Ψηφιακή Επεξεργασία Σήματος

Διαμάντη Ιωάννα(AM:03115036)
Μαυροθαλασσίτης Ιωάννης(AM:03115059)

Μέρος 1ο

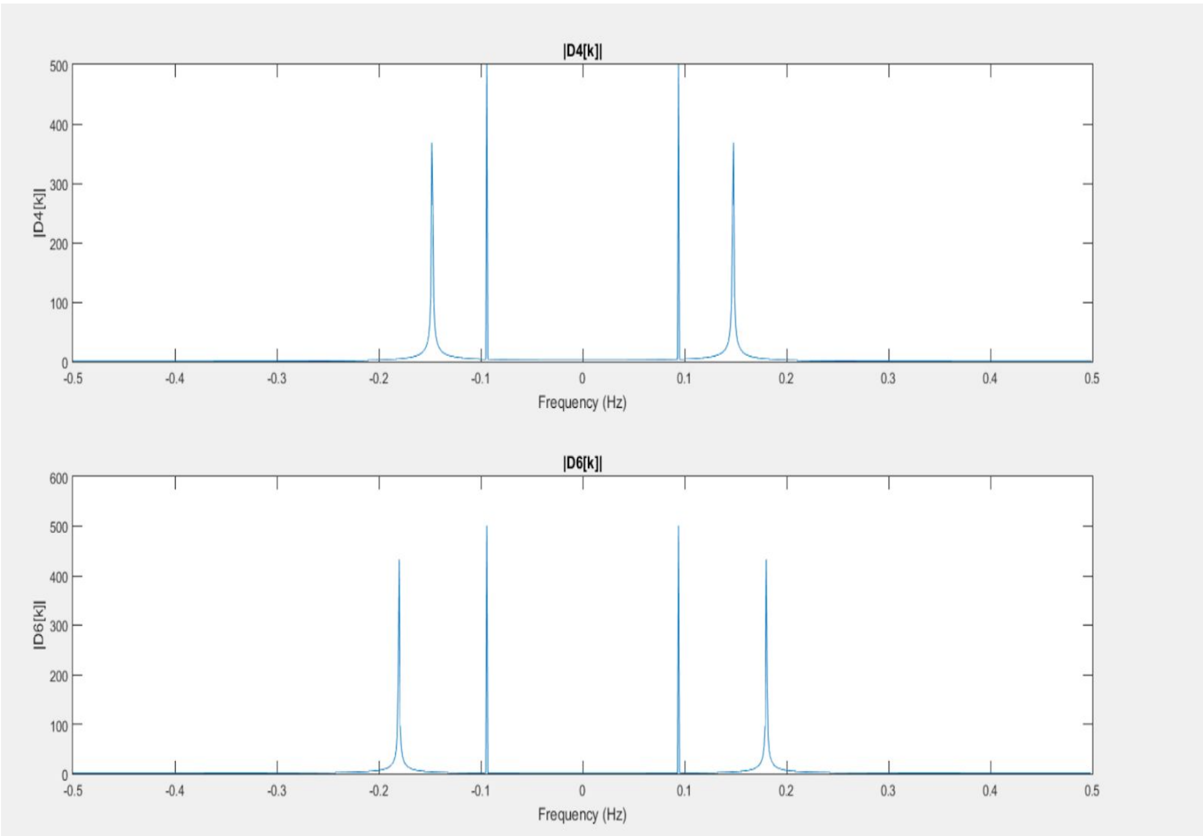
Ερώτημα 1.1

Με βάση τις συχνότητες που δίνονται στον πίνακα της εκφώνησης συνθέσαμε τους 10 τόνους ως εξής:

```
n=0:999;
d0= sin(0.7217.*n)+sin(1.0247.*n);
d1= sin(0.5346.*n)+sin(0.9273.*n);
d2= sin(0.5346.*n)+sin(1.0247.*n);
d3= sin(0.5346.*n)+sin(1.1328.*n);
d4= sin(0.5906.*n)+sin(0.9273.*n);
d5= sin(0.5906.*n)+sin(1.0247.*n);
d6= sin(0.5906.*n)+sin(1.1328.*n);
d7= sin(0.6535.*n)+sin(0.9273.*n);
d8= sin(0.6535.*n)+sin(1.0247.*n);
d9= sin(0.6535.*n)+sin(1.1328.*n);
```

Ερώτημα 1.2

Μέσω της συνάρτησης fft() που παρέχεται από το matlab, υπολογίσαμε τον DFT των σημάτων d4 και d6 που αντιστοιχούν στους τόνους των πλήκτρων 4 και 6 αντίστοιχα. Στη συνέχεια μέσω της συνάρτησης fftshift() και abs() κεντράραμε την γραφική γύρω από το 0 και υπολογίσαμε το μέτρο του αποτελέσματος, το οποίο φαίνεται στις παρακάτω γραφικές παραστάσεις:



Ερώτημα 1.3

Για AM1 = 03115035 και AM2 = 03115059 προκύπτει άθροισμα 0 6 2 3 0 0 9 4. Έτσι φτιάξαμε το διάνυσμα [d0 z d6 z d2 z d3 z d0 z d0 z d9 z d4] (όπου z είναι διάνυσμα με 100 μηδενικά στοιχεία) και μέσω της συνάρτησης wavwrite() το ηχογραφήσαμε με συχνότητα 8192Hz στο αρχείο “tone_sequence.wav” που περιέχεται στο συμπιεσμένο αρχείο της εργασίας, παρεμβάλλοντας 100 μηδενικά δείγματα ανάμεσα στα ψηφία.

