Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Εργαστήριο Επεξεργασίας Σημάτων & Τηλεπικοινωνιών
Τομέας Υλικού και Αρχιτεκτονικής Υπολογιστών
Πανεπιστήμιο Πατρών

Εργαστήριο Ψηφιακής Επεξεργασίας Σημάτων

Σετ Βοηθητικών Ασκήσεων #2

(Σχεδιασμός Φίλτρων & Αποθορυβοποίηση Σημάτων)

Επιβλέπων Καθηγητής:
Δημήτριος Κοσμόπουλος

Σύνταξη – Επιμέλεια:
Αλέξανδρος-Οδυσσέας Φαρμάκης

Πάτρα, Εαρινό Εξάμηνο, 2022-23

Θέματα: Εισαγωγή στη MATLAB για ΨΕΣ

Θέμα 1ο) Μελέτη Συναρτήσεων Παραθύρων

Τόσο στον σχεδιασμό ψηφιακών φίλτρων όσο και στην εκτίμηση κάποιας φασματικής ανάλυσης που κάνουμε, η επιλογή μιας συνάρτησης παραθύρου παίζει σημαντικό ρόλο στον προσδιορισμό της ποιότητας των συνολικών αποτελεσμάτων που εξάγουμε. Ο κύριος ρόλος του παραθύρου είναι να εξαλείψει τις επιπτώσεις του φαινομένου Gibbs που προκύπτει κατά την περικοπή μιας άπειρης σειράς. Με βάση αυτά, ζητείται:

α) Να συγκρίνετε το συμμετρικό παράθυρο Hamming μήκους $N = 64$ σε σχέση με το περιοδικό παράθυρο Hamming μήκους $N = 63$, στα οποία να εξηγείστε τις διαφορές που παρατηρούνται μεταξύ τους, δείχνοντας και τα κατάλληλα γραφήματα.

β) Να σχεδιάσετε κατάλληλα ένα παράθυρο Chebyshev μήκους $N = 64$ με τέτοιο τρόπο ώστε ο κύριος λοβός του να βρίσκεται στο ίδιο ύψος με αυτό των παραθύρων Hamming που υλοποιήσατε στο προηγούμενο ερώτημα. Οι ζητούμενες αποκρίσεις να βρίσκονται στα ίδια γραφήματα αντίστοιχα. Πότε είναι καλύτερο να χρησιμοποιήσουμε παράθυρο Chebyshev και πότε παράθυρο Hamming;

γ) Να σχεδιάσετε κατάλληλα ένα τετραγωνικό παράθυρο και σχολιάστε τα αποτελέσματά του σε σχέση με τις προηγούμενες διαδικασίες που υλοποιήσατε.

Λύση:

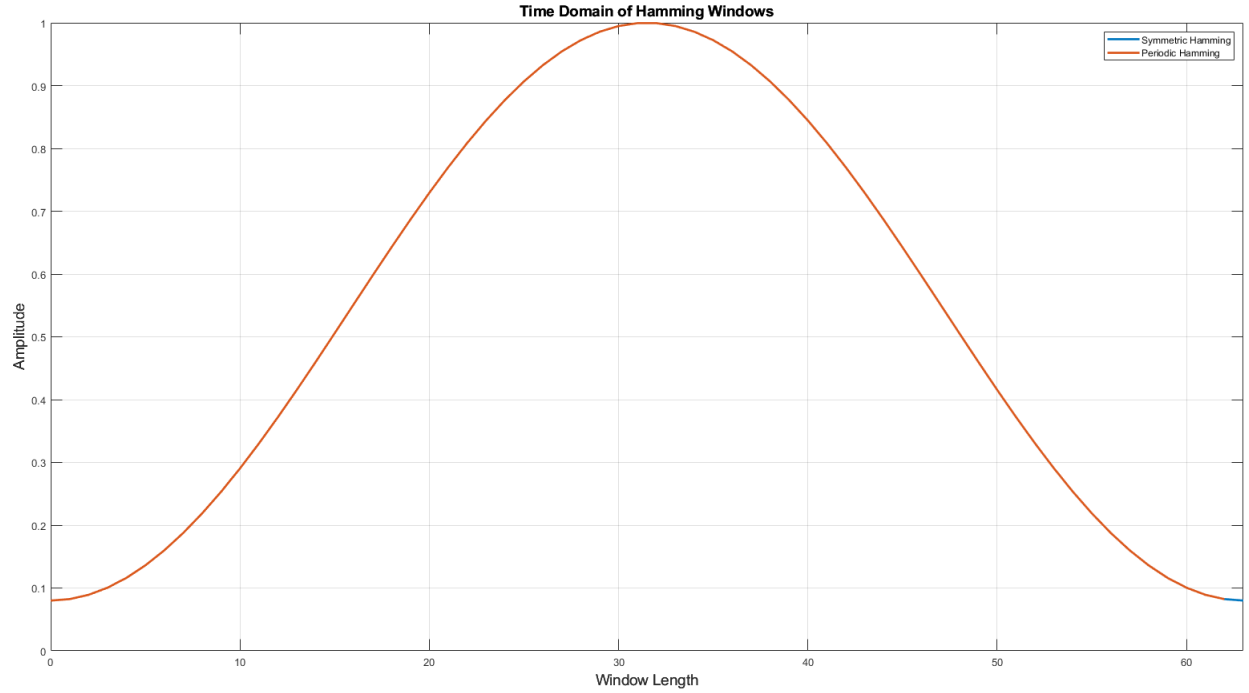
α) Ο λόγος για τον οποίο το περιοδικό παράθυρο Hamming έχει ρυθμιστεί να έχει μήκος μικρότερο κατά ένα από αυτό του συμμετρικού παραθύρου Hamming είναι για να διασφαλιστεί ότι τα δύο παράθυρα έχουν το ίδιο συνολικό σχήμα όταν χρησιμοποιούνται για τη παραθυροποίηση ενός σήματος.

Το περιοδικό παράθυρο Hamming τυλίγεται στις άκρες και αντιμετωπίζει το σήμα σαν να επαναλαμβάνεται άπειρα και προς τις δύο κατευθύνσεις. Αυτό μπορεί να είναι χρήσιμο σε ορισμένες εφαρμογές όπου το σήμα θεωρείται ότι είναι περιοδικό ή όταν το παραθυροποιημένο σήμα πρέπει να επεξεργαστεί με κυκλικό τρόπο. Δεδομένου ότι το περιοδικό παράθυρο Hamming τυλίγεται στις άκρες, δεν έχει αποτέλεσμα περιορισμού (tapering) στις άκρες όπως το αντίστοιχο συμμετρικό παράθυρο Hamming, του οποίου τιμές είναι συμμετρικές ως το κέντρο του παραθύρου, που οδηγεί σε ζυγό αριθμό λοβών στον χώρο της συχνότητας.

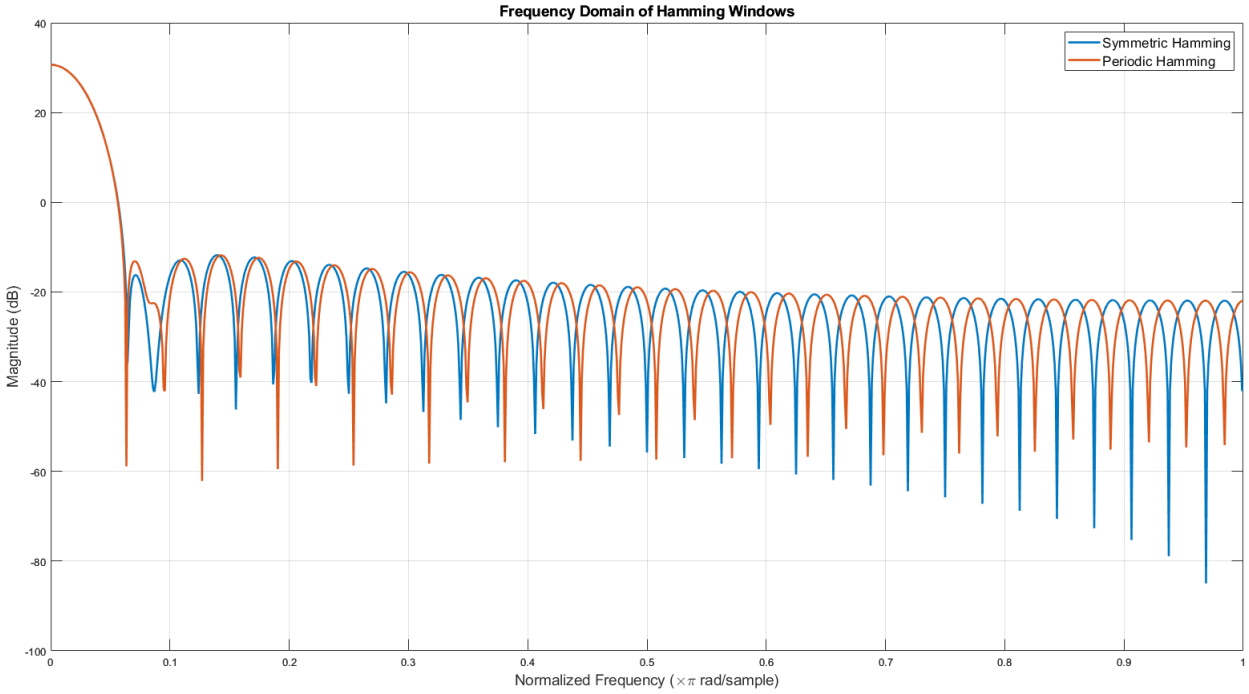
Ως εκ τούτου, ουσιαστικά μετατοπίζεται το περιοδικό παράθυρο κατά ένα δείγμα, ευθυγραμμίζοντας τις κορυφές και τις κοιλάδες του με εκείνες του συμμετρικού παραθύρου. Έχοντας το ίδιο σχήμα, επιτρέπει την εύκολη σύγκριση της απόδοσής τους σε διαφορετικές εφαρμογές, όπως στη φασματική ανάλυση. Σημειώνεται επίσης ότι το συμμετρικό παράθυρο Hamming έχει σχεδιαστεί για να ελαχιστοποιεί τη φασματική διαρροή (spectral leakage), η οποία είναι η διαρροή ενέργειας του σήματος από τον έναν "καλάθι" συχνοτήτων στον άλλο κατά την ανάλυση Fourier, την οποία επιτυγχάνει μέσω του tapering.

Τα παραγόμενα γραφήματα βρίσκονται στην επόμενη σελίδα:

Τα παράθυρα Hamming στο χώρο του χρόνου:



Τα παράθυρα Hamming στο χώρο της συχνότητας:

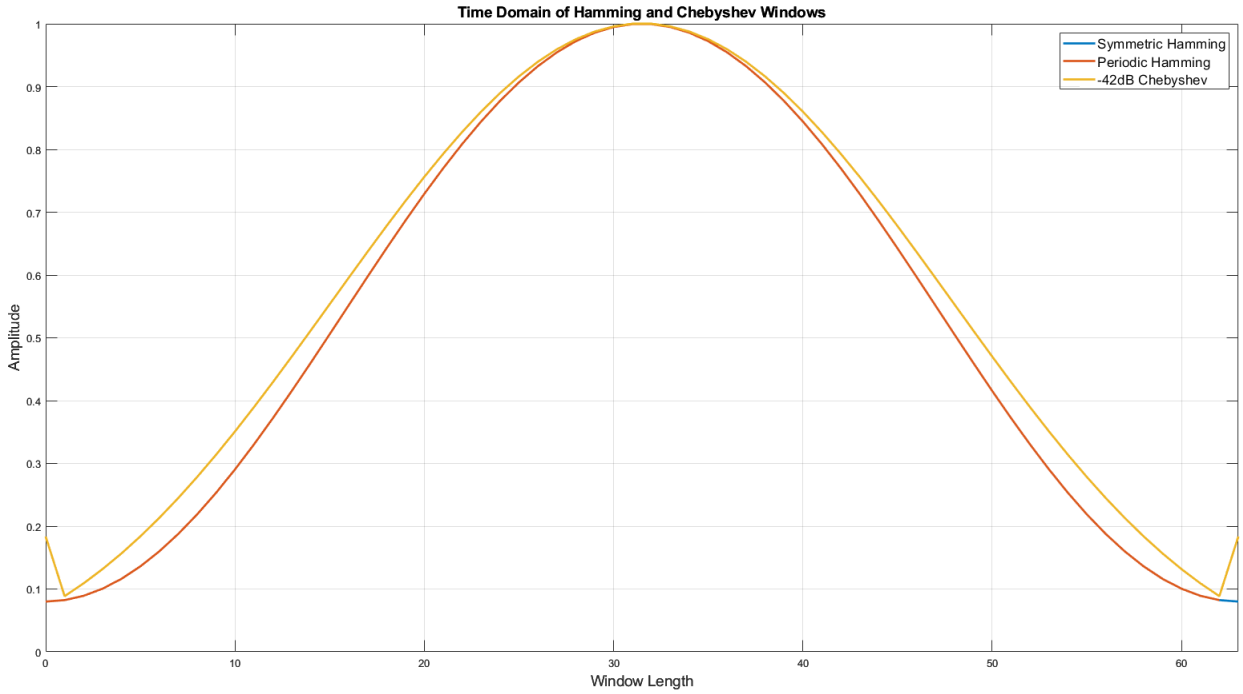


β) Δοκιμαστικά, βλέπουμε ότι πρέπει να ορίσουμε απόσβεση 42dB στο παράθυρο Chebyshev ώστε ο κύριος λοβός του να βρίσκεται στο ίδιο ύψος με αυτό των παραθύρων Hamming των προηγουμένων ερωτημάτων. Παρακάτω υπάρχει ένας πίνακας που δείχνει μερικές πληροφορίες για το πότε να επιλέγουμε παράθυρο Chebyshev και πότε παράθυρο Hamming.

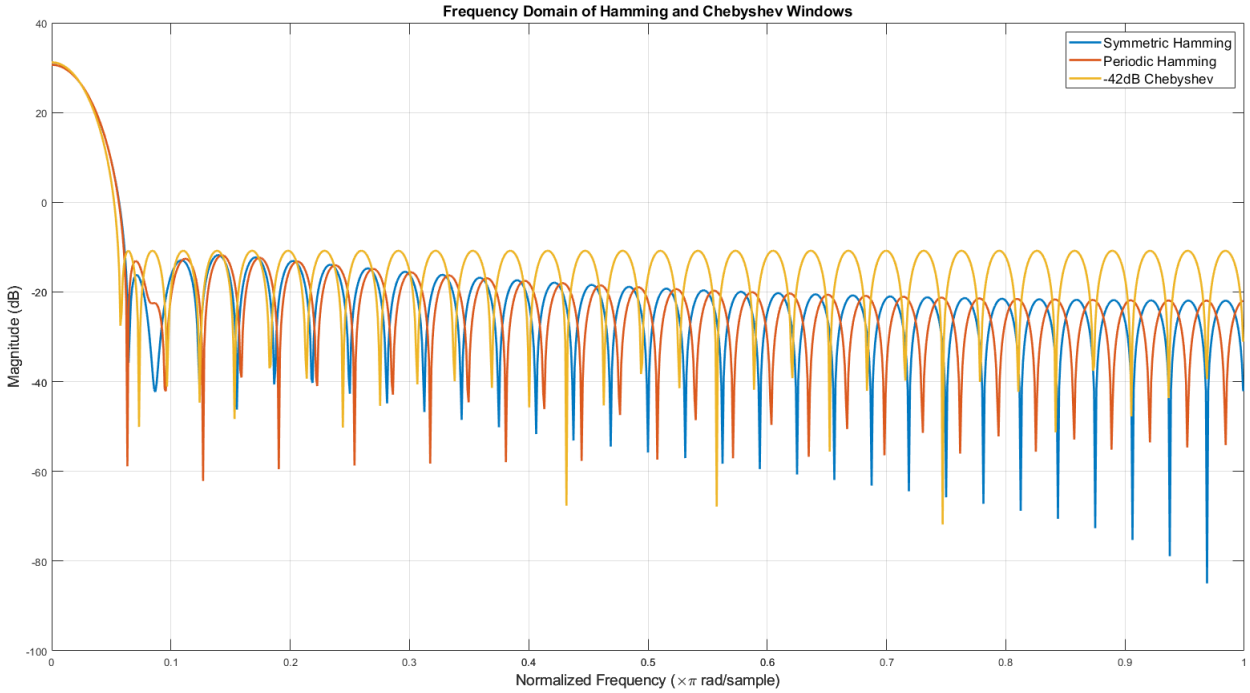
Παράθυρα Chebyshev	Παράθυρα Hamming
Ανάγκη απότομων μεταβατικών ζωνών	Είναι επιθυμητή η μέτρια εξασθένηση του πλευρικού λοβού (sidelobe attenuation)
Η απόσβεση της ζώνης αποκοπής είναι προτιμητέα σε σχέση με το κυματισμό της ζώνης διέλευσης	Η απλότητα και η ευκολία χρήσης είναι μεγαλύτερες προτεραιότητες
Ο λόγος του σήματος ως προς τον θόρυβο (signal-to-noise ratio, SNR) μπορεί να προκαλέσει προβλήματα	Η ανάλυση συχνότητας είναι ο πιο σημαντικός παράγοντας για την επιλογή παραθύρου

Τα παραγόμενα γραφήματα δίνονται σε αυτή και την επόμενη σελίδα:

Τα παράθυρα Chebyshev και Hamming στο χώρο του χρόνου:



Τα παράθυρα Chebyshev και Hamming στο χώρο της συχνότητας:



