Όνομα: Άγγελος

Επώνυμο: Ευστρατίου

ΑΜ: 03121113

Ψηφιακή Επεξεργασία Σημάτων

1η Εργαστηριακή Άσκηση

Μέρος 1ο – Σύστημα Εντοπισμού Τηλεφωνικών Τόνων

Περιγραφή

Σκοπός της άσκησης είναι να δημιουργήσουμε σήματα που είναι το άθροισμα 2 ημιτόνων συγκεκριμένης συχνότητας και στη συνέχεια από τα σήματα να ανακατασκευάσουμε τους αριθμούς που αντιπροσoπεύουν. Κάθε δυάδα συχνοτήτων αντιστοιχεί σε έναν και μοναδικό αριθμό του δεκαδικού συστήματος σύμφωνα με τον παρακάτω πίνακα.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ω\_column | | |
| 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 1.1 - Διακριτές Συχνότητες για Τηλεφωνικούς Τόνους για Συχνότητα Δειγματοληψίας 8192 Hz.

* 1. Δημιουργία 10 διαφορετικά σήματα με 1000 δείγματα

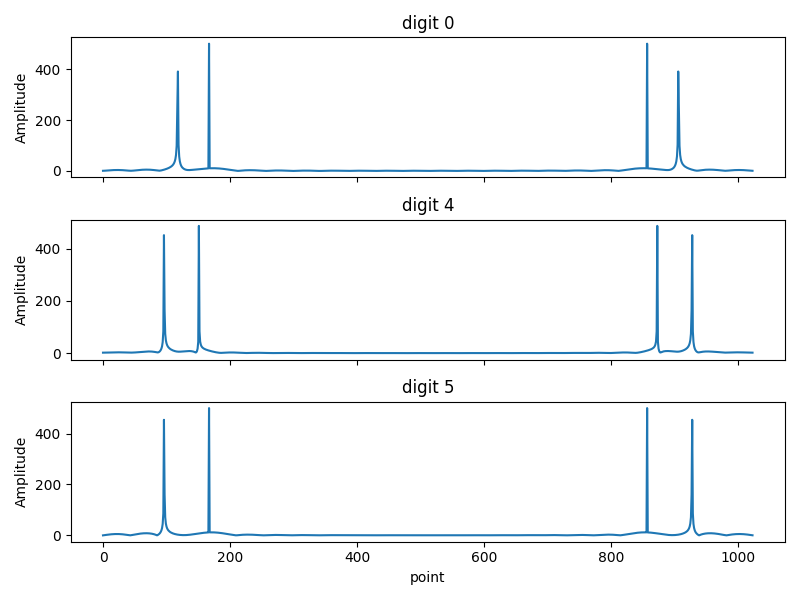
Για κάθε αριθμό στο δεκαδικό σύστημα μπορούμε να φτιάξουμε το αντίστοιχο σήμα που προκύπτει σύμφωνα με τον παραπάνω πίνακα. Έπειτα, μπορούμε να αποθηκεύσουμε όλα αυτά τα σήματα σε έναν 10x1000 πίνακα που θα τον ονμάσουμε tone, όπου το στοιχείο tone[i][j] είναι το j+1-οστό δείγμα του σήματος που κωδικοποιεί τον αριθμό i.

* 1. Υπολογισμός DFT των σημάτων

Χρησιμοποιώντας την συνάρτηση fft της βιβλιοθήκης numpy στα σήματα tone[4], tone[5], tone[0] υπολογίζουμε τον διακριτό μετασχηματισμό fourier των σημάτων. Δηλαδή ένα διάνυσμα με τους συντελεστές του μετασχηματισμένου σήματος. Επειδή τα σήματα έχουν 1000 δείγματα και ζητείται ο DFT 1024 σημείων, στα σήματα προσθέτουμε 24 μηδενικά δείγματα. Στη συνέχεια, υπολογίζουμε το μέτρο για κάθε ένα από τα 1024 σημεία κάθε διανύσματος, οπότε βρίσκουμε τα |D0[k]|, |D4[k]| και |D5[k]|. Απεικονίζουμε τα αποτελέσματα στο διάγραμμα 1.1. Παρατηρούμε πως υπάρχουν δύο μεγάλες κορυφές σε κάθε εικόνα, που αντιστοιχούν στις συχνότητες των σημάτων. Πιο συγκεκριμένα, η κορυφή που αντικατοπτρίζει τη συχνότητα ενός ημιτινοειδούς σήματος sin(Ω\*n), βρίσκεται περίπου στο (1024 / 2π)\*Ω σημείο.

* 1. Δημιουργία αρχείου tone\_sequence.wav

Για τον σκοπό του ερωτήματος, φτιάχνουμε ένα διάνυσμα με αριθμούς τα ψηφία του αριθμού μητρώου και το ονομάζουμε identification\_number. Έπειτα αντιστοιχούμε το i – οστό δείγμα του διανύσματος, identification\_number[i], με το σήμα tone[identification\_number]. Προσθέτουμε όλα αυτά τα σήματα σε ένα μεγάλο σήμα, το ένα μετά το άλλο, και τα διαχωρίζουμε με ένα διάνυσμα



Διάγραμμα 1.1 – Το μέτρο του fft των σημάτων tone[0], tone[4], tone[5]

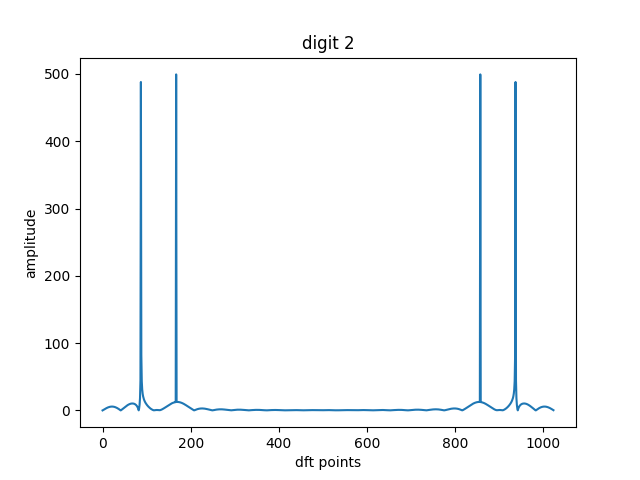
που αποτελείται από 100 μηδενικά. Τέλος, αποθηκεύουμε το προκύπτον σήμα σε ένα .wav αρχείο με όνομα tone\_sequence.wav.

* 1. Υπολογισμός fft παραθυροποιημένων σημάτων

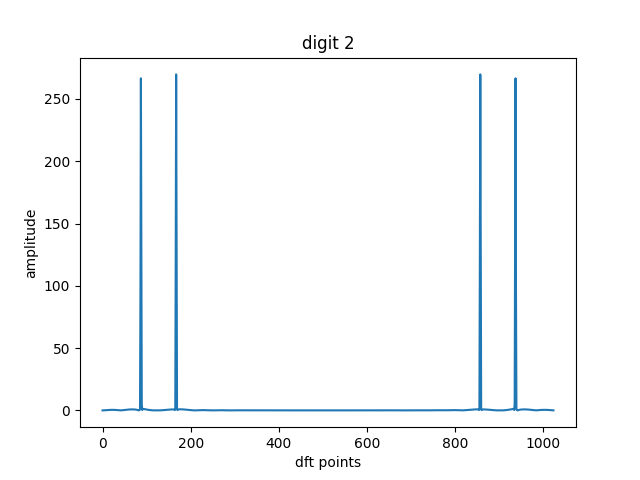
Θέλουμε να ανακατασκευάσουμε τα ψηφία του σήματος που δημιουργήσαμε στο προηγούμενο ερώτημα. Για να το καταφέρουμε αυτό πρέπει αρχικά να αποκόψουμε τα σήματα που αντιστοιχούν σε κάθε ψηφίο, δηλαδή τα tone[identification\_number]. Αυτό θα το καταφέρουμε με τη χρήση παραθύρων. Πολλαπλασιάζουμε το παράθυρο με το μέρος του σήματος που θέλουμε να αποκόψουμε και έτσι διαχωρίζουμε τα σήματα. Παίρνουμε πάλι τον fft του αποκομένου σήματος και στη συνέχεια παίρνουμε το μέτρο των δειγμάτων του σήματος τα οποία μπορούμε να απεικονίσουμε. Στα διαγράμματα 1.2 και 1.3 φαίνονται τα αποτελέσματα για το 4ο ψηφίο του αριθμού μητρώου (03121113) δηλαδή το 2. Την πρώτη φορά χρησιμοποιούμε ορθογώνια παράθυρα και τη δεύτερη παράθυρα hamming. Παρατηρούμε ότι τα αποτελέσματα είναι αρκετά καλύτερα με την χρήση hamming window επειδή οι κορυφές φαίνονται εντονότερα ενώ το πλάτος γύρω από τις κορυφές έχει μηδενιστεί.

1.6. Υπολογισμός ακολουθίας ψηφίων από το κωδικοποιημένο σήμα

Για να φτιάξουμε μια συνάρτηση που αποκοδικοποιεί ένα σήμα που δημιουργείται σύμφωνα με το ερώτημα 1.3. και επιστρέφει τα ψηφία σε ένα διάνυσμα, ακολουθούμε την εξής διαδικασία.



Διάγραμμα 1.2 – Μέτρο του fft του σήματος που προκύπτει από παραθυροποίηση το σήματος tone\_sequence.wav με χρήση rectangular window



Διάγραμμα 1.3 – Μέτρο του fft του σήματος που προκύπτει από παραθυροποίηση το σήματος tone\_sequence.wav με χρήση hamming window

Αρχικά, όπως κάναμε στο ερώτημα 1.4. απομονώνουμε τα σήματα που κωδικοποιούν το κάθε ψηφίο με τη χρήση παραθύρων. Έπειτα, υπολογίζουμε τον fft για κάθε παραθυροποιημένο σήμα και στη συνέχεια υπολογίζουμε το μέτρο του κάθε σημείου του μετασχηματισμένου σήματος. Όπως είδαμε στο προηγούμενο ερώτημα το αποτέλεσμα είναι ένα διάνυσμα με κορυφές σε σημεία που αντιστοιχούν στις συχνότητες των ημιτόνων του αρχικού σήματος. Χρησιμοποιώντας την συνάρτηση find\_peaks της scipy βρίσκουμε τα σημεία που αντιστοιχούν σε αυτές τις κορυφές. Από το ερώτημα 1.2. είδαμε πως πρέπει να κανονικοποιήσουμε τα σημεία αυτά με την τιμή (1024 / 2π) για να βρούμε τις πραγματικές συχνότητες των αρχικών σημάτων. Τώρα μένει μονάχα να αντιστοιχήσουμε τις συχνότητες αυτές με το σωστό ψηφίο σύμφωνα με τον πίνακα 1.1. Για να το κάνουμε αυτό, δημιουργούμε μια συνάρτηση, την frequencies\_to\_digit. Αυτή η συνάρτηση επιστρέφει το ψηφίο που η δυάδα συχνοτήτων του ταιριάζει καλύτερα με τις συχνότητες που υπολογίσαμε. Εδώ με τη λέξη ταιριάζει, εννοούμε πως ελαχιστοποιεί την ευκλείδια απόσταση των δύο ζευγαριών συχνοτήτων. Κάνοντας αυτήν τη διαδικασία για κάθε παραθυροποιημένο σήμα, βρίσκουμε την ζητούμενη ακολουθία ψηφίων, την οποία και επιστρέφουμε. Δοκιμάζουμε αυτή τη συνάρτηση (ttdecode) με όρισμα εισόδου το σήμα tone\_sequence που φτιάξαμε στο ερώτημα 1.3 και χαιρόμαστε όταν βλέπουμε πως πράγματι επιστρέφει τον αριθμό μητρώου 0 3 1 2 1 1 1 3.

1.7. Αποκοδικοποίηση σημάτων αποθηκευμένων σε αρχεία με χρήση της ttdecode.

Φορτώνουμε τα σήματα easy\_sig, medium\_sig και hard\_sig και τα αποκωδικοποιούμε με τη συνάρτηση που φτιάξαμε στο προηγούμενο ερώτημα. Τελικά βρίσκουμε τα ακόλουθα αποτελέσματα:

easy\_sig: 9 3 9 2 6 8 4 8

medium\_sig: 2 7 3 3 2 5 4 2 5 4

hard\_sig: 6 9 2 6 3 1 9 9 1 0