Ερώτημα 1.4

Χρησιμοποιώντας τα δύο είδη παραθύρων που ζητά η εκφώνηση (τετραγωνικό και hamming) με μήκος N=1000 και Overlap= -100 παραθυροποιήσαμε το συνολικό σήμα στα επιμέρους σήματα που το αποτελούν, παραβλέποντας τα 100 μηδενικά που παρεμβάλλονται ανάμεσα τους. Στη συνέχεια μέσω του αλγόριθμου fft() υπολογίζουμε τον DFT του κάθε παραθυροποιημένου σήματος.

Ερώτημα 1.5

Χρησιμοποιώντας την συνάρτηση findpeaks() του matlab βρίσκουμε για το παραπάνω σήμα όλα τα τοπικά ακρότατα (τιμή και θέση). Στη συνέχεια κρατάμε αυτά που έχουν πλάτος μεγαλύτερο από ένα κατώφλι, διασφαλίζοντας την εύρεση μόνο των επιθυμητών τοπικών μεγίστων. Συγκεκριμένα η λίστα που προέκυψε για το σήμα 0 6 2 3 0 0 9 4 είναι η εξής:

Λίστα με δείκτες κ = [86 95 105 116 149 164 181]
Αντίστοιχες συχνότητες = [0.5404 0.5969 0.6597 0.7288 0.9362 1.0304 1.1373] σε rad/sec

Ερώτημα 1.6

Η συνάρτηση ttdecode() περιέχεται στο αρχείο ttdecode.m που υπάρχει στο συμπιεσμένο αρχείο της εργασίας. Σαν όρισμα δέχεται ένα τονικό σήμα και επιστρέφει την ακολουθία των ψηφίων που πατήθηκαν με τον εξής τρόπο:
Γνωρίζοντας πως το τονικό σήμα κάθε ψηφίου αποτελείται από 1000 δείγματα και πως ανάμεσα στα σήματα παρεμβάλλονται 100 μηδενικά, επεξεργαζόμαστε το σήμα ανα 1000 δείγματα και παραβλέποντας κάθε φορά τα επόμενα 100. Φτιάχνουμε αρχικά έναν πίνακα,του οποίου κάθε γραμμή αντιστοιχεί στις χαμηλές συχνότητες των τόνων, ενώ η στήλη αντιστοιχεί στις υψηλές συχνότητες των τόνων. Σε κάθε στοιχείο του πίνακα υπάρχει ο αριθμός που αντιστοιχεί στις συχνότητες της γραμμής και της στήλης. Έτσι διευκολύνεται η εύρεση των ψηφίων μόλις ανιχνευθούν οι δύο συχνότητες. Για την ανίχνευση των συχνοτήτων επαναλαμβάνουμε για κάθε ψηφίο (1000 δείγματα) του σήματος την εξής διαδικασία:
Παίρνουμε αρχικά τον DFT του σήματος με τον αλγόριθμο fft(). Στη συνέχεια υπολογίζουμε την ενέργεια του σήματος και αναζητούμε τις δύο μέγιστες τιμές. Σαν κατώφλι χρησιμοποιούμε την συχνότητα 2π0.8 Hz , καθώς όπως φαίνεται από τον πίνακα οι χαμηλές συχνότητες χωρίζονται από τις υψηλές κοντά σε αυτήν. Έτσι αναζητούμε το πρώτο μέγιστο για συχνότητα μέχρι και 2π0.8 Hz και το δεύτερο για συχνότητες από 2π0.8 Hz μέχρι περίπου 2π1.14 Hz, καθώς η μέγιστη συχνότητα που έχουν τα ημίτονα είναι 2π1.1328 Hz (αφήνουμε ένα περιθώριο λόγω σφαλμάτων). Παίρνουμε τα αντίστοιχα ψηφία από τον πίνακα που αναφέραμε παραπάνω.

Ερώτημα 1.7

Διαβάζουμε τα δύο σήματα hardSig, easySig με την συνάρτηση load() και καλούμε την συνάρτηση ttdecode() με όρισμα αυτά. Το αποτέλεσμα είναι το εξής:

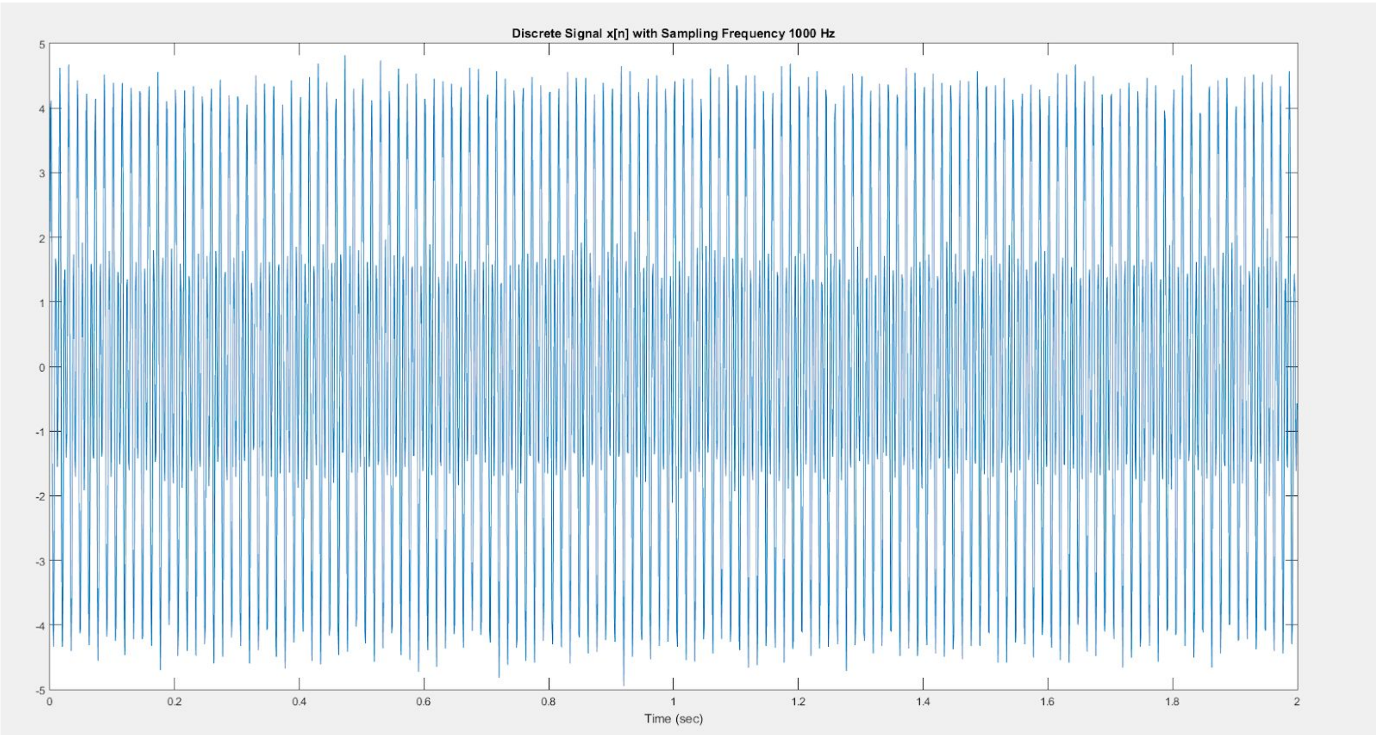
hardSig: 1 3 2 6 3 9 0 0

easySig: 9 0 9 6 3 2 1 1 9 1

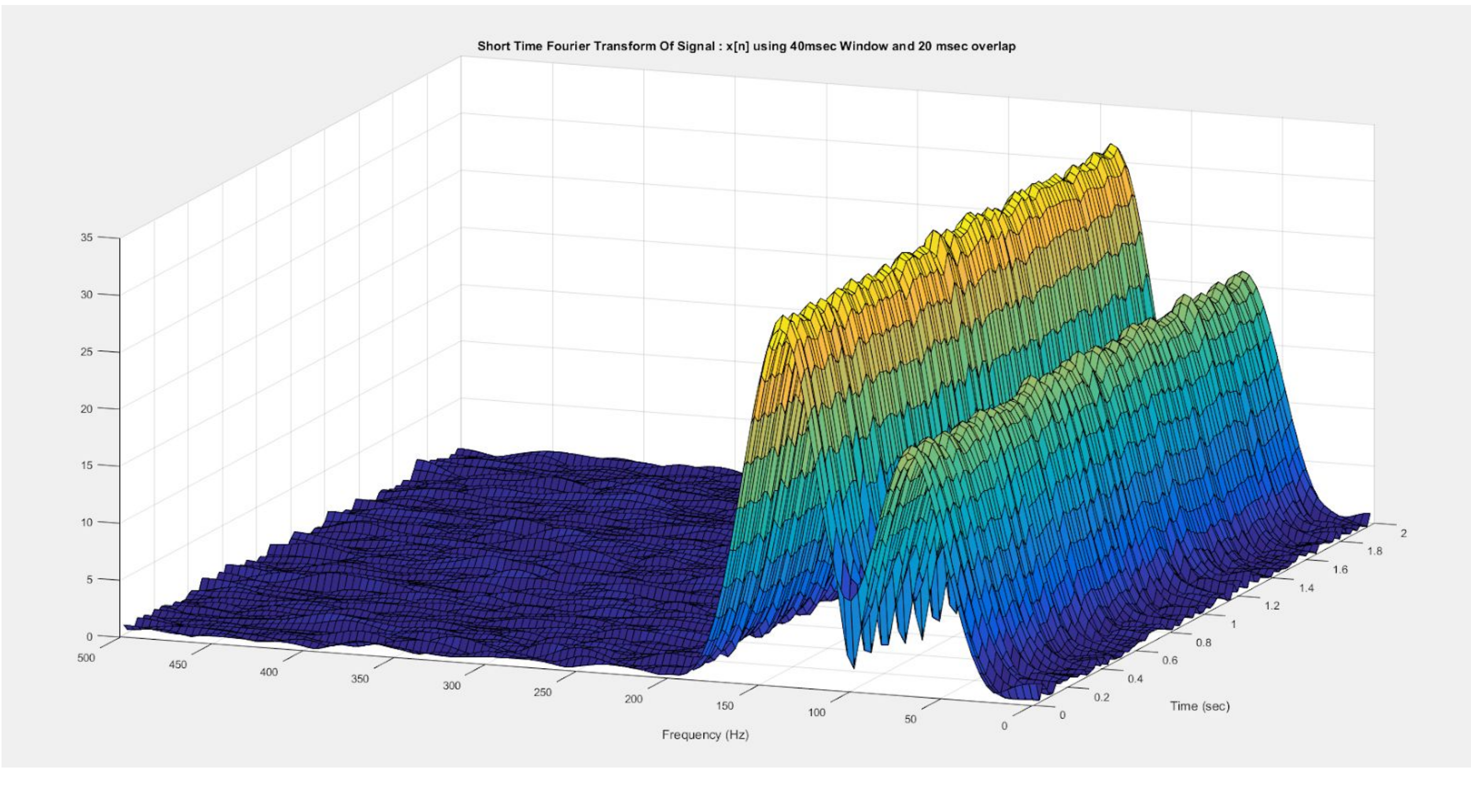
Μέρος 2ο

Ερώτημα 2.1:

α)Η γραφική παράσταση του διακριτού x χρησιμοποιώντας συχνότητα δειγματοληψίας Fs=1000Hz είναι