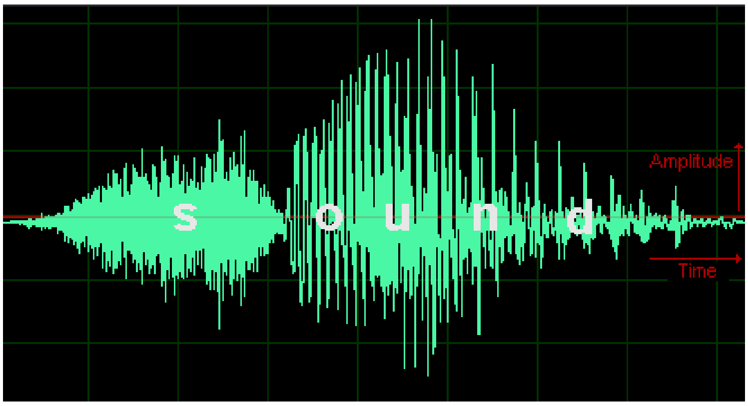
Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα: Ψηφιακή Επεξεργασία Σημάτων

1η Εργαστηριακή Άσκηση

Θέμα: Εισαγωγή στην Ψηφιακή Επεξεργασία Σημάτων με MATLAB και Εφαρμογές σε Ακουστικά Σήματα



Ονοματεπώνυμο: **Βαβουλιώτης Γεώργιος** AM: **03112083**

Ονοματεπώνυμο: **Σταυρακάκης Δημήτριος** AM: **03112017**

Ημερομηνία Παράδοσης: 7/4/2015

Μέρος 1°
Σύστημα Εντοπισμού Τηλεφωνικών Τόνων
(Telephone Touch-Tones)

Σκοπός της 1ης άσκησης είναι η παρουσίαση της λειτουργίας του τηλεφωνικού τονικού συστήματος, χρησιμοποιώντας σήματα διαφορετικών συχνοτήτων, με σκοπό να εντοπίσει ποιο από τα πλήκτρα έχει πατηθεί. Με το πάτημα ενός πλήκτρου στο τηλέφωνο παράγεται ένας ήχος ο οποίος είναι άθροισμα 2 ημιτόνων, ενός χαμηλόσυχνου κι ενός υψίσυχνου. Το χαμηλόσυχνο υποδεικνύει τη γραμμή και το υψίσυχνο τη στήλη του touch-pad του τηλεφώνου. Οι συχνότητες γραμμών και στηλών του touch-pad φαίνονται στον παρακάτω πίνακα :

	Ω_{col}		
Ω_{row}	0.9273	1.0247	1.1328
0.5346	1	2	3
0.5906	4	5	6
0.6535	7	8	9
0.7217		0	

Πίνακας 1: Διακριτές συχνότητες για τηλεφωνικούς τόνους σε συχνότητα δειγματοληψίας 8192Hz.

Ερώτημα 1.1. :

Αρχικά δημιουργείται ένας 1000x10 πίνακας με όνομα d ο οποίος περιέχει 10 διαφορετικούς τόνους(στήλες), καθένας από τους οποίους έχει μήκος 1000 δείγματα και αποτελείται από το άθροισμα 2 ημιτόνων, ενός υψίσυχνου(στήλη του Πίνακα 1.1) κι ενός χαμηλόσυχνου(γραμμή του Πίνακα 1.1). Αυτό έγινε πράξη με τη βοήθεια των συναρτήσεων ***givetones()*** και ***tone()***, οι οποίες φαίνονται αναλυτικά στον κώδικα της άσκησης.
Για να επαληθεύσω ότι όντως έχω καταφέρει να δημιουργήσω αυτό που περιγράφω παραπάνω χρησιμοποιώ το παρακάτω τμήμα κώδικα ώστε να ακούσω τους αντίστοιχους ήχους μέσω της συνάρτησης sound του matlab, το οποίο όμως υπάρχει σαν σχόλιο στο κώδικα της άσκησης 1 :

```
for(i=1:10)
        sound(d(:,i));
        pause ;
end
```

Ερώτημα 1.2. :

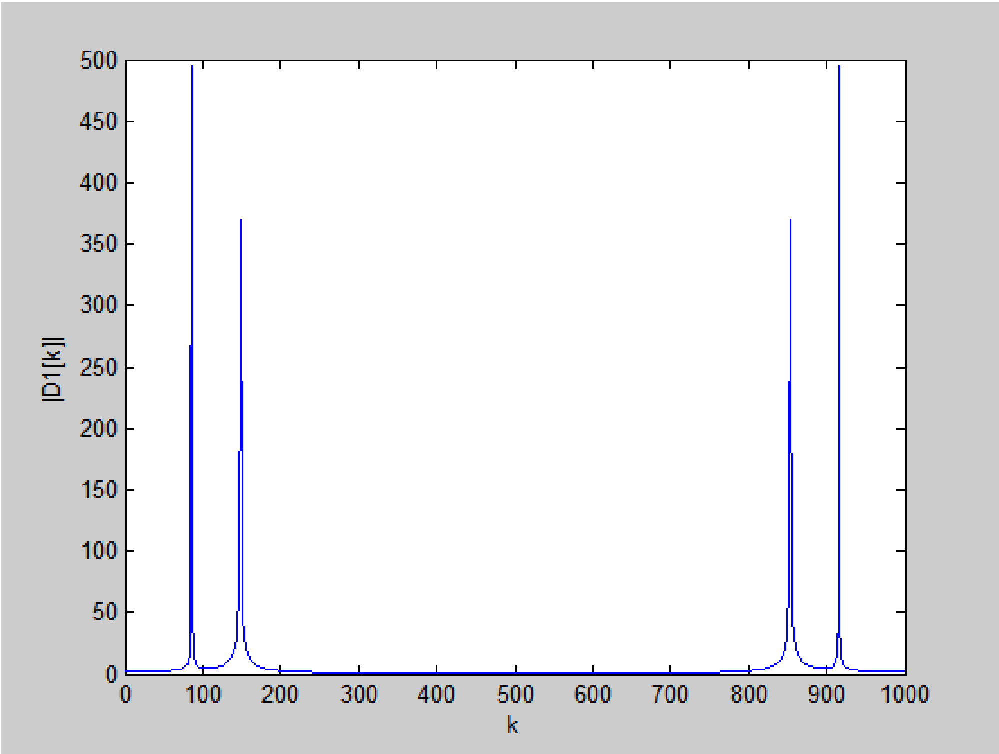
Στο δεύτερο ερώτημα ζητείται ο υπολογισμός του DFT των σημάτων $d_1[n]$ και $d_8[n]$. Γνωρίζω ότι ο DFT ενός σήματος προκύπτει σύμφωνα με τη σχέση :

$$X[k]=\sum_{n=0}^{N-1} x[n]\cdot e^{-j\frac{2\pi}{N}nk}, 0\le k\le N-1 \ .$$

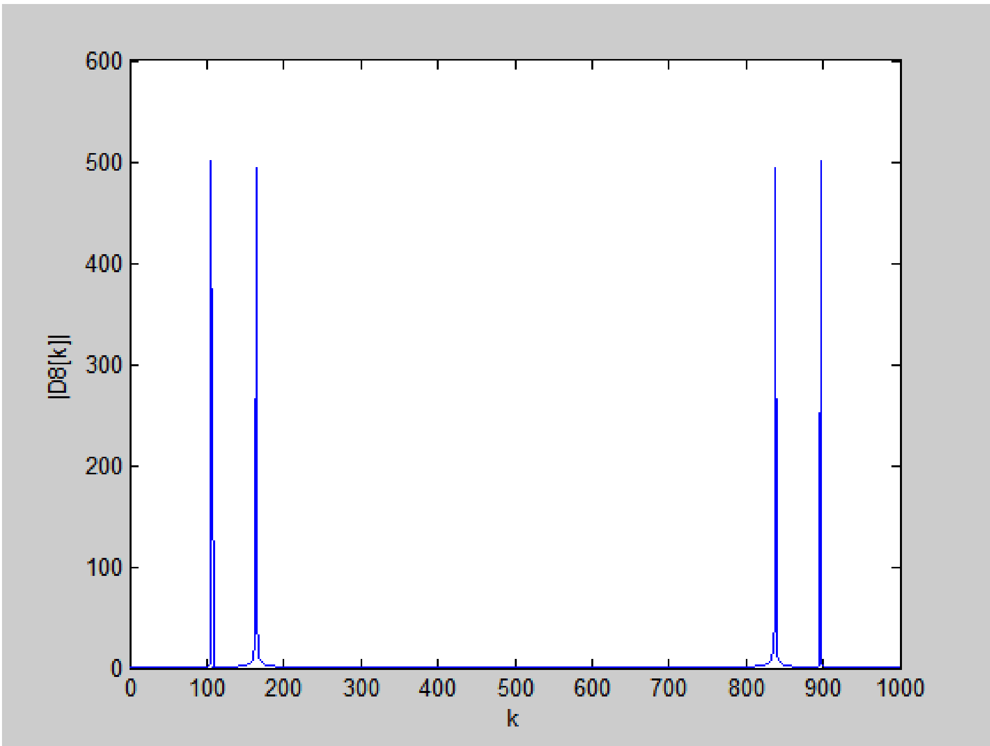
Ωστόσο στο Matlab ο DFT ενός σήματος υλοποιείται μέσω της συνάρτησης ***fft()***. Ενδεικτικά παρατίθεται η εντολή που δίνει τον DFT του $d_1[n]$:

```
D1=abs(fft(d(:,1)));
```

Στη συνέχεια παρουσιάζονται τα γραφήματα που αφορούν τις γραφικές παραστάσεις των μέτρων των DFT των παραπάνω σημάτων($d_1[n]$ και $d_8[n]$):



Εικόνα 1.2.1: Γραφική παράσταση του πλάτους του DFT του τόνου $d_1[n]$



Εικόνα 1.2.2: Γραφική παράσταση του πλάτους του DFT του τόνου $d_8[n]$

Σύμφωνα με τον ορισμό του DFT ενός σήματος προκύπτει η εξής σχέση μεταξύ ω,k:

$$\omega = \frac{2\pi}{\mathrm{N}} \cdot k = \frac{2\pi}{1000} \cdot k$$

Παρατηρούμε από τα παραπάνω γραφήματα ότι για κάθε σήμα υπάρχουν 4 τιμές(ουσιαστικά 2 λόγω συμμετρίας του DFT) για το k που δίνουν 2 μέγιστα στο πλάτος, δηλαδή αντιστοιχούν στις 2 συχνότητες που απαρτίζουν το κάθε σήμα(μια χαμηλόσυχνη και μια υψίσυχνη). Αυτό θα επαληθευτεί και στη συνέχεια στο *ερώτημα 1.5* όπου θα βρεθούν με χρήση του Matlab τα k που είναι εγγύτερα στις touch-tone συχνότητες.

<u>Ερώτημα 1.3. :</u>

Εφόσον η αναφορά αυτή γίνεται από 2 άτομα υπολογίζεται το άθροισμα των αριθμών μητρώου των 2 ατόμων (03112083 + 03112017 = ***<u>06224100</u>***) , το οποίο θα μεταφραστεί σε τονικά σήματα με την βοήθεια του Matlab και στη συνέχεια θα ηχογραφηθεί σε ένα αρχείο wav.

Συγκεκριμένα δημιουργείται ο πίνακας ***s*** ο οποίος περιέχει τους τόνους του αντιστοιχούν στο παραπάνω ΑΜ, οι οποίοι πρέπει να τονιστεί ότι διαχωρίζονται με 100 μηδενικά δείγματα ο ένας από τον άλλο, όπως αυτό φαίνεται και στο κώδικα που έχει παραδοθεί στο συμπιεσμένο αρχείο.

Τέλος, το σήμα που είναι αποθηκευμένο στον πίνακα ***s*** ηχογραφείται στο αρχείο με όνομα ***tone_sequence.wav*** με τη βοήθεια της συνάρτησης ***wavwrite()*** του Matlab με συχνότητα δειγματοληψίας ***Fs=8192 Hz***. Το αρχείο αυτό μπορείτε να το βρείτε ώστε να το ακούσετε στο συμπιεσμένο φάκελο με το παραπάνω όνομα.

<u>Ερώτημα 1.4. :</u>

Στο ερώτημα αυτό αρχικά δημιουργώ ένα τετραγωνικό παράθυρο μήκους 1000 δειγμάτων όπως επίσης κι ένα παράθυρο hamming ίδιου μήκους μέσω των εντολών :

```
w_rec= ones(1,1000);
w_ham= hamming(1000);
```

Σκοπός του ερωτήματος αυτού είναι η παραθυροποίηση του σήματος του προηγούμενου ερωτήματος, δηλαδή του σήματος που είναι μέσα στον πίνακα ***s***.

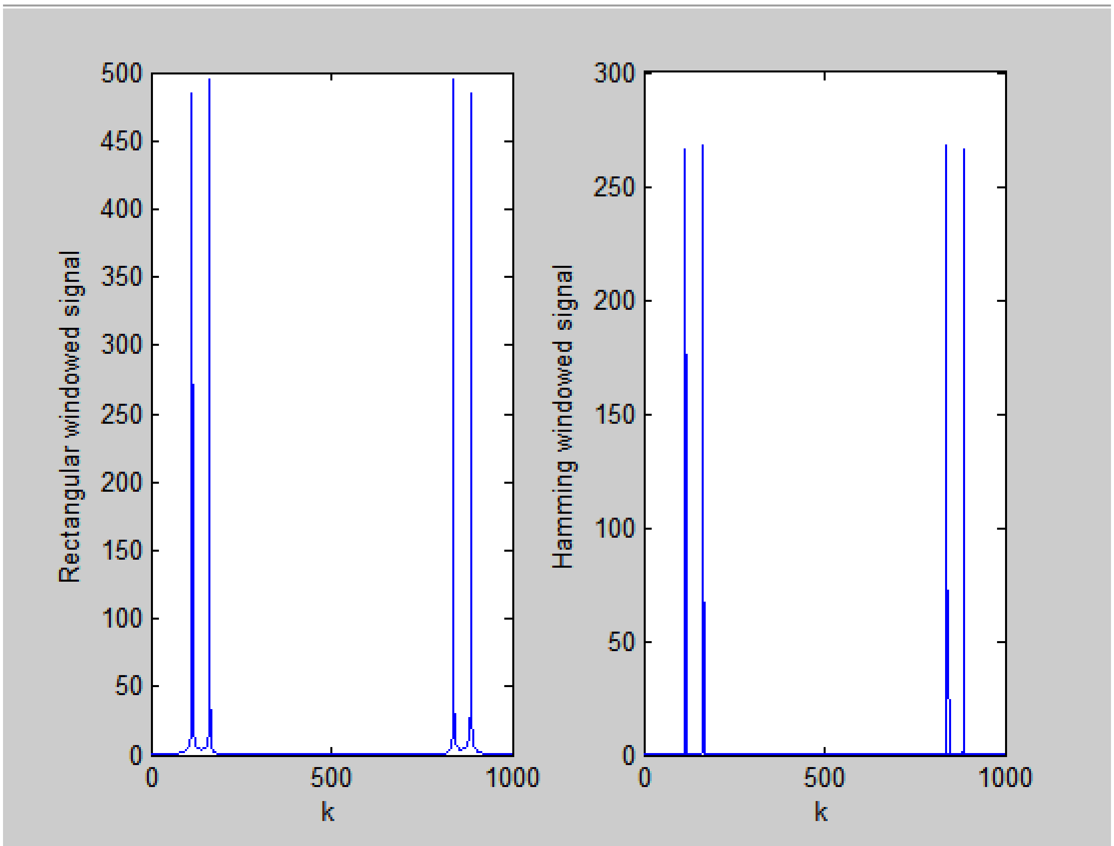
Για να το καταφέρω αυτό, αρχικά πολλαπλασιάζω κάθε τόνο του πίνακα ***s*** με ένα παράθυρο, *στοιχείο προς στοιχείο* αφού έχουν ίσο μήκος. Αυτό το κάνω και για τους 8 τόνους του πίνακα s και για τα δυο είδη παραθύρων που ανέφερα παραπάνω.

Μια αφηρημένη αναπαράσταση του κώδικα που γράφτηκε στο Matlab για τη δημιουργία των παραθυροποιημένων σημάτων είναι η εξής:

```
Παραθυροποιημένο_Σήμα_1 = Αρχικό_Σήμα(1:1000).*Παράθυρο
Παραθυροποιημένο_Σήμα_2 = Αρχικό_Σήμα(1101:2100).*Παράθυρο
…
Παραθυροποιημένο_Σήμα_8 = Αρχικό_Σήμα(7701:8700).*Παράθυρο
```

Συνολικά προκύπτουν *<u>16 παραθυροποιημένα σήματα</u>* αφού έχω 2 είδη παραθύρων, των οποίων εγώ παίρνω τον Μετ/σμό Fourier με χρήση εκ νέου της συνάρτησης fft() του Matlab.

