Σήματα & Συστήματα

Υπολογιστική Εργασία σε MATLAB

Εαρινό εξάμηνο 2019

Νίκος Πλέσσας

ΑΕΜ:615

Στο πρώτο κομμάτι της εργασίας γράφονται οι παρατηρήσεις και τα σχόλια. Στο τέλος επισυνάπτεται ο κώδικας με τα σχόλιά του και όλα τα plots όπως έγινε publish από το matlab.

Παρατηρήσεις - σχόλια

Μέρος Α.

Α.1,2.3

Λόγω έλλειψης μικροφώνου στο laptop μου ηχογράφησα στο κινητό μου το ηχητικό με το όνομά μου, το πέρασα στο Goldwave, το μετέτρεψα σε .wav και έκανα resample στα 8kHz.

Το .wav φορτώθηκε στο matlab και σχεδιάστηκε, μετατράπηκε σε double και φασματογραφήθηκε.

Α.4

Έπειτα σχεδιάστηκαν στο ίδιο παράθυρο τα δύο μέτρα του συχνοτικού του περιεχομένου σε κανονική και λογαριθμική κλίμακα με τον άξονα x να έχει όρια (0,2π).

Α.5

Κατά την αναπαραγωγή του αρχείου, με τις εντολές audioreader & play() το ηχητικό δεν "έπαιζε". Έτσι χρησιμοποιήθηκε (εδώ και όπου ξαναχρειαστηκε) η εντολή sound().

Μέρος Β.

Β.1

Το σύστημά που δίνεται $y[n]=x[n]+a*x[n-n0]$ έχει συνάρτηση μεταφοράς $H(z)=1+0.5*z^{-10}$ για α = 0.5 και n0 = 10. Οι συναρτήσεις **impz**() & **stepz**() έχουν ως ορίσματα διανύσματα που αναφέρονται στον αριθμητή και παρονομαστή της συνάρτησης μεταφοράς. Τα όρισα ως διανύσματα 10 θέσεων (a1,b1) που αναπαριστούν τα 2 πολυώνυμα. Οι δύο αποκρίσεις, βηματική και κρουστική σχεδιάστηκαν στο διάστημα n=[0:1:20]

B.2

Για τις διάφορες τιμές του α (0.1 , 0.01, 0.001) σχεδιάστηκαν με την **zplane()** τα διαγράμματα πόλων και μηδενικών του συστήματος, χρησιμοποιώντας ξανά σαν ορίσματα τα διανύσματα a1,b1 με διαφορετικές τιμές του α για το κάθε διάγραμμα. Όσο οι τιμές του α μικραίνουν, τόσο "πλησιάζουν" τα μηδενικά

Για την τελευταία τιμή του α (0.001) σχεδιάστηκαν και οι ρίζες αριθμητή & παρονομαστή με την **roots**(). Οι θέσεις των μηδενικών και τον πόλων συμπιπτουν στα 2 διαγράμματα (με την **roots** και την **zplane**).

Μέρος Γ.

Γ.1,2

Το σύστημά που δίνεται $y[n]=x[n]+a*x[n-n0]$ έχει συνάρτηση μεταφοράς $H(z)=1+0.5*z^{-2000}$ για α = 0.5 και n0 =2000. Τα ορίσματα στην filter() χρειάζεται να είναι διανύσματα 2000 θέσεων όπου το διάνυσμα του αριθμητή θα έχει στην τελευταία του θέση την τιμή 0.5. Όταν καλεστεί η filter() με ορίσματα τα παραπάνω διανύσματα και το ηχητικό και αναπαραχθεί το ηχητικό παρατηρούμε ότι πλεον έχει echo.

Γ.3

Το διάγραμμα του autocorrelated ηχητικού με echo βλέπουμε ότι "ανεβαίνει" περισσότερο (μεγαλύτερο μέτρο) από αυτό του πρωτότυπου (χωρίς echo). Η ηχώ επίσης, είναι φανερή στο διάγραμμα (περιεχόμενο που επαναλαμβάνεται).

Μέρος Δ.

Δ.1

Το αντίστροφο σύστημα βρίσκεται ως εξής: Εφόσον x → h → y , y → h(inv) → x. Στο πεδίο του μετασχηματισμού θα ισχύει H(z) * Hinv(z) = 1 => Hinv (z) = 1/H(z)
=> $Hinv=1/1+0.5*z^{-2000}$ που μας δίνει το σύστημα: $y[n]+0.5*y[n-2000]=x[n]$

Για να το σχεδιάσουμε αρκεί να πάρουμε αντίστροφα με πριν τα διανύσματα παρονομαστή/αριθμητή a1,b1. Με χρήση της impz() βρίσκουμε και σχεδιάζουμε την κρουστική του απόκριση για τιμές του n [0:1:20000].

Δ.2,3

Σε αυτά τα ερωτήματα θέλουμε να "φιλτράρουμε" το σήμα με τις πρώτες 3.000 και 15.000 της κρουστικής απόκρισης που βρήκαμε με την **impz**() στο ερώτημα Δ1 . Η εντολή filter() θέλει σαν ορίσματα τον αριθμητή και τον παρονομαστή της συνάρτησης μεταφοράς. Εδώ για να υλοποιήσω το σύστημα με βάση την κρουστική απόκριση του, αποφάσισα να χρησιμοποιήσω την **conv**() για να κάνω συνέλιξη του φιλτραρισμένου σήματος με τα πρώτα 3.000 και 15.000 στοιχεία του διανύσματος που περιέχει τιμές της κρουστικής.

Παρατηρούμε ότι στην περίπτωση όπου παίρνουμε τις πρώτες 3.000 τιμές της κρουστικής το φαινόμενο της ηχούς δεν αναιρείται, ενώ χρησιμοποιώντας τις 15.000 τιμές το σήμα όταν αναπαραχθεί ακούγεται παρόμοιο με το αρχικό. Το ίδιο παρατηρούμε και στα διαγράμματα των 2 σημάτων (μετά από το αντίστροφο φιλτράρισμα).
Στο 2ο διάγραμμα φαίνεται ότι η επανάληψη του περιεχομένου (echo) έχει σχεδόν εξαφανιστεί.