η παρακάτω όπως προέκυψε και απο την συνάρτηση plot:

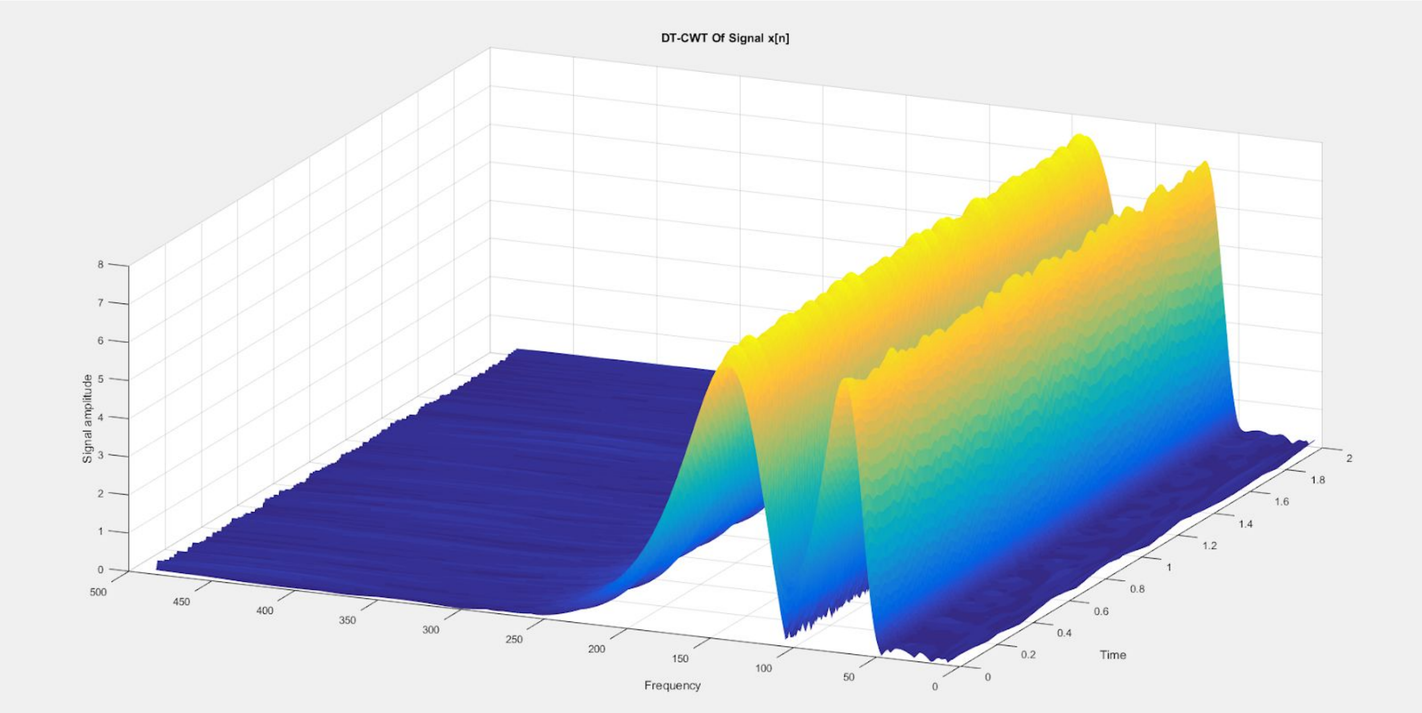


β)Η γραφική παράσταση για τον STFT όπως μας προκύπτει είναι η παρακάτω:



Έχουμε λάβει τα όρια του χρόνου και της συχνότητας από την συνάρτηση spectogram().