Ωστόσο για να μπορέσουμε να εξάγουμε κάποιο συμπέρασμα σχετικά με το ποιο παράθυρο από τα δυο δίνει καλύτερο αποτέλεσμα, παρουσιάζεται παρακάτω το πρώτο παραθυροποιημένο σήμα που προέκυψε τόσο από το τετραγωνικό παράθυρο, όσο και από το παράθυρο hamming:



Εικόνα 1.3.1: Παραθυροποιημένα σήματα από τετραγωνικό παράθυρο (αριστερά) και παράθυρο hamming (δεξιά)

Συμπέρασμα: Συγκρίνοντας τα δυο γραφήματα *καταλήγω ότι το παράθυρο hamming δίνει ακριβέστερα αποτελέσματα σε σχέση με το τετραγωνικό παράθυρο.*

Ερώτημα 1.5. :

Στο ερώτημα αυτό δανείζομαι τη σχέση του ερωτήματος 1.2. Λύνοντας ως προς k προκύπτει η σχέση: $k = \frac{1000\omega}{2\pi}$

Χρησιμοποιώντας την παραπάνω σχέση μπορώ πλέον να αντιστοιχίσω τις χαμηλές και τις υψηλές συχνότητες του αρχικού πίνακα σε αντίστοιχους δείκτες k.

Έτσι υπολογίζονται για τις χαμηλές και για τις ψηλές συχνότητες του πίνακα 1 οι αντίστοιχοι δείκτες k. Εφαρμόζοντας τον τύπο αυτό 7 φορές βρίσκω τα ζητούμενα k, τα οποία φαίνονται αναλυτικά στον επόμενο πίνακα:

Συχνότητες γραμμής		Συχνότητες στήλης	
Ω_{row}	k_{row}	Ω_{col}	k_{col}
0.5346	86	0.9273	149
0.5906	95	1.0247	164
0.6535	105	1.1328	181
0.7217	116		

Πίνακας 1.5.1: Πίνακας αντιστοίχισης συχνότητων Ω σε δείκτες k

Στον κώδικα Matlab που υπάρχει στο συμπιεσμένο αρχείο έχω κάνει το εξής για την υλοποίηση του ερωτήματος 1.5:

- Δημιούργησα ένα νέο πίνακα ***s_new*** οποίος περιέχει τους τόνους του touch-pad με την εξής σειρά: ***s_new=[1234567890]*** όπου ανάμεσα σε κάθε τόνο υπάρχουν 100 μηδενικά δείγματα για να υπάρχει ομοιομορφία με το *ερώτημα 1.3* .
- Στη συνέχεια πολλαπλασιάζω κάθε τόνο του πίνακα αυτού με ένα παράθυρο hamming μήκους 1000 δειγμάτων και δημιουργώ 10 παραθυροποιημένα σήματα των οποίων βρίσκω τον DFT μέσω της συνάρτησης fft() και τον αποθηκεύω στις μεταβλητές με ονόματα `fft_ham_tone1 έως fft_ham_tone10.` Θα πρέπει να τονιστεί ότι επέλεξα ως παράθυρο το hamming αφού όπως ανέφερα και παραπάνω δίνει καλύτερα αποτελέσματα σε σχέση με το τετραγωνικό.
- Παρατηρώντας όλα τα διαγράμματα των παραπάνω μεταβλητών, καταλήγω ότι μια συνθήκη ώστε να πάρω τα ζητούμενα k είναι να αναθέσω στο Matlab να μου βρει όλα τα k για τα οποία το πλάτος των σημάτων αυτών είναι μεγαλύτερο από το 205(αποτελεί μια κατά προσέγγιση τιμή η οποία όμως δίνει το σωστό αποτέλεσμα και προέκυψε μετά από λεπτομερή παρατήρηση). Αυτό το κάνει με την εντολή `p1= find(abs(fft_ham_tone1)>205);` (αυτή η εντολή είναι μονο για τον τόνο 1) Έτσι δημιουργούνται τα p1 έως p10 τα οποία είναι διανύσματα γραμμή και έχουν 4 στοιχεία το καθένα. Ωστόσο επειδή έχω τις θεωρητικές τιμές των k στον πίνακα 1.5.1 οι οποίες είναι όλες μικρότερες από 200 κι επειδή ο DFT είναι συμμετρικός μπορώ να ισχυριστώ ότι τα ζητούμενα k για κάθε τόνο είναι αυτά που είναι μικρότερα από 500 το οποίο είναι μια πιο γενική συνθήκη. Άρα με ένα for-loop για κάθε μεταβλητή p αποθηκεύω τις 2 από τις 4 τιμές του k (αυτά που είναι μικρότερα από 500) της σε ένα πίνακα ***k_list***.
- Τελικά ο πίνακας ***k_list*** είναι μια λίστα από k όπου ανά δυο τιμές θα είναι τα ζητούμενα k για κάθε τόνο που βρίσκετε μέσα στον πίνακα ***s_new***. Επίσης στις περιττές θέσεις του πίνακα x είναι τα k που αντιστοιχούν στις χαμηλές συχνότητες(γραμμές του πίνακα 1), ενώ στις άρτιες τα k που αντιστοιχούν στις υψηλές συχνότητες(στήλες του πίνακα 1).
- Έχει δημιουργηθεί για πληρότητα και ο πίνακας ***k_list_new*** ο οποίος περιέχει τις αντίστοιχες κυκλικές συχνότητες, αφού χρησιμοποιήθηκε η γνωστή σχέση $k = \frac{1000\omega}{2\pi}$. Οι κυκλικές συχνότητες που προκύπτουν παρατηρώ ότι είναι πολύ κοντά με τις θεωρητικές που φαίνονται στον πίνακα 1 και πολλές φορές ταυτίζονται με αυτές.

<u>Ερώτημα 1.6. :</u>

Στο ερώτημα αυτό δημιουργώ το μια συνάρτηση με το όνομα ***ttdecode()*** η οποία δέχεται ως είσοδο ένα τονικό σήμα με το όνομα *signIn* και επιστρέφει ένα διάνυσμα με τα αντίστοιχα ψηφία του τονικού σήματος. Ας εξετάσουμε όμως τι κάνει βήμα προς βήμα η συνάρτηση αυτή :

- Αρχικά επειδή το τονικό σήμα εισόδου δεν ξέρω πως θα έχει δοθεί, δηλαδή αν υπάρχουν στην αρχή μηδενικά ή αν υπάρχει κάποια άλλη ιδιομορφία, δημιουργώ ένα πίνακα με όνομα ***ShowBegin*** ο οποίος δείχνει που είναι το πρώτο δείγμα κάθε τόνου. Η μόνη σύμβαση που έχει γίνει εδώ είναι ότι από τη στιγμή που θα βρει ένα μη μηδενικό δείγμα τότε από το σημείο αυτό ξεκινάει ένα τονικό σήμα που θα έχει μήκος 1000 δείγματα. Προφανώς το μέγεθος του πίνακα ***ShowBegin*** ισούται με το πλήθος των τόνων του σήματος εισόδου.
- Στη συνέχεια κάθε τόνο τον βάζουμε σε ένα πίνακα με σκοπό να τον επεξεργαστούμε. Υπολογίζουμε τον DFT κάθε τόνου και κρατάμε όπως και στο προηγούμενο ερώτημα το $1^{ο}$ μέχρι και το $500^{στό}$ δείγμα, λόγω συμμετρίας του DFT. Το σήμα αυτό προφανώς στο πεδίο της συχνότητας θα έχει 2 σημεία στα οποία θα εμφανίζει μέγιστο πλάτος τα οποία θα αντιστοιχούν στις 2 διαφορετικές συχνότητες των δυο ημιτόνων τα οποία όταν αθροιστούν αποτελούν το σήμα. Τα σημεία αυτά τα βρίσκω με την βοήθεια της συνάρτησης ***find()*** του Matlab.
- Στο σημείο αυτό έχω υπολογίσει τους δυο δείκτες k που αντιστοιχούν σε μια γραμμή και μια στήλη του πίνακα 1.5.1. Για να μπορέσω να βρω ποιο πλήκτρο αντιστοιχεί στους 2 αυτούς δείκτες δημιουργώ μια βοηθητική συνάρτηση ***Search()*** η οποία παίρνει ως ορίσματα αυτούς τους 2 δείκτες και επιστρέφει ποιο πλήκτρο αντιστοιχεί σε αυτούς. Η υλοποίηση αυτής της συνάρτησης είναι πολύ απλή αφού γνωρίζω τις τιμές των δεικτών k και ουσιαστικά πρόκειται για μια απλή εφαρμογή της εντολής **switch-case** όπως φαίνεται και στον κώδικα σε Matlab.
- Τέλος η έξοδος της συνάρτησης ***Search()*** αποθηκεύεται σε ένα νέο διάνυσμα με το όνομα *Vector*, το οποίο είναι και η έξοδος της συνάρτησης ***ttdecode***() και περιέχει τα ψηφία του τονικού σήματος εισόδου.

Για να διαπιστωθεί αν η συνάρτηση ***ttdecode()*** λειτουργεί σωστά θέτω σαν είσοδο το σήμα που δημιούργησα στο ερώτημα 1.3 και το αποτέλεσμα που λαμβάνω είναι το εξής:

Vector = 0 6 2 2 4 1 0 0

Άρα το αποτέλεσμα της συνάρτησης είναι αυτό που περίμενα, γεγονός το οποίο επαληθεύει την ορθή λειτουργία της.

Ερώτημα 1.7. :

Στο τελευταίο ερώτημα της πρώτης άσκησης ουσιαστικά βάζω ως είσοδο δυο άλλα σήματα στην συνάρτηση ***ttdecode()***. Συγκεκριμένα φορτώνουμε το αρχείο *"my_touchtones.mat"* από το συμπληρωματικό υλικό της άσκησης με χρήση της εντολής ***load()***. Πλέον δυο νέα σήματα εμφανίστηκαν στο workspace με ονόματα ***easySig*** και ***hardSig***. Εισάγω καθένα από τα δυο σήματα στη συνάρτηση ***ttdecode()*** και παίρνω ως αποτέλεσμα τα εξής:

Για το ***easySig***:

2 1 1 8 5 0 8 0 6 9

Για το ***hardSig***:

6 6 2 2 6 6 2 2 6 6 0 0 0 0 9 9 3 3 9 9 3 3 9 9 0 0 2 2

Μέρος 2ο

Φασματική Ανάλυση και Ανίχνευση Ημιτονοειδών με τον Διακριτό Μετασχηματισμό Fourier (DFT)

Σκοπός της άσκησης αυτής είναι η κατανόηση και ο πειραματισμός με την έννοια της *Διακριτικής Ικανότητας* ενός συστήματος. Δύο πολύ σημαντικά στοιχεία που πρέπει να χρησιμοποιήσουμε για να το καταφέρουμε αυτό είναι ο διακριτός μετασχηματισμός Fourier (DFT) και τα "παράθυρα" που χρησιμοποιούνται για την ανάλυση των σημάτων.

Ο Διακριτός μετασχηματισμός Fourier προκύπτει από δειγματοληψία του μετασχηματισμού Fourier διακριτού χρόνου (DTFT) στις συχνότητες $\omega = 2\pi k / N$ με 0<k<N-1 όπου N το μήκος του DTF που θέλουμε. Ο Διακριτός μετασχηματισμός Fourier (DFT) δίνεται λοιπόν από τον ακόλουθο τύπο:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}nk}, 0 \le k \le N-1$$

Όσον αφορά τα παράθυρα, πρέπει να αναφερθεί πως στο μεγαλύτερο μέρος αυτής της εργαστηριακής άσκησης θα κάνουμε χρήση του παραθύρου τύπου Hamming. Ο μαθηματικός τύπος που το ορίζει είναι ο εξής:

$$w_{hamm}[n] = 0.54 - 0.46\cos\left(2\pi n / N\right)$$

Για τη μελέτη και την κατανόηση της έννοιας της διακριτικής ικανότητας ενός συστήματος θα κάνουμε χρήση δύο μιγαδικών ημιτονοειδών σημάτων των οποίων τα πλάτη και οι συχνότητες παίρνουν κατά την πορεία της άσκησης κατάλληλες τιμές(λειτουργούν ως παράμετροι) ικανές ώστε να μας δώσουν τα κατάλληλα συμπεράσματα όπως θα φανεί και στη συνέχεια. Τα δυο αυτά σήματα αθροίζονται και πολλαπλασιάζοντάς τα με το παράθυρο προκύπτει το σήμα $y[n] = w[n] \cdot (x_1[n] + x_2[n])$, όπου $x_1[n] = A_1 e^{j(\omega_1 n + \varphi_1)}$ και $x_2[n] = A_2 e^{j(\omega_2 n + \varphi_2)}$. Αυτό είναι το σήμα που θα επεξεργαστούμε στην πορεία.

Ερώτημα 2.1 :

Για το πρώτο ερώτημα της άσκησης αυτής υλοποιήσαμε μια συνάρτηση με όνομα ***Manual_DFT()*** που δέχεται 2 παραμέτρους, ένα σήμα και έναν ακέραιο N και μας επιστρέφει τον DFT N σημείων του σήματος που δώσαμε. Μες στη συνάρτηση έχει ληφθεί και η περίπτωση που χρειάζεται να γίνει η γνωστή διαδικασία *zero-padding* αν το μήκος του σήματος είναι μικρότερο του αριθμού N. Επίσης στο τέλος της συγγραφής της ρουτίνας αυτής ελέγξαμε αν δίνει ως αποτέλεσμα ίδιες ακριβώς τιμές με τη συνάρτηση **fft()** του

Matlab πράγμα που επαληθεύτηκε. Η συνάρτηση αυτή παρατίθεται μαζί με τους υπόλοιπους κώδικες.

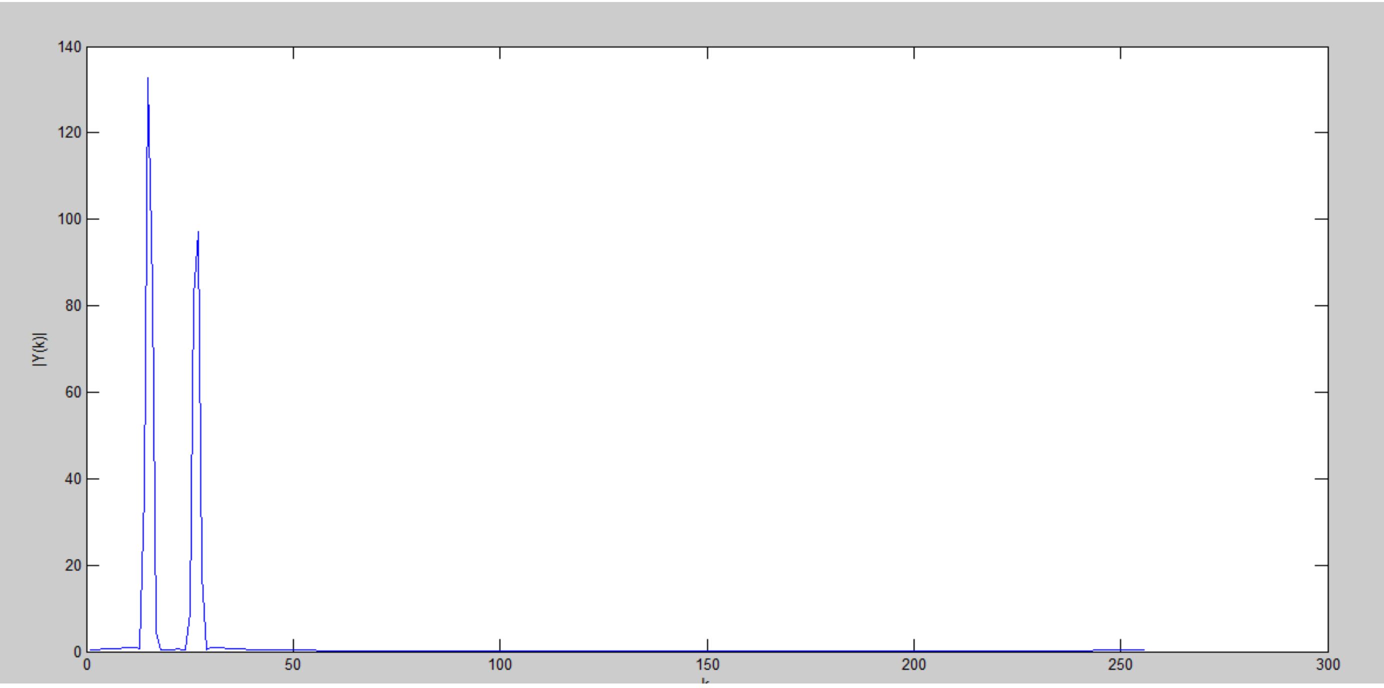
Ερώτημα 2.2:

Θέτουμε στα παραπάνω σήματα : $A_1 = 1, A_2 = 0.8$ (δίνεται)

$\omega_1 = \pi/9, \omega_2 = \pi/5$ (δίνεται)

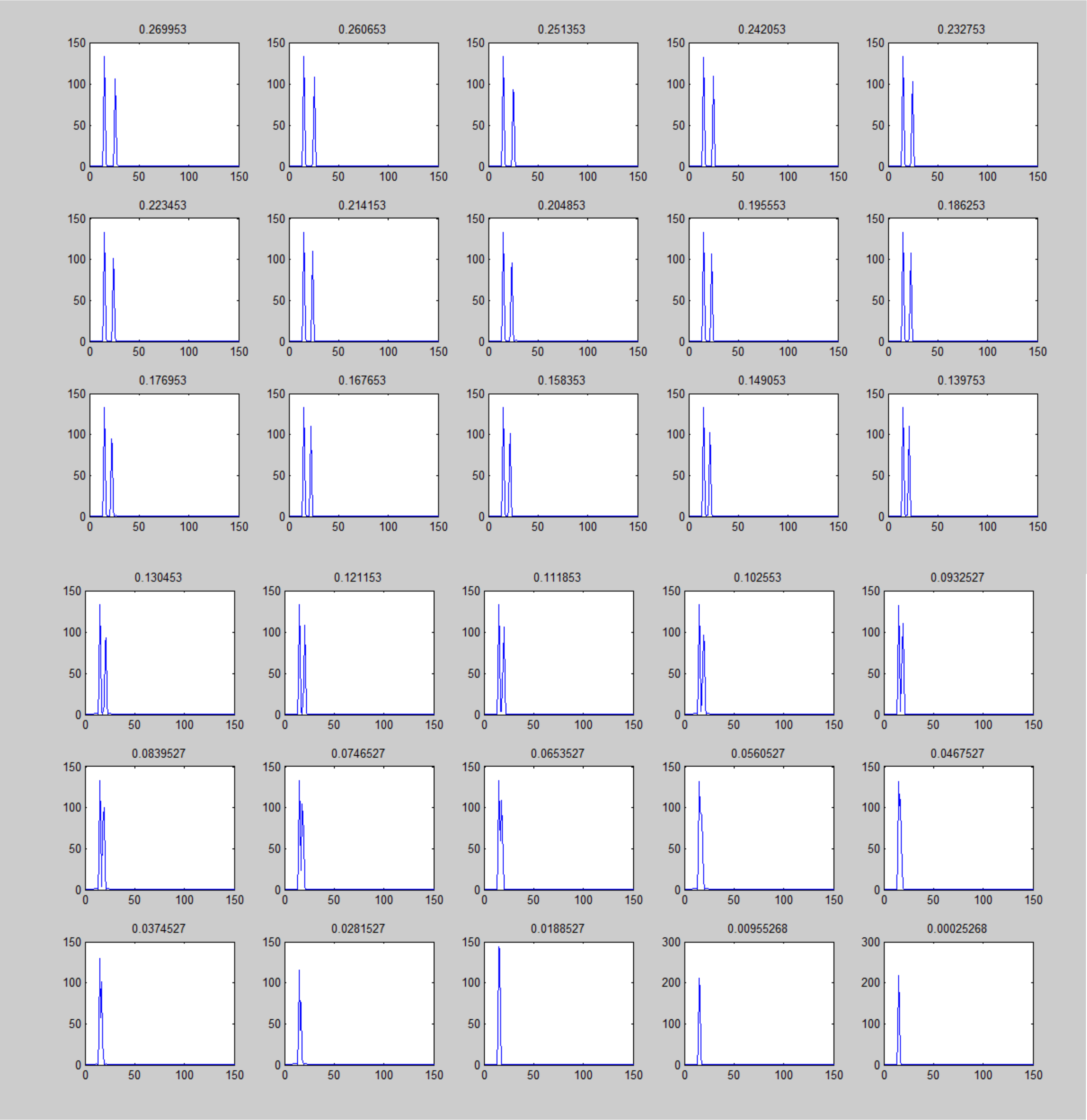
$\varphi_1 = \pi/5, \varphi_2 = \pi/36$ (αυθαίρετα)

Αθροίζουμε τα δύο σήματα (με τις παραπάνω τιμές των μεταβλητών) και περνώντας τα από το παράθυρο hamming με μήκος 256 δείγματα παίρνουμε το σήμα y που έχει και αυτό 256 δείγματα. Στη συνέχεια λαμβάνουμε τον DFT του σήματος y με χρήση της συνάρτησης ***fft()*** του Matlab και σχεδιάζουμε τη γραφική παράσταση του πλάτους του μετασχηματισμού αυτού η οποία παρουσιάζεται παρακάτω:



Στο παραπάνω σχήμα είναι εμφανής ο διαχωρισμός των δύο κορυφών των ημιτόνων. Ωστόσο, στο ερώτημα αυτό καλούμαστε να βρούμε την ελάχιστη τιμή της διαφοράς των συχνοτήτων των 2 ημιτονοειδών σημάτων , ώστε η διάκριση των κορυφών να είναι δυνατή, καθώς όπως θα δούμε **όσο ελαττώνουμε το Δω οι 2 κορυφές τείνουν να συγχωνευθούν σε μία**. Προκειμένου να φέρουμε εις πέρας αυτή την εργασία μεταβάλλουμε με μικρά βήματα την τιμή της συχνότητα ω2 φέρνοντας την πιο κοντά στην ω1. Εμείς για να το επιτύχουμε αυτό επιλέξαμε ως βήμα την τιμή 0.0093rad/sec $= \dfrac{\left(\pi/5 - \pi/9\right)}{30}$ (η επιλογή της οποίας εξηγείται και στον αντίστοιχο κώδικα). Επειδή παρόμοια με αυτή τη διαδικασία θα πρέπει να ακολουθηθεί και σε επόμενα ερωτήματα κατασκευάσαμε μια συνάρτηση ***plot_maker()*** που κατασκευάζει τα γραφήματα για κάθε τιμή του Δω και έτσι μας δίνει τη δυνατότητα να διακρίνουμε την οριακή τιμή του Δω. Η συνάρτηση αυτή λαμβάνει ως ορίσματα τον αριθμό N των δειγμάτων του DFT, την αρχική συχνότητα ω2 του σήματος που θέλουμε να

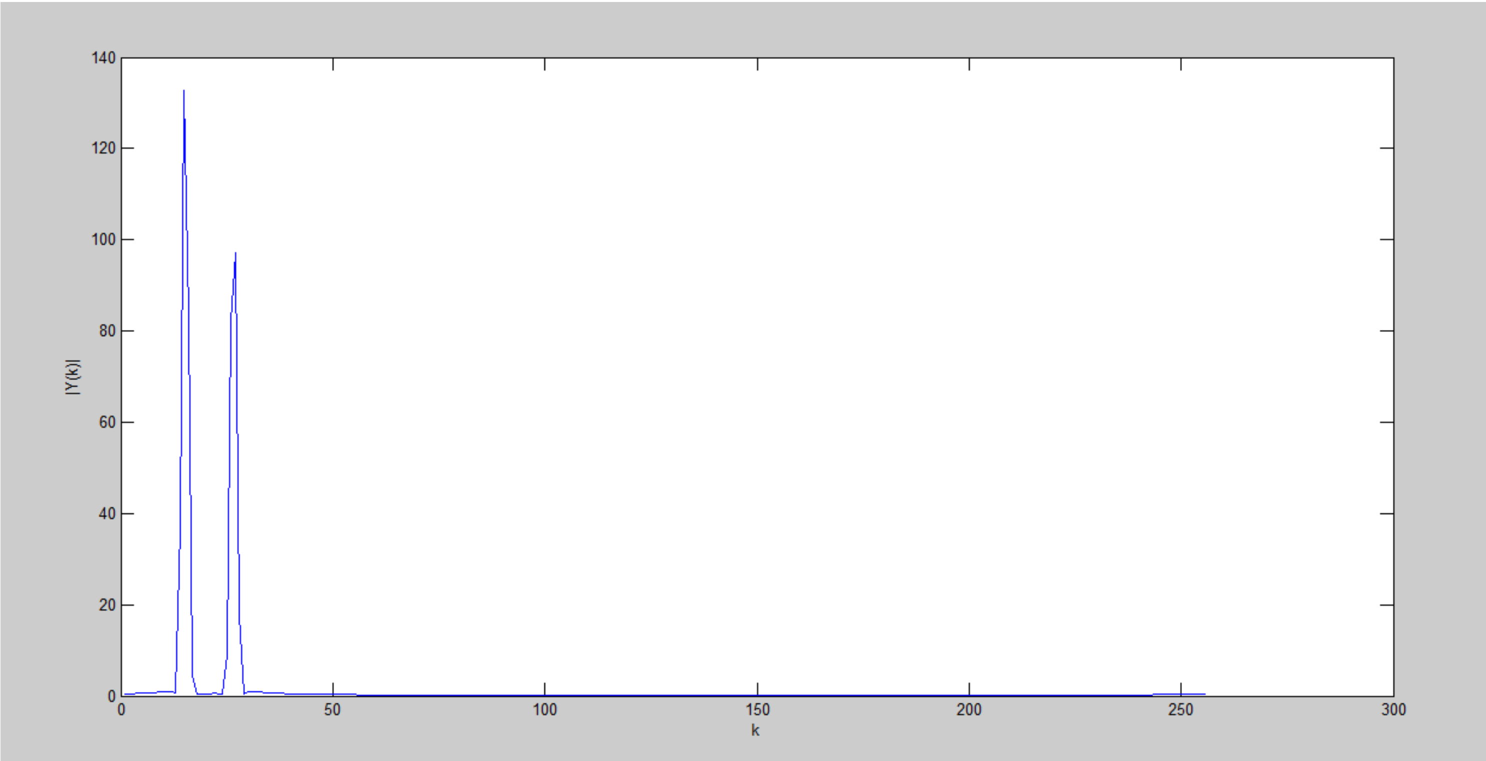
μεταβάλλουμε, το πρώτο σήμα και την τιμή του βήματος. Έτσι , στο ερώτημα αυτό η συνάρτηση ***plot_maker()*** μειώνει σε κάθε επανάληψη την τιμή του ω2 κατά 0.0093rad/sec και σχεδιάζει τη γραφική παράσταση ***αναγράφοντας πάνω από κάθε γράφημα την τιμή του Δω τη δεδομένη στιγμή***.
Έτσι λαμβάνουμε τα παρακάτω γραφήματα:



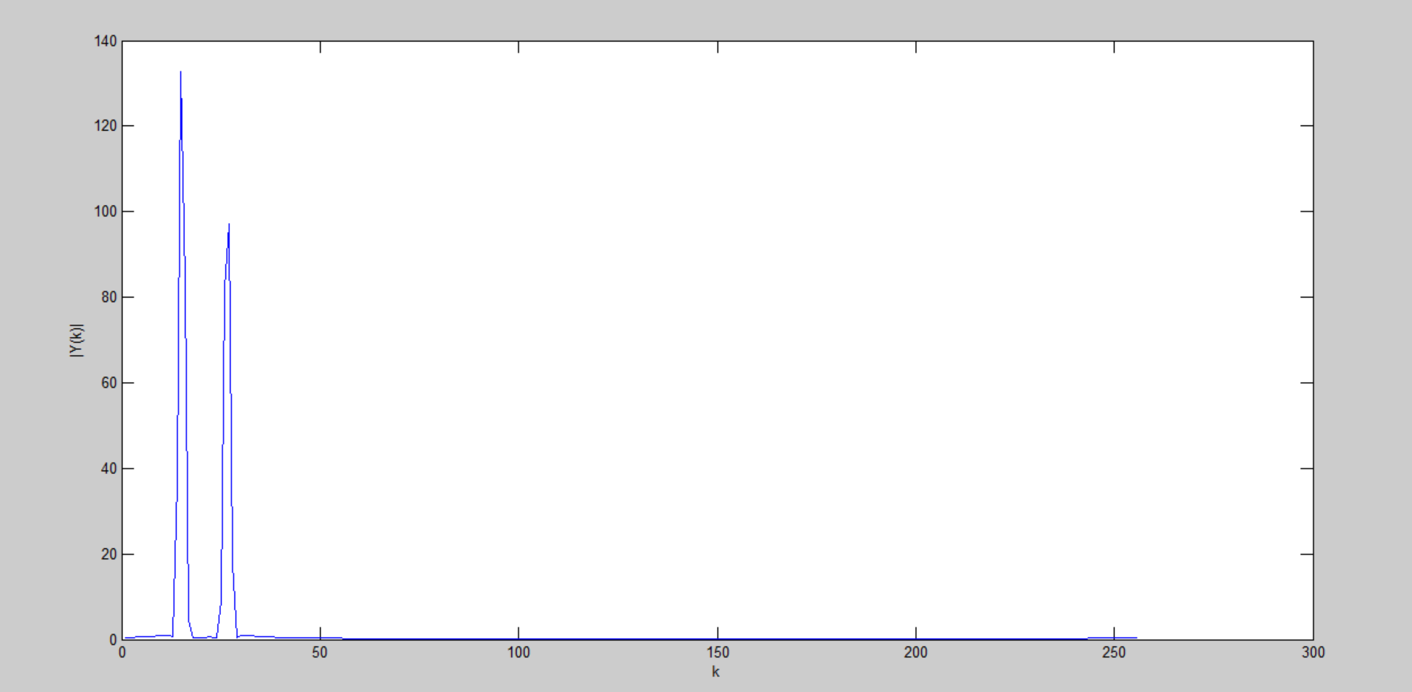
Κοιτώντας προσεχτικά τα παραπάνω διαγράμματα, είμαστε σε θέση να διαπιστώσουμε ότι <u>για τιμές του Δω μεγαλύτερες από 0.056rad/sec γίνεται διάκριση των κορυφών ενώ για μικρότερες του 0.056rad/sec υπάρχει επικάλυψη και καθιστά αδύνατη τη διάκριση των δύο κορυφών (γίνονται μία).</u>

<u>Ερώτημα 2.3 :</u>

Σε αυτό το ερώτημα της άσκησης, ακολουθούμε την ίδια ακριβώς διαδικασία με το ερώτημα 2.2, με μόνη διαφορά το μήκος του DFT. Εδώ το μήκος του ζητούμενου DFT είναι μεγαλύτερο από το μήκος του σήματός μας με αποτέλεσμα να πρέπει να χρησιμοποιήσουμε τη διαδικασία του zero-padding (προσθέτουμε μηδενικά δείγματα στο σήμα μας μέχρι το μήκος του να γίνει ίσο με το μήκος του ζητούμενου DFT). Εδώ θα μελετήσουμε πως αυτή η διαφοροποίηση θα επηρεάσει την ευκρίνεια και την ικανότητα φασματικής διάκρισης του συστήματος μας.
Στο πρώτο σκέλος ζητάμε να βρούμε τον DFT 512 σημείων. Έτσι εφαρμόζουμε zero-padding στο σήμα μας και αποκτά και αυτό 512 δείγματα. (Υπενθυμίζεται πως το παράθυρο απ’ το οποίο περνάμε το σήμα μας εξακολουθεί να έχει μήκος 256 δειγμάτων)
Παρακάτω παρουσιάζεται όπως προέκυψε απ’ το Matlab ο DFT του σήματος αυτού (οι αρχικές παράμετροι παραμένουν ίδιες με αυτές του προηγούμενου ερωτήματος).



