

**Θεοδωρίδης Αριστομένης | 03115632**

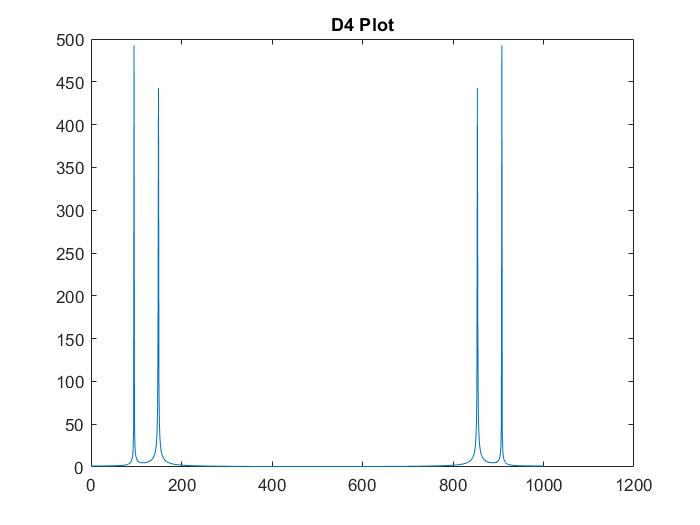
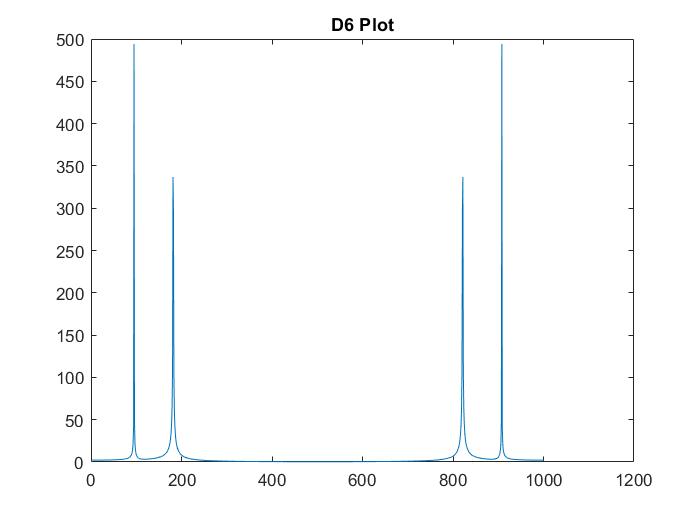
**Αβδελάς Λεωνίδας | 03115632**

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

1η εργαστηριακη ασκηση

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

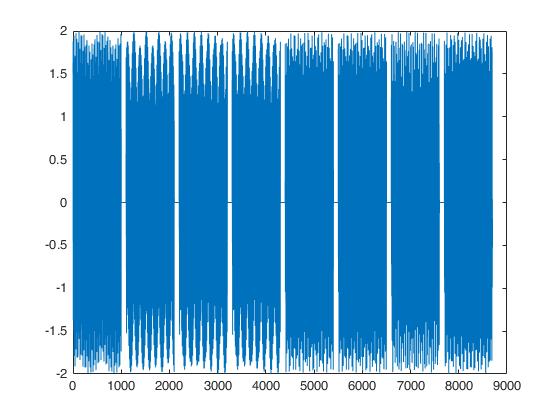
## 1.2

Από τα σήματα d4[n] και d6[n] δημιουργήσαμε τις εξής γραφικές παραστάσεις:

Παρατηρούμε ότι μπορούμε εύκολα να διακρίνουμε τα δύο σήματα μεταξύ τους, λόγω των peaks στις χαρακτηριστικές τους συχνότητες. Ακόμα, λόγω του αλγορίθμου της συνάρτησης **fft()** που χρησιμοποιούμε, οι αρνητικές συχνότητες έχουν μετατοπιστεί κυκλικά στις συχνότητες μεγαλύτερες των 500Hz. Οι θετικές συχνότητες, οι οποίες μας ενδιαφέρουν εδώ, είναι αυτές από 0-500Hz.