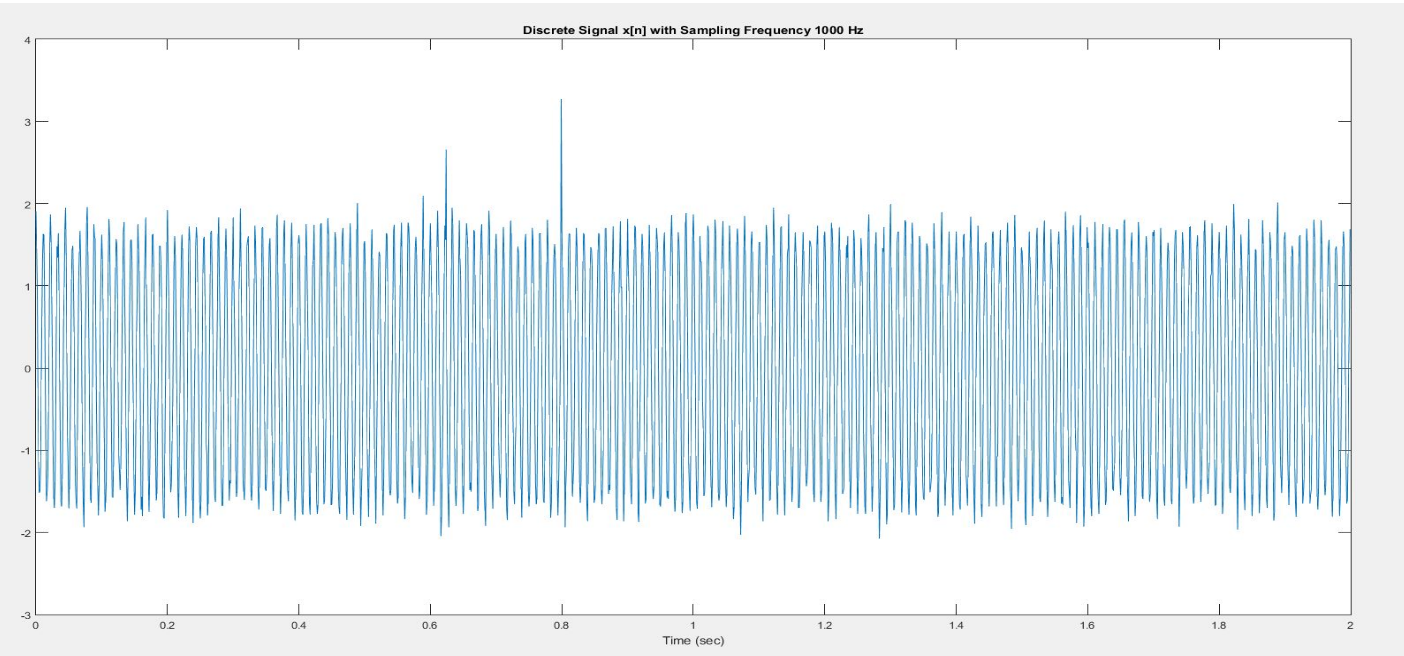
γ)Υπολογίσαμε τον μετασχηματισμό Morlet κυματιδίου διακριτού χρόνου του σήματος x[n] μέσω της συνάρτησης wavescales() που δίνεται στο συμπληρωματικό υλικό. Η γραφική παράσταση από τον μετασχηματισμό DT-CWT είναι η παρακάτω όπως προκύπτει από την συνάρτηση surf με κατάλληλα όρια, τα οποία λάβαμε από την συνάρτηση wavescales().



δ)Όπως παρατηρούμε και απο τις γραφικές παραστάσεις δεν υπάρχει εμφανής διαφορά πληροφορίας από τους δύο μετασχηματισμούς αντιθέτως λαμβάνουμε την ίδια πληροφορία όπως και θα περιμέναμε, καθώς πρόκειται για μία καθαρά ημιτονική συνάρτηση .