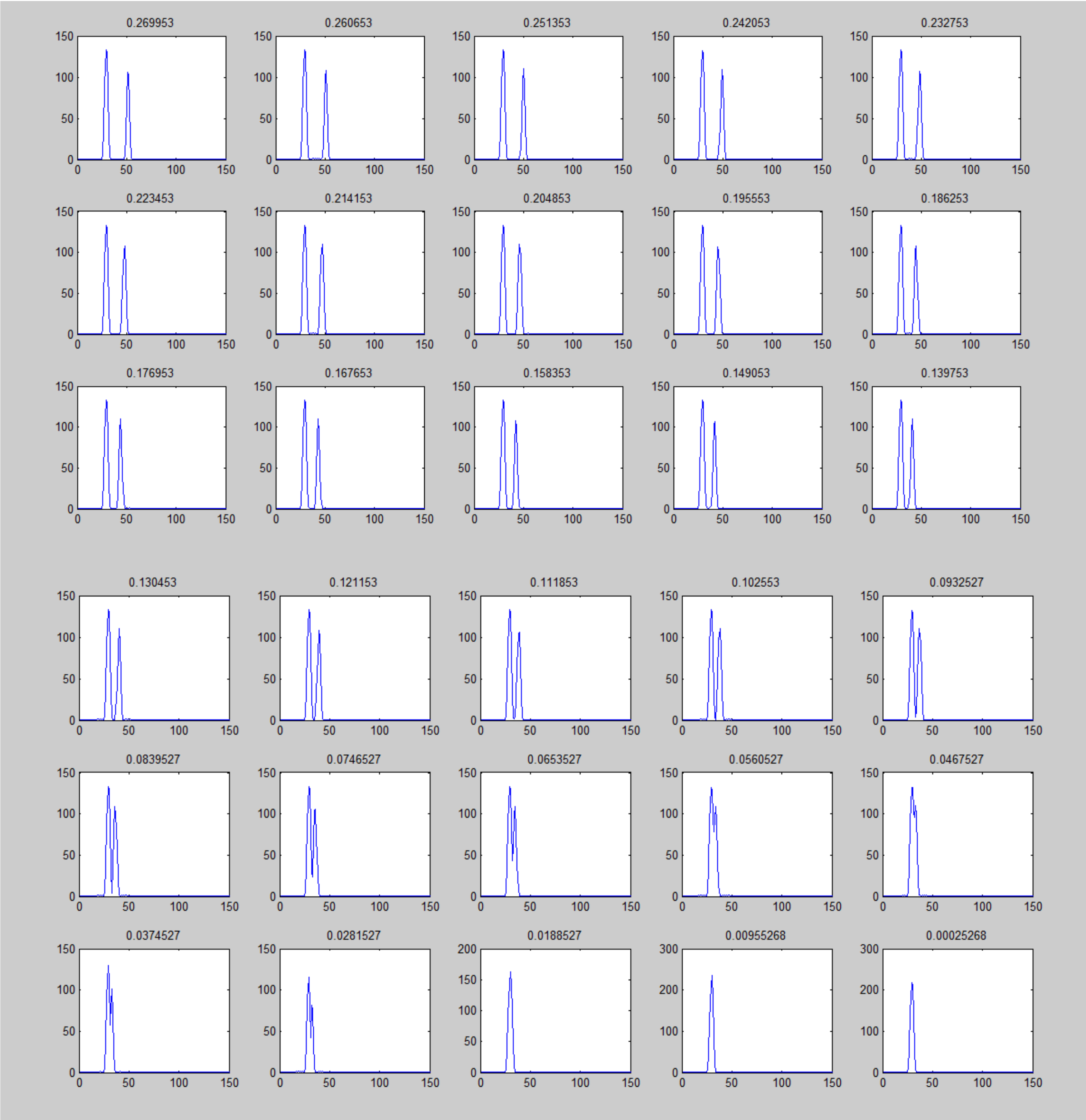
Την ίδια ακριβώς διαδικασία ακολουθούμε για να υπολογίσουμε και τον DFT 1024 σημείων (που ζητείται στο 2ο σκέλος του ερωτήματος αυτού)του παραθυροποιημένου σήματος με αρχικό μήκος 256 δείγματα. Το μέτρο αυτού του DFT 1024 σημείων ανάλογα με την τιμή του k παρατίθεται ακριβώς από κάτω:



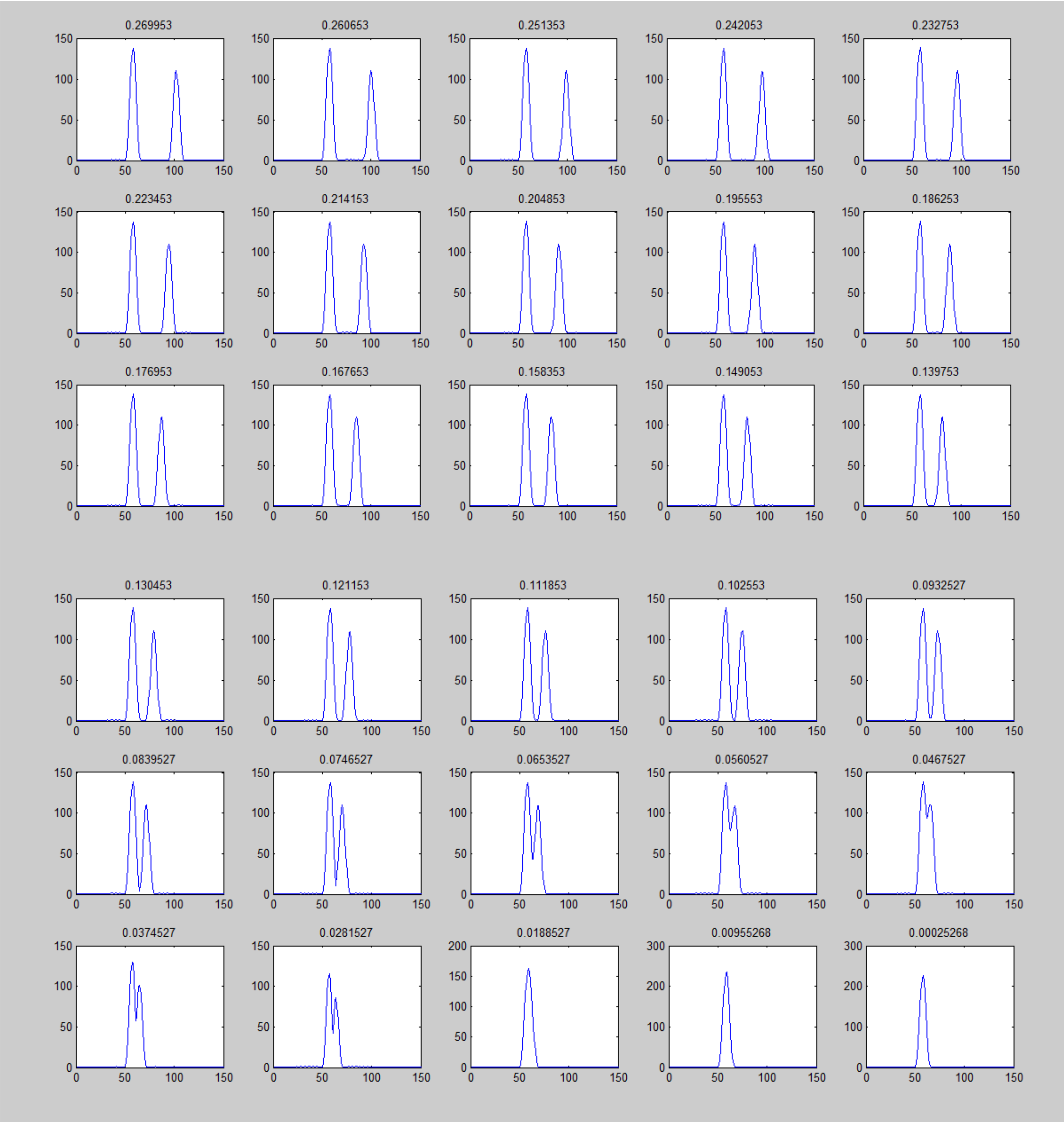
Κάνοντας μια σύγκριση μεταξύ των γραφικών παραστάσεων των 3 DFT του ίδιο σήματος y[n] κάνουμε τη διαπίστωση πως όσο περισσότερα δείγματα έχει ο DFT μας, τόσο πιο “καθαρή” και ευκρινής είναι η αναπαράσταση του Διακριτού Μετασχηματισμού Fourier. Επομένως είναι προτιμότερο (από άποψη ευκρίνειας της αναπαράστασης του φάσματος) να εφαρμόζουμε την τεχνική του Zero-Padding και να λαμβάνουμε DFT με μήκος μεγαλύτερο από το μήκος του ίδιου του σήματος .

Ωστόσο, στο ερώτημα αυτό πέρα από την ευκρίνεια θέλουμε να ελέγξουμε και τον τρόπο που επηρεάζει το μήκος του DFT την ικανότητα φασματικής διάκρισης. Γι’ αυτό ακολουθούμε την ίδια διαδικασία με το ερώτημα 2.2 μόνο που λαμβάνουμε DFT 512 αρχικά και 1024 δειγμάτων στη συνέχεια. Αυτό υλοποιείται ξανά με τη συνάρτηση ***plot_maker()*** με μόνη διαφορά τον αριθμό N που δίνεται ως όρισμα κατά την κλήση της. Η διαδικασία του zero-padding υλοποιείται μες σε αυτή τη συνάρτηση. Το βήμα με το οποίο μεταβάλλουμε τη συχνότητα ω2 εξακολουθεί να παίρνει την τιμή 0.0093 rad/sec. Παρακάτω παρατίθενται τα διαγράμματα για τις διάφορες τιμές της διαφοράς των συχνοτήτων Δω:

- DFT 512 δειγμάτων:



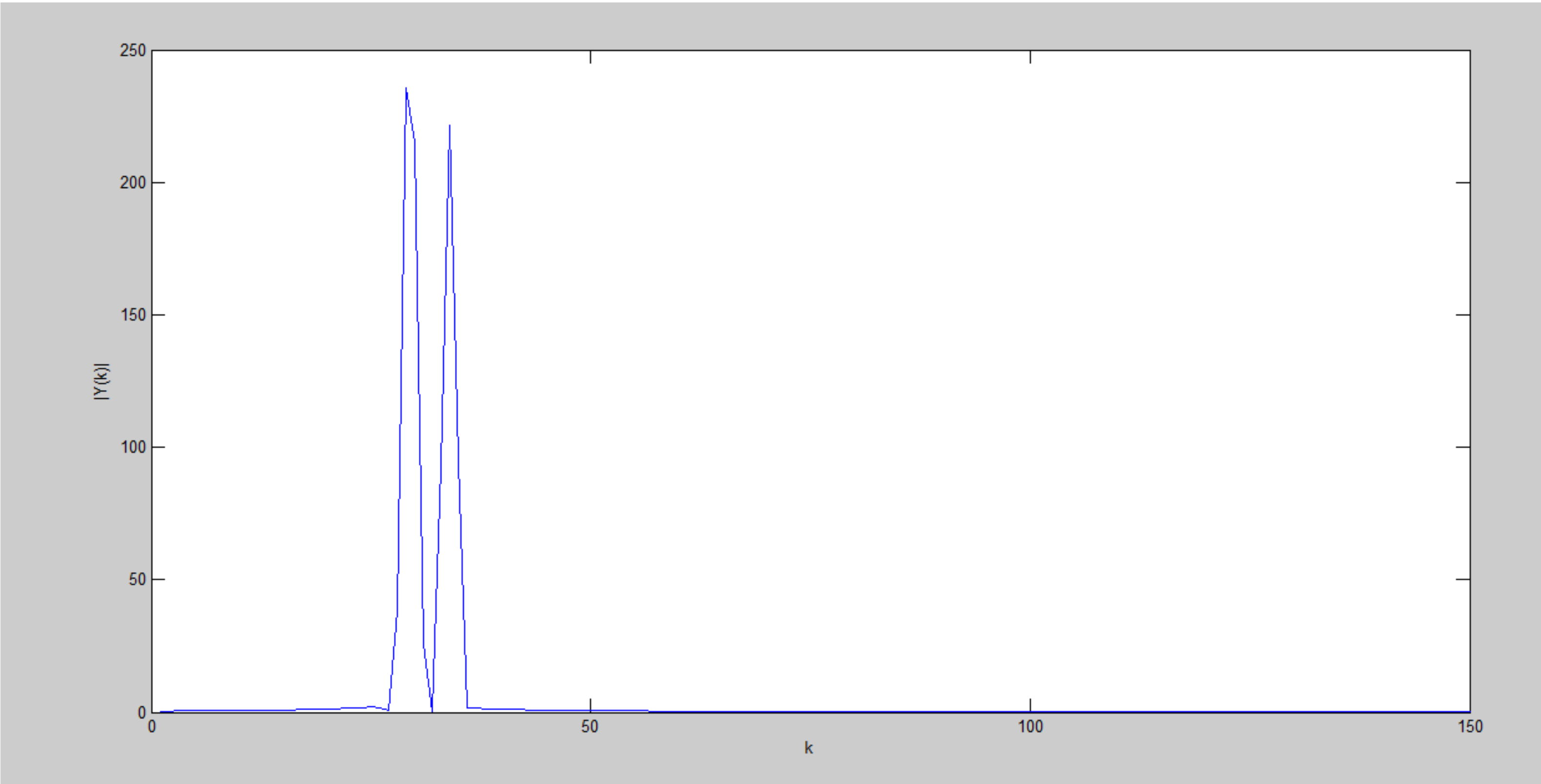
- DFT 1024 σημείων:



Με βάση τα παραπάνω διαγράμματα μπορούμε να παρατηρήσουμε ότι παρόλο που αλλάξαμε τον αριθμό δειγμάτων του DFT, δεν παρατηρούμε διαφοροποίηση στην οριακή διαφορά συχνοτήτων για την οποία μπορούμε να διακρίνουμε τις δύο κορυφές. Πάλι κυμαίνεται περίπου στα 0.056rad/sec. Επομένως είμαστε σε θέση να συμπεράνουμε πως μπορούμε να αυξάνουμε τον αριθμό των δειγμάτων του DFT κάνοντας χρήση της διαδικασίας zero-padding στο αρχικό σήμα και αυτό έχει ως αποτέλεσμα υψηλότερη ευκρίνεια στη σχεδίαση του φάσματος και ταυτόχρονα δεν ασκεί καμία επιρροή στη δυνατότητα φασματικής διάκρισης του συστήματός μας.

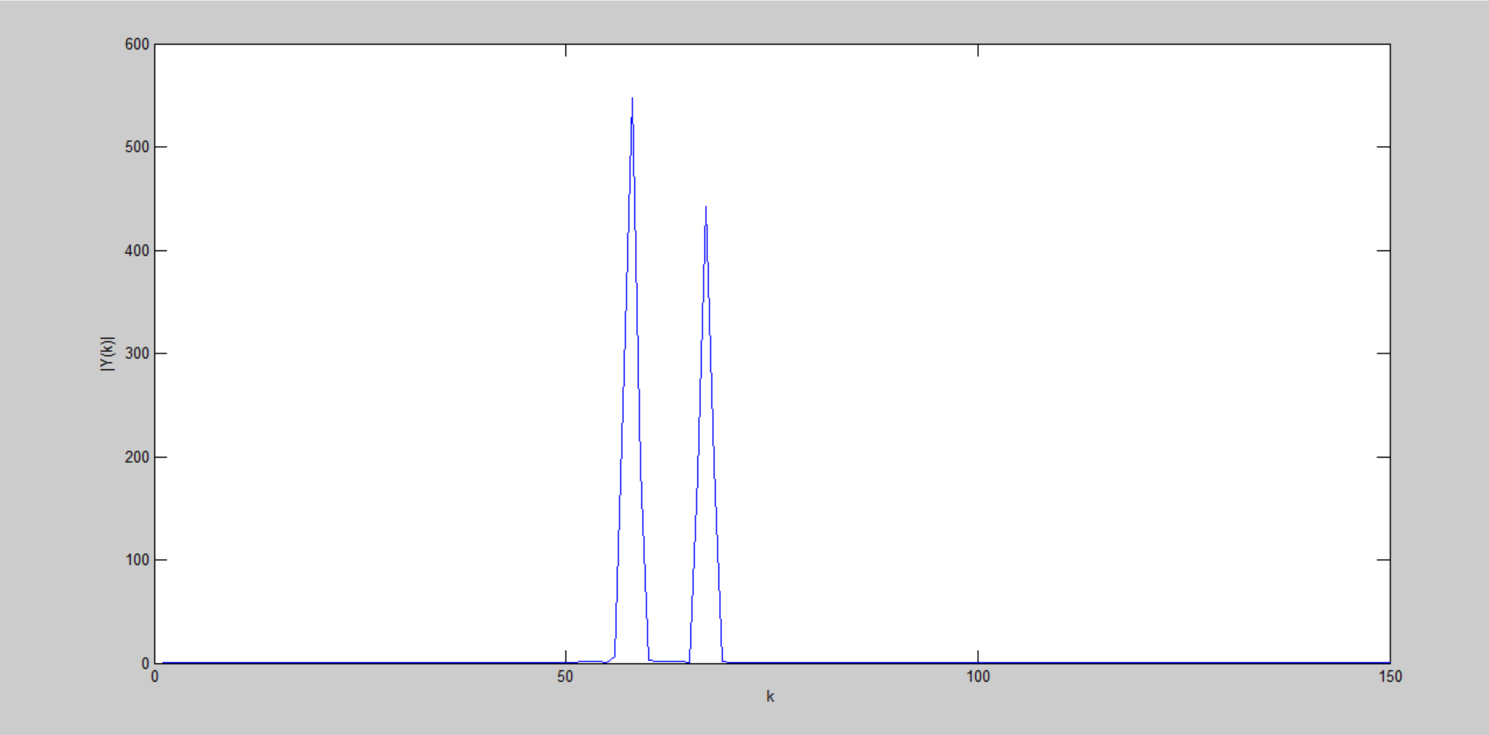
<u>Ερώτημα 2.4:</u>

Στο 4^{ο} αυτό ερώτημα της άσκησης, σκοπός μας είναι να δούμε πως και αν επηρεάζει ο αριθμός των δειγμάτων του παραθυροποιημένου σήματος την ικανότητα φασματικής διάκρισης. Για να το μελετήσουμε αυτό χρησιμοποιούμε μεγαλύτερο παράθυρο hamming. Επίσης στο ερώτημα αυτό θέτουμε τη συχνότητα του 2^{ου} ημιτονοειδούς σήματος ίση με την οριακή όπως βρέθηκε παραπάνω, δηλαδή ω2=ω1+0.056 rad/s, διατηρώντας την ω1 σταθερή. Τώρα εισάγουμε το σήμα μας αυτό σε ένα παράθυρο 512 δειγμάτων. Παρακάτω παρατίθεται ο DFT 512 σημείων στην περίπτωση αυτή:



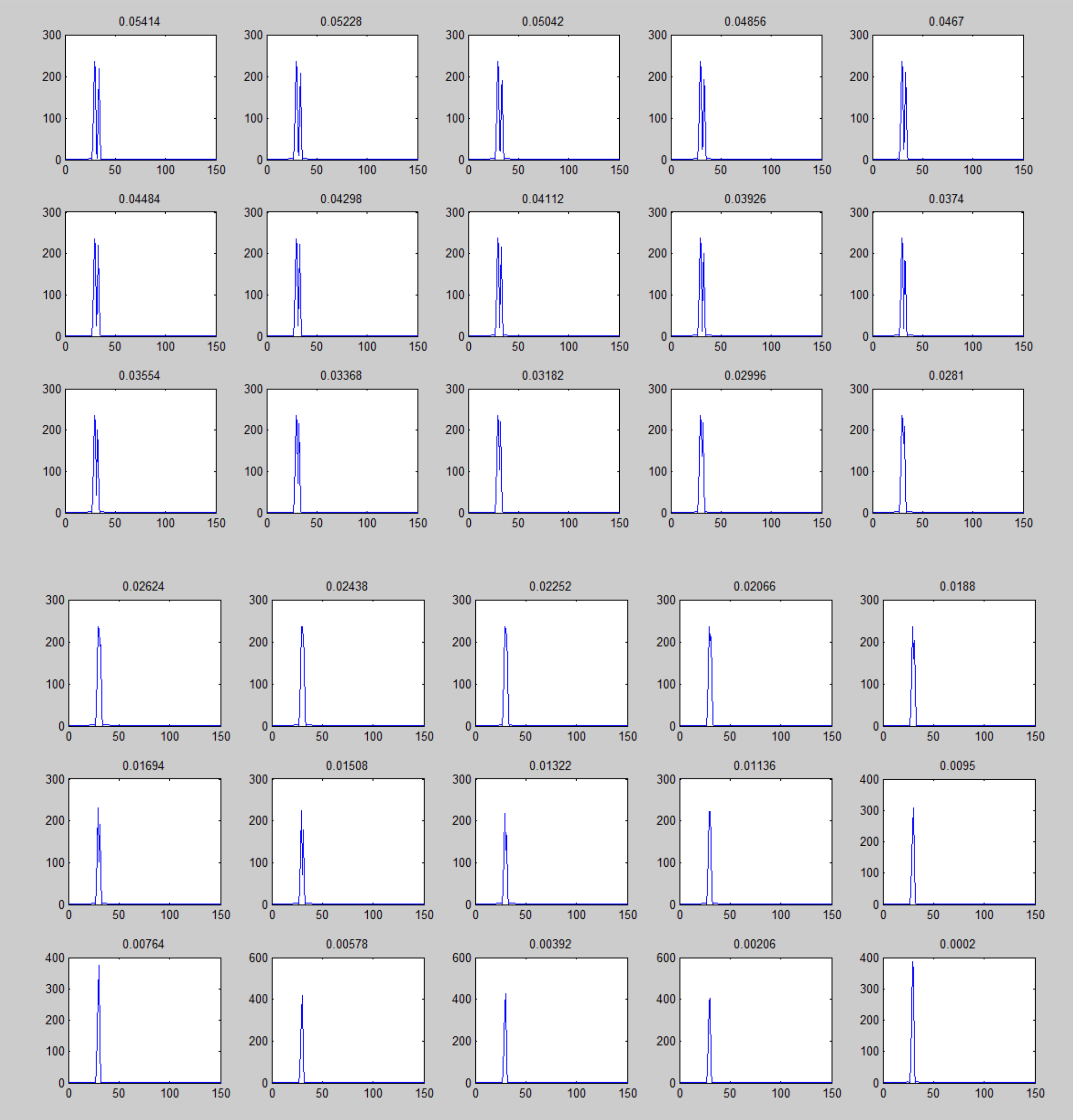
Βλέπουμε πως τώρα που μεταβάλλαμε τον αριθμό των δειγμάτων του αρχικού μας σήματος, οι 2 κορυφές είναι ξεκάθαρα διαχωρισμένες. <u>Πλέον η συχνότητα αυτή δε δίνει οριακή διάκριση του φάσματος</u>.

Αντίστοιχα, ο DFT 1024 σημείων του σήματος για παράθυρο hamming 1024 σημείων και για συχνότητα ω2 ίση με την οριακή που βρέθηκε προηγουμένως είναι ο ακόλουθος:

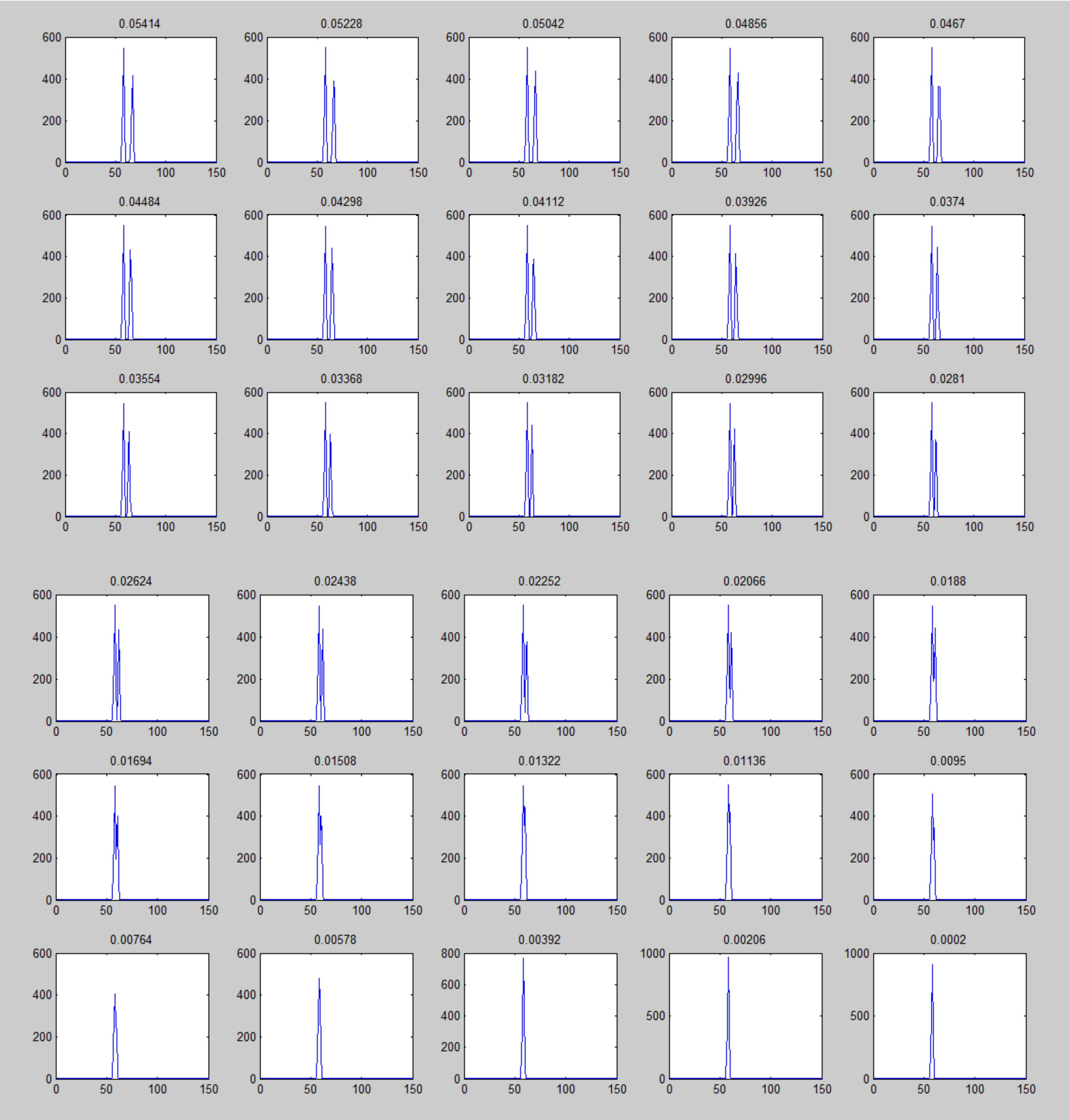


Παρατηρώντας προσεχτικά τα παραπάνω διαγράμματα μπορούμε να συμπεράνουμε πως όσο μεγαλύτερος είναι ο αριθμός των δειγμάτων του αρχικού σήματος που επεξεργαζόμαστε, τόσο καλύτερα και ευκολότερα μπορούμε να διακρίνουμε τις δύο ξεχωριστές κορυφές των ημιτόνων. Παρατηρούμε πλέον πως η παλιά "οριακή" διαφορά συχνοτήτων δεν ισχύει πια. Γι' αυτό στη συνέχεια καλούμαστε να βρούμε την νέα αυτή "οριακή" Δω. Αυτό επιτυγχάνεται με τη χρήση της συνάρτησης ***plot_maker()*** πάλι δίνοντας ως ορίσματα την παλιά οριακή συχνότητα ω2=ω1+0.056, τον αριθμό N των δειγμάτων του DFT (την $1^{η}$ φορά 512 και τη $2^{η}$ 1024), το σήμα x1 και το βήμα το οποίο αλλάζει σε αυτή την περίπτωση και γίνεται 0.056/30=0.00186 rad/sec. Με αυτές τις παραμέτρους λαμβάνουμε τα ακόλουθα σετ διαγραμμάτων:

- Για DFT 512 σημείων σε παραθυροποιημένο σήμα 512 σημείων:



- Για DFT 1024 σημείων σε παραθυροποιημένο σήμα 1024 σημείων:



Από τα σετ γραφικών παραστάσεων των 2 προηγούμενων σελίδων μπορούμε να βγάλουμε το συμπέρασμα πως στην περίπτωση του αρχικού σήματος με 512 δείγματα η οριακή διαφορά συχνοτήτων για τη διάκριση των δυο κορυφών είναι 0.02624rad/sec ενώ αντίστοιχα για την περίπτωση του αρχικού σήματος με τα 1024 δείγματα η οριακή Δω είναι περίπου 0.01508rad/sec.
Από τις παραπάνω παρατηρήσεις είμαστε σε θέση να βγάλουμε το συμπέρασμα πως όσο περισσότερα δείγματα του αρχικού σήματος έχουμε τόσο ευκολότερη είναι η διάκριση δυο ημιτόνων για πολύ κοντινές συχνότητες. Επομένως εάν έχουμε τόνους που βρίσκονται πολύ κοντά από άποψη συχνότητας πρέπει να λαμβάνουμε αρκετά δείγματα του αρχικού σήματος προκειμένου να έχουμε καλύτερη ικανότητα φασματικής διάκρισης αφού αυτά τα δύο μεγέθη είναι ανάλογα όπως παρατηρήσαμε.

<u>Ερώτημα 2.5:</u>

Φτάνοντας στο τελευταίο ερώτημα της άσκησης, θέλουμε να δούμε πως μεταβάλλεται το πλάτους του DFT όταν το αρχικό μας σήμα περνάει από διαφορετικού τύπου παράθυρο. Εδώ θα χρησιμοποιήσουμε στην πρώτη περίπτωση τετραγωνικό παράθυρο και στη δεύτερη παράθυρο τύπου hamming (όπως αυτό που χρησιμοποιήσαμε κατά κόρον ως τώρα).
Για τα σήματα x1[n] και x2[n] ορίζονται σε αυτό το ερώτημα οι παράμετροι ως εξής:

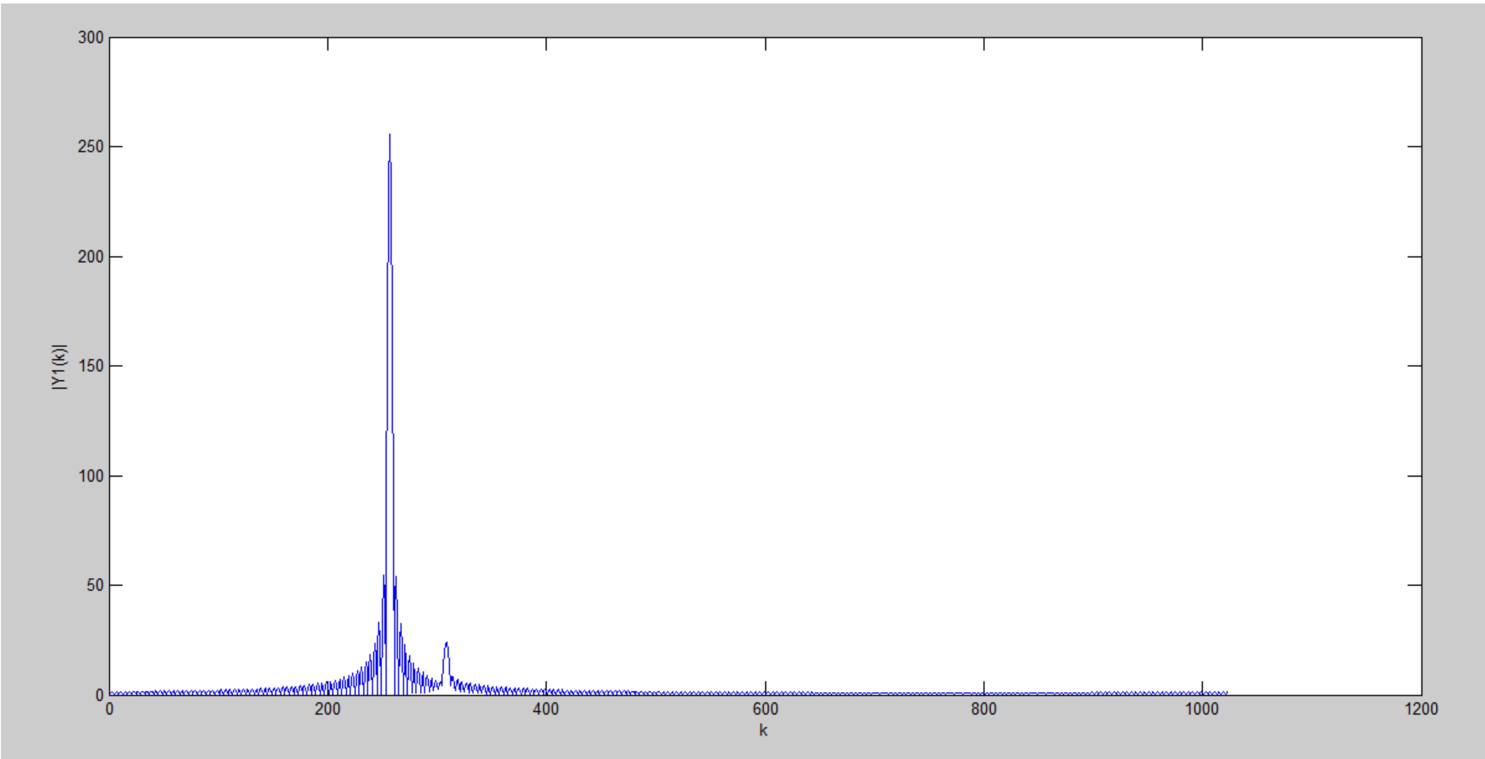
$A_1 = 1, A_2 = 0.1$ (δίνεται)

$\omega_1 = 0.5\pi, \omega_2 = 0.4\pi$ (δίνεται)

