Εργαστηριακή Άσκηση 2
Σχεδιασμός/υλοποίηση ψηφιακών φίλτρων FIR με το MATLAB®

Σκοπός της δεύτερης σειράς ασκήσεων είναι η εξοικείωση με τις συναρτήσεις σχεδιασμού φίλτρων πεπερασμένης κρουστικής απόκρισης (FIR) και την υλοποίησή τους στο MATLAB. **Προτού ξεκινήσετε την άσκηση θα πρέπει να μελετήσετε με προσοχή το Κεφάλαιο 1 και, ειδικότερα, την παράγραφο 1.3 του τεύχους του μαθήματος[1].** Το MATLAB (www.mathworks.com) είναι ένα διαδραστικό εμπορικό πρόγραμμα (Windows, Linux, Unix) με το οποίο μπορείτε να κάνετε εύκολα αριθμητικές πράξεις με πίνακες. Μπορεί να το έχετε εγκατεστημένο τοπικά, στον προσωπικό σας υπολογιστή ή να εργάζεστε σε κάποιο Εργαστήριο Προσωπικών Υπολογιστών (ΕΠΥ) της Σχολής σας που διαθέτει το συγκεκριμένο λογισμικό.

Για να εργαστείτε στην πλατφόρμα MATLAB από το σπίτι σας, με τις άδειες που διαθέτει το Ίδρυμα, δείτε σχετικό έγγραφο *Χρήση του MATLAB από το σπίτι*.

Μέρος 1: Εισαγωγή

Στο MATLAB οι συναρτήσεις `fft` και `ifft` υποθέτουν ζεύγος μετασχηματισμού Fourier $x(t)$ και $X(f)$ υπολογισμένων σε μη αρνητικά διαστήματα t=[0:N-1]t_s και f=[0:N-1]f_o. Όπως έχετε ήδη δει στην Εργαστηριακή άσκηση 1, το άνω μισό μέρος του διαστήματος συχνοτήτων αντιστοιχεί στις αρνητικές συχνότητες του σήματος, όταν υπολογίζουμε το $X(f)$ με τη βοήθεια της συνάρτησης `fft`. Ακριβώς το ίδιο ισχύει και για το άνω μισό μέρος του χρονικού διαστήματος, όταν το σήμα $x(t)$ προκύπτει από τον αντίστροφο μετασχηματισμό Fourier μέσω της `ifft`.

Το MATLAB διαθέτει τη συνάρτηση `fftshift` για να ολισθήσει κυκλικά τις τιμές του σήματος ή του μετασχηματισμού Fourier, ώστε να αντιστοιχούν σε κεντραρισμένα στο μηδέν αμφίπλευρα διαστήματα, δηλαδή, στις χρονικές στιγμές t_b=[-ceil((N-1)/2): floor((N-1)/2)]t_s ή στις συχνότητες f_b=[–ceil((N-1)/2): floor((N-1)/2)]f_o. Με τον τρόπο αυτό μπορούμε να παράγουμε τα $x_b(t)$ και $X_b(f)$ που αντιστοιχούν στην αμφίπλευρη αναπαράσταση του σήματος και του μετασχηματισμού Fourier.

Για να κατανοήσετε τα ανωτέρω θεωρείστε το διάνυσμα [1 2 3 4] ως το αποτέλεσμα του FFT μήκους 4. Τότε, το πρώτο στοιχείο (1) είναι ο όρος dc, το τρίτο στοιχείο (3) είναι το σημείο στο μισό της συχνότητας δειγματοληψίας $f_s/2$, που μπορεί να εκληφθεί ότι αντιστοιχεί είτε στην $-f_s/2$ είτε στην $+f_s/2$. Τα στοιχεία 2 και 4 αντιστοιχούν στις συχνότητες $+f_s/4$ και $-f_s/4$. Εφαρμόζοντας την `fftshift`, το στοιχείο 3 εμφανίζεται πρώτο, που σημαίνει ότι στο MATLAB αντιστοιχεί στην αρνητική συχνότητα $-f_s/2$, το επόμενο στοιχείο 4 αντιστοιχεί στη συχνότητα $-f_s/4$ ακολουθούμενο από το dc και τη συχνότητα $+f_s/4$. Για ένα μετασχηματισμό περιττού μήκους, δεν υφίσταται σημείο για το $\pm f_s/2$. Έτσι για το διάνυσμα [1 2 3], η εφαρμογή της `fftshift` θα δώσει τα στοιχεία που αντιστοιχούν στις συχνότητες $-f_s/3$, 0, $+f_s/3$.