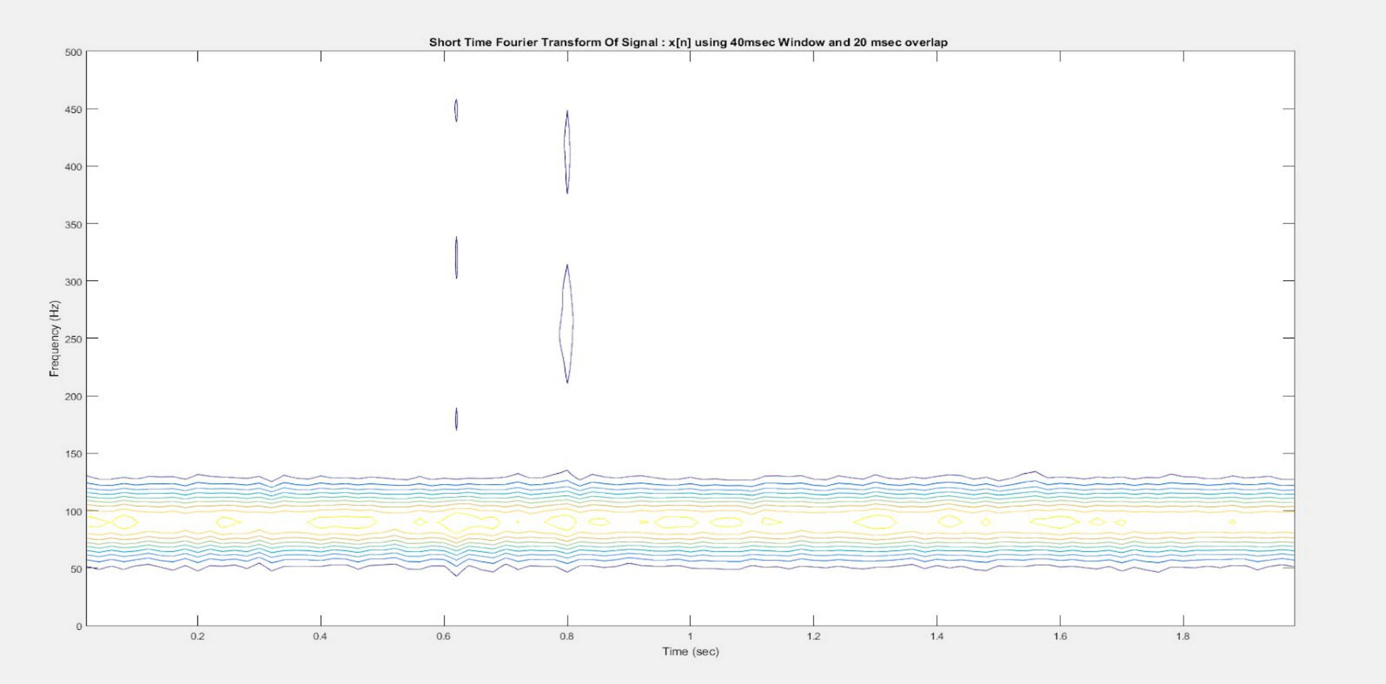
<u>Ερώτημα 2.2:</u>

α)Η γραφική παράσταση του σήματος x για συχνότητα δειγματοληψίας $Fs = 1000Hz$ όπως προκύπτει από την εντολή plot είναι η παρακάτω:

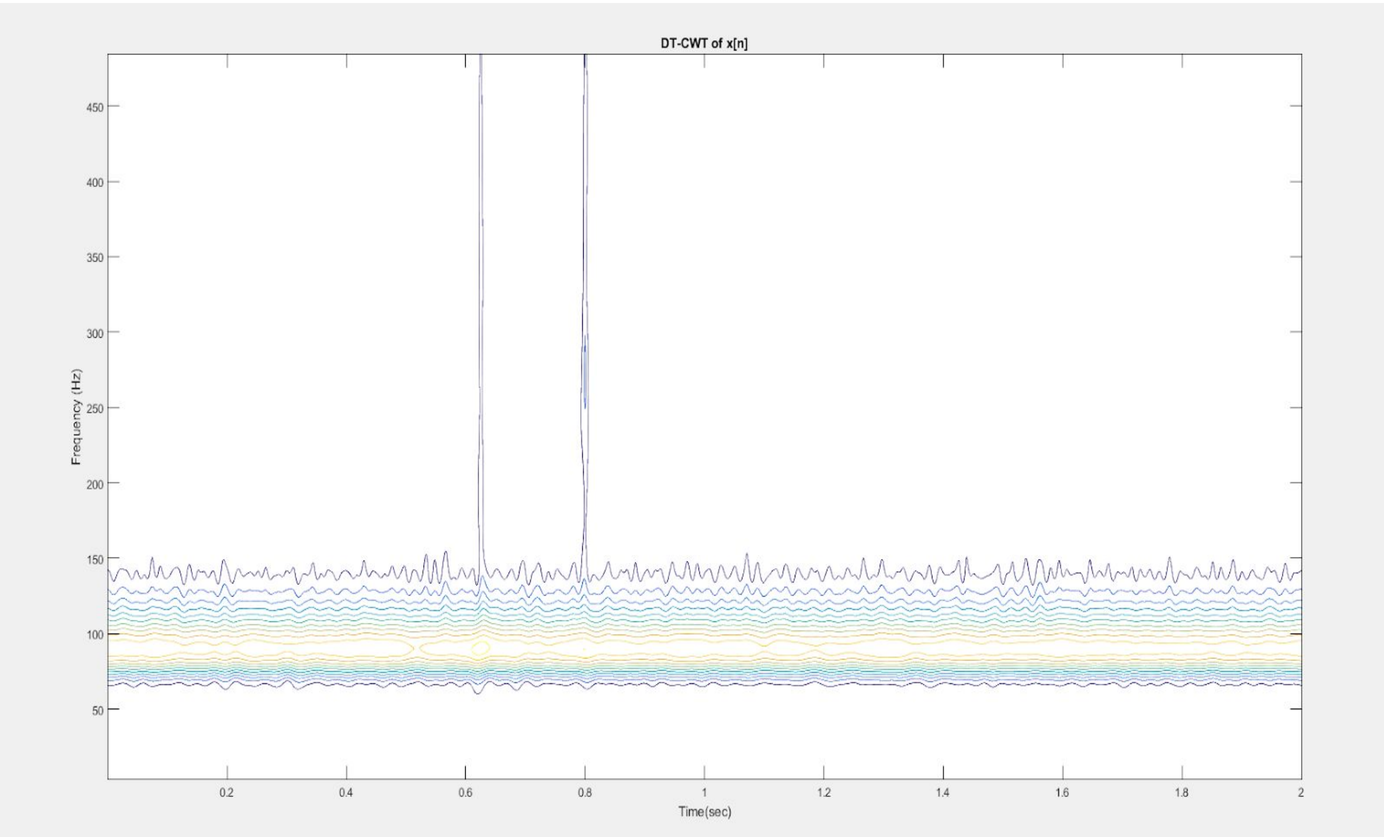


β)Έπειτα με χρήση της συνάρτησης spectogram() λάβαμε την παρακάτω γραφική παράσταση για το παράθυρο που χρησιμοποιήσαμε, την οποία αποτυπώσαμε με την βοήθεια της συνάρτησης contour() . Η γραφική αυτή όπως φαίνεται αποτελεί μια κάτοψη της surf().