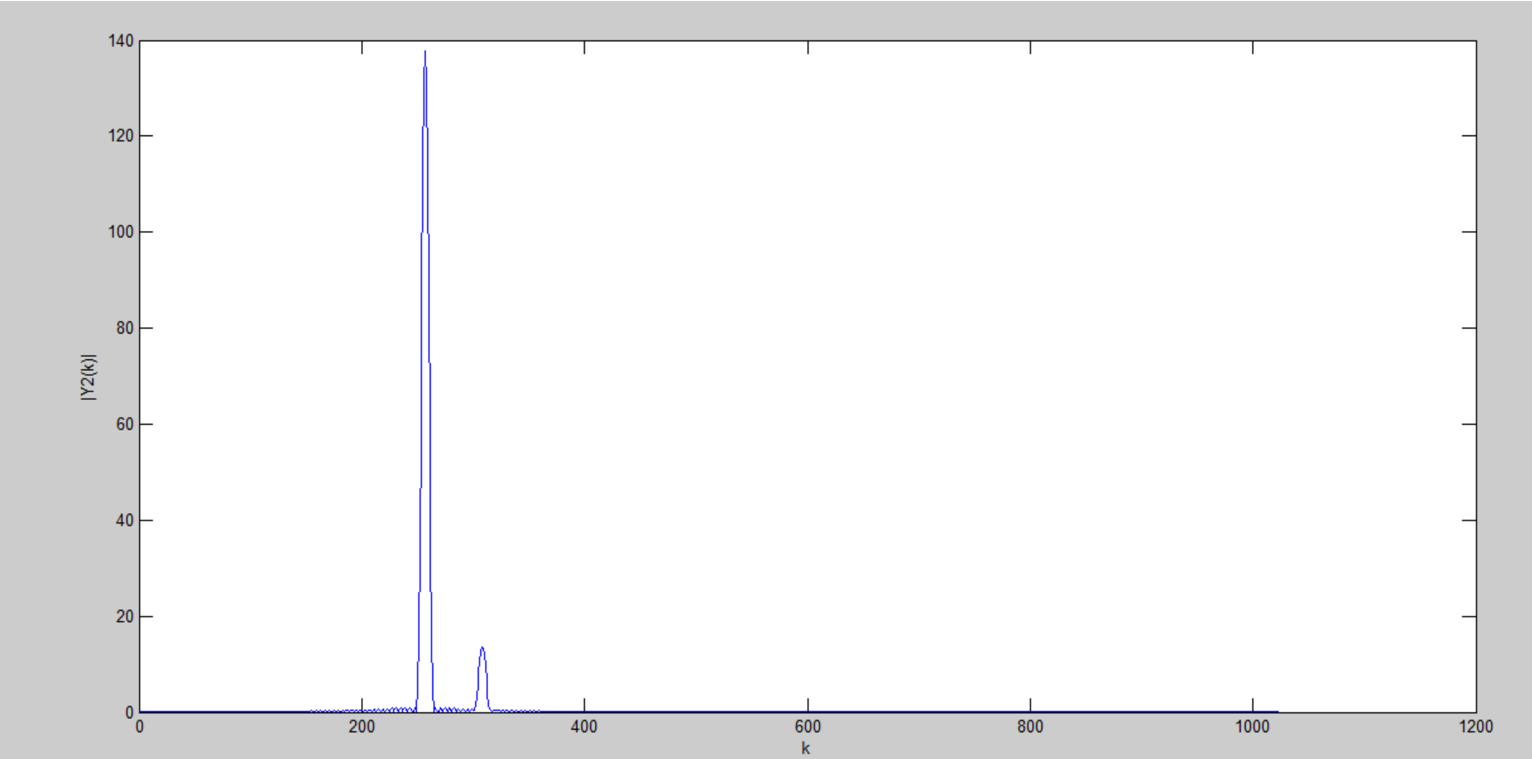
$\varphi_1 = \pi/5, \varphi_2 = \pi/36$ (αυθαίρετα)

Παρακάτω παραθέτουμε το μέτρο των μετασχηματισμών DFT που μας έδωσε το Matlab για :

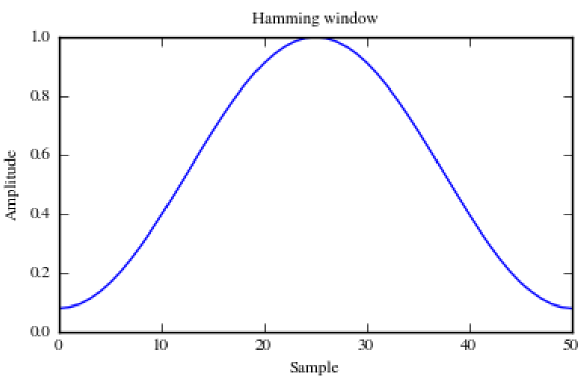
- DFT 1024 Σημείων Σήματος y[n] που εξέρχεται από τετραγωνικό παράθυρο 256 σημείων :

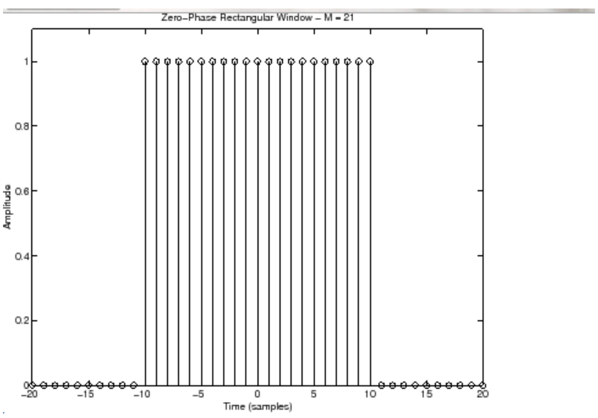


- DFT 1024 Σημείων Σήματος y[n] που εξέρχεται από Hamming παράθυρο 256 σημείων :



Η κύρια διαφορά που παρατηρούμε μεταξύ των 2 παραπάνω γραφικών παραστάσεων είναι η έντονη κυμάτωση που κάνει το φάσμα στην περίπτωση του τετραγωνικού παραθύρου. Στη περίπτωση του παραθύρου hamming είναι εμφανείς μονάχα 2 κρουστικές στην ουσία, πράγμα που μας κάνει ξεκάθαρο πως το παράθυρο hamming δεν επηρεάζει ιδιαίτερα το φάσμα των συναρτήσεων που περνούν από αυτό καθώς όπως και να είχε 2 κρουστικές αναμέναμε να δούμε εμείς ακόμη και αν δεν υπήρχε το παράθυρο. Αντίθετα το τετραγωνικό παράθυρο δείχνει να επηρεάζει το φάσμα και αυτό κυρίως πηγάζει από το γεγονός ότι το τετραγωνικό παράθυρο έχει παρόμοιο πλάτος σε όλο το εύρος του πράγμα που δε συμβαίνει στην περίπτωση του παραθύρου hamming όπως παρουσιάζεται και στα ακόλουθα σχήματα. Έτσι δικαιολογούνται οι κυματισμοί που φαίνονται παραπάνω στην περίπτωση του τετραγωνικού παραθύρου.





Συγκεντρωτικά από αυτή την εργαστηριακή άσκηση μάθαμε ότι:

- Όσο μεγαλύτερος σε μήκος είναι ο DFT που λαμβάνουμε από ένα σήμα τόσο πιο ευκρινές είναι το φάσμα. Ωστόσο αυτό δεν επηρεάζει την ικανότητα διάκρισης του συγκεκριμένου συστήματος.
- Η διακριτική ικανότητα ενός συστήματος βελτιώνεται με τη σειρά της σημαντικά αν αυξήσουμε τα δείγματα του παραθύρου που χρησιμοποιούμε για τη λήψη του σήματος που θα επεξεργαστούμε.
- Είναι προτιμότερο να χρησιμοποιούμε Hamming windows παρά Rectangular καθώς τα δεύτερα επηρεάζουν το φάσμα του σήματος που εισέρχεται σε αυτά ενώ τα πρώτα το κρατούν σχεδόν ανέπαφο.

Μέρος 3ο
Χαρακτηριστικά Βραχέος Χρόνου Σημάτων Φωνής και Μουσικής (Ενέργεια και Ρυθμός Εναλλαγής Προσήμου)

Σκοπός της άσκησης αυτή είναι ο υπολογισμός της ενέργειας βραχέος χρόνου ενός σήματος καθώς και η εύρεση του ρυθμού εναλλαγής προσήμου του.
Από θεωρία γνωρίζουμε πως οι μετρήσεις βραχέος χρόνου είναι μετρήσεις που λαμβάνονται σε ένα μετακινούμενο παράθυρο του σήματος.
Βάσει αυτού, η ενέργεια βραχέος χρόνου δίνεται από τον τύπο:

$$E_n = \sum_{m=-\infty}^{+\infty} \left[x(m) \cdot w(n-m)\right]^2$$

Ο παραπάνω τύπος αν το παρατηρήσουμε στην ουσία είναι η εξής συνέλιξη:

$$E_n = x^2(n) * w^2(n)$$

Αντίστοιχα ο ρυθμός Εναλλαγής Προσήμου ορίζεται από τον τύπο:

$$Z_n = \sum_{m=-\infty}^{+\infty} \left|\operatorname{sgn}\left[x(m)\right] - \operatorname{sgn}\left[x(m-1)\right]\right| \cdot w(n-m)$$

Ο παραπάνω τύπος αν το παρατηρήσουμε πάλι στην ουσία είναι η εξής συνέλιξη:

$$Z_n = \left|\operatorname{sgn}\left[x(n)\right] - \operatorname{sgn}\left[x(n-1)\right]\right| * w(n)$$

Γνωρίζουμε ωστόσο πως αν κάνουμε μετασχηματισμό DFT η συνέλιξη γίνεται ένας απλός πολλαπλασιασμός στο πεδίο της συχνότητας επομένως αυτό είναι που μας βολεύει. Στην ουσία θα πάρουμε τον διακριτό μετασχηματισμό Fourier κάνοντας χρήση της συνάρτησης ***fft()*** του Matlab και αφού βρούμε το αποτέλεσμα που θέλουμε σε μορφή DFT θα πάρουμε τον αντίστροφο μετασχηματισμό DFT (***ifft()***). Βέβαια προτού πάρουμε το μετασχηματισμό αυτόν θα εφαρμόσουμε τη διαδικασία του zero-padding μέχρις ότου τα σήματα αποκτήσουν μήκος ίσο με το άθροισμα των αρχικών μηκών τους μειωμένο κατά 1 δείγμα.
Πιο μαθηματικοποιημένα ισχύει:

$$x_1[n] * x_2[n] \overset{DFT}{\rightleftarrows} X_1[k] \cdot X_2[k]$$

Ερώτημα 3.1. :

Οι παραπάνω τύποι θα εφαρμοστούν μέσω του Matlab στο σήμα φωνής speech_utterance.wav που μας δίνεται. Θα ελέγξουμε πώς επηρεάζονται οι τιμές της ενέργεια και του ρυθμού εναλλαγής προσήμου ανάλογα με το μήκος του παραθύρου. Επίσης θα προσπαθήσουμε να εντοπίσουμε μέσω αυτών του έμφωνους και άφωνους ήχους που συμπεριλαμβάνονται στο σήμα μας.
Πρακτικά ξεκινάμε διαβάζοντας το αρχείο μας μέσω της συνάρτησης ***wavread()***. Όπως αναφέρεται και στην εκφώνηση το αρχείο ήχου αυτό έχει συχνότητα δειγματοληψίας ίση με 16kHz. Επιθυμούμε το παράθυρο hamming που θα επιλέξουμε να έχει δείγματα τα οποία θα έχουν προέλθει από τη συνάρτηση συνεχούς χρόνου του hamming με ίδια συχνότητα δειγματοληψίας με αυτή του αρχείου ήχου. Επομένως έτσι βρίσκουμε την περίοδο δειγματοληψίας από τον γνωστό τύπο:

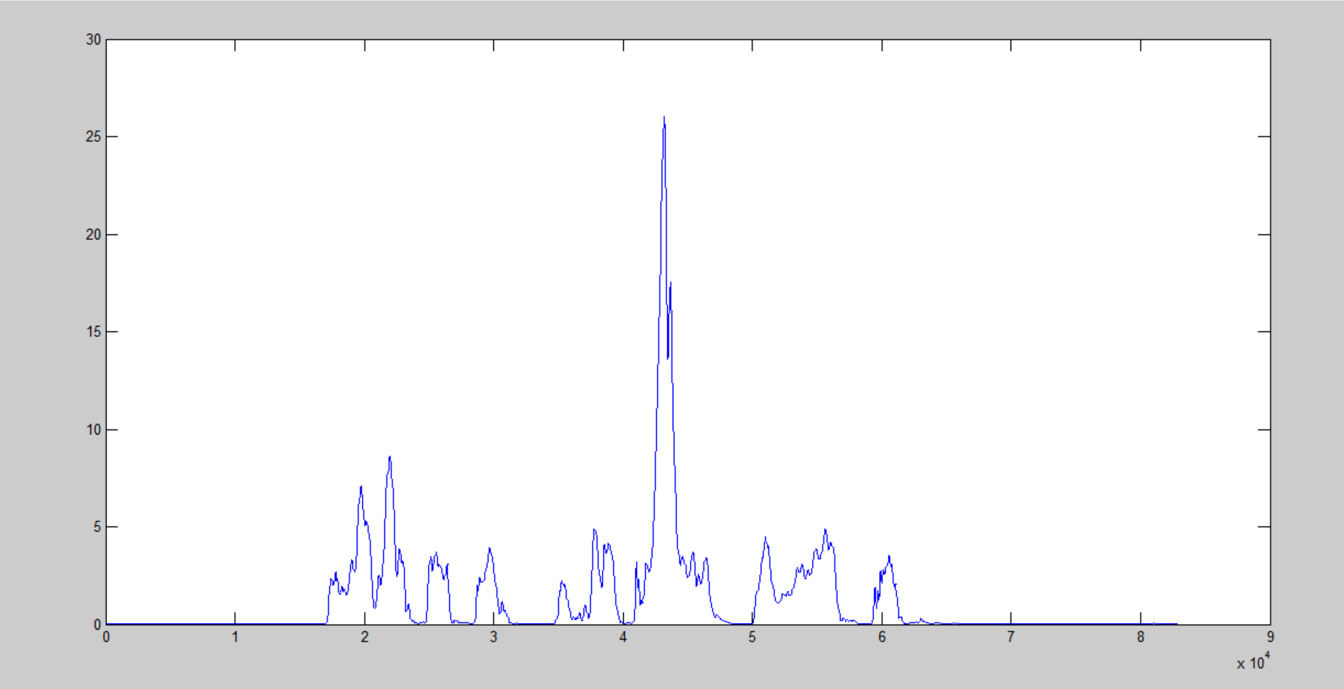
$$T_s = {}^{1}\!/_{f_s}$$

Με βάση λοιπόν αυτό έχουμε: Περίοδος δειγματοληψίας$= T_s = 0.0625 m\sec$

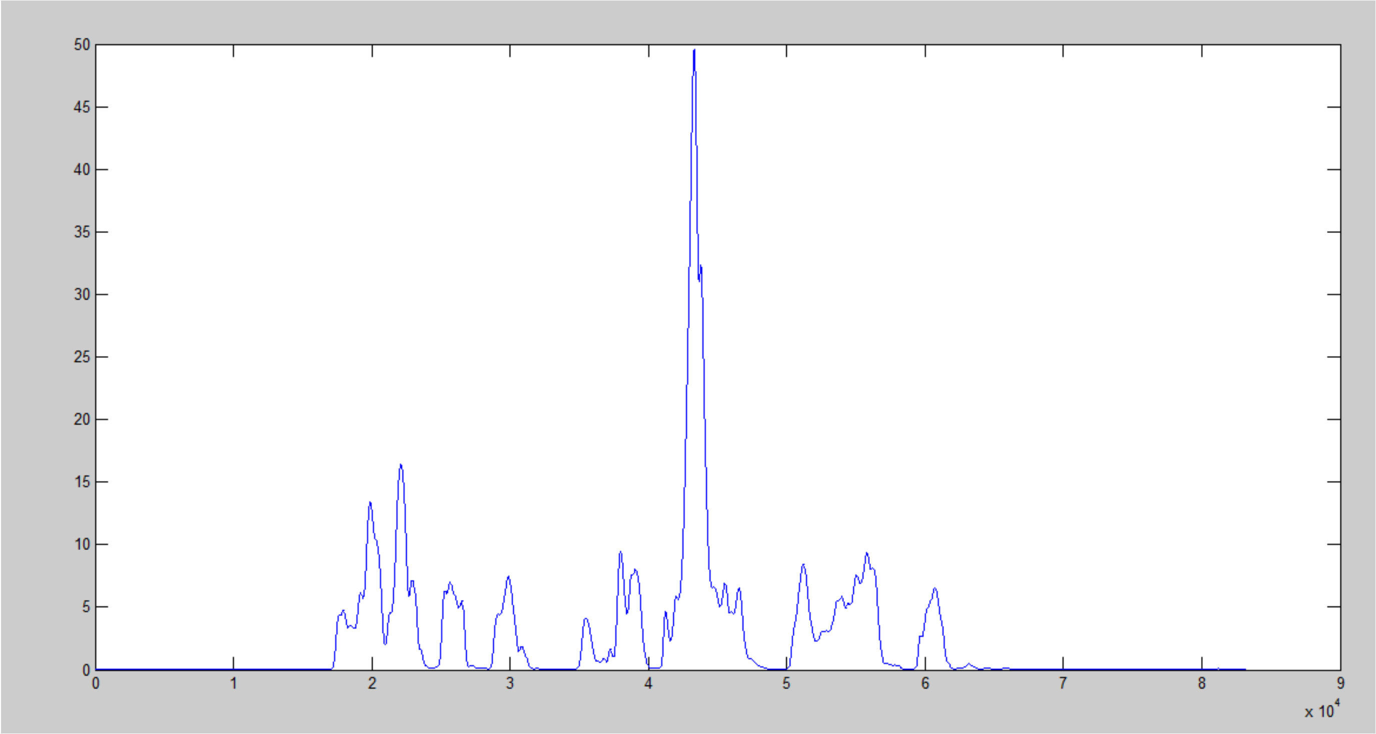
Στη συνέχεια αν διαιρέσουμε τη διάρκεια του παραθύρου με αυτή την τιμή βρίσκουμε το επιθυμητό πλήθος των δειγμάτων.
Για τον υπολογισμό όλων όσων αναφέρθηκαν παραπάνω γράφουμε στο Matlab 2 ρουτίνες. Μια που ονομάζουμε ***energy_find()*** που παίρνει ως ορίσματα το σήμα ήχου καθώς και τη διάρκεια σε ms του παραθύρου hamming που θα χρησιμοποιηθεί, και επιστρέφει την ενέργεια βραχέος χρόνου. Η δεύτερη ονομάζεται ***zcr()*** και είναι υπεύθυνη για τον υπολογισμό του ρυθμού εναλλαγής προσήμου. Παίρνει ως ορίσματα τα ίδια με την παραπάνω συνάρτηση και επιστρέφει το ρυθμό εναλλαγής προσήμου(zero crossing rate).
Οι συναρτήσεις καλούνται για τα διάφορα σε μήκος παράθυρα που ζητούνται στην άσκηση. Παρακάτω παρουσιάζονται οι ανάλογες γραφικές παραστάσεις:

Γραφική παράσταση **Ενέργειας Βραχέος Χρόνου** για:

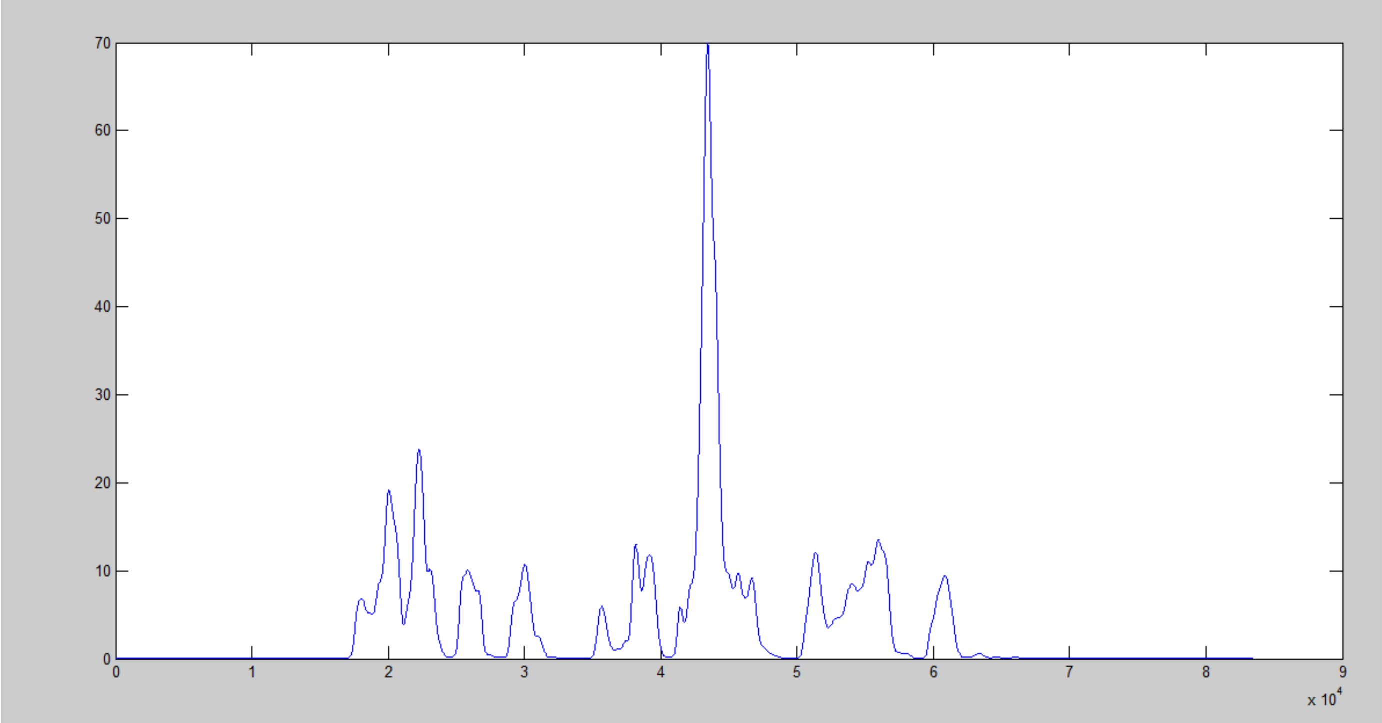
- Παράθυρο hamming διάρκειας ***20msec***:



- <u>Παράθυρο hamming διάρκειας ***40msec***:</u>

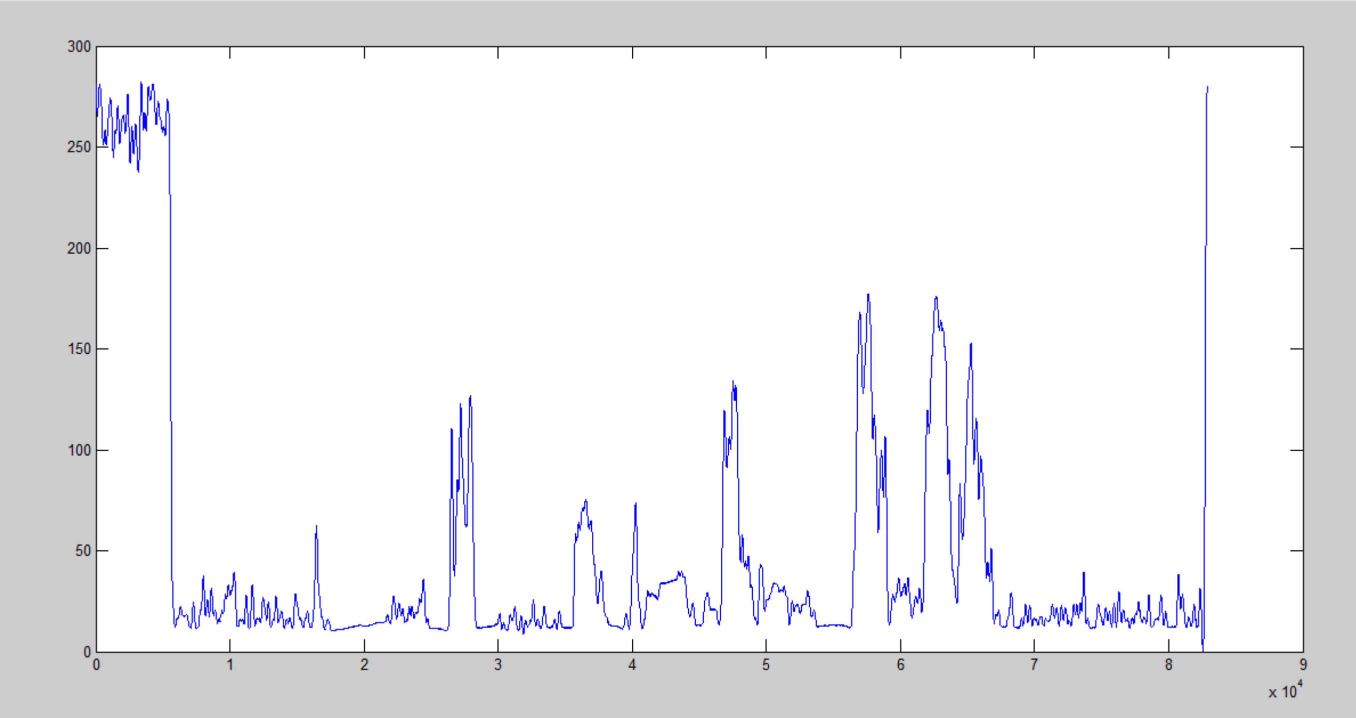


- <u>Παράθυρο Hamming διάρκειας ***60msec***:</u>

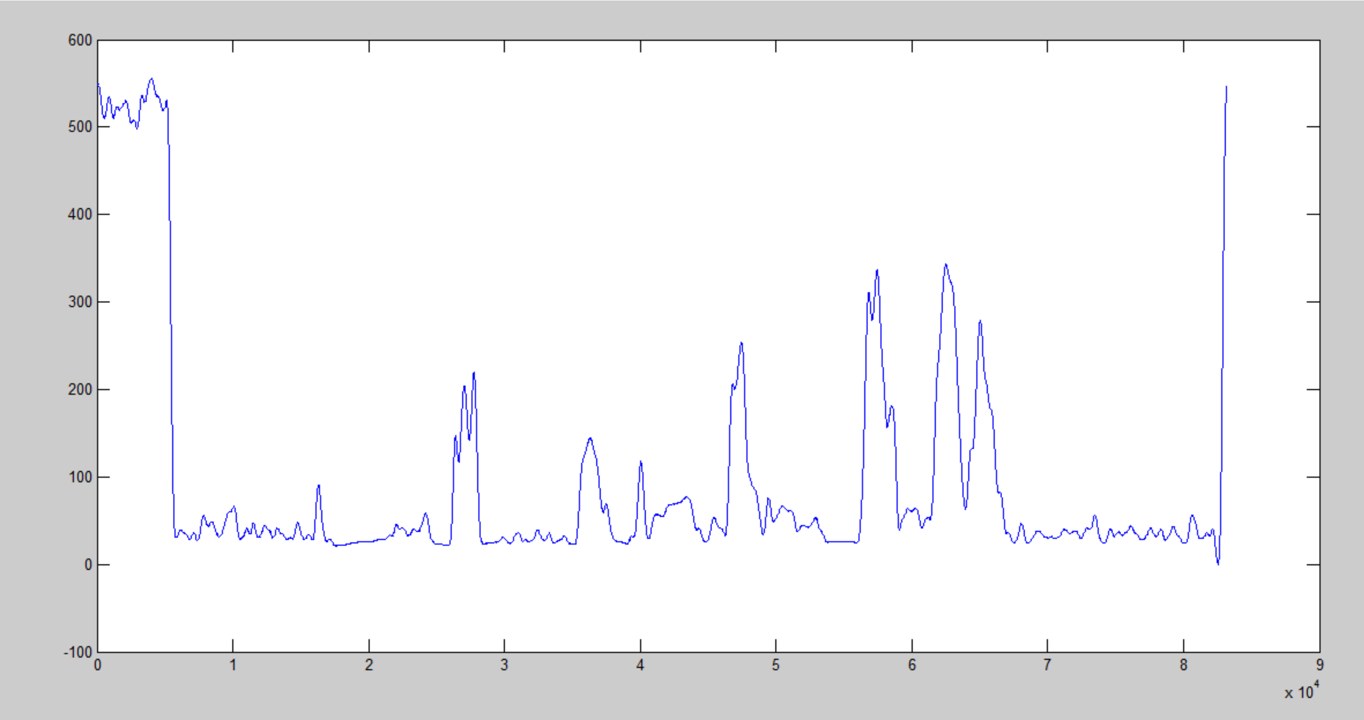


Γραφική παράσταση **Ρυθμού Εναλλαγής Προσήμου** για:

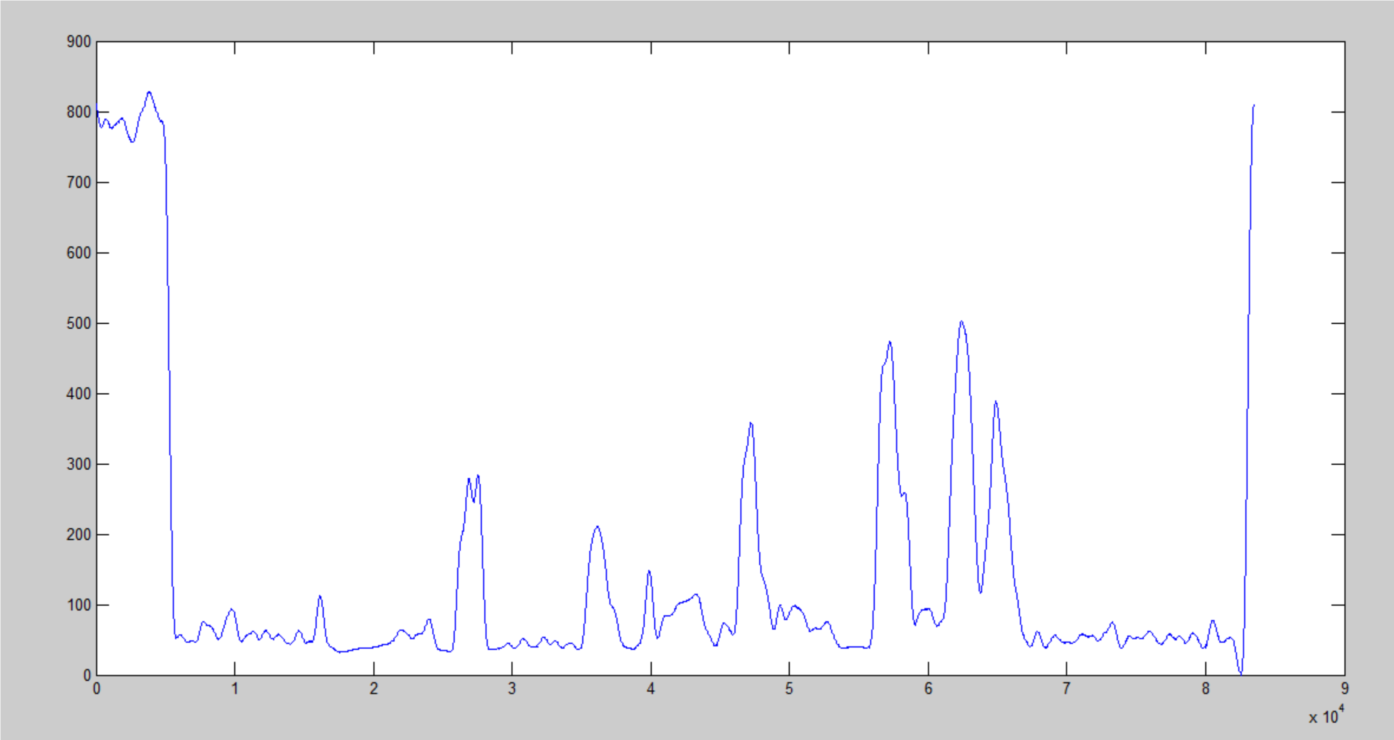
- Παράθυρο hamming διάρκειας ***20msec***:



- Παράθυρο hamming διάρκειας ***40msec***:



- <u>Παράθυρο hamming διάρκειας ***60msec***:</u>

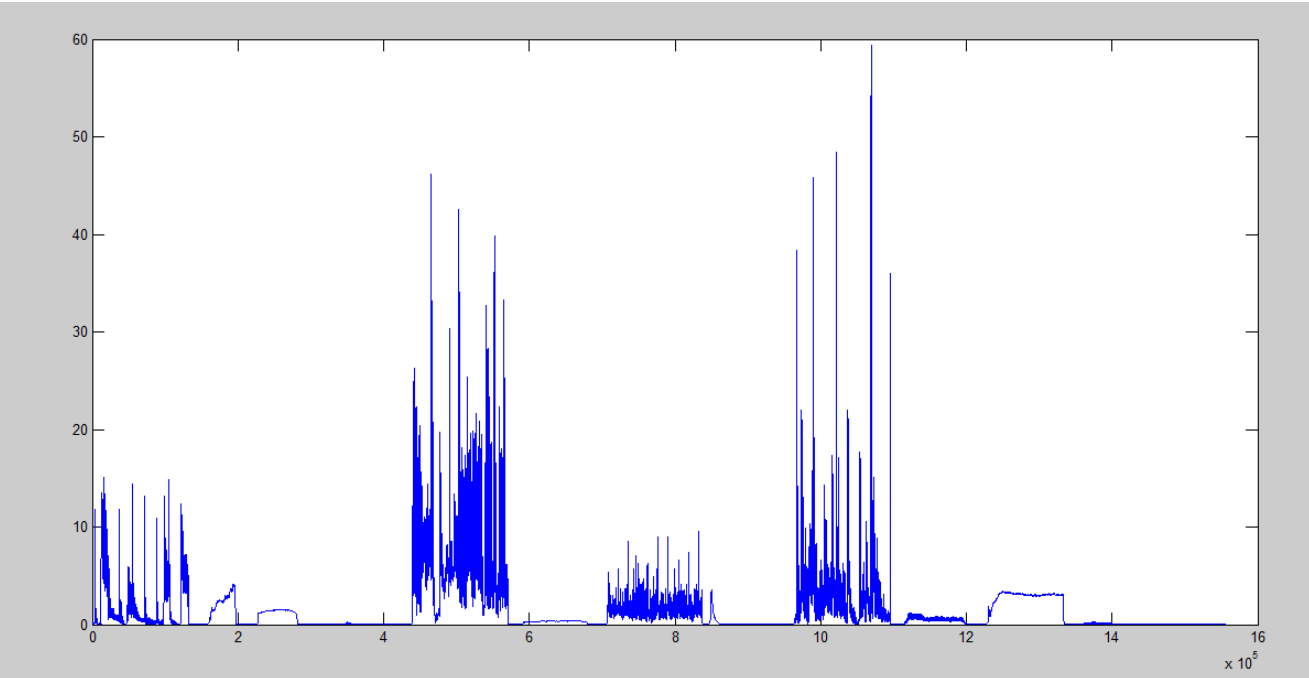


Κοιτώντας προσεχτικά τα παραπάνω σχήματα μπορούμε να βγάλουμε διάφορα συμπεράσματα.
Αρχικά παρατηρούμε ότι όσο μεγαλύτερα σε διάρκεια είναι τα παράθυρα τόσο πιο "ομαλές" είναι οι γραφικές παραστάσεις τόσο της ενέργειας όσο και της εναλλαγής προσήμου.
Βλέπουμε για παράδειγμα στην περίπτωση που έχουμε παράθυρο διάρκειας 20msec πως υπάρχουν πολλά οξεία σημεία.
Επίσης, οι παραπάνω γραφικές παραστάσεις μπορούν να μας δώσουν πληροφορίες για τα σημεία του σήματος που έχουν έμφωνους και άφωνους ήχους(σιωπή).
Ειδικότερα, από τις γραφικές παραστάσεις τις ενέργειας μπορούμε να πούμε ότι όπου υπάρχει υψηλή ενέργεια βραχέος χρόνου έχουμε έμφωνους ήχους (πχ /aa/,/ih/). Το ακριβώς αντίθετο ισχύει για τους άφωνους ήχους (αντιστοιχούν σε σημεία με χαμηλή ενέργεια βραχέος χρόνου). Για τη σιωπή είναι εύκολη η διάκριση. Σε όσα σημεία η ενέργεια είναι μηδενική, δεν υπάρχει φωνή. Επικρατεί σιγή.
Παρόμοιο συμπέρασμα μπορεί να βγει και από την παρατήρηση του ρυθμού εναλλαγής προσήμου. Οι άφωνοι ήχοι έχουν αυτή τη μέτρηση αυξημένη σε σχέση με τους έμφωνους.

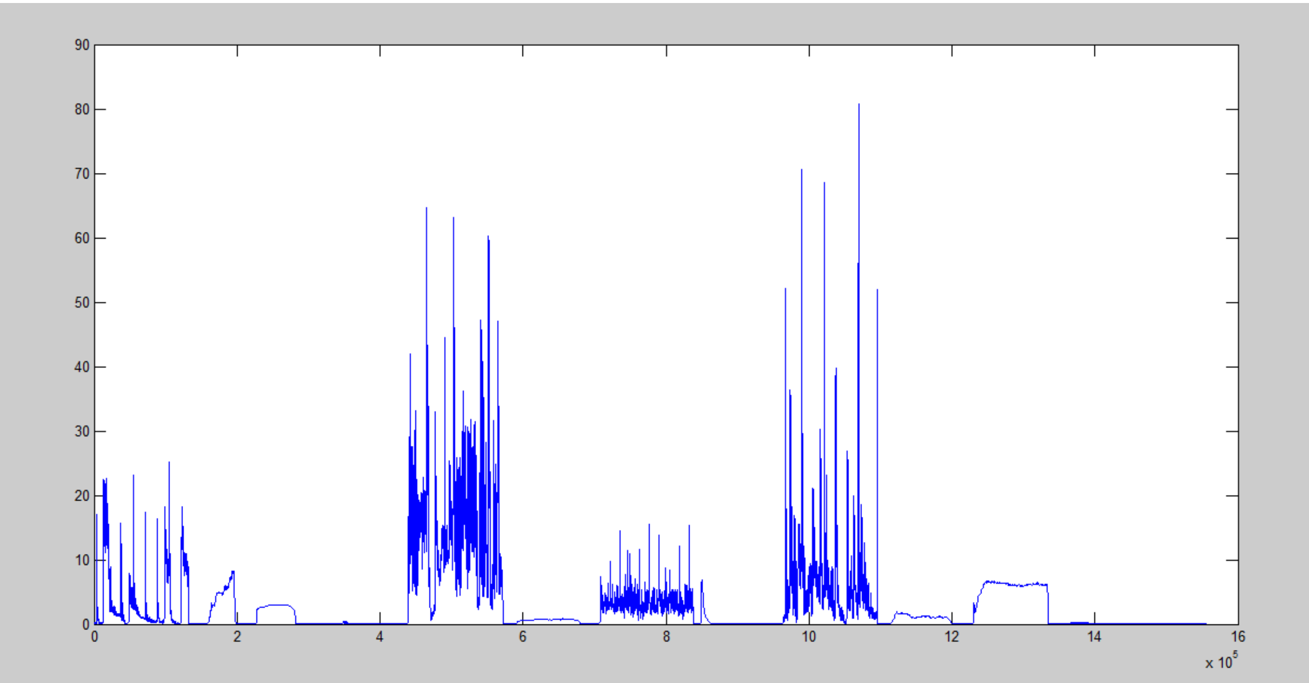
Ερώτημα 3.2.:
Την ίδια διαδικασία με παραπάνω ακολουθούμε αλλά αυτή τη φορά επεξεργαζόμαστε το σήμα music.wav που μας δίνεται. Παρακάτω δίδονται οι αντίστοιχες γραφικές παραστάσεις όπως προέκυψαν από το Matlab για τα διάφορα μήκη παραθύρων.

Γραφική παράσταση **Ενέργειας Βραχέος Χρόνου** για:

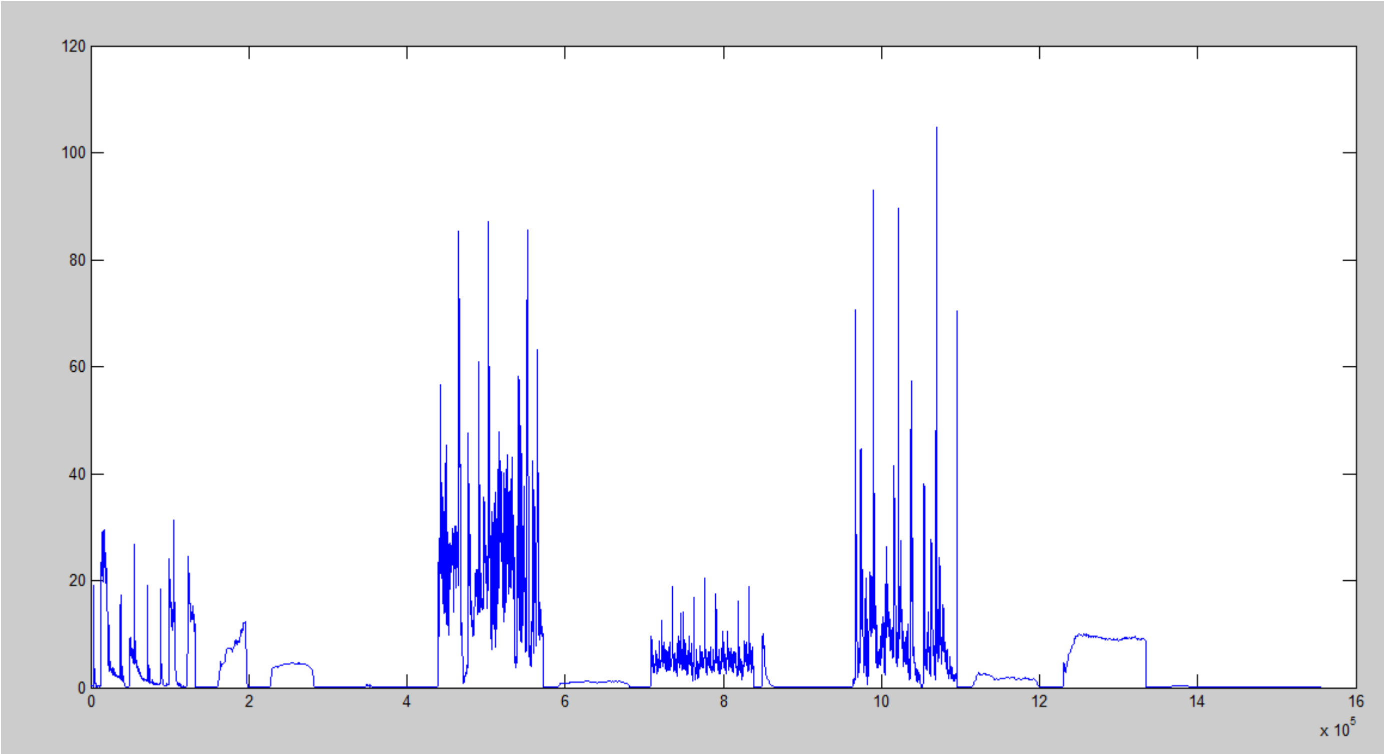
- Παράθυρο hamming διάρκειας ***20msec***:



- Παράθυρο hamming διάρκειας ***40msec***:

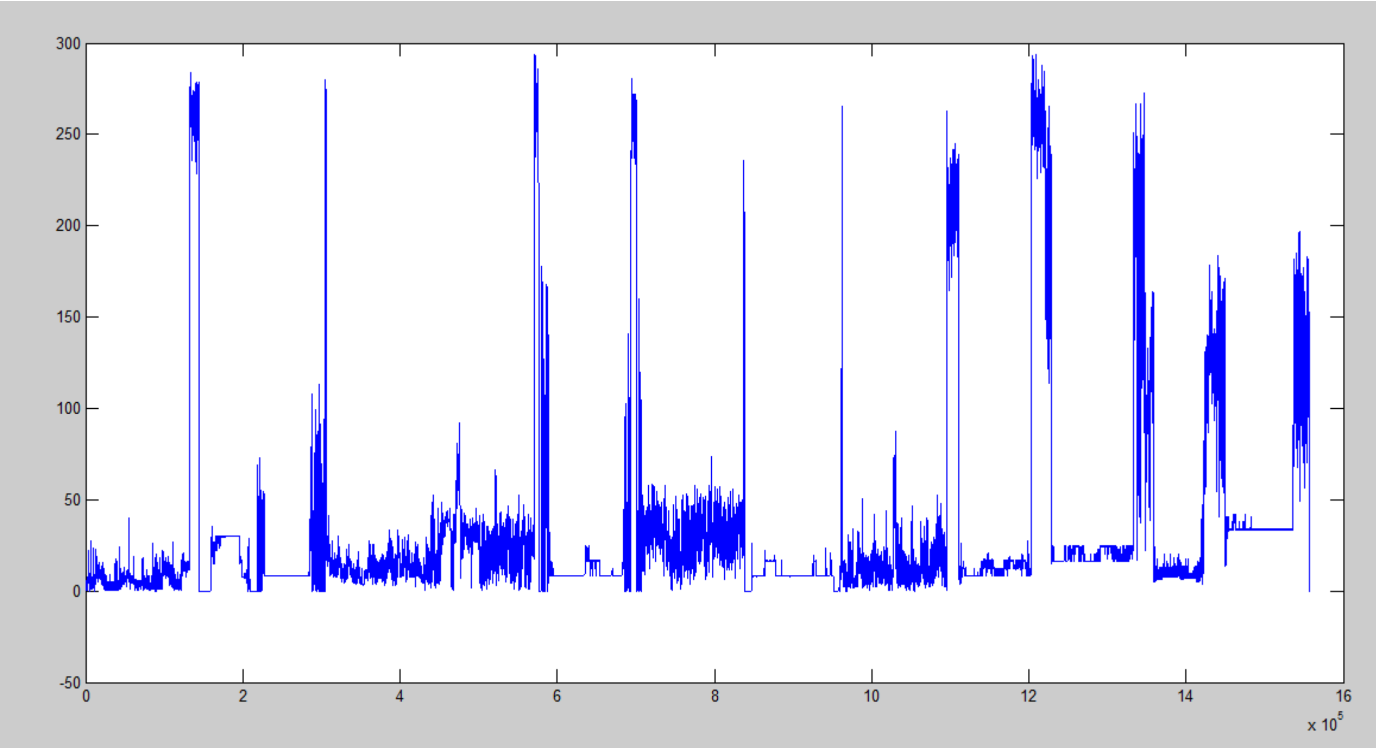


- <u>Παράθυρο hamming διάρκειας ***60msec***:</u>

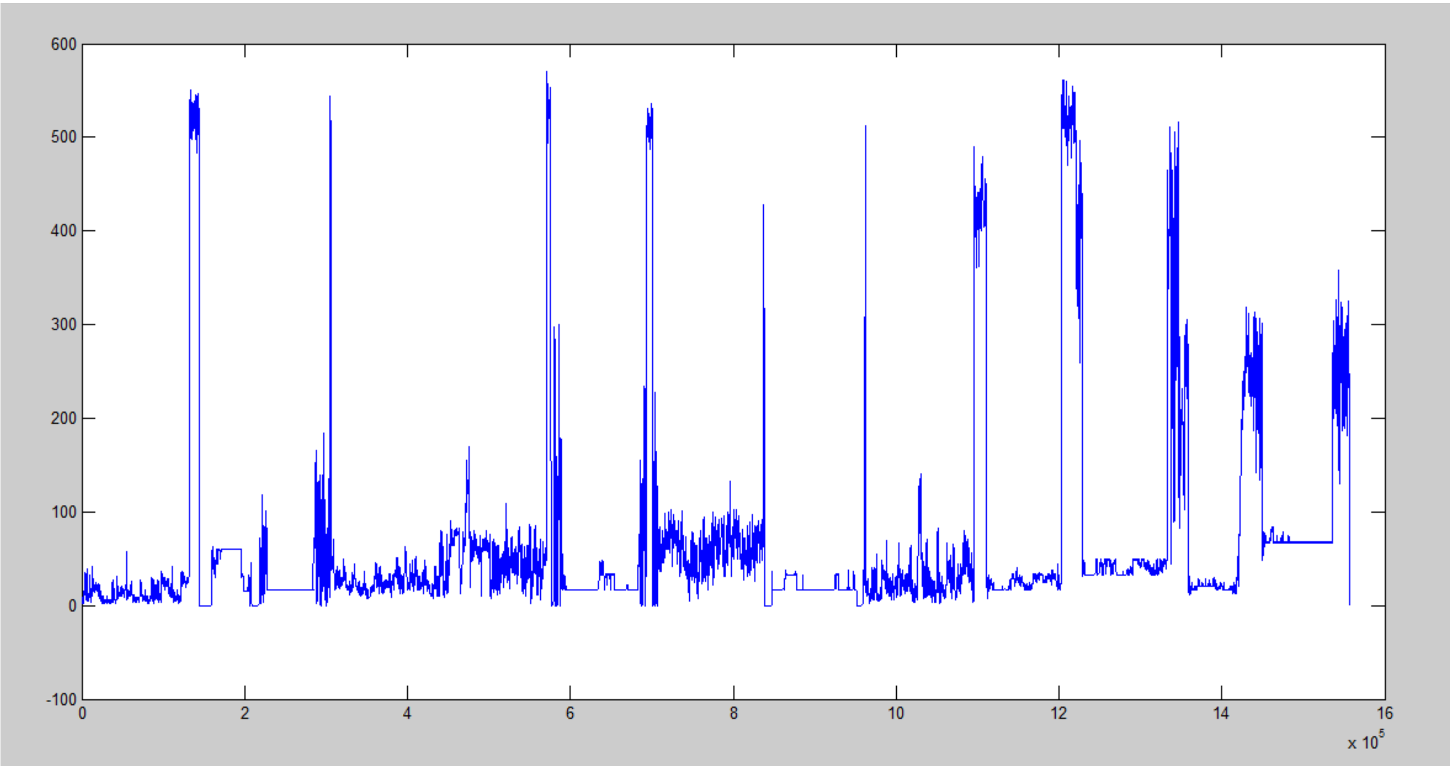


Γραφική παράσταση **Ρυθμού Εναλλαγής Προσήμου** για:

- <u>Παράθυρο hamming διάρκειας ***20msec***:</u>



- <u>Παράθυρο hamming διάρκειας ***40msec***:</u>



- <u>Παράθυρο hamming διάρκειας ***60msec***:</u>