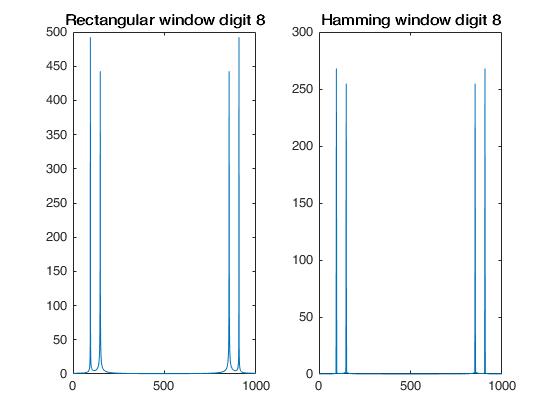
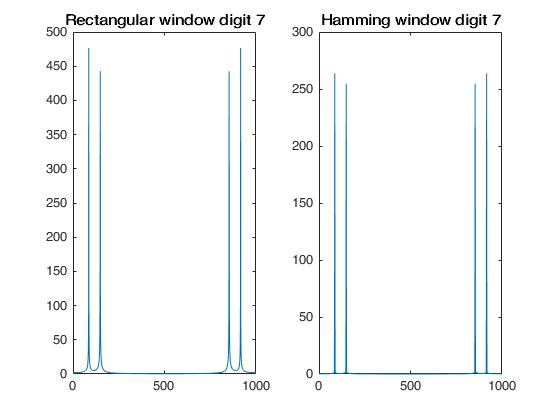
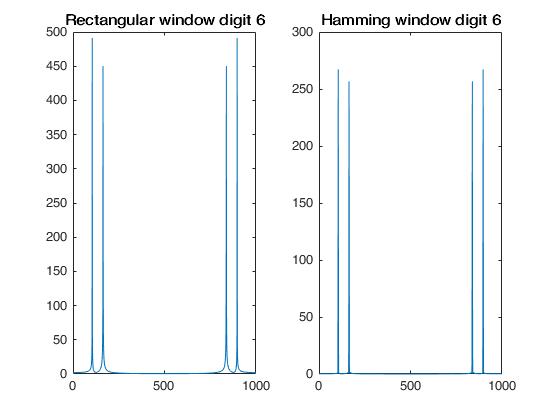
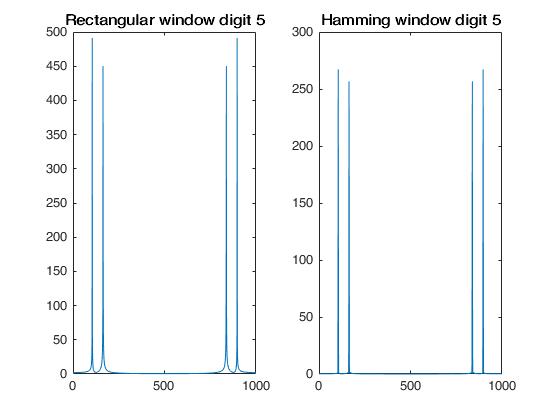
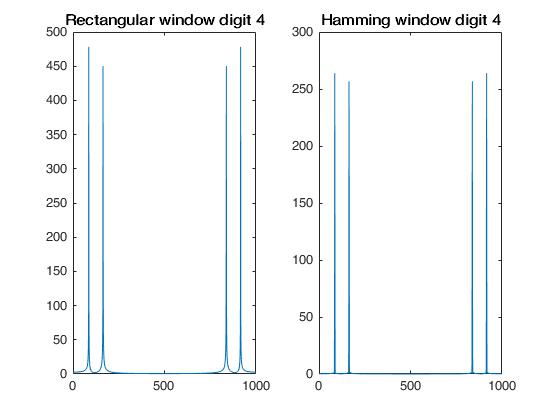
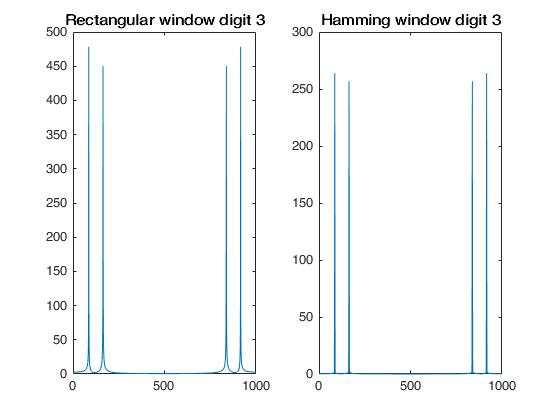
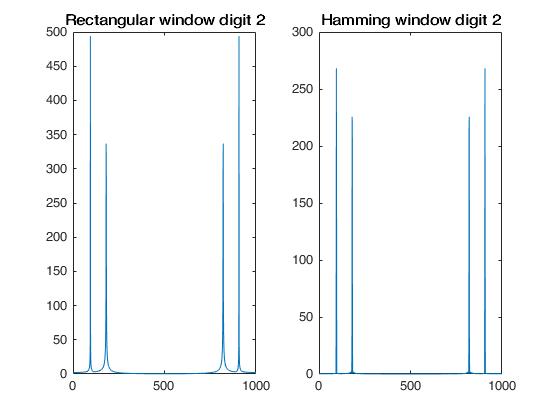
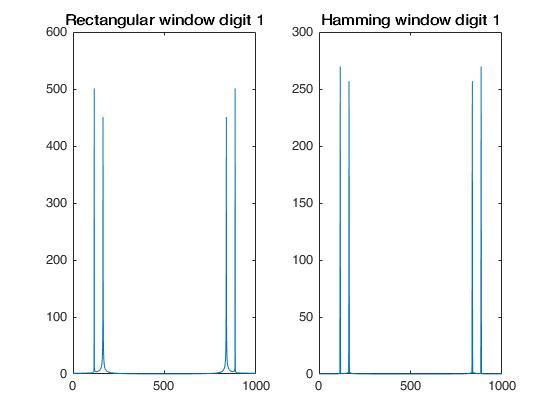
## 1.3

Το συνδυασμένο ΑΜ μας είναι 06228814. Για να δημιουργήσουμε το αρχείο ‘tone\_sequence.wav’, δημιουργήσαμε έναν πίνακα με τα tone sequences που δημιουργήσαμε στο ερώτημα 1.1 και 100 μηδενικά μεταξύ τους.



## 1.4

Παρακάτω βλέπουμε τις γραφικές που παίρνουμε από την παραθυροποίηση και εφαρμογή της συνάρτησης **fft()** στα παράθυρα του σήματος AM\_sound.



Όπως βλέπουμε, χρησιμοποιώντας παράθυρα Hamming, τα peaks συγκεντρώνονται γύρω από συγκεκριμένες συχνότητες και η διάκριση των κύριων συχνοτήτων είναι πιο ξεκάθαρη.

### 1.5

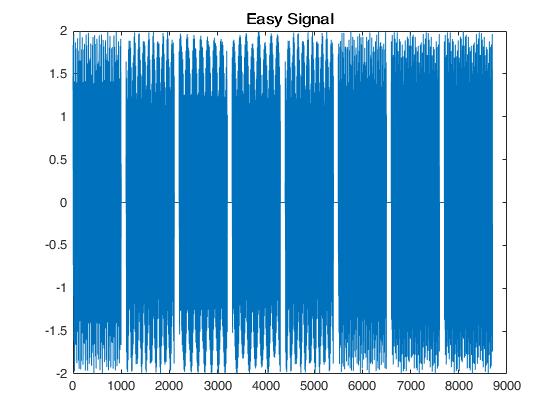
Για να υπολογίσουμε την λίστα με τις touch-tone συχνότητες, οπτικά αυτό που κάνουμε είναι να βρούμε τα δείγματα στα οποία αντιστοιχούν τα μέγιστα των συχνοτήτων. Λόγω του ότι δεν υπάρχουν απόλυτες συναρτήσεις Dirac στον πραγματικό κόσμο, οι μέγιστες τιμές δεν είναι σε ένα σημείο, αλλά σε ένα μικρό πεδίο γύρω από την κύρια συχνότητα. Για να αντιμετωπίσουμε αυτό το πρόβλημα, ελέγχουμε για το δεύτερο peak, σε περιοχή που είναι τουλάχιστον 10 δείγματα μακριά από το πρώτο.

## 1.6

Η συνάρτηση **ttdecode()** δέχεται ως παραδοχή, ότι όλοι οι τόνοι είναι 1000 δείγματα και ότι ανάμεσα σε κάθε τόνο υπάρχει ένα κενό 100 δειγμάτων. Έτσι, με αυτό το μοτίβο σπάει το σήμα σε παράθυρα μέχρι να φτάσει στο τέλος του και μετά χρησιμοποιεί μετασχηματισμό fft σε κάθε ένα από αυτά. Χρησιμοποιώντας παράθυρα Hamming, έχουμε καλύτερη διακριτική ικανότητα σχετικά με τις κορυφές των σημάτων μας, όπως αναφέρθηκε και παραπάνω. Έπειτα, αποσπούμε την πληροφορία σχετικά με το που βρίσκονται οι κορυφές (δηλαδή τα σημεία με μέγιστη ενέργεια) με τον ίδιο τρόπο που αναφέρθηκε στο βήμα 1.5. Για να βρούμε σε ποιο ψηφίο αντιστοιχούν αυτές οι κορυφές, συγκρίνουμε τα σημεία που πήραμε με τον πίνακα που δημιουργήσαμε στον βήμα 1.5, έχοντας ένα περιθώριο λάθους 3 δειγμάτων. Η συνάρτηση αυτή προσδιορίζει τέλεια τους τόνους που δημιουργήσαμε στο βήμα 1.3.

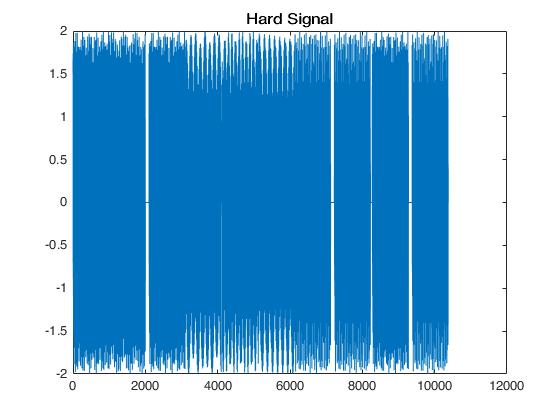
## 1.7

Το εύκολο σήμα, προσδιορίζεται από την **ttdecode()** που έχουμε δημιουργήσει χωρίς πρόβλημα, καθώς ακολουθεί τις παραδοχές που είχαμε ορίσει νωρίτερα. Αυτό φαίνεται εύκολα και από την γραφική του σήματος που ακολουθεί.



Η ακολουθία ψηφιών στην οποία καταλήγουμε είναι η 1 3 2 6 3 9 0 0.

Το hardSig, βλέπουμε εύκολα ότι δεν ακολουθεί τις παραδοχές που είχαμε ορίσει για την συνάρτηση μας στο βήμα 1.6. Αυτό που θα κάνουμε λοιπόν είναι να την αλλάξουμε ώστε να μπορεί να αντιμετωπίσει μεταβλητό μήκος κενού μεταξύ των τόνων, δεδομένου ότι οι τόνοι συνεχίζουν να είναι 1000 δειγμάτων.



Ο τρόπος που το κάνουμε αυτό είναι να βρίσκουμε κάθε φορά που τελειώνει ένας τόνος, το πρώτο δείγμα που δεν είναι 0 στο σήμα μας και να παίρνουμε ένα παράθυρο 1000 δειγμάτων, το οποίο ξεκινάει από εκείνο το δείγμα, ως τον επόμενο τόνο. Δοκιμάζοντας την νέα υλοποίηση της **ttdecode()** στο σήμα του 1.3 και στο easySig παίρνουμε τα ίδια αποτελέσματα με πριν. Δοκιμάζοντας το στο hardSig, παίρνουμε την ακολουθία ψηφίων 9 0 9 6 3 2 1 1 9 1.