) Eirini Donti");
sima_bpw=conv(s,hbpw);
figure; pwelch(sima_bpw,[],[],[],Fs); hold on; legend("(after windowing) Eirini Donti");
% (β)
% Bandpass Parks-MacClellan
clear all; close all; load sima;
flow = 600; fhigh=2000; %Hz
f=2*[0 flow*0.95 flow*1.05 fhigh*0.95 fhigh*1.05 Fs/2]/Fs; %high side lobe
f=2*[0 570 630 1970 2030 4096]/Fs; %normal side lobe
hbp_pm=firpm(256,f,[0 0 1 1 0 0]);
%figure; freqz(hpm,1);
wvtool(hbp_pm);
sima_pm=conv(s,hbp_pm);
figure; pwelch(sima_pm,[],[],[],Fs); hold on; legend("(after filtering) Eirini Donti");
```

Παρακάτω, παραθέτουμε τα ζητούμενα διαγράμματα που προέκυψαν αφότου εκτελέσαμε τον παραπάνω κώδικα:

(1) Απόκριση συχνότητας των δύο φίλτρων

Μέθοδος 1^η: Μέθοδος Παραθύρων

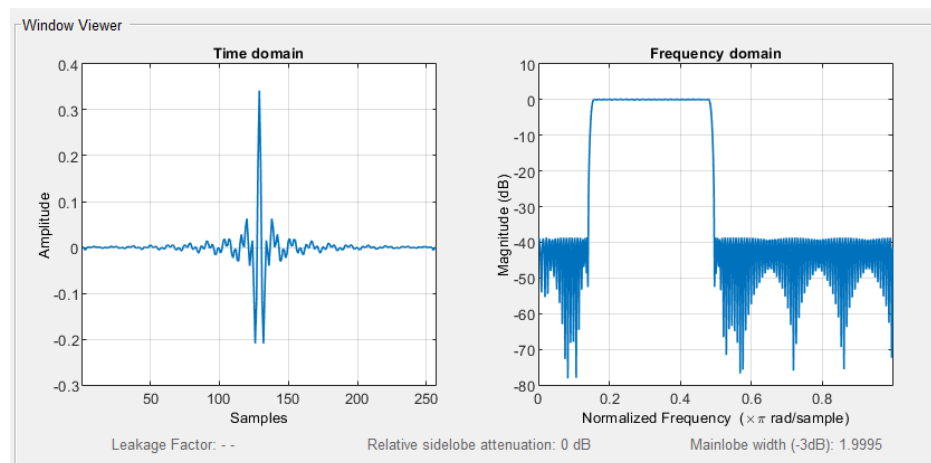
Παρακάτω, παραθέτουμε την απόκριση συχνότητας (δεξιά, πορτοκαλί γραφική παράσταση) με τη μέθοδο παραθύρων (Kaiser), σύμφωνα με την εντολή wvtool():



Η γραφική παράσταση που παρουσιάζεται παραπάνω με μπλε χρώμα, είναι η απόκριση συχνότητας του σήματος πριν φιλτραριστεί από το παράθυρο Kaiser. Παραπάνω, επαληθεύεται ο σωστός σχεδιασμός του φίλτρου, καθώς η απόκριση συχνότητας του σήματος πριν φιλτραριστεί από το παράθυρο Kaiser (μπλε γραφική παράσταση) πλησιάζει, σε αρκετό βαθμό, την απόκριση συχνότητας μετά το φιλτράρισμα από το παράθυρο Kaiser (πορτοκαλί γραφική).

Μέθοδος 2^η: Μέθοδος Ισοϋψών κυματώσεων

Παρακάτω, παραθέτουμε την απόκριση συχνότητας (δεξιά γραφική παράσταση) με τη μέθοδο των ισοϋψών κυματώσεων, σύμφωνα με την εντολή `wvtool()`:

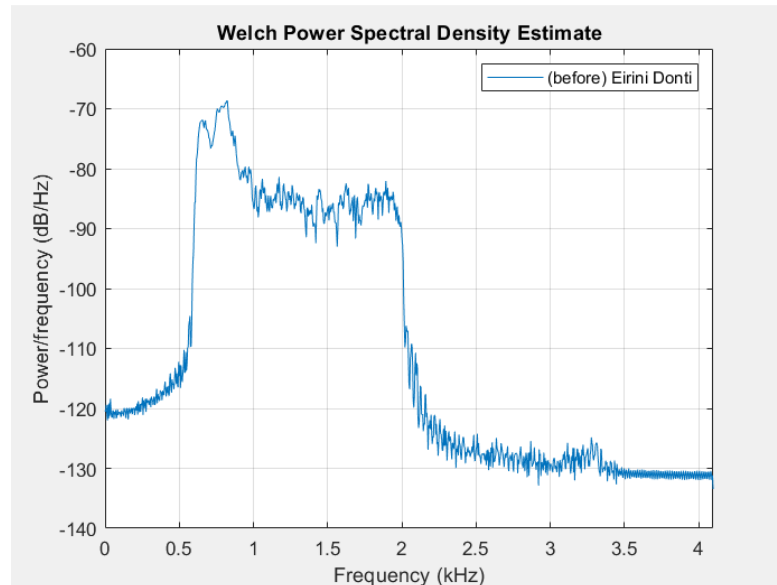


Παρατηρούμε ότι οι μορφές των αποκρίσεων συχνότητας, με τις δύο μεθόδους, είναι παρόμοιες. Όμως, η γραφική που προκύπτει με τη μέθοδο των ισοϋψών κυματώσεων, πλησιάζει περισσότερο την απόκριση συχνότητας σε θεωρητικά πλαίσια. Συγκεκριμένα, η εφαρμογή του φίλτρου εκτείνεται στον κατακόρυφο άξονα κατά 40 dB πλάτους (σε απόλυτη τιμή).

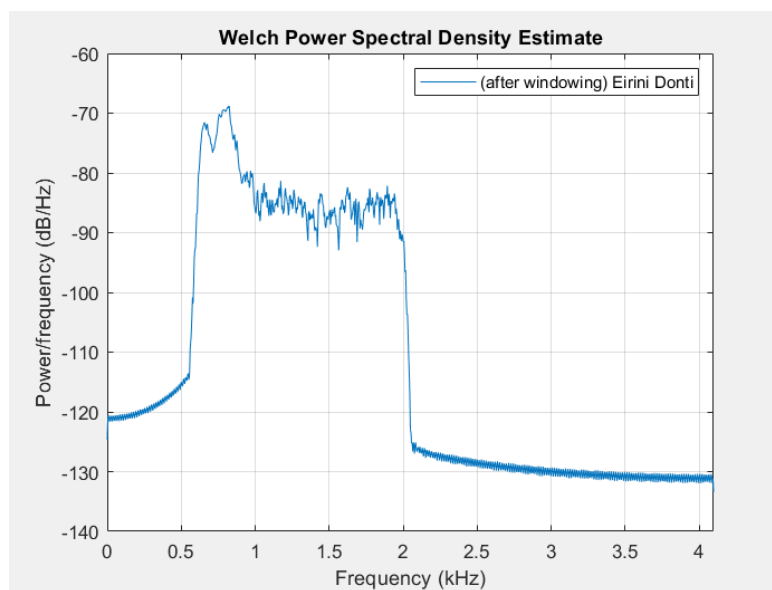
(2) Φασματική Πυκνότητα του σήματος, πριν και μετά το φιλτράρισμα

Μέθοδος 1^η: Μέθοδος Παραθύρων

Παρακάτω, παραθέτουμε τη φασματική πυκνότητα του σήματος πριν το φιλτράρισμα, σύμφωνα με τις εντολές `sima_bp=conv(s,hbp); figure; pwelch(sima_bp,[],[],[],Fs);`

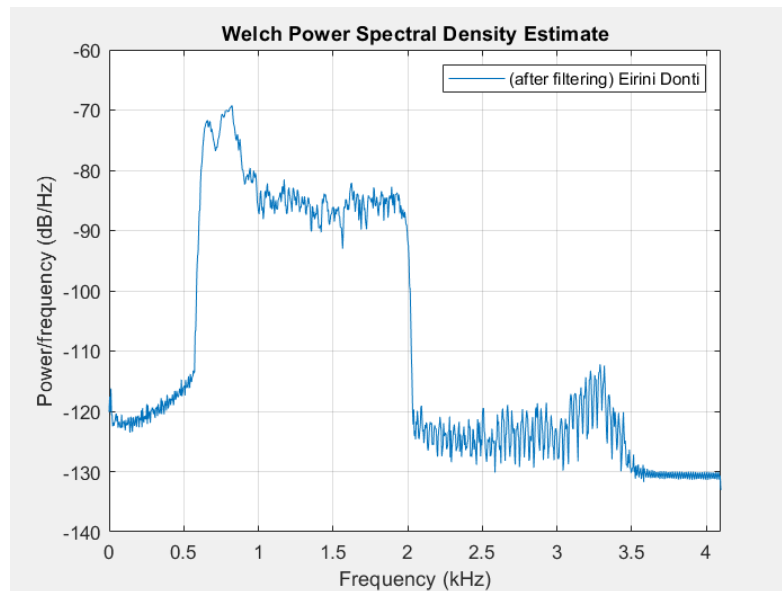


Παρακάτω, παραθέτουμε τη φασματική πυκνότητα του σήματος μετά το φιλτράρισμα με τη μέθοδο των παραθύρων (Kaiser), σύμφωνα με τις εντολές `sima_bpw=conv(s,hbpw); figure; pwelch(sima_bpw,[],[],[],Fs);`



Μέθοδος 2^η: Μέθοδος Ισοϋψών κυματώσεων

Παρακάτω, παραθέτουμε τη φασματική πυκνότητα του σήματος μετά το φιλτράρισμα με τη μέθοδο των ισοϋψών κυματώσεων, σύμφωνα με τις εντολές `wvtool(hbp_pm); sima_pm=conv(s,hbp_pm); figure; pwelch(sima_pm,[],[],[],Fs);`



Παρατηρούμε ότι το διάγραμμα της εκτίμησης φασματικής πυκνότητας ισχύος κατά Welch είναι παρόμοιο και για τις δύο μεθόδους. Παρατηρούμε, επίσης, ότι το διάγραμμα που προκύπτει με τη μέθοδο των παραθύρων, είναι λιγότερο αλλοιωμένο, ενώ το διάγραμμα που προκύπτει με τη μέθοδο των ισοϋψών κυματώσεων μοιάζει περισσότερο με το διάγραμμα πριν το φιλτράρισμα.