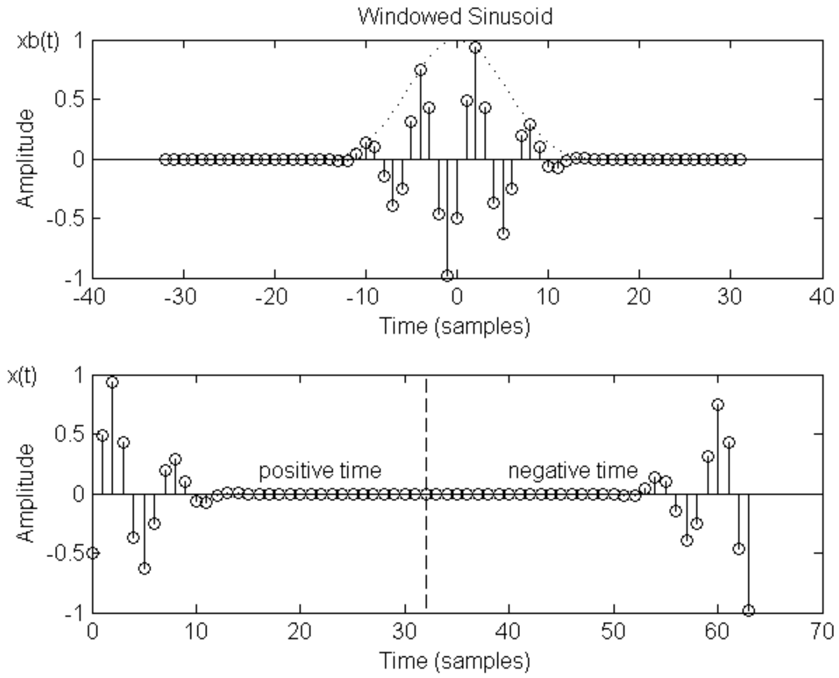
Εκτός του ότι παράγουν εξόδους με τις αρνητικές συχνότητες ή χρόνους στο άνω μισό του διανύσματος, αμφότερες οι συναρτήσεις `fft` και `ifft` αναμένουν ως είσοδο διάνυσμα με την ίδια μορφή, αφού προφανώς ισχύουν οι ταυτότητες

`h == ifft(fft(h))` και `H == fft(ifft(H))`

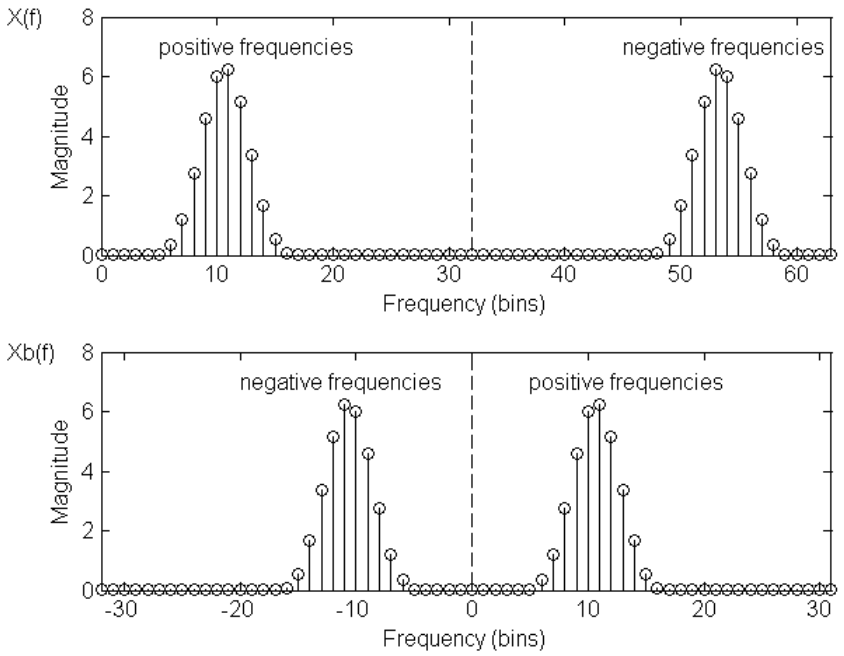
Η πρώτη υποδεικνύει ότι η είσοδος της `ifft` πρέπει να είναι αντεστραμμένη, όπως την παράγει η `fft`, και η δεύτερη ότι η είσοδος της `fft` πρέπει να είναι αντεστραμμένη, όπως την παράγει η `ifft`.

Στο επόμενο σχήμα βλέπετε παραστατικά ένα ημιτονικό σήμα που έχει πολλαπλασιασθεί με παράθυρο Blackman τόσο στην αμφίπλευρη, όσο και την μονόπλευρη αναπαράστασή του.

[1] *ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ – Συνοπτική Θεωρία και Εργαστήριο, Ν. Μήτρου, ΚΑΛΛΙΠΟΣ 2016,* *http://hdl.handle.net/11419/6044*



Οι αντίστοιχοι μετασχηματισμοί Fourier, σε αμφίπλευρη και μονόπλευρη αναπαράσταση, φαίνονται στο επόμενο σχήμα:



Όταν τα $x(t)$ και $X(f)$ παράγονται από το MATLAB, δεν χρειάζεται κάποια ιδιαίτερη προσοχή, πλην της κυκλικής ολίσθησης σε περίπτωση που θέλουμε π.χ. να σχεδιάσουμε το αμφίπλευρο φάσμα ή σήμα. Όταν όμως ένα εκ των $x(t)$ ή $X(f)$ ορίζεται από τον χρήστη απαιτείται περισσότερη προσοχή, διότι, συνήθως χρησιμοποιούνται τα αμφίπλευρα σήματα ή φάσματα. Μπορείτε να μεταβείτε από τη μία αναπαράσταση στην άλλη ως εξής:

`x = ifftshift(xb), X = fft(x), Xb = fftshift(X)`, εάν ξεκινάτε από αμφίπλευρο σήμα και θέλετε να καταλήξετε σε αμφίπλευρο φάσμα, και

`X = ifftshift(Xb), x = ifft(X), xb = fftshift(x)`, εάν ξεκινάτε από αμφίπλευρο φάσμα και θέλετε να καταλήξετε σε αμφίπλευρο σήμα, όπου η συνάρτηση `ifftshift` του MATLAB εκτελεί την αντίστροφη λειτουργία της `fftshift`. Όταν το *N* είναι άρτιο, οι `fftshift` και `ifftshift` δίνουν το ίδιο αποτέλεσμα. Όταν όμως το *N* είναι περιττό αυτό δεν ισχύει και χρειάζεται προσοχή στη χρήση τους. Στην πράξη, η προσεκτική εφαρμογή των ανωτέρω έχει σημασία όταν υπολογίζεται η **φάση** του φάσματος. Το πλάτος του φάσματος δεν επηρεάζεται από την κυκλική ολίσθηση των στοιχείων που προκαλούν οι `fftshift` και `ifftshift` (δείτε ιδιότητες DFT).

Εξάσκηση

Δοκιμάστε στο παράθυρο εντολών τα ακόλουθα προκειμένου να εμπεδώσετε τη χρήση των συναρτήσεων `fftshift` και `ifftshift`.

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

Ερώτηση 1: Ποιο εκ των διανυσμάτων Y και Z ισούται με το X; Γράψτε την απάντησή σας σε ένα αρχείο κειμένου lab2_nnnnn.txt, όπου nnnnn τα πέντε τελευταία νούμερα του αριθμού μητρώου σας, χρησιμοποιώντας το Notepad από το μενού των Windows (*Start* → *Programs* → *Accessories* → *Notepad*) και αποθηκεύστε το στον φάκελο My Documents. Θα υποβάλετε το αρχείο αυτό ηλεκτρονικά στο τέλος, αφού απαντήσετε και τις επόμενες ερωτήσεις, οπότε μπορείτε να τα αφήσετε ανοικτό.

Ερώτηση 2: Επαναλάβατε με `X=[-1:2]`. Τι παρατηρείτε; Γράψτε την απάντησή σας στο αρχείο κειμένου lab2_nnnnn.txt.

Δοκιμάστε στο παράθυρο εντολών τα ακόλουθα δύο παραδείγματα για να εμπεδώσετε τη χρήση των συναρτήσεων `fftshift` και `ifftshift` σε συνδυασμό με τις `fft` και `ifft`.

```
>> close all; clear all;clc;
>> xb=[1 2 3 4 5 4 3 2 1]  % πραγματικό σήμα με άρτια συμμετρία
>> figure; subplot (2,1,1); plot([-4:4],xb); ylabel('xb');
>> x=ifftshift(xb)         % το σήμα με τις αρνητικές συνιστώσες στο άνω μέρος
>> X=fft(x)                % FFT
>> Xb=fftshift(X)          % το φάσμα με τη dc συνιστώσα στο κέντρο, πραγματικές
>>                         % τιμές με άρτια συμμετρία όπως αναμένεται
>> subplot (2,1,2); plot([-4:4],Xb);ylabel('Xb');
```

```
>> 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');
```

Ερώτηση 3: Τροποποιείστε το προηγούμενο παράδειγμα ώστε να ξεκινήσετε απευθείας με τον ορισμό του φάσματος του βαθυπερατού σήματος X όπως το αναμένει η `ifft`. Γράψτε την απάντησή σας στο αρχείο κειμένου lab2_nnnnn.txt.

Μέρος 2: Σχεδιασμός φίλτρων

Θα ασχοληθείτε με το παράδειγμα 1.2 της παραγράφου 1.5 του τεύχους Μαθήματος. Το παράδειγμα αυτό παρουσιάζει δύο εναλλακτικούς τρόπους σχεδιασμού FIR φίλτρων:
α) τη μέθοδο των παραθύρων και
β) τη μέθοδο των ισοϋψών κυματώσεων
τους οποίους εφαρμόζει για την περίπτωση του σχεδιασμού ενός βαθυπερατού φίλτρου.

Για την εκτέλεση του παραδείγματος 1.2, αντιγράψτε τον Κώδικα 1.3 από το τεύχος σε ένα καινούριο αρχείο M-file και αποθηκεύστε το στο φάκελο εργασίας σας (My Documents\MATLAB). Χρησιμοποιήστε για το αρχείο το όνομα lab2_1_nnnnn.m, όπου nnnnn τα πέντε τελευταία νούμερα του αριθμού μητρώου σας. Επίσης, κατεβάστε από την ιστοσελίδα του μαθήματος το αρχείο sima (sima.mat) και ομοίως αποθηκεύστε το στο φάκελο εργασίας σας.

Τα αρχεία με κατάληξη .mat χρησιμοποιούνται στο MATLAB για την αποθήκευση μεταβλητών του χώρου εργασίας (workspace) και δεν πρέπει να συγχέονται με τα M-files, που περιέχουν εντολές MATLAB. Η φόρτωση ενός .mat αρχείου πραγματοποιείται με την εντολή `load 'filename'`, όπου `'filename'` το όνομα του αρχείου χωρίς την κατάληξη .mat ή εναλλακτικά από το *Open*[2] στο tab *Home*. Το sima.mat περιέχει δύο μεταβλητές: το διάνυσμα s που περιέχει ένα σήμα σόναρ, το φάσμα του οποίου εκτείνεται μέχρι περίπου τα 4 KHz, και την τιμή της μεταβλητής F_s, που είναι η συχνότητα με την οποία έγινε η δειγματοληψία του σήματος σόναρ. Παρατηρήστε ότι, στον κώδικα του παραδείγματος, προηγείται η φόρτωση του sima.mat (γραμμή 6), γεγονός που επιτρέπει τη χρήση των μεταβλητών s, F_s στο υπόλοιπο τμήμα του (γραμμές 7-44).

Η μέθοδος των παραθύρων

Η μέθοδος των παραθύρων εφαρμόζεται στις γραμμές 8-38 του κώδικα. Πρώτο βήμα αποτελεί ο ορισμός της απόκρισης συχνότητας H ενός ιδανικού βαθυπερατού φίλτρου με συχνότητα αποκοπής $F_s/8$ (γραμμή 9), μέσω ενός διανύσματος μήκους $N=F_s$, γεγονός που οδηγεί σε ανάλυση συχνότητας $f_o = F_s /N = 1$ Hz. Το διάνυσμα H αποτελείται από μια αλληλουχία μονάδων και μηδενικών που δημιουργούνται από την κλήση των συναρτήσεων `ones` και `zeros`, αντίστοιχα. Για περισσότερες πληροφορίες σχετικά με αυτές τις συναρτήσεις μπορείτε να συμβουλευτείτε την τεκμηρίωση του MATLAB, πληκτρολογώντας `doc 'function_name'` στο παράθυρο εντολών, όπου `'function_name'` το όνομα της συνάρτησης. Τα $F_s/8$ πρώτα στοιχεία του H, που αντιστοιχούν στις συχνότητες $[0,F_s/8)$, είναι μονάδες, ακολουθούν $3F_s/4$ μηδενικά και άλλες $F_s/8$ μονάδες που αντιστοιχούν στη ζώνη συχνοτήτων[3] $[–F_s/8, 0)$.

Επόμενο βήμα είναι ο υπολογισμός του αντίστροφου διακριτού μετασχηματισμού Fourier (IDFT), που υπολογίζεται από τη συνάρτηση `ifft` του MATLAB (γραμμή 12). Ακολουθεί μια αναδιάταξη του αποτελέσματος του αντίστροφου DFT (γραμμές 13-14), που ισοδυναμεί με ολίσθηση της κρουστικής απόκρισης του φίλτρου κατά το μισό της μήκος. Στη συνέχεια, η κρουστική απόκριση h περικόπτεται σε μήκη 32+1, 64+1 και 128+1 δειγμάτων (γραμμές 15-17). Με τη βοήθεια του wvtool (Window Visualization Tool), απεικονίζονται στο ίδιο διάγραμμα η κρουστική απόκριση και η απόκριση συχνότητας του βαθυπερατού φίλτρου και για τα 3 παραπάνω μήκη. Το wvtool είναι ένα παρόμοιο εργαλείο με το wintool και χρησιμεύει για την οπτικοποίηση παραθύρων στο πεδίο του χρόνου και της συχνότητας. Παρατηρήστε ότι όσο μεγαλύτερο το μήκος του φίλτρου τόσο μικρότεροι είναι οι πλευρικοί λοβοί στην απόκριση συχνότητας.

Για να μειωθούν ακόμα περισσότερο αυτοί οι πλευρικοί λοβοί και οι επιπτώσεις του ορθογωνικού παραθύρου, κατασκευάζονται παράθυρα Hamming και Kaiser (μέσω των συναρτήσεων `hamming`

[2] Στην έκδοση R2011b μενού *File* → *Open*

[3] Όπως είδατε και στην Εργαστηριακή Άσκηση 1, η δειγματοληψία στο πεδίο του χρόνου προκαλεί περιοδική επανάληψη του φάσματος του σήματος με περίοδο F_s

και `kaiser` αντίστοιχα, γραμμές 24-25) τα οποία εφαρμόζονται στο φίλτρο μήκους 64+1 σημείων *h64* (γραμμές 27-30). Παρατηρήστε μέσω του wvtool (γραμμή 31) τη σαφώς χαμηλότερη στάθμη των πλευρικών λοβών στην απόκριση συχνότητας των παραθύρων Hamming και Kaiser σε σχέση με το ορθογωνικό. Τέλος, το σήμα *s* φιλτράρεται με καθένα από τα τρία φίλτρα (ορθογωνικό, Hamming, Kaiser), χρησιμοποιώντας τη συνάρτηση `conv` που υπολογίζει τη συνέλιξη μεταξύ του σήματος και της κρουστικής απόκρισης του εκάστοτε φίλτρου (γραμμές 33-38). Παράλληλα, υπολογίζεται και σχεδιάζεται η πυκνότητα φάσματος ισχύος κατά Welch με τη βοήθεια της συνάρτησης `pwelch`. Το αποτέλεσμα δείχνει εμφανώς την ραγδαία μείωση της φασματικής ισχύος για συχνότητες άνω της $F_s/8$.

Μέθοδος ισοϋψών κυματώσεων

Η μέθοδος των ισοϋψών κυματώσεων εφαρμόζεται στις γραμμές 40-44 του κώδικα για την κατασκευή ενός φίλτρου Parks-McClellan. Η κατασκευή του φίλτρου πραγματοποιείται με την κλήση της συνάρτησης `firpm` (γραμμή 41). Για τον ορισμό των παραμέτρων εισόδου της `firpm` συμβουλευτείτε την τεκμηρίωση του MATLAB. Η εφαρμογή του φίλτρου στο σήμα γίνεται όπως και στην προηγούμενη μέθοδο, με χρήση της συνάρτησης `conv` (γραμμή 43).

<u>Πειραματισθείτε</u>

1. Τροποποιείστε τον κώδικα στην γραμμή 14 ώστε να επιτύχετε το ίδιο αποτέλεσμα με τη χρήση μιας εκ των συναρτήσεων `ifftshift` ή `fftshift`.
2. Τροποποιείστε τον κώδικα ώστε να χρησιμοποιηθεί βαθυπερατό φίλτρο μήκους 256+1 αντί του 64+1. Σχεδιάστε την απόκριση συχνότητας του φίλτρου με σχεδιασμό παραθύρου, όπως στο παράδειγμα, για δύο περιπτώσεις: ορθογωνικού και Hamming.
3. Σχεδιάστε φίλτρο Parks-McClellan μήκους 128+1, με τις ίδιες οριακές συχνότητες, όπως στο παράδειγμα (0.1, 0.15). Σχεδιάστε την απόκριση συχνότητας και αυτού του φίλτρου και σχολιάστε τις διαφορές από το ίδιο φίλτρο μήκους 64+1 (του παραδείγματος 1.2). Γράψτε την απάντησή σας στο αρχείο κειμένου lab2_nnnnn.txt.
4. Αλλάξτε τις οριακές συχνότητες του φίλτρου του ερωτήματος 3 σε (0.11, 0.12) και συγκρίνετε τις αποκρίσεις συχνότητας των δύο φίλτρων. Γράψτε τα σχόλιά σας στο αρχείο κειμένου lab2_nnnnn.txt.
5. Αντικαταστήστε το σήμα *s* με άθροισμα τεσσάρων ημιτονικών συναρτήσεων μοναδιαίου πλάτους, συχνότητας 800, 1000, 2000 και 3000 Hz και διάρκειας 1.0 sec. Διατηρήστε την ίδια $F_s = 8192$ Hz.

Μέρος 3: Εφαρμογή Α

Εκτελέστε την άσκηση 1.2 της παραγράφου 1.6 του τεύχους του μαθήματος, μεταβάλλοντας κατάλληλα τον κώδικα του παραδείγματος 1.2 (που έχετε αποθηκεύσει στο αρχείο lab2_1_nnnnn.m). Σκοπός της άσκησης είναι ο σχεδιασμός ενός ζωνοπερατού φίλτρου ζώνης διέλευσης (1.5 KHz, 3 KHz) με τις δύο μεθόδους που περιγράφηκαν πιο πάνω και η εφαρμογή τους στο σήμα *s* του παραδείγματος 1.2 (δείτε και το Παράδειγμα 2 στο τέλος της παρουσίασης «Η ΨΕΣ στις τηλεπικοινωνίες»). Επαληθεύστε το σωστό σχεδιασμό του φίλτρου σας, ελέγχοντας τόσο την απόκριση συχνότητάς του όσο και το αποτέλεσμα του φιλτραρίσματος στο σήμα *s*.

<u>Υποβάλατε την εργασία σας</u>

Αποθηκεύσατε τον κώδικά σας ως αρχείο M-file στο φάκελο εργασίας σας (My Documents\MATLAB). Χρησιμοποιήστε για το αρχείο το όνομα lab2_2_nnnnn.m, όπου nnnnn τα πέντε τελευταία νούμερα του αριθμού μητρώου σας.

Να υποβληθεί συμπιεσμένος φάκελος με τα αρχεία lab2_nnnnn.txt, lab2_1_nnnnn.m και lab2_2_nnnnn.m.

Μέρος 4: Εφαρμογή Β[4]

Να σχεδιαστεί και υλοποιηθεί σε MATLAB φίλτρο, όπως στην Εφαρμογή Α, με δύο ζώνες διέλευσης: (750 Hz, 950 Hz) και (2000 Hz, 3000 Hz).

[4] Προαιρετικό, αντί της Εφαρμογής Α.