Για παράθυρο μήκους 40ms και overlap 20ms έχουμε:



γ)Η γραφική παράσταση του μετασχηματισμό DT-CWT του σήματος αυτού με χρήση της συνάρτησης contour() είναι η παρακάτω:

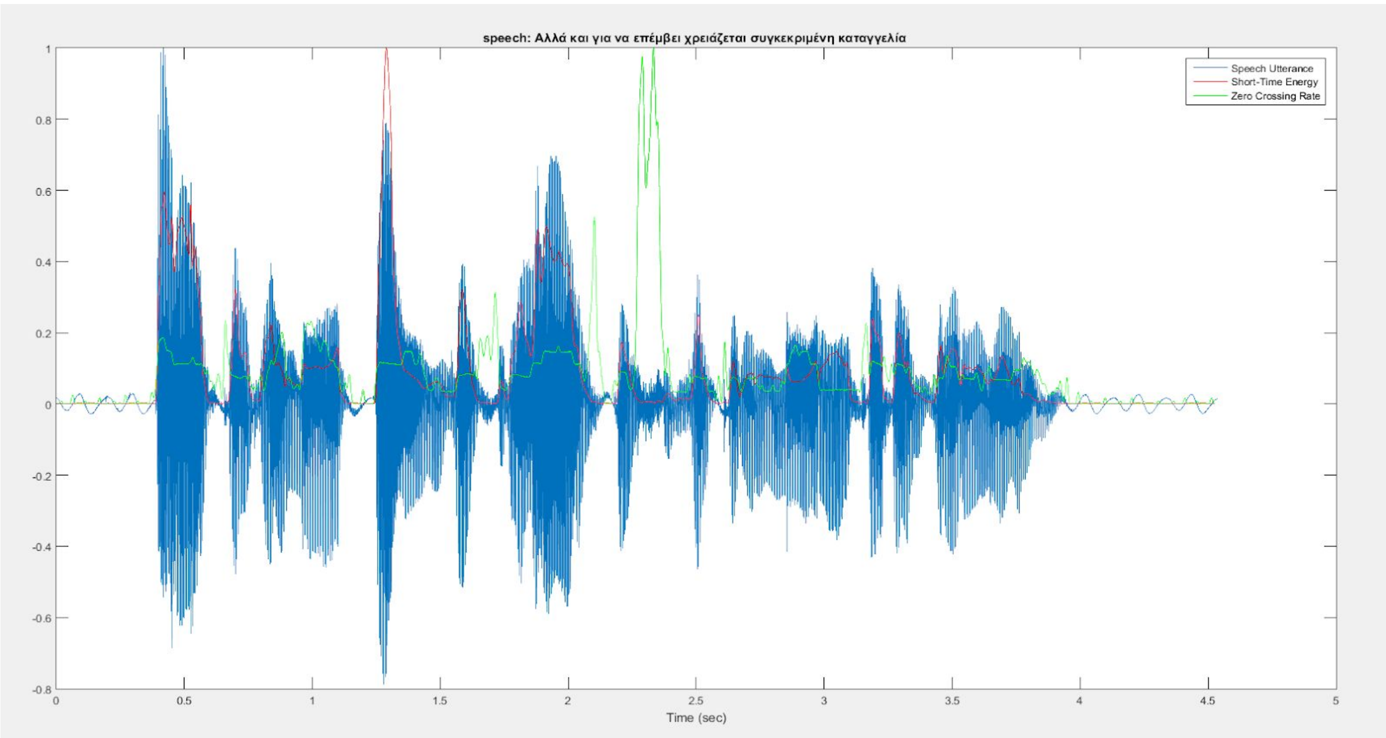


δ) Λόγω της αρχής της αβεβαιότητας του Heisenberg στην επεξεργασία σημάτων, η ανάλυση (resolution) των σημάτων στα πεδία του χρόνου και της συχνότητας είναι ένα trade-off. Δηλαδή δεν μπορεί κανείς να έχει πολύ καλή ανάλυση ταυτόχρονα και στα δύο πεδία. Τα ημίτονα που χρησιμοποιεί ο μετασχηματισμός Fourier, σε αντίθεση με τα κυματίδια είναι άπειρου μήκους. Έτσι μια συχνότητα υπάρχει σε όλο το πεδίο του χρόνου. Αντίθετα τα κυματίδια έχουν περιορισμένο μήκος στο χρόνο και την συχνότητα (αποσβένουν γρήγορα). Έτσι ο μετασχηματισμός κυματιδίων δίνει καλύτερη ανάλυση στο πεδίο του χρόνου σε σχέση με τον Fourier μετασχηματισμό. Όπως είναι φανερό λοιπόν στον μετασχηματισμό DT-CWT παρατηρούμε αρκετά καλά και τις 2 dirac καθώς και τις κεντρικές τους συχνότητες. Οι dirac αυτές φαίνονται και στον STFT με μικρότερη ακρίβεια, ενώ αποτυπώνονται καλύτερα οι αρμονικές. Ο wavelet μετασχηματισμός δίνει καλύτερη ανάλυση στο φάσμα για υβριδικές συναρτήσεις καθώς έχει σταθερό σχετικό (και όχι απόλυτο) σφάλμα και δεν δίνει ομοιόμορφη ανάλυση σε ολόκληρο το πεδίο της συχνότητας (καθώς το μήκος του παραθύρου δεν είναι σταθερό σε αντίθεση με τον STFT), “θυσιάζοντας” την ανάλυση στο πεδίο του χρόνου.

