ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
Τμήμα Επιστήμης Υπολογιστών

ΗΥ-370: Ψηφιακή Επεξεργασία Σήματος
Χειμερινό Εξάμηνο 2016
Διδάσκοντες: Γ. Στυλιανού - Γ. Καφεντζής
Δεύτερο Εργαστήριο

Σημείωση: Για βοήθεια σχετικά με τις εντολές/συναρτήσεις MATLAB
γράψτε: "*doc εντολή/συνάρτηση*" στο Command Window του MATLAB.

Άσκηση 1 - Μετασχηματισμός Fourier Διακριτού Χρόνου

Ας αρχίσουμε την ενασχόλησή μας με το μετασχ. Fourier, αφού δούμε λίγο μερικά πράγματα για "ζέσταμα". ☺

i. Θεωρήστε τα σήματα

$$x_1[n] = \begin{cases} 1, & 0 \leq n \leq 4 \\ 0, & \text{αλλού} \end{cases} \quad (1)$$

$$x_2[n] = \begin{cases} 1, & 0 \leq n \leq 29 \\ 0, & \text{αλλού} \end{cases} \quad (2)$$

(α΄) Υπολογίστε θεωρητικά το Μετασχ. Fourier των δυο σημάτων με χρήση του ορισμού

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}$$

Γράψτε το αποτέλεσμά σας σε μορφή λόγου ημιτόνων επί ένα εκθετικό, όπως κάναμε στις διαλέξεις. Σημειώστε την τιμή του μέγιστου στη θέση $\omega = 0$, καθώς και τα σημεία μηδενισμού.

(β΄) Στο MATLAB, τα παραπάνω σήματα παράγονται ως εξής

```
N1 = 5;
N2 = 30;
x1 = ones(1,N1);
x2 = ones(1,N2);
```

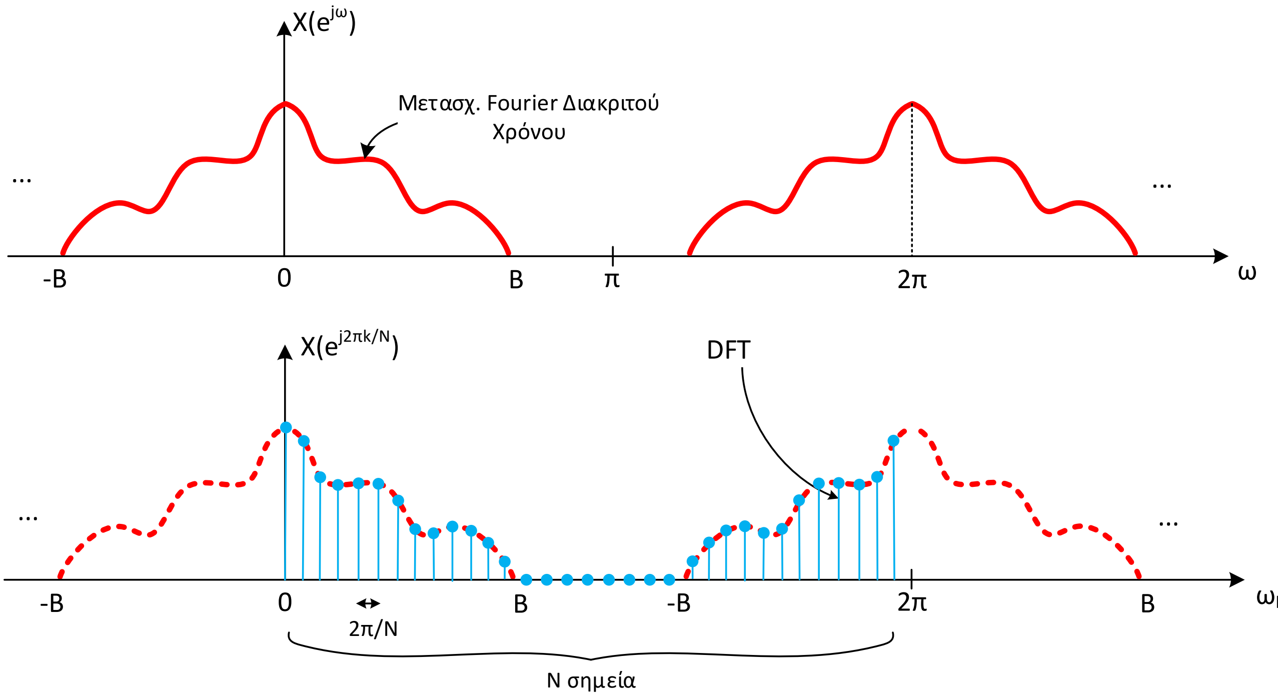
και δημιουργούν δυο μεταβλητές σε μορφή διανύσματος οι οποίες περιέχουν 5 και 30 άσσους αντίστοιχα. Ας βρούμε το μετασχ. Fourier στο MATLAB.

Η εντολή `fft(x, N)` υπολογίζει τιμές του μετασχ.Fourier διακριτού χρόνου (DTFT) για N συγκεκριμένες τιμές του ω, τις $\omega_k = 2\pi k/N,\ \ k = 0, \cdots, N-1$. Το διάστημα που δειγματοληπτείται είναι το $[0, 2\pi)$, αφού γνωρίζουμε πως έξω από αυτό, το φάσμα επαναλαμβάνεται - μην ξεχνάτε ότι ο μετασχ. Fourier διακριτού χρόνου είναι περιοδικός ως προς ω! Ο μετασχηματισμός αυτός λέγεται **Διακριτός Μετασχηματισμός Fourier - DFT** και υλοποιείται μέσω της συνάρτησης `fft`[1]. Το Σχήμα 1 δείχνει παραστατικά τη σχέση DTFT και DFT για ένα πραγματικό σήμα που έχει φάσμα στο διάστημα $[-B, B]$. Υπολογίστε τον DFT των παραπάνω σημάτων με τις εντολές

```
X1 = fft(x1, N1);
X2 = fft(x2, N2);
```

(γ΄) Οι παραπάνω μετασχηματισμοί είναι μιγαδικοί, οπότε και οι αντίστοιχες τιμές των διανυσμάτων `X1, X2` είναι μιγαδικές. Χρησιμοποιώντας τις συναρτήσεις `abs, angle, stem,`

[1] Η οποία υλοποιεί τον Fast Fourier Transform, ο οποίος είναι απλά μια υπολογιστικά αποδοτική υλοποίηση του DFT.



Σχήμα 1: Σχέση DTFT και DFT για ένα πραγματικό σήμα.

υπολογίστε και σχεδιάστε το φάσμα πλάτους και φάσης των σημάτων. Προφανώς θα είναι διακριτά φάσματα, αφού ο μετασχ. Fourier διακριτού χρόνου (DTFT)- ο οποίος είναι συνεχής συνάρτηση ως προς ω - έχει δειγματοληπτηθεί στις συχνότητες ω_k, δηλ. έχει υπολογιστεί μέσω του DFT. Προτείνεται ο παρακάτω κώδικας για το φάσμα `X1`:

```
NFFT = N1;
omega = 2*pi*(0:NFFT-1)./NFFT;
mag = abs(X1);
ph = angle(X1);
figure; stem(omega, mag);
title('Magnitude Spectrum'); xlabel('Frequency (\omega_k)');
figure; stem(omega, ph);
title('Phase Spectrum'); xlabel('Frequency (\omega_k)');
```

(δ') Μοιάζουν τα φάσματα πλάτους και φάσης των δυο σημάτων που υπολογίσατε παραπάνω με αυτό που περιμένατε; Τελικά, τα σήματα είναι ισοδύναμα ή διαφορετικά; Κάντε μια εκτίμηση.

(ε') Υπολογίστε τον DFT του σήματος $x_1[n]$ με τις εντολές

```
fX1 = fft(x1, 25);
fX2 = fft(x1, 26);
```

Χρησιμοποιώντας τις συναρτήσεις `abs`, `angle`, `stem`, υπολογίστε και σχεδιάστε το φάσμα πλάτους και φάσης των σημάτων, όπως παραπάνω. Μοιάζουν τα φάσματα πλάτους και φάσης των δυο σημάτων που υπολογίσατε μόλις; Αυτό που βλέπετε μοιάζει με αυτό που περιμένατε θεωρητικά; Σχολιάστε τα αποτελέσματα.

(ϛ') Προσέξτε τι σημαίνουν οι τιμές 25 και 26 στις παραπάνω εντολές και παρατηρήστε προσεκτικά τις τιμές του φάσματος πλάτους των δυο παραπάνω μετασχηματισμών. Εξηγήστε την παρουσία μηδενικών στο φάσμα του πρώτου μετασχηματισμού. Σε ποιές συχνότητες παρουσιάζονται τα μηδενικά αυτά; Εξηγήστε την απουσία αυτών στον δεύτερο.

(ζ) Συγκρίνετε τις πρώτες 10 τιμές του φάσματος πλάτους και φάσης όπως τις έχει υπολογίσει το MATLAB, και τις τιμές που εσείς βρίσκετε θεωρητικά (χρησιμοποιήστε MATLAB και για τους υπολογισμούς του θεωρητικού σας κομματιού, δηλ. υλοποιήστε το θεωρητικό σας κομμάτι στο MATLAB ρητά). Σχολιάστε.

(η') Ας επιστρέψουμε πίσω στο πεδίο του χρόνου. Υπολογίστε τον αντίστροφο μετασχ. Fourier με τη συνάρτηση `ifft`, και σχεδιάστε αυτό που παίρνετε με τη συνάρτηση `stem` ως ακολούθως:

```
xx1 = real(ifft(fX1));
stem(0:24, xx1);
```

(θ') Στα θεωρητικά σήματα που έχετε αναλύσει στο χαρτί σας υπάρχει μια γραμμική φάση (αν δεν την έχετε βρει, ελέγξτε τα αποτελέσματά σας!). Από που προέρχεται η φάση αυτή; Πως μπορείτε να την αφαιρέσετε θεωρητικά, με χρήση μιας από τις ιδιότητες του μετασχ. Fourier διακριτού χρόνου;

(ι') Υλοποιήστε τη σκέψη σας και υπολογίστε ξανά τον αντίστροφο μετασχ. Fourier του νέου φάσματος. Για το ερώτημα αυτό, προτείνεται ο ακόλουθος κώδικας:

```
N = 25;
w = 2*pi/N;
lp = 2;
wn = lp*w*(0:N-1);
xx1 = real(ifft(fX1 .* exp(j*wn)));
stem(0:24, xx1);
```

Σχεδιάστε τα σήματα `xx1, xx2`. Συγκρίνετε το σήμα `xx1` που έχετε τώρα με αυτό που βρήκατε στο προηγούμενο ερώτημα. Τι (πολύ!) απρόοπτο παρατηρείτε ότι συνέβη;; Με βάση αυτό το απρόοπτο που συνέβη, πώς καταλαβαίνετε ότι αντιμετωπίζει ο `fft/ifft` του MATLAB τα σήματα στο χρόνο;

ii. Έστω τώρα το σήμα

$$x[n] = 3\cos(2\pi n/5) \tag{3}$$

i. Βρείτε το μετασχ. Fourier του σήματος και σχεδιάστε το φάσμα πλάτους και φάσης για συχνότητες $0 \le \omega \le 2\pi$ στο χαρτί.

ii. Δημιουργήστε μόνο 15 δείγματα του παραπάνω σήματος, γύρω από το $n = 0$. Έστω $x_1[n]$ αυτό το σήμα. Ουσιαστικά πολλαπλασιάζετε το $x[n]$ με ένα τετραγωνικό παράθυρο $w[n]$ μοναδιαίου πλάτους και διάρκειας 15 δειγμάτων, με κέντρο το $n = 0$. Υπολογίστε και σχεδιάστε το φάσμα πλάτους και φάσης του $x_1[n]$ χρησιμοποιώντας τον παρακάτω κώδικα:

```
No = 5;
N = 15;
x1 = 3 * cos(2*pi*(-7:7)/No);
figure(1); stem(-7:7, x1); xlabel('Time (samples');
fx1 = fft(x1, N);
mag = abs(fx1);
ph = angle(fx1);
omega = 2*pi*(0:N-1)./N;
figure(2);
subplot(211); stem(omega, mag);
title('Magnitude Spectrum'); xlabel('Frequency (\omega_k)');
subplot(212); stem(omega, ph);
title('Phase Spectrum'); xlabel('Frequency (\omega_k)');
```

iii. Το σήμα στο χρόνο που δημιουργήσατε είναι πραγματικό και *μη* αιτιατό. Όμως είναι περιττό ή άρτιο; Με βάση τη θεωρία, τι συνέπεια έχει αυτό στα φάσματα του μετασχ. Fourier;

iv. Τα γραφήματα που πήρατε από τον κώδικα συμφωνούν με τη θεωρητική σας παρατήρηση; Ποιό γράφημα δεν είναι όπως αναμενόταν;[2]

v. Πώς μπορείτε να αλλάξετε τον παραπάνω κώδικα για να διορθώσετε το "προβληματικό" γράφημα; Σκεφτείτε ότι η συνάρτηση `fft` λαμβάνει ως όρισμα ένα διάνυσμα `x1` θεωρώντας ότι η χρονική στιγμή $n = 0$ συμπίπτει με το πρώτο στοιχείο του διανύσματος (ενώ εσείς θέλετε να επεξεργαστείτε ένα μη αιτιατό σήμα που το πρώτο δείγμα του συμβαίνει τη χρονική στιγμή $n = -7$).

vi. Ας θεωρήσουμε τώρα τον παρακάτω κώδικα:

```
No = 5;
x1 = 3 * cos(2*pi*(-7:7)/No);
N = 99;
fx1 = fft(x1, N);
mag = abs(fx1);
ph = angle(fx1);
omega = 2*pi*(0:N-1)./N;
figure(3);
subplot(211); stem(omega, mag);
title('Magnitude Spectrum'); xlabel('Frequency (\omega_k)');
subplot(212); stem(omega, ph);
title('Phase Spectrum'); xlabel('Frequency (\omega_k)');
```

Εδώ ζητούμε από τη συνάρτηση `fft` να υπολογίσει το μετασχ. Fourier σε 99 σημεία. Αυτό σημαίνει ότι ο θεωρητικός μετασχ. Fourier διακριτού χρόνου θα δειγματοληπτηθεί σε 99 σημεία-συχνότητες ω_k.

vii. Ποιές είναι οι νέες συχνότητες στις οποίες δειγματοληπτείται ο μετασχ. Fourier διακριτού χρόνου;

viii. Σε ποιές συχνότητες εμφανίζεται τώρα να είναι κατανεμημένη η ενέργεια του σήματος;

ix. Διορθώστε ξανά τη φάση όπως (πρέπει να) κάνατε στον προηγούμενο κώδικα.

iii. Αφού είδαμε πως λειτουργεί πρακτικά ο μετασχ. Fourier μέσω του fft, ας προσπαθήσουμε να προσομοιώσουμε ακριβώς τον μετασχ. Fourier διακριτού χρόνου σε διάφορα σήματα στο MATLAB. Ξέρουμε στη θεωρία, λοιπόν, πώς να υπολογίζουμε ένα μετασχ. Fourier διακριτού χρόνου. Όμως όπως είπαμε, ο μετασχ. Fourier είναι *συνεχής* συνάρτηση του ω, άρα ορίζεται σε άπειρα σημεία του άξονα ω. Αυτό είναι πρόβλημα για την αναπαράστασή του στο MATLAB, και είδαμε νωρίτερα πως αντιμετωπίζεται αυτό μέσω του DFT.

Για περισσότερη ευκολία, έχει γραφεί για σας η συνάρτηση `myDTFT`, την οποία καλείστε να χρησιμοποιήσετε για να βρίσκετε φάσμα πλάτους και φάσμα φάσης σημάτων γρηγορότερα απ' ότι κάνατε νωρίτερα. Γράψτε `help myDTFT` για να δείτε τη σύνταξη και τα παραδείγματα. Προσέξτε, ο κώδικας αυτός προσομοιώνει τον μετασχ. Fourier διακριτού χρόνου, άρα τα γραφήματα που θα παίρνετε θα αποτελούνται από συνεχείς καμπύλες συναρτήσει του ω. Φυσικά η προσομοίωση γίνεται με κλήση της συνάρτησης `fft`.

Υπολογίστε το φάσμα πλάτους και το φάσμα φάσης των παρακάτω σημάτων:

(α') $x[n] = \frac{1}{M+1}\sum_{k=0}^{M}\delta[n-k]$, για $n = 0, \cdots, 19$ και $M = 5$. Επαναλάβετε για $n = 0, \cdots, 299$ και $M = 5$. Συγκρίνετε και σχολιάστε.

[2] Μην κοιτάζετε τις τιμές, κοιτάζετε το σχήμα, τη μορφή των γραφημάτων.

(β') $x[n] = \cos(0.4\pi n)$, για $n = 0, \cdots, 19$. Σχολιάστε. Επαναλάβετε για $n = 0, \cdots, 299$. Συγκρίνετε και σχολιάστε. Επίσης, περιγράψτε ποιοτικά τι θα συμβεί αν το παραπάνω σήμα εμφανιστεί στην είσοδό ενός συστήματος που περιγράφεται από το σήμα του πρώτου ερωτήματος.

(γ') $x[n] = (0.5)^n u[n]$ για $n = 0, \cdots, 19$. Σχολιάστε. Επαναλάβετε για $n = 0, \cdots, 299$. Συγκρίνετε και σχολιάστε. Επίσης, σχολιάστε ποιοτικά τι θα συμβεί αν το παραπάνω σήμα περιγράφει ένα ΓΧΑ σύστημα, και στην είσοδό του εμφανιστεί το σήμα του δεύτερου ερωτήματος.

Άσκηση 2 - Αποθορυβοποίηση στην Aegean ☺

Στα αεροπλάνα, η ηλεκτρική ισχύς είναι στα 400 Hz (ενώ στα σπίτια είναι στα $50 - 60$ Hz). Για τον λόγο αυτό, όταν γίνεται κάποια ανακοίνωση από τα ηχεία του αεροπλάνου, στο σήμα της φωνής προστίθεται ένα συνημίτονο συχνότητας 400 Hz. Το αποτέλεσμα δεν είναι ευχάριστο στο αυτί μας. Καλείστε να φτιάξετε ένα φίλτρο το οποίο να αφαιρεί το παραπάνω ενοχλητικό σήμα χωρίς να καταστρέφει το σήμα της φωνής.

Για το πείραμά σας, χρησιμοποιήστε το αρχείο 4781_8k.wav, το οποίο θα φορτώσετε και ακούσετε στο MATLAB ως εξής:

```
[s,fs] = wavread('4781_8k.wav');
soundsc(s, fs);
```

Ακολουθήστε τα παρακάτω βήματα:

i. Προφανώς καταλαβαίνετε ότι για ένα σήμα με απόκριση σε συχνότητα

$$H(e^{j\omega}) = 1 - e^{-j(\omega-\omega_0)} \tag{4}$$

ισχύει $H(e^{j\omega}) = 0$ όταν $\omega = \omega_0$. Θεωρήστε ότι έχετε ένα σήμα της μορφής

$$x[n] = \cos(\omega_0 n) \tag{5}$$

(α') Βρείτε στο χαρτί το σήμα στο χρόνο $h[n]$ του παραπάνω φάσματος $H(e^{j\omega})$.

(β') Η συνάρτηση `freqz` υπολογίζει την απόκριση σε συχνότητα $H(e^{j\omega})$ μιας εξίσωσης διαφορών. Δέχεται τους συντελεστές της εισόδου B και της εξόδου A σε δυο ξεχωριστά διανύσματα-ορίσματα, και επιστρέφει δυο διανύσματα, ένα με τον άξονα των συχνοτήτων σε rad/sample, W, και ένα με τις τιμές της απόκρισης σε συχνότητα, H. Παράδειγμα σύνταξης είναι το `[H,W] = freqz(B,A)`.

Βρείτε τη φασματική απόκριση του παραπάνω φίλτρου για την τιμή του ω_0 που απαιτείται και χρησιμοποιήστε τη συνάρτηση `plot` για να σχεδιάσετε το *μέτρο* της φασματικής απόκρισης (συνάρτηση `abs`). Σας μοιάζει σωστό;

(γ') Σχεδιάστε το φάσμα φάσης (συνάρτηση `plot`), χρησιμοποιώντας το διάνυσμα W και τη συνάρτηση `angle` επάνω στο διάνυσμα H.

(δ') Θεωρώντας το $x[n]$ που σας δίνεται, βρείτε στο χαρτί το μετασχ. Fourier του.

(ε') Ποιά θα είναι η έξοδος $y[n]$ όταν το παραπάνω σήμα $x[n]$ περάσει από ένα **τυχαίο** σύστημα (όχι απαραίτητα το συγκεκριμένο που δίνεται) με φασματική απόκριση $H(e^{j\omega})$; Εκμεταλλευτείτε το γεγονός ότι το σήμα εισόδου είναι άθροισμα *ιδιοσυναρτήσεων* του συστήματος και εκφράστε την απάντησή σας συναρτήσει των ιδιοτιμών του συστήματος, $H(e^{\pm j\omega_0})$.

(ϛ') Βρείτε στο χαρτί ένα φίλτρο $H_z(e^{j\omega})$ που να μηδενίζει το παραπάνω σήμα εισόδου (το οποίο πρέπει να βρήκατε ότι έχει **δυο** συνιστώσες στο χώρο της συχνότητας). Σκεφτείτε παρόμοια με τη Σχέση (4), και οτι το σύστημα αυτό μπορεί να αναλυθεί σε δυο υποσυστήματα σε σειρά.

ii. Μετατρέψτε το φίλτρο $H_z(e^{j\omega})$ που βρήκατε παραπάνω σε εξίσωση διαφορών. Χρησιμοποιήστε τη συνάρτηση `freqz` για να δείτε αν όντως κάνει αυτό που πρέπει.

iii. Χρησιμοποιήστε τη συνάρτηση `filter` για να το υλοποιήσετε στο MATLAB, δίνοντας ως είσοδο το πραγματικό σήμα φωνής που έχετε ανοίξει παραπάνω. Η filter συντάσσεται ως `y = filter(B,A,x)`, με B τους συντελεστές της εισόδου στην εξίσωση διαφορών, A τους συντελεστές της εξόδου στην εξίσωση διαφορών, και x το διάνυσμα του σήματος εισόδου.

iv. Ένα πιο καλό φίλτρο από το προηγούμενο που βρήκατε περιγράφεται με την εξίσωση διαφορών

$$y[n] - 2\alpha\cos(\omega_0)y[n-1] + \alpha^2 y[n-2] = x[n] - 2\cos(\omega_0)x[n-1] + x[n-2] \tag{6}$$

(α΄) Χρησιμοποιώντας $\alpha = 0.99$ και $\alpha = 0.8$, εφαρμόστε το φίλτρο με χρήση της `filter` για να καθαρίσετε το σήμα φωνής. Σχολιάστε τις επιδόσεις του φίλτρου για κάθε τιμή του α.

(β΄) Χρησιμοποιήστε τη συνάρτηση `freqz` για να σχεδιάσετε και να παραδώσετε τα φάσματα πλάτους και φάσης.

(γ΄) Βρείτε στο χαρτί σας τη φασματική απόκριση $H(e^{j\omega})$ της παραπάνω εξίσωσης διαφορών.

(δ΄) Υλοποιήστε το μέτρο και τη φάση της στο MATLAB ως συνάρτηση του ω - μη χρησιμοποιήσετε τη `freqz`, αντίθετα χρησιμοποιήστε τις συναρτήσεις `abs, angle`. Απεικονίστε τις.

(ε΄) Τα γραφήματά σας είναι ίδια με αυτά που παρήγαγε η `freqz`;

Άσκηση 3 - Κωδικός 007**: ανίχνευση αριθμού τηλεφώνου**

Όταν πληκτρολογούμε ένα τηλεφωνικό νούμερο στο κινητό μας ή σε ένα οποιοδήποτε τηλέφωνο με πλήκτρα, δημιουργούμε για κάθε αριθμό δυο συνημίτονα. Για παράδειγμα, όταν πληκτρολογούμε τον αριθμό 0, δημιουργούμε δυο συνημίτονα με συχνότητες 941 και 1336 Hz.

Ο Πίνακας που αντιστοιχεί στο πρότυπο (standard) DTMF - Dual-Tone Multi-Frequency, φαίνεται παρακάτω στον πίνακα του Σχήματος 2. Οι συχνότητες που αντιστοιχούν σε κάθε αριθμό δεν

Frequencies	$1209Hz$	$1336Hz$	$1477Hz$
$697Hz$	1	2	3
$770Hz$	4	5	6
$852Hz$	7	8	9
$941Hz$	$\star$	0	$\sharp$

Σχήμα 2: Πίνακας Πρότυπου DTMF.

έχουν επιλεγεί τυχαία. Καμιά από τις συχνότητες δεν είναι πολλαπλάσιο κάποιας άλλης, άθροισμα ή διαφορά οποιωνδήποτε δυο συχνοτήτων κλπ. Η συνθήκη αυτή διευκολύνει πολύ τον εντοπισμό των συχνοτήτων (και άρα τον αριθμό).

Κάθε άθροισμα συνημιτόνων (δηλ. κάθε τηλεφωνικός *τόνος*) διαρκεί 0.5 δευτερόλεπτα, ενώ υπάρχει μια παύση (σιωπή) μεταξύ των τόνων διάρκειας 0.1 δευτερολέπτων. Η παύση δεν είναι τίποτε άλλο από ένα διάνυσμα γεμάτο μηδενικά, κατάλληλης διάρκειας. Η συχνότητα δειγματοληψίας είναι 8000 Hz για όλο το σύστημα. Υποθέτουμε ότι ο αριθμός που ψάχνουμε έχει συντεθεί από συνενώσεις τόνου+σιωπής διάρκειας $0.5 + 0.1 = 0.6$ δευτερολέπτων. Άρα το σκεπτικό μας είναι να χωρίσουμε το σήμα σε “κομμάτια” (frames) των 0.6 δευτερολέπτων, χωρίς αυτά να επικαλύπτονται, και να ελέγχουμε σε κάθε frame αν υπάρχει κάποιος τόνος από τους παραπάνω. Σκοπός αυτής της άσκησης είναι να σχεδιάσετε ένα σύστημα ζωνοπερατών (bandpass) φίλτρων που θα φιλτράρει ένα frame κάθε φορά, και θα ελέγχει ποιά έξοδος φίλτρου έχει περισσότερη πληροφορία σε σχέση με τις υπόλοιπες. Αυτή η έξοδος - που αντιστοιχεί σε κάποιο ζωνοπερατό φίλτρο - θα μας υποδείξει ποιό φίλτρο χρησιμοποιήθηκε, και άρα ποιές συχνότητες επιτρέπει να περάσουν. Αυτές οι συχνότητες θα πρέπει να είναι συχνότητες που υπάρχουν στον Πίνακα 2.

Το σύστημα ζωνοπερατών φίλτρων θα σχεδιαστεί από ένα βασικό (ιδανικό) βαθυπερατό (lowpass) φίλτρο το οποίο έχει απόκριση σε συχνότητα στο διάστημα $(-\pi, \pi)$ ως

$$H_{lp}(e^{j\omega}) = \begin{cases} 1, & -\dfrac{\pi}{200} < \omega < \dfrac{\pi}{200} \\[2ex] 0, & \dfrac{\pi}{200} \le |\omega| < \pi \end{cases} \tag{7}$$

Προφανώς για να υλοποιήσετε τα ζωνοπερατά φίλτρα, θα πρέπει να μετατοπίζετε αυτό το φίλτρο γύρω από τις κατάλληλες συχνότητες.

Ως παράδειγμα σας δίνουμε ένα αρχείο ήχου που περιέχει ένα νούμερο από ένα κινητό τηλέφωνο. Σκοπός σας είναι να εντοπίσετε τον αριθμό. Στο MATLAB μπορείτε να χρησιμοποιήσετε τις παρακάτω εντολές για να το ακούσετε.

```
[s,fs] = wavread('cell_num.wav');
soundsc(s,fs);
```

Σας δίνονται επίσης τρεις συναρτήσεις που πρέπει να συμπληρώσετε. Ακολουθήστε τις παρακάτω οδηγίες για κάθε συνάρτηση.

1. **makeFB.m**

   Σε αυτήν την συνάρτηση θα φτιάξετε το σύστημα ζωνοπερατών φίλτρων. Η συνάρτηση αυτή σας επιστρέφει έναν πίνακα που η κάθε στήλη του περιέχει 201 δείγματα της κρουστικής απόκρισης από ένα φίλτρο. Π.χ. η 3η στήλη του πίνακα περιέχει την κρουστική απόκριση του ζωνοπερατού φίλτρου που έχει σαν κέντρο του τη συχνότητα 852 Hz.

   (α΄) Βρείτε αρχικά στο χαρτί ποιό είναι το σήμα στο χρόνο στο οποίο αντιστοιχεί το χαμηλοπερατό φίλτρο που αναφέρεται παραπάνω στη Σχέση (7).

   (β΄) Γράψτε την έκφραση που βρήκατε στη γραμμή 33 της συνάρτησης. Το MATLAB διαθέτει συνάρτηση `sinc` η οποία μπορεί να σας φανεί χρήσιμη.

   (γ΄) Βρείτε το μετασχηματισμό Fourier του σήματος

   $$y[n] = \cos(\omega_0 n) h_{lp}[n] \tag{8}$$

   στο χαρτί σας. Εκφράστε τον ως συνάρτηση του μετασχ. Fourier του $h_{lp}[n]$, που συμβολίζεται με $H_{lp}(e^{j\omega})$. Τι παρατηρείτε ότι συμβαίνει στο $H_{lp}(e^{j\omega})$ όταν πολλαπλασιάζεται στο χρόνο με ένα συνημίτονο;

   (δ΄) Με βάση την παραπάνω παρατήρησή σας, συμπληρώστε τη γραμμή 38 της συνάρτησης, επιλέγοντας κατάλληλες ω_0.

   (ε΄) Τέλος, στη γραμμή 40, κανονικοποιήστε το κάθε φίλτρο απλά διαιρώντας το με το άθροισμα των τιμών των δειγμάτων του. Η συνάρτηση `sum` μπορεί να σας φανεί χρήσιμη.

2. **dDTMF.m**

   Σε αυτήν την συνάρτηση, θα φιλτράρετε ένα σήμα τηλεφωνικού αριθμού και θα εντοπίσετε τον τηλεφωνικό αριθμό που μεταφέρει το σήμα. Σημειώστε ότι γίνεται κλήση της συνάρτησης `makeFB.m`. Η συνάρτηση `dDTMF.m` σας επιστρέφει έναν πίνακα δυο στηλών ο οποίος περιέχει τις συχνότητες που μεταφέρει το σήμα, καθώς και έναν πίνακα χαρακτήρων που περιέχει τον αριθμό που ανιχνεύτηκε. Συγκεκριμένα:

   (α΄) Γραμμή 27: Μετατρέψτε τη διάρκεια κάθε τόνου από δευτερόλεπτα σε δείγματα.

   (β΄) Γραμμή 33: Κάντε το ίδιο για τις σιωπές.

(γ΄) Γραμμή 36: Βρίσκουμε πόσα frames τόνου+σιωπής υπάρχουν σε όλο το τηλεφωνικό σήμα.

(δ΄) Γραμμή 48: Ο βρόχος επανάληψης διατρέχει το σήμα εισόδου ανά `sh` δείγματα και αποθηκεύει κάθε φορά στο διάνυσμα `fr` ένα κομμάτι διάρκειας όσο είναι *μόνο* η διάρκεια του τόνου (`st`), η οποία θεωρούμε ότι προηγείται της σιωπής πάντα.

(ε΄) Γραμμή 54: Υπολογίστε την ενέργεια του κάθε κομματιού που δεσμεύεται, με βάση την εξίσωση

$$E = \sum_{n=-\infty}^{+\infty} x^2[n] \tag{9}$$

Ξανά, η συνάρτηση sum θα σας φανεί χρήσιμη.

(ϛ΄) Γραμμή 58: Ο βρόχος επανάληψης θα πρέπει να διατρέχει όλα τα ζωνοπερατά φίλτρα που έχουμε φτιάξει στον πίνακα `h` και να κάνει συνέλιξη με το σήμα που βρίσκεται στο διάνυσμα `fr`.

(ζ΄) Γραμμή 60: Θα κάνετε φιλτράρισμα (συνέλιξη δηλαδή) του σήματος με το j-οστό ζωνοπερατό φίλτρο. Η συνάρτηση `conv` θα σας χρειαστεί.

(η΄) Γραμμή 63: Υπολογίστε την ενέργεια του φιλτραρισμένου σήματος.

(θ΄) Γραμμή 69: Πρέπει να θέσετε ένα όριο ώστε να αναγνωρίζετε πότε ένα ζωνοπερατό φίλτρο έχει πέσει σε συχνότητα που περιέχεται στο σήμα `fr`. Σας προτείνεται ένα τέτοιο κριτήριο μέσα στον κώδικα ως σχόλιο. Πειραματιστείτε αν έχετε χρόνο.

(ι΄) Γραμμές $84 - 118$: Με βάση τον Πίνακα 2, συμπληρώστε τα ζεύγη συχνοτήτων για κάθε αριθμό.

3. **call.m**
   Σε αυτήν την συνάρτηση, μπορείτε να δημιουργήσετε το δικό σας ήχο από οποιοδήποτε τηλεφωνικό νούμερο, να το ακούσετε, και στο τέλος να το ανιχνεύσετε με τη βοήθεια των συναρτήσεων που γράψατε. Για παράδειγμα, μπορείτε να γράψετε

```
x = call([2 8 1 0 3 9 3 5 3 3]);
[Freq, Num] = dDTMF(x);
```

   Για να δουλέψει αυτή η συνάρτηση, πρέπει να συμπληρώσετε μέσα τις συχνότητες του Πίνακα 2 στις γραμμές $27 - 50$.

Αν καταφέρετε να ανακτήσετε σωστά τον αριθμό που σας δίνεται, στείλτε ένα SMS - όχι κλήση! - σε αυτόν τον αριθμό[3] με το όνομά σας και το Α.Μ. σας ☺. Αν όλα πάνε καλά, θα σας σταλεί μια απάντηση...

Για την παράδοση της άσκησης, γράψτε **πλήρη** αναφορά, συμπεριλαμβάνοντας απαντήσεις σε όλα τα ερωτήματα του εργαστηρίου, καθώς και διαγράμματα/γραφήματα/εικόνες με τα αποτελέσματά σας, και συμπεριλάβετε τον κώδικα MATLAB σε *ξεχωριστά* .m files. Φροντίστε να έχετε στείλει και το SMS! ☺

Η παράδοση γίνεται **αποκλειστικά** με το πρόγραμμα TURNIN.

Προθεσμία: 19 / 10 / 2016, timestamp: 23:59:59

[3] Ναι, πρέπει να ρισκάρετε :-)