Ο κώδικας αυτού και του προηγούμενου ερωτήματος δίνονται παρακάτω:

```
% Μήκος παραθύρου και συντελεστής απόσβεσης
L = 64;
m = 42;

% Συναρτήσεις παραθύρου
windows = {'sym_hamm', 'per_hamm', 'cheb'};

% Custom naming scheme για κάθε συνάρτηση παραθύρου
window_names = containers.Map(windows, {'Symmetric Hamming', ...
    'Periodic Hamming', '-42dB Chebyshev'});

% Αρχικοποίηση των handle για τη δημιουργία των κατάλληλων γραφημάτων
figure_handles = zeros(2, 1);

% Δέσμευση μνήμης για τα legend των γραφημάτων
legend_names = cell(2, numel(windows));

% Βρόχος για τη σχεδίαση των γραφημάτων των παραθύρων στο χώρο
% του χρόνου και της συχνότητας
for i = 1:numel(windows)
    window_function = windows{i};
    if strcmp(window_function, 'sym_hamm')
        chosen_window = hamming(L, 'symmetric');
        x_axis = 0:L-1;
    elseif strcmp(window_function, 'per_hamm')
```

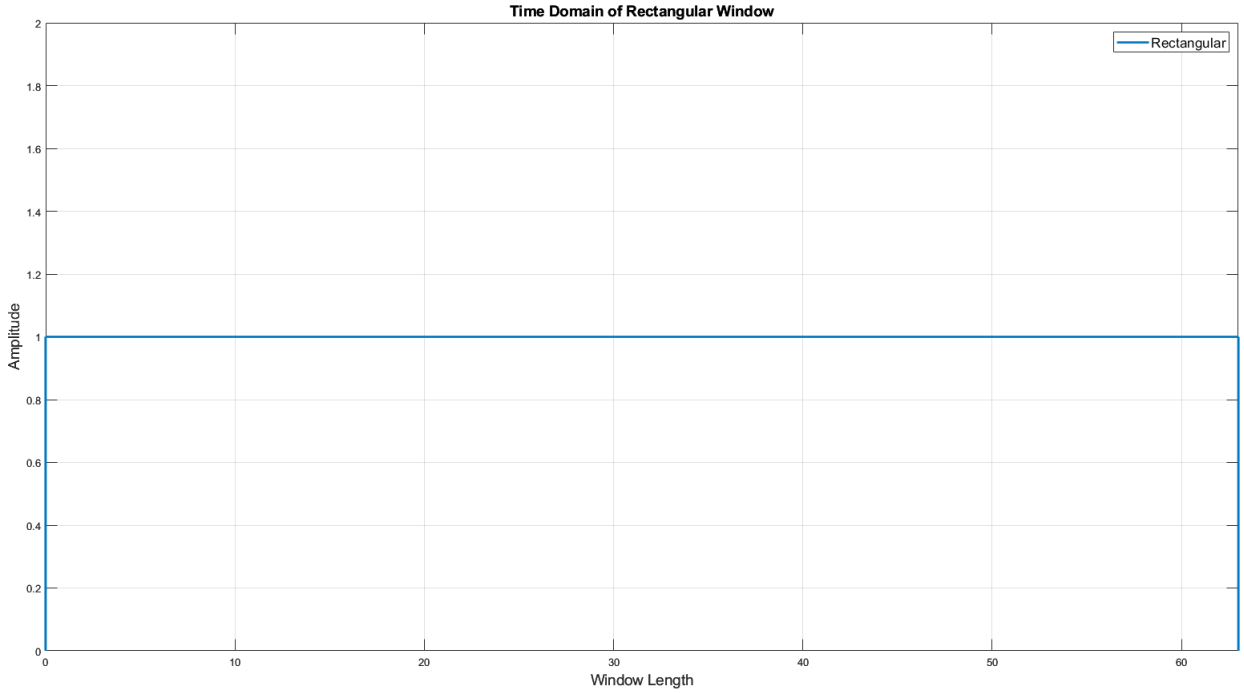
```
        chosen_window = hamming(L-1, 'periodic');
        x_axis = 0:L-2;
    else
        chosen_window = chebwin(L, m);
        x_axis = 0:L-1;
    end

    [H, w] = freqz(chosen_window, 1, 1024);

    % Εύρεση του ονόματος της συνάρτησης παραθύρου από λίστα
    window_func_name = window_names(window_function);

    % Σχεδίαση γραφήματος του παραθύρου στο χώρο του χρόνου
    figure_handles(1) = figure(1);
    plot(x_axis, chosen_window, 'LineWidth', 2);
    grid on;
    hold on;

    % Σχεδίαση γραφήματος του παραθύρου στο χώρο της συχνότητας
    figure_handles(2) = figure(2);
    plot(w/pi, 20*log10(abs(H)), 'LineWidth', 2);
    grid on;
    hold on;

    % Αποθήκευση των ονομάτων των legend για κάθε συνάρτηση παραθύρου
    legend_names{1,i} = window_func_name;
    legend_names{2,i} = window_func_name;
end

% Προσθήκη legend στα γραφήματα των παραθύρων Hamming και Chebyshev
% στο χώρο του χρόνου
figure(1);
legend(legend_names{1, 1:3});
xlim([0 63]);
title('Time Domain of Hamming and Chebyshev Windows');
xlabel('Window Length');
ylabel('Amplitude');
hold off;

% Προσθήκη legend στα γραφήματα των παραθύρων Hamming και Chebyshev
% στο χώρο της συχνότητας
figure(2);
legend(legend_names{2, 1:3});
title('Frequency Domain of Hamming and Chebyshev Windows');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
hold off;
```

γ) Οι συναρτήσεις παραθύρου εφαρμόζονται συνήθως με πολλαπλασιασμό ανά στοιχείο του σήματος. Αυτό σημαίνει ότι κάθε δείγμα της συνάρτησης παραθύρου πολλαπλασιάζεται με το αντίστοιχο δείγμα του σήματος. Συνεπώς, αν θέλαμε να κρατήσουμε το γινόμενο αυτό “ως έχει”, αρκεί να το πολλαπλασιάζαμε με τη μονάδα, δηλαδή εδώ το τετραγωνικό παράθυρο. Αν και το τετραγωνικό παράθυρο πλήττεται από χαμηλή ανάλυση συχνότητας και υψηλούς πλευρικούς λοβούς, γεγονός που μπορεί να οδηγήσει σε φασματική διαρροή και μειωμένη απόδοση σε ορισμένες εφαρμογές σε σύγκριση με άλλες λειτουργίες παραθύρου, έχει και τις χρήσεις του.

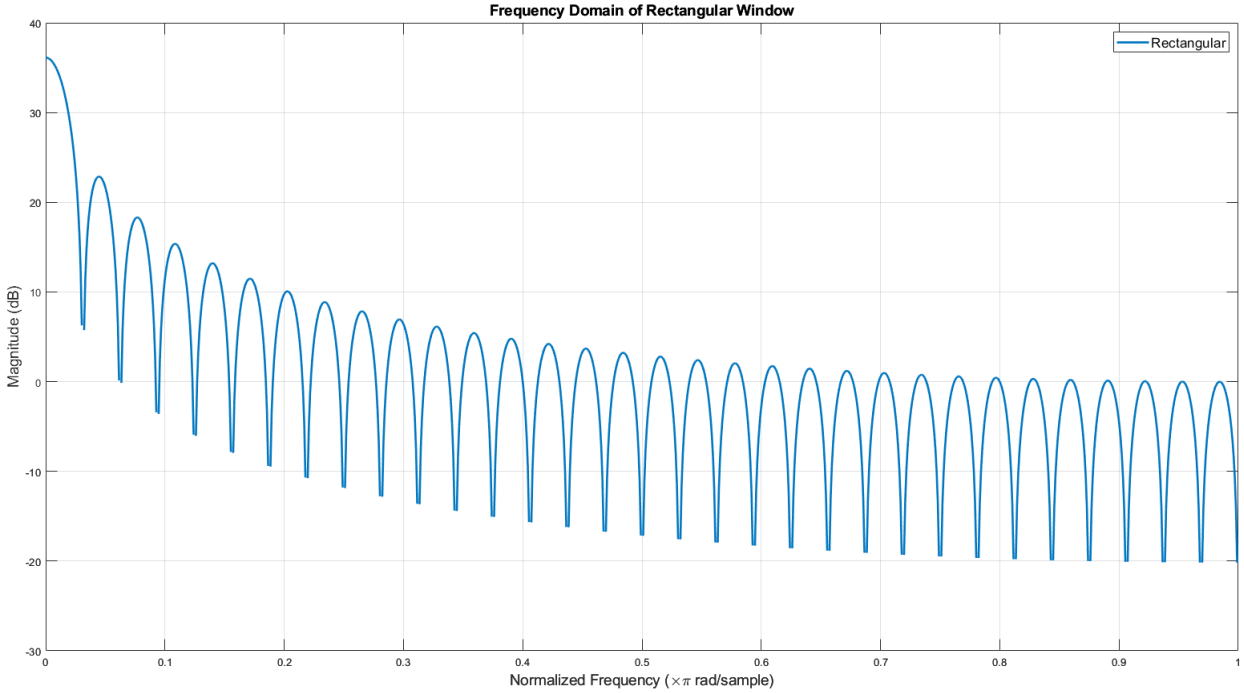
Για παράδειγμα, συγκρίνοντας την απόδοση μιας συνάρτησης παραθύρου ή ενός αλγόριθμου επεξεργασίας σήματος έναντι του ιδανικού τετραγωνικού παραθύρου, μπορούν να αποκτηθούν πληροφορίες για τα θετικά και αρνητικά μεταξύ διαφορετικών συναρτήσεων παραθύρου ή τεχνικών επεξεργασίας σήματος όσον αφορά χαρακτηριστικά όπως η ανάλυση συχνότητας, τα επίπεδα των πλευρικών λοβών και τη φασματική διαρροή. Επιπλέον, το τετραγωνικό παράθυρο μπορεί να χρησιμοποιηθεί για την απομόνωση κάποιου τμήματος ενός σήματος για ανάλυση. Το σήμα εκτός του καθορισμένου διαστήματος ουσιαστικά μηδενίζεται, απομονώνοντας αποτελεσματικά το επιθυμητό τμήμα. Αυτό μπορεί να είναι χρήσιμο σε εφαρμογές όπως η φασματική ανάλυση, όπου στόχος είναι η ανάλυση του περιεχομένου συχνότητας ενός συγκεκριμένου τμήματος ενός σήματος.

Τα παραγόμενα γραφήματα δίνονται σε αυτή και την επόμενη σελίδα:

Το τετραγωνικό παράθυρο στο χώρο του χρόνου:



Το τετραγωνικό παράθυρο στο χώρο της συχνότητας:



Ο κώδικας αυτού του ερωτήματος δίνεται παρακάτω:

```
% Μήκος παραθύρου και συντελεστής απόσβεσης
L = 64;
m = 42;

% Συναρτήσεις παραθύρου
windows = {'sym_hamm', 'per_hamm', 'cheb', 'rect'};

% Custom naming scheme για κάθε συνάρτηση  παραθύρου
window_names = containers.Map(windows, {'Symmetric Hamming', ...
    'Periodic Hamming', '-42dB Chebyshev', 'Rectangular'});

% Αρχικοποίηση των handle για τη δημιουργία των κατάλληλων γραφημάτων
figure_handles = zeros(4, 1);

% Δέσμευση μνήμης για τα legend των γραφημάτων
legend_names = cell(2, numel(windows));

% Βρόχος για τη σχεδίαση των γραφημάτων των παραθύρων στο χώρο
% του χρόνου και της συχνότητας
for i = 1:numel(windows)
    window_function = windows{i};
    if strcmp(window_function, 'sym_hamm')
        chosen_window = hamming(L, 'symmetric');
        x_axis = 0:L-1;
    elseif strcmp(window_function, 'per_hamm')
```

```
        chosen_window = hamming(L-1, 'periodic');
        x_axis = 0:L-2;
    elseif strcmp(window_function, 'rect')
        chosen_window = rectwin(L);
        x_axis = 0:L-1;
    else
        chosen_window = chebwin(L, m);
        x_axis = 0:L-1;
    end

    [H, w] = freqz(chosen_window, 1, 1024);

    % Εύρεση του ονόματος της συνάρτηση παραθύρου από λίστα
    window_func_name = window_names(window_function);

    % Σχεδίαση γραφήματος του παραθύρου στο χώρο του χρόνου
    if i <= 3
        figure_handles(1) = figure(1);
        plot(x_axis, chosen_window, 'LineWidth', 2);
        grid on;
        hold on;
    else
        figure_handles(3) = figure(3);
        plot(x_axis, chosen_window, 'LineWidth', 2);
        grid on;
        hold on;
    end

    % Σχεδίαση γραφήματος του παραθύρου στο χώρο της συχνότητας
    if i <= 3
        figure_handles(2) = figure(2);
        plot(w/pi, 20*log10(abs(H)), 'LineWidth', 2);
        grid on;
        hold on;
    else
        figure_handles(4) = figure(4);
        plot(w/pi, 20*log10(abs(H)), 'LineWidth', 2);
        grid on;
        hold on;
    end

    % Αποθήκευση των ονομάτων των legend για κάθε συνάρτηση παραθύρου
    legend_names{1,i} = window_func_name;
    legend_names{2,i} = window_func_name;
end

% Προσθήκη legend στα γραφήματα των παραθύρων Hamming και Chebyshev
% στο χώρο του χρόνου
figure(1);
legend(legend_names{1, 1:3});
xlim([0 63])
```

```
title('Time Domain of Hamming and Chebyshev Windows');
xlabel('Window Length');
ylabel('Amplitude');
hold off;

% Προσθήκη legend στα γραφήματα των παραθύρων Hamming και Chebyshev
% στο χώρο της συχνότητας
figure(2);
legend(legend_names{2, 1:3});
title('Frequency Domain of Hamming and Chebyshev Windows');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
hold off;

% Προσθήκη legend στο γράφημα του τετραγωνικού παραθύρου
% στο χώρο του χρόνου
figure(3);
legend(legend_names{1, 4});
xlim([0 63]);
ylim([0 2]);
line([0 0], [0 1], 'Color', [0 0.4470 0.7410], 'LineWidth', 2, ...
    'HandleVisibility','off');
line([63 63], [0 1], 'Color', [0 0.4470 0.7410], 'LineWidth', 2, ...
    'HandleVisibility','off');
title('Time Domain of Rectangular Window');
xlabel('Window Length');
ylabel('Amplitude');
hold off;

% Προσθήκη legend στο γράφημα του τετραγωνικού παραθύρου
% στο χώρο της συχνότητας
figure(4);
legend(legend_names{2, 4});
title('Frequency Domain of Rectangular Window');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
hold off;
```

Θέματα: Εισαγωγή στη MATLAB για ΨΕΣ

Θέμα 2ο) Υλοποίηση FIR Φίλτρων

α) Χρησιμοποιώντας τη συνάρτηση ***fir2()*** της MATLAB, η οποία βασίζεται στην τεχνική σχεδίασης φίλτρων με επιλογή των συντελεστών απόκρισης συχνότητας (ελαχιστοποίηση της στάθμης L_2 της συνάρτησης σφάλματος), σχεδιάστε ένα ζωνοπερατό FIR φίλτρο μήκους 26 με συχνότητα διάβασης $\omega_p = 0.36\pi$, και δείξτε την κρουστική απόκριση, την απόκριση μέτρου, φάσης καθώς και την συνάρτηση κόστους από την οποία προκύπτει το φίλτρο.

β) Επαναλάβετε το προηγούμενο ερώτημα για παράθυρα μήκους $N = 27$, και συγκεκριμένα:

1) παράθυρο Chebyshev και απόσβεση 42dB,
2) συμμετρικό παράθυρο Blackman,
3) περιοδικό παράθυρο Blackman,
4) συμμετρικό παράθυρο Hamming,
5) περιοδικό παράθυρο Hamming,
6) τριγωνικό παράθυρο,
7) και τετραγωνικό παράθυρο

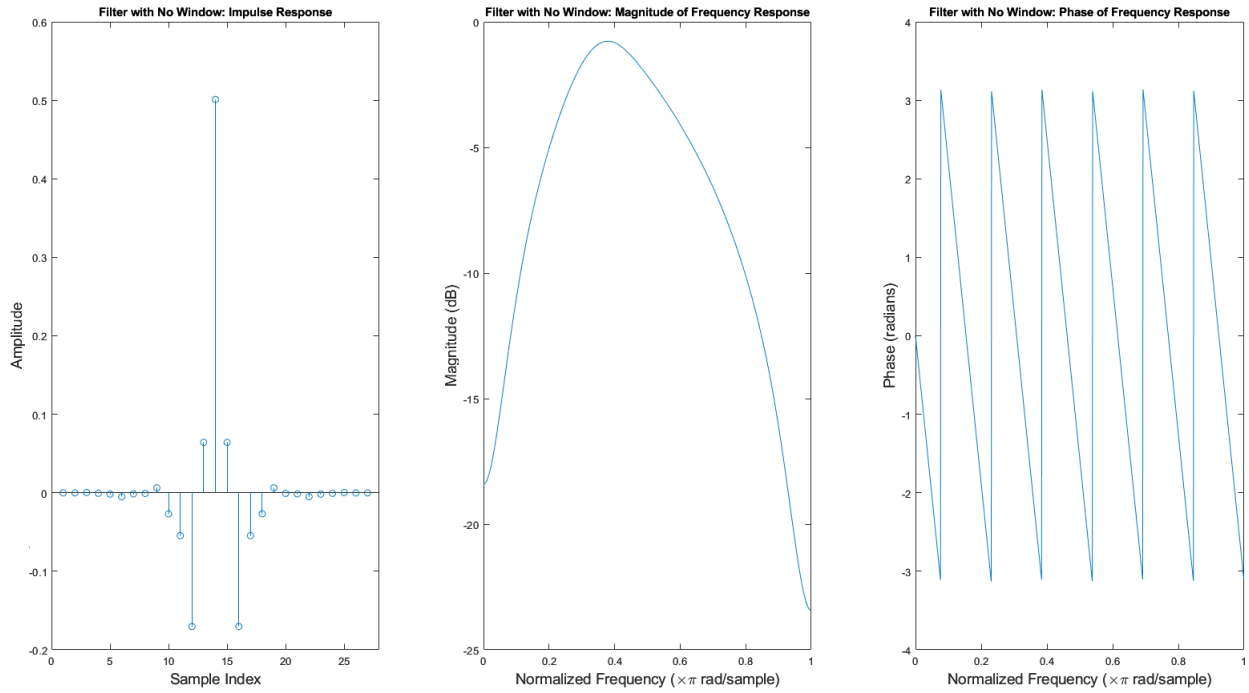
και σχολιάστε τα αποτελέσματά τους.

γ) Χρησιμοποιώντας τη συνάρτηση ***fircls()*** της MATLAB, η οποία βασίζεται στην τεχνική σχεδίασης ελαχίστων τετραγώνων "don't care" (ελαχιστοποίηση της στάθμης L_2 της συνάρτησης σφάλματος εκτός της μεταβατικής ζώνης) με την προσθήκη ορισμένων παραπάνω περιορισμών από μεριάς του χρήστη για την παραγωγή πιο ευέλικτων φίλτρων, να σχεδιάσετε ένα ζωνοπερατό FIR φίλτρο πολλαπλών ζωνών μήκους 50 με συχνότητες διάβασης $\omega_{p1} = 0.13\pi$, $\omega_{p2} = 0.34\pi$, $\omega_{p3} = 0.78\pi$, και συχνότητες αποκοπής $\omega_{s1} = 0.27\pi$, $\omega_{s2} = 0.46\pi$ και $\omega_{s3} = 0.89\pi$. Επιπλέον, να ενισχύσετε την ενέργεια της συχνότητας διάβασης $\omega_{p2} = 0.34\pi$ μέχρι της συχνότητας διάβασης $\omega_{s2} = 0.46\pi$ κατά περίπου 10dB, και δείξτε την κρουστική απόκριση, την απόκριση μέτρου, φάσης καθώς και την συνάρτηση κόστους από την οποία προκύπτει το φίλτρο.

δ) Συγκρίνετε τα αποτελέσματα της συνάρτησης ***fircls()*** της MATLAB με το αντίστοιχο φίλτρο που προκύπτει από τη συνάρτηση ***firls()***, αν δοθούν τα ίδια ορίσματα για τους συντελεστές πλάτους και τις ζώνες διέλευσης / αποκοπής. Σημειώνεται πως θα χρειαστεί να επεκτείνετε κατά μία τιμή το διάνυσμα των συντελεστών πλάτους για τη ***firls()***.

<u>Λύση:</u>

α) Το ζητούμενο φίλτρο δίνει τις γραφικές παραστάσεις που φαίνονται στην επόμενη σελίδα:



β) Βλέποντας τις γραφικές παραστάσεις, μεταξύ άλλων, παρατηρούμε τα εξής:

- Το γράφημα της φάσης της απόκρισης συχνότητας δε διαφέρει από περίπτωση σε περίπτωση.
- Τα γραφήματα του μέτρου της απόκρισης συχνότητας και της κρουστικής απόκρισης του συμμετρικού Hamming παραθύρου ταυτίζονται με τα αντίστοιχα γραφήματα της περίπτωσης που δεν ορίζουμε παράθυρο. Αυτό συμβαίνει επειδή η ***fir2()***, όπως και η ***fir1()***, έχουν προεπιλέξει τέτοιο παράθυρο.
- Το τετραγωνικό παράθυρο δε παρέχει εξομάλυνση του σήματος, γεγονός που οδηγεί σε “κακά” χαρακτηριστικά απόκρισης συχνότητας, το οποίο φαίνεται από τις μικρές αλλαγές κλίσης που παρουσιάζονται στο μέτρο απόκρισης συχνότητας. Παρουσιάζει όμως και τη μεγαλύτερη απόσβεση των χαμηλών και υψηλών συχνοτήτων που περνάνε από το φίλτρο που σχεδιάσαμε.
- Τα περιοδικά παράθυρα Blackman και Hamming δεν καθρεπτίζονται ως προς τη μεσαία τιμή στο γράφημα της κρουστικής απόκρισής τους, αλλά τα μέτρα απόκρισης συχνότητάς τους μοιάζουν αρκετά με τα γραφήματα των αντίστοιχων συμμετρικών εκδοχών τους.
- Το παράθυρο Chebyshev με απόσβεση 42dB, λόγω της προσαρμογής που του έχει γίνει, έχει γράφημα μέτρου απόκρισης συχνότητας πολύ παρόμοιο με αυτό του περιοδικού Hamming και αμέσως μετά με του συμμετρικού Hamming.

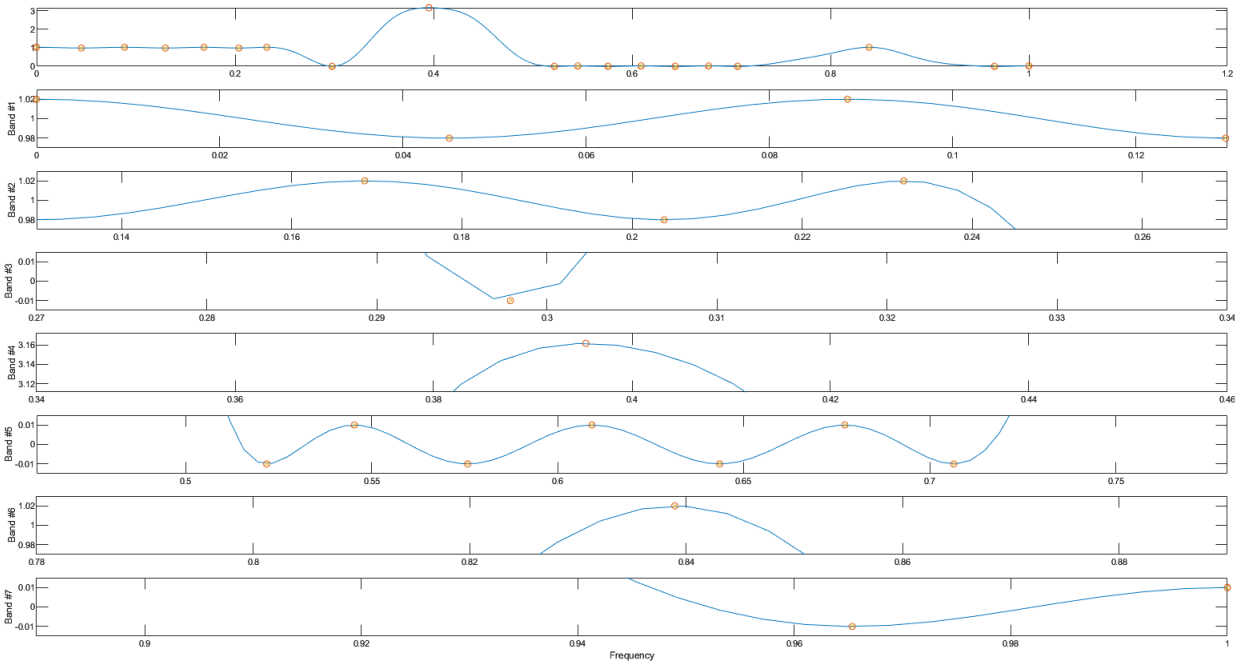
Ο κώδικας αυτού του ερωτήματος και του προηγούμενου δίνονται στις επόμενες σελίδες:

```
% Κανονικοποιημένες συχνότητες
frequencies = [0, 0.36, 1];

% Επιθυμητά πλάτη ανά συχνότητα και τάξη φίλτρου
magnitudes = [0, 1, 0];
n = 26;

% Συναρτήσεις Παραθύρου
windows = {'ones(n+1)', 'chebwin(n+1, 42)', ...
    'blackman(n+1, ''symmetric'')', 'blackman(n+1, ''periodic'')', ...
    'hamming(n+1, ''symmetric'')', 'hamming(n+1, ''periodic'')', ...
    'triang(n+1)', 'rectwin(n+1)'};

% Custom naming scheme for each window function
window_names = containers.Map(windows, {'No', '-42dB Chebyshev', ...
    'Symmetric Blackman', 'Periodic Blackman', ...
    'Symmetric Hamming', 'Periodic Hamming', ...
    'Triangular', 'Rectangular'});

% Βρόχος επανάληψης για τη χάραξη των γραφημάτων της κρουστικής απόκρισης,
% το μέτρο και τη φάση της απόκρισης συχνότητας για κάθε παράθυρο
for i = 1:numel(windows)
    window_function = windows{i};
    if strcmp(window_function, 'ones(n+1)')
        fir2_filter = fir2(n, frequencies, magnitudes); % no window
    else
        fir2_filter = fir2(n, frequencies, magnitudes, ...
            eval(window_function));
    end

    [H, w] = freqz(fir2_filter, 1, 1024);

    % Εύρεση ονόματος για τη συνάρτηση παραθύρου από λίστα
    window_func_name = window_names(window_function);

    figure(i);
    % Υπογράφημα κρουστικής απόκρισης
    subplot(131);
    impulse_response = impz(fir2_filter);
    stem(impulse_response);
    title(['Filter with ', char(window_func_name) ' Window:', ...
        ' Impulse Response']);
    xlabel('Sample Index');
    ylabel('Magnitude');
    xlim([0, n+2]);

    % Υπογράφημα μέτρου απόκρισης συχνότητας
    subplot(132);
    plot(w/pi, 20*log10(abs(H)));
    title(['Filter with ', char(window_func_name) ' Window:', ...
        ' Magnitude of Frequency Response']);
```

```
    xlabel('Normalized Frequency (\times\pi rad/sample)')
    ylabel('Magnitude (dB)')

    % Υπογράφημα φάσης απόκρισης συχνότητας
    subplot(133);
    plot(w/pi, angle(H));
    title(['Filter with ', char(window_func_name) ' Window:', ...
        ' Phase of Frequency Response']);
    xlabel('Normalized Frequency (\times\pi rad/sample)');
    ylabel('Phase (radians)');
end
```

γ) Η συνάρτηση ***fircls()*** της MATLAB, όταν χρησιμοποιηθεί, επιστρέφει ένα προειδοποιητικό μήνυμα που υποδεικνύει τον αριθμό των παραβιάσεων δεσμεύσεων, εφόσον υπάρχουν, και που συναντήθηκαν κατά τη διαδικασία σχεδιασμού του φίλτρου. Αυτά τα “bound violations” αναφέρονται σε περιπτώσεις όπου το σχεδιασμένο φίλτρο δεν πληρεί τους καθορισμένους περιορισμούς στην απόκριση πλάτους σε ορισμένες ζώνες συχνοτήτων. Αυτά τα άνω και κάτω όρια ορίζουν το αποδεκτό εύρος τιμών πλάτους σε κάθε συχνότητα στο διάνυσμα συχνοτήτων. Αν η απόκριση πλάτους του σχεδιασμένου φίλτρου υπερβαίνει αυτά τα άνω ή κάτω όρια σε οποιαδήποτε συχνότητα, θεωρείται ότι έχει "παραβίαση ορίου" σε αυτή τη συχνότητα. Αυτό το προειδοποιητικό μήνυμα μπορεί να καταγραφεί και να εμφανιστεί στην έξοδο για να ενημερώσει τον χρήστη σχετικά με πιθανά προβλήματα με τη συμμόρφωση του σχεδιασμένου φίλτρου με τους καθορισμένους περιορισμούς.

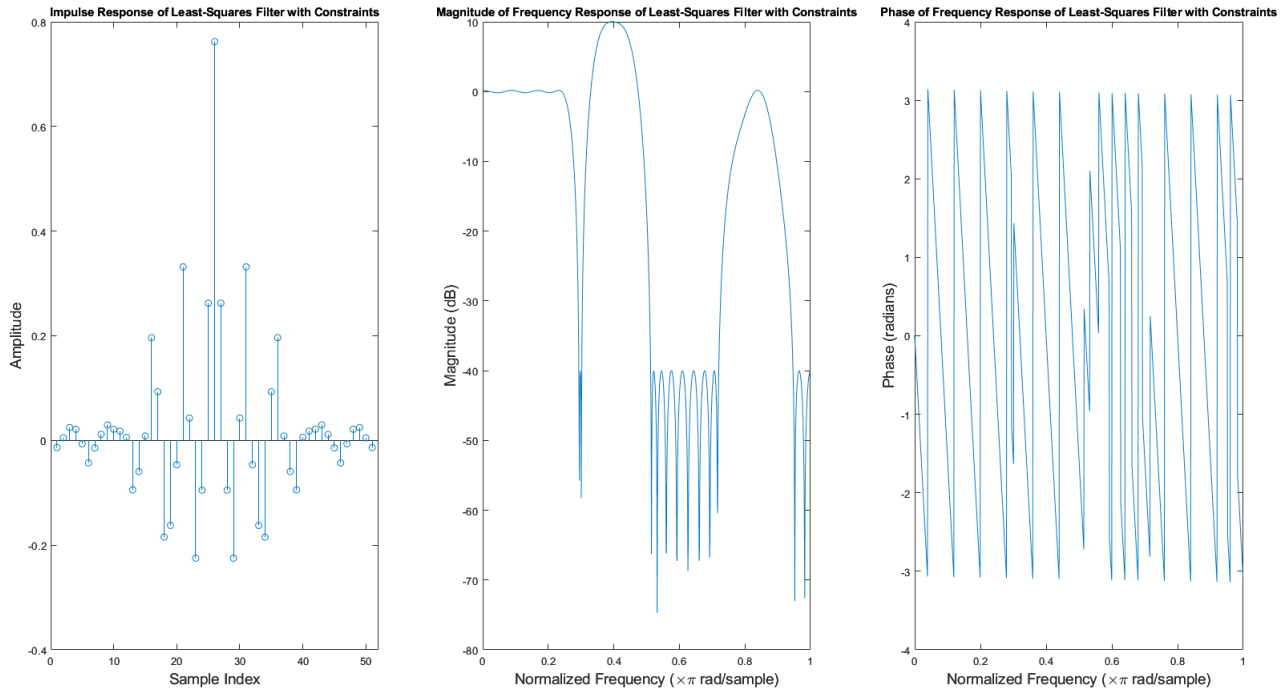
Για τη συγκεκριμένη υλοποίηση, το παραγόμενο γράφημα είναι:



Επιπλέον, στο command line μας βγάζει τα ακόλουθα μηνύματα:

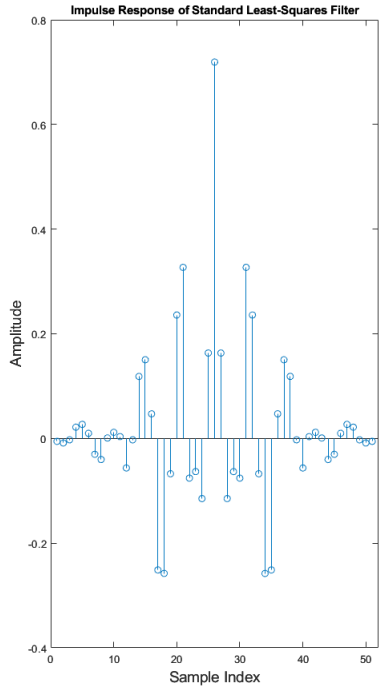
```
Bound Violation = 0.3918104664476
Bound Violation = 0.1118335031356
Bound Violation = 0.0226560488368
Bound Violation = 0.0056702489781
Bound Violation = 0.0005792610948
Bound Violation = 0.0001450674071
Bound Violation = 0.0000098477988
Bound Violation = 0.0000024604619
Bound Violation = 0.0000006094791
Bound Violation = 0.0000001524279
Bound Violation = 0.0000000304371
Bound Violation = 0.0000000076091
```

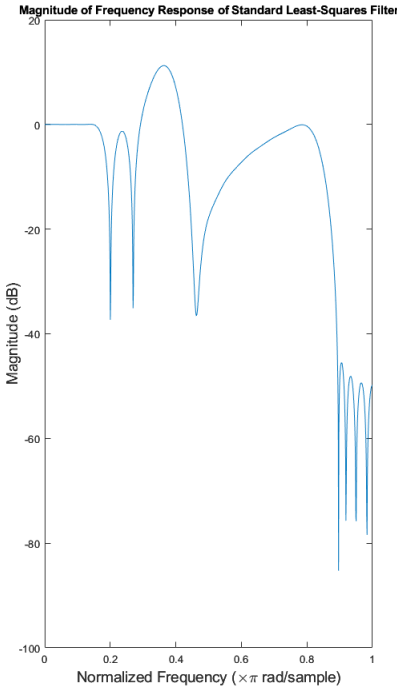
δ) Τα παραγόμενα γραφήματα των ***fircls()*** της MATLAB:

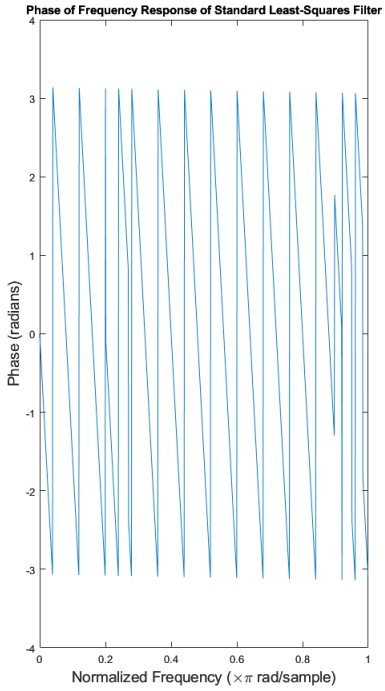


Βλέπουμε εδώ το μέτρο της απόκρισης συχνότητας μοιάζει αρκετά με αυτό που θέλουμε, αφού βελτιστοποιεί τους συντελεστές του μέτρου της απόκρισης συχνότητας του φίλτρου για να ανταποκρίνεται στους καθορισμένους περιορισμούς πλάτους, ενώ ελαχιστοποιεί τη συνολική απόκλιση από την επιθυμητή απόκριση του μέτρου της απόκρισης συχνότητας. Ωστόσο, αυτή η διαδικασία βελτιστοποίησης οδηγεί σε μη γραμμική απόκριση φάσης. Έτσι έχουν εισαχθεί παραμορφώσεις φάσης προκειμένου να επιτευχθεί η επιθυμητή μέτρο της απόκρισης συχνότητας εντός των καθορισμένων περιορισμών, οπότε η φάση της απόκρισης συχνότητας παρουσιάζει διακυμάνσεις, όπως γρήγορες αλλαγές ή κυματισμούς, ειδικά κοντά στις ζώνες μετάβασης όπου η απόκριση πλάτους αλλάζει γρήγορα.

Τα παραγόμενα γραφήματα των ***firls()*** της MATLAB:







Βλέπουμε εδώ το μέτρο της απόκρισης συχνότητας έχει τη πιο “γενική” μορφή που θέλαμε, αλλά οι ζώνες αποκοπής ειδικά που βρίσκονται ανάμεσα στις ζώνες διέλευσης δεν είναι τόσο “ισχυρές”, και συγκεκριμένα οι ω_{s1} = 0.27π και ω_{s2} = 0.46π. Ειδικά η ω_{s1} = 0.27π, η οποία αποτελεί μια στενή ζώνη αποκοπής, δεν λειτουργεί όπως θα επιθυμούσαμε, μιας και παρουσιάζει λοβό με κέρδος παρόμοιο αυτό της μονάδας, δηλαδή σαν αυτό που έχει η ζώνη διέλευσης ω_{p1} = 0.13π.

Συνεπώς, σε περιπτώσεις που θέλουμε πιο στενές ζώνες διέλευσης / αποκοπής, ή σχεδιάζουμε πολύπλοκα φίλτρα, μπορεί να είναι προτιμότερο να έχουμε κάποιες διακυμάνσεις στη φάση της απόκρισης συχνότητας, έστω κι αν εισάγουν χρονική καθυστέρηση που ποικίλλει ανάλογα με τη συχνότητα, για να έχουμε καλύτερο φιλτράρισμα ανεπιθύμητων συχνοτήτων.

Σημειώνεται βέβαια, πως σε ορισμένες εφαρμογές, όπως συστήματα ήχου ή επικοινωνίας, αυτές οι παραμορφώσεις φάσης μπορεί να είναι κρίσιμες και απαιτούν προσεκτική εξέταση κατά τη σχεδίαση του φίλτρου και την ανάλυση του συστήματος. Αν, για παράδειγμα, είναι επιθυμητή μια γραμμική φάση απόκρισης συχνότητας, άλλες μέθοδοι σχεδιασμού φίλτρων, όπως σχεδιασμοί ισοκυματισμού (equiripple) FIR, μπορεί να είναι πιο κατάλληλες.

Ο κώδικας αυτού του ερωτήματος και του προηγούμενου δίνονται στην επόμενη σελίδα:

```
% Κανονικοποιημένες συχνότητες
frequencies = [0, 0.13, 0.27 0.34, 0.46, 0.78, 0.89 1];

% Επιθυμητά πλάτη ανά συχνότητα
amplitude = [1, 1, 0, pi, 0, 1, 0];

% Τάξη φίλτρου και περιορισμοί μέγιστης απόκλισης στα όρια πλάτος και
n = 50;
upper_bound = [1.02, 1.02, 0.01, pi+0.02, 0.01, 1.02, 0.01];
lower_bound = [0.98, 0.98, -0.01, pi-0.02, -0.01, 0.98, -0.01];

% Σχεδίαση των φίλτρων least-squares με και χωρίς περιορισμούς μέσω βρόχου
% και του γραφήματος παραβίασης περιορισμών ανά ζώνη για το fircls()
for filter_type = 1:2
    if filter_type == 1
        filter_name = 'Least-Squares Filter with Constraints';
        filter = fircls(n, frequencies, amplitude, upper_bound, ...
            lower_bound, "both");
        figure(1);
    else
        filter_name = 'Standard Least-Squares Filter';
        filter = firls(n, frequencies, [amplitude, 0]);
    end

    % Δημιουργία αντίστοιχα αριθμημένου γραφήματος με βάση τη περίπτωση
    figure(filter_type + 1);

    % Σχεδίαση των υπογραφημάτων της κρουστικής απόκρισης και του μέτρου
    % και της φάσης της απόκρισης συχνότητας των φίλτρων fircls() και
    % firls() αντίστοιχα
    [H, w] = freqz(filter, 1, 1024);
    subplot(131);
    impulse_response = impz(filter);
    stem(impulse_response);
    title(['Impulse Response of ' filter_name]);
    xlabel('Sample Index');
    ylabel('Magnitude');
    xlim([0, n+2]);
    subplot(132);
    plot(w/pi, 20*log10(abs(H)))
    title(['Magnitude of Frequency Response of ' filter_name]);
    xlabel('Normalized Frequency (\times\pi rad/sample)');
    ylabel('Magnitude (dB)');
    subplot(133);
    plot(w/pi, angle(H));
    title(['Phase of Frequency Response of ' filter_name]);
    xlabel('Normalized Frequency (\times\pi rad/sample)');
    ylabel('Phase (radians)');
end
```

Θέματα: Εισαγωγή στη MATLAB για ΨΕΣ

Θέμα 3ο) Μέθοδοι Αποθορυβοποίησης Σήματος με FIR και IIR Φίλτρα

Κατά τη λήψη κάποιου σήματος, υπάρχει πιθανότητα να μολυνθεί από διάφορα είδη θορύβου, το οποίο μπορεί να είναι ο τυχαίος λευκός θόρυβος, κάποια σταθερή συχνότητα ή ακόμα και κάποιο άλλο ανεπιθύμητο σήμα το οποίο αποκρύπτει την πληροφορία που θέλουμε να εξετάσουμε. Τέτοια είδη θορύβου μπορεί να υποβαθμίσει την ποιότητα ενός σήματος, καθιστώντας πιο δύσκολη την ανάλυση, την ερμηνεία ή τη χρήση του. Συνεπώς, σας δίνονται δύο (2) σήματα δειγματοληπτημένα με συχνότητα f_s = 44.1kHz, το πρώτο είναι το κανονικό, το οποίο αποτελεί ένα μήνυμα σε κώδικα Morse συχνότητας μεταξύ 600Hz και 1.1kHz, και ένα άλλο το οποίο έχει μολυνθεί με θόρυβο.
α) Χρησιμοποιώντας τη συνάρτηση ***sound()*** της MATLAB, ακούστε το θορυβοποιημένο και καθαρό σήμα που σας δίνονται. Με τι είδος / είδη θορύβου έχει μολυνθεί το σήμα;
β) Αφού βρείτε το είδος / τα είδη του θορύβου, να δείξετε που γίνεται η κατανομή της ενέργειας του σήματος και που του θορύβου και ξεκινήστε τη διαδικασία φιλτραρίσματος, ώστε να καταλήξετε σε ένα όσο το δυνατόν πιο καθαρό σήμα.
γ) Χρησιμοποιώντας πάλι τη συνάρτηση ***sound()*** της MATLAB, ακούστε το αποθορυβοποιημένο και το καθαρό σήμα που σας δίνονται. Διαφέρουν με κάποιο τρόπο;

Λύση:

α) Ακούγοντας το θορυβοποιημένο σήμα, μπορούμε να διακρίνουμε δύο (2) είδη θορύβου, τα οποία είναι ο λευκός θόρυβος, μαζί με κάποιο σήμα σταθερής συχνότητας που έχει μεγάλη ενέργεια.
β) Η κατανομή του λευκού θορύβου, εξ ορισμού, βρίσκεται σε όλο το συχνοτικό πεδίο, μιας και ακολουθεί την ομοιόμορφη κατανομή. Διακρίνουμε δύο (2) αιχμές, η ψηλότερη στα 850Hz και η αμέσως μικρότερη στα 900Hz, τα οποία αντιστοιχούν με κάποιο τρόπο στο σταθερό σήμα και στον κώδικα Morse που θέλουμε να αποθορυβοποιήσουμε. Συνεπώς, για την καλύτερο φιλτράρισμα του σήματος, θα το κάνουμε σε τρία στάδια, τα οποία είναι:

- Εφαρμογή ζωνοπερατού (bandpass) φίλτρου μεταξύ 600Hz και 1.1kHz για να μπορούμε να διακρίνουμε καλύτερα ποιο είναι το σταθερό σήμα και ποιο το επιθυμητό.
- Εφαρμογή στενού ζωνοφραχτού (bandstop) φίλτρου για την αφαίρεση του σταθερού σήματος.
- Εφαρμογή στενού ζωνοπερατού φίλτρου για την αφαίρεση του υπόλοιπου θορύβου, ώστε να κρατήσουμε όσο γίνεται μόνο την συχνότητα στην οποία βρίσκεται το αρχικό σήμα.

Για να καταφέρουμε τα παραπάνω, θα ορίσουμε τα ακόλουθα φίλτρα:

- Ζωνοπερατό φίλτρο με χρήση της μεθόδου ***fir2()*** της MATLAB. Με αντίστοιχο τρόπο μπορεί να γίνει και με τη μέθοδο fircls() ή κάποια άλλη αντίστοιχη συνάρτηση.
- Στενό ζωνοφραχτό Butterworth φίλτρο.
- Στενό ζωνοπερατό Butterworth φίλτρο.

Τα Butterworth φίλτρα, όπως γνωρίζουμε, είναι IIR φίλτρα, δηλαδή η κρουστική απόκρισή τους έχουν άπειρο μήκος. Είναι γνωστά για τη πιο επίπεδη απόκριση συχνότητας στη ζώνη διέλευσης σε σχέση με άλλα φίλτρα αντίστοιχης απόσβεσης, που σημαίνει ότι το μέτρο της απόκρισης του φίλτρου παραμένει σχεδόν σταθερό σε όλη τη ζώνη διέλευσης. Στο Signal Processing Toolbox της MATLAB υπάρχει ένα ιδιαίτερα χρήσιμο εργαλείο, το οποίο ονομάζεται Filter Designer.

Αφού ορίσουμε τα χαρακτηριστικά του φίλτρου που θέλουμε, θα προτιμήσουμε εδώ να το εξάγουμε ως κώδικα MATLAB, ώστε εύκολα μετά να μπορούμε να μετονομάσουμε χαρακτηριστικά του, αλλάξουμε τιμές ή γενικώς να το επεξεργαστούμε με οποιονδήποτε τρόπο. Για να κάνουμε αυτό, θα πάμε στο File → Generate MATLAB Code → Filter Design Function, όπου μετά θα ανοίξει ένα παράθυρο για να ορίσουμε το σημείο αποθήκευσης και το όνομα του φίλτρου που δημιουργήσαμε, το οποίο αποθηκεύεται ως .m αρχείο, όπως και κάθε άλλο αρχείο κώδικα MATLAB. Στην περίπτωση που θέλουμε να εξάγουμε το φίλτρο μας ως .mat αρχείο, καλή ιδέα θα ήταν να αποθηκεύσουμε το συγκεκριμένο session, μιας και δεν είναι δυνατή η επεξεργασία ενός .mat αρχείου, αφού καθώς περιέχει δυαδικά δεδομένα μη αναγνώσιμης μορφής από άνθρωπο. Για την αποθήκευση του session, File → Save Session As… και επιλέγουμε αντίστοιχα το σημείο αποθήκευσης και το όνομα του session που δημιουργήσαμε, το οποίο αποθηκεύεται ως .fda αρχείο.

γ) Παρατηρούμε, ενώ η ενέργεια του αρχικού σήματος κατανέμεται σε μία πολύ στενή ζώνη συχνοτήτων, δε γίνεται το φιλτραρισμένο σήμα μας να ακούγεται σαν το αρχικό. Αν και έχουμε μειώσει σημαντικά τον θόρυβο, το πλάτος του σήματος έχει μειωθεί, το οποίο είναι αναμενόμενο, αλλά πλέον υπάρχει και μια μικρή ηχώ, που αποτελεί τα υπόλοιπα του θορύβου που μόλυνε το σήμα μας αρχικά. Επιπλέον, ο λευκός θόρυβος που υπήρχε στην ίδια ζώνη με το αρχικό μας σήμα δεν είναι δυνατόν να αφαιρεθεί ποτέ, μιας και αυτό προϋποθέτει και την αφαίρεση του σήματός μας.

Ο κώδικας MATLAB συνολικά της άσκησης αυτής βρίσκεται στις επόμενες σελίδες:

Η κυρίως συνάρτησή μας:

```
%======================== Φόρτωση Αρχείων & Φίλτρων ========================%

[audio, fs] = audioread('morse_code.wav');
load morse_code_noisy.mat
NumFFT = 4096;
F = linspace(-fs/2, fs/2, NumFFT);

% Σχεδιασμός φίλτρου με τη μέθοδο ορισμού συντελεστών απόκρισης συχνότητας
fs = 44100;     % Συχνότητα δειγματοληψίας
f1 = 600;       % Όριο αποκοπής χαμηλών συχνοτήτων (lower frequency cutoff)
f2 = 1100;      % Όριο αποκοπής υψηλών συχνοτήτων (upper frequency cutoff)
n = 20;         % Τάξη φίλτρου

% Κανονικοποίηση συχνοτήτων
frequencies = [0, f1-100, f1, f2, f2+100, fs/2] ./ (fs/2);

% Επιθυμητά πλάτη ανά συχνότητα
magnitudes = [0, 0, 3, 3, 0, 0];

% Φίλτρα
filter_1 = fir2(n, frequencies, magnitudes);
filter_2 = Butterworth_Bandstop();
filter_3 = Butterworth_Bandpass();


%============================ Γραφήματα Φίλτρων ============================%

% Σχεδίαση γραφημάτων κρουστικής απόκρισης, μέτρου και φάσης της απόκρισης
% συχνότητας του πρώτου φίλτρου
[H1, w] = freqz(filter_1, 1, NumFFT, fs);
figure (1);
subplot(131);
impulse_response_1 = impz(filter_1);
stem(impulse_response_1);
title('First Filter''s Impulse Response');
xlabel('Sample Index');
ylabel('Amplitude');
xlim([0, n+2]);
subplot(132);
plot(w/pi, 20*log10(abs(H1)));
title('First Filter''s Magnitude of Frequency Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
subplot(133);
plot(w/pi, angle(H1));
title('First Filter''s Phase of Frequency Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');
```

```
% Σχεδίαση γραφημάτων κρουστικής απόκρισης, μέτρου και φάσης της απόκρισης
% συχνότητας του δεύτερου φίλτρου
[H2, w] = freqz(filter_2, NumFFT, fs);
figure (2);
subplot(131);
impulse_response_2 = impz(filter_2);
plot(impulse_response_2);
title('Second Filter''s Impulse Response');
xlabel('Sample Index');
ylabel('Amplitude');
subplot(132);
plot(w/pi, 20*log10(abs(H2)));
title('Second Filter''s Magnitude of Frequency Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
subplot(133);
plot(w/pi, angle(H2));
title('Second Filter''s Phase of Frequency Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');


% Σχεδίαση γραφημάτων κρουστικής απόκρισης, μέτρου και φάσης της απόκρισης
% συχνότητας του τρίτου φίλτρου
[H3, w] = freqz(filter_3, NumFFT, fs);
figure (3);
subplot(131);
impulse_response_3 = impz(filter_3);
plot(impulse_response_3);
title('Third Filter''s Impulse Response');
xlabel('Sample Index');
ylabel('Amplitude');
subplot(132);
plot(w/pi, 20*log10(abs(H3)));
title('Third Filter''s Frequency Response Magnitude');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
subplot(133);
plot(w/pi, angle(H3));
title('Third Filter''s Frequency Response Phase');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)');


%============================ Γραφήματα Σημάτων ============================%

f1_output = filtfilt(filter_1, 1, noisy_audio);
f2_output = filter(filter_2, f1_output);
f3_output = filter(filter_3, f2_output);
signals = {audio, noisy_audio, f1_output, f2_output, f3_output};
```

```
titles = {'Audio Signal', 'Noisy Audio Signal', 'First Filter Output', ...
    'Second Filter Output', 'Third Filter Output'};

% Σχεδίαση γραφημάτων κατανομής ενέργειας του κάθε παραγόμενου σήματος
% ανά συχνότητα (μετασχηματισμός Fourier) και ανά δείγμα
for i = 1:length(signals)
    figure(i+3);
    subplot(121);
    plot(F, abs(fftshift(fft(signals{i}, NumFFT))));
    title(['Fourier Transform of ', titles{i}]);
    xlabel('Frequency');
    ylabel('Amplitude');
    subplot(122);
    plot(signals{i});
    title(['Waveform of ', titles{i}]);
    xlabel('Samples');
    ylabel('Amplitude');
end


%========== Σήματα Ήχου (αρχικό, θορυβοποιημένο, φιλτραρισμένα) ==========%

sound(audio, fs);
pause(18);
sound(noisy_audio, fs);
pause(18);
sound(f1_output, fs);
pause(18);
sound(f2_output, fs);
pause(18);
sound(f3_output, fs);
```

Η παραγόμενη συνάρτηση από το Filter Designer για το Bandstop Butterworth φίλτρο:

```
function Hd = Butterworth_Bandstop
% BUTTERWORTH_BANDSTOP Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.13 and Signal Processing Toolbox 9.1.
% Butterworth Bandstop filter designed using FDESIGN.BANDSTOP.

% All frequency values are in Hz.
Fs = 44100;  % Sampling Frequency

N   = 10;   % Order
Fc1 = 895;  % First Cutoff Frequency
Fc2 = 905;  % Second Cutoff Frequency

% Construct an FDESIGN object and call its BUTTER method.
h  = fdesign.bandstop('N,F3dB1,F3dB2', N, Fc1, Fc2, Fs);
Hd = design(h, 'butter');

% [EOF]
```

Η παραγόμενη συνάρτηση από το Filter Designer για το Bandpass Butterworth φίλτρο:

```
function Hd = Butterworth_Bandpass
% BUTTERWORTH_BANDPASS Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.13 and Signal Processing Toolbox 9.1.
% Butterworth Bandpass filter designed using FDESIGN.BANDPASS.

% All frequency values are in Hz.
Fs = 44100;  % Sampling Frequency

N   = 10;   % Order
Fc1 = 800;  % First Cutoff Frequency
Fc2 = 900;  % Second Cutoff Frequency

% Construct an FDESIGN object and call its BUTTER method.
h  = fdesign.bandstop('N,F3dB1,F3dB2', N, Fc1, Fc2, Fs);
Hd = design(h, 'butter');

% [EOF]
```