Μέρος 3ο

Ερώτημα 3.1:

Αρχικά πήραμε το σήμα και το χωρίσαμε σε παράθυρα μήκους 20ms. Για συχνότητα δειγματοληψίας 16kHz , αυτό αντιστοιχεί σε 320 δείγματα ανά παράθυρο. Έτσι στην προκειμένη περίπτωση θέσαμε μήκος παραθύρου L=320 δείγματα και επικάλυψη R=319 δείγματα (προχωράμε κατά ένα δείγμα κάθε φορά). Στη συνέχεια μέσω της συνάρτησης hamming() δημιουργήσαμε ένα παράθυρο Hamming, με αυτά τα χαρακτηριστικά και το πολλαπλασιάσαμε με το σήμα μας. Έτσι λάβαμε το παραθυροποιημένο σήμα και στη συνέχεια με εφαρμογή των σχέσεων που δίνονται στην εκφώνηση φτιάξαμε την εξής γραφική παράσταση, στην οποία απεικονίζεται στο χρόνο με μπλε χρώμα το αρχικό σήμα,με κόκκινο η ενέργεια βραχέος χρόνου και με πράσινο ο ρυθμός εναλλαγής προσήμου. Όλα τα μεγέθη εμφανίζονται κανονικοποιημένα.



Ο διαχωρισμός φωνής από σιωπή από τις παραπάνω γραφικές είναι ιδιαίτερα εύκολος. Χρησιμοποιώντας την ενέργεια μπορούμε να αναζητούμε περιοχές με υψηλή ενέργεια για την φωνή και χαμηλή για την σιωπή. Από το zero crossing rate επίσης μπορούμε να διαχωρίσουμε φωνή από σιωπή διότι στην σιωπή υπάρχει θόρυβος ο οποίος έχει μεγάλο zero crossing rate λόγο τυχαιότητας ενώ σε φωνή η ένταση της φωνής είναι μεγαλύτερη και συνεπώς επειδή στην κανονική φωνή δεν έχουμε τόσο μεγάλο zero crossing rate έχουμε αισθητή διαφορά.

Οφείλουμε να αναφέρουμε πως οι έμφωνοι χαρακτήρες μπορούν να παρασταθούν από περιοδικές συναρτήσεις ενώ οι άφωνοι μοιάζουν με τυχαίο θόρυβο(white noise) με υψηλότερο πλάτος.Γενικότερα με την χρήση των γραφικών που πήραμε είναι δύσκολο να διαχωρίσουμε έμφωνους απο άφωνους διότι κάθε χαρακτήρας μας έρχεται για ιδιαίτερα βραχέα περίοδο και επιπλέον η ένταση ενός χαρακτήρα ενεργειακά είτε αυτός είναι έμφωνος είτε άφωνος είναι περίπου ίδια. Ενώ το zero crossing rate και πάλι δεν φαίνεται να έχει μεγάλες διαφορές γραφικά. Αν θέλουμε υπάρχει η δυνατότητα να διαχωρίσουμε όχι μόνο έμφωνους από άφωνους αλλά και έμφωνους μεταξύ τους και άφωνους μεταξύ τους. Μπορούμε να ηχογραφήσουμε εκ των προτέρων όλους τους χαρακτήρες που μας ενδιαφέρουν και μετά διατρέχοντας για τον καθένα τον πίνακα να τον αντιστοιχίζουμε και αν ταυτίζονται τότε αντιστοιχίζουμε το διάστημα αυτό στον χαρακτήρα αυτό.

Μεγαλώνοντας το μήκος του παραθύρου παρατηρούμε ότι οι καμπύλες που αντιστοιχούν στην ενέργεια βραχέος χρόνου του σήματος αλλα και στο ρυθμό εναλλαγής προσήμου εμφανίζονται εξομαλυμένες, δεν υπάρχουν δηλαδή τόσο απότομες μεταβολές στο πλάτος τους με την πάροδο του χρόνου όσο πριν. Αυτό είναι λογικό και αναμενόμενο, καθώς μεγαλώνοντας το μήκος του παραθύρου συγκρίνουμε όλο και περισσότερα δείγματα του σήματος μεταξύ τους. Μεγαλύτερο μήκος παραθύρου οδηγεί σε καλύτερη ανάλυση στο πεδίο της συχνότητας, αλλά χειρότερη ανάλυση στο πεδίου του χρόνου, καθώς συγκρίνονται όλο και περισσότερα δείγματα, δηλαδή όλο και μεγαλύτερο μέρος του σήματος μαζί, πράγμα που καθιστά την εύρεση της ακριβούς χρονικής στιγμής δυσκολότερη.

<u>Ερώτημα 3.2:</u>

Διαβάζουμε το σήμα music_cut.wav με χρήση της συνάρτησης audioread() του matlab, η οποία μας επιστρέφει εκτός από την ψηφιακή μορφή του σήματος και μία συχνότητα δειγματοληψίας ίση με 44100Hz. Έτσι για παράθυρο μήκους 20ms και αυτήν την συχνότητα δειγματοληψίας, χρησιμοποιήσαμε παράθυρο Hamming με μήκος 882 δείγματα και επικάλυψη 881 δείγματα. Στη συνέχεια χρησιμοποιώντας τις ίδιες συναρτήσεις και σχέσεις με το Ερώτημα 3.1 παράγουμε τις παρακάτω γραφικές για το αρχείο music_cut.wav: