Σήματα και Συστήματα – Εργασία MATLAB (2020-21)

Φοιτητής: Ανδρέας Καλαβάς

ΑΜ: 03119709

1.1 Σχεδίαση Φίλτρων Ηχούς και Αντήχησης (Echo, Reverb)

Ένα αιτιατό φίλτρο διακριτού χρόνου ορίζεται μέσω της εξίσωσης διαφορών:

$$y[n] + \sum_{i=1}^{N} a_i y[n-i] = \sum_{i=0}^{M} b_i x[n-i]$$

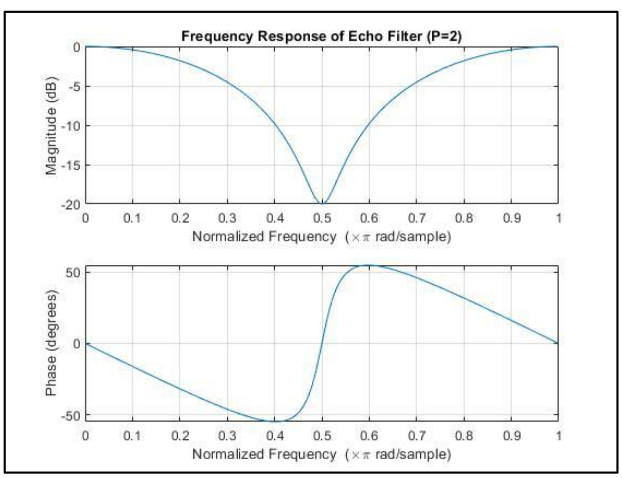
Το φίλτρο που θα χρησιμοποιήσουμε περιγράφεται από την εξίσωση:

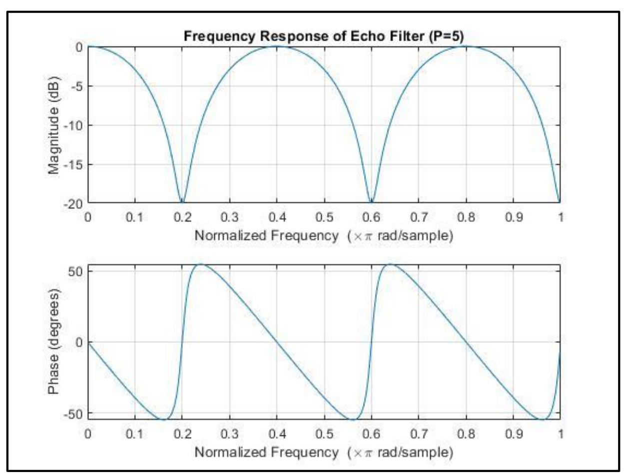
$$y[n] = cx[n] + (1-c)x[n-P], c \in [0,1], P \in \mathbb{N}$$

α) Για τιμές P=2 ή P=5, και c=0.55 ορίζουμε τα διανύσματα $a = [a_i]$ και $b = [b_i]$ των εξισώσεων διαφορών που αντιστοιχούν στο κάθε φίλτρο:

- Για P=2, c=0.55, η εξίσωση γίνεται $y[n] = 0.55x[n] + 0.45x[n-2]$ και άρα $a = [a_0] = [1]$, $b = [b_0\ b_1\ b_2] = [0.55\ 0\ 0.45]$.
- Για P=5, c=0.55, η εξίσωση γίνεται $y[n] = 0.55x[n] + 0.45x[n-5]$ και άρα $a = [a_0] = [1]$, $b = [b_0\ b_1\ b_2\ b_3\ b_4\ b_5] = [0.55\ 0\ 0\ 0\ 0\ 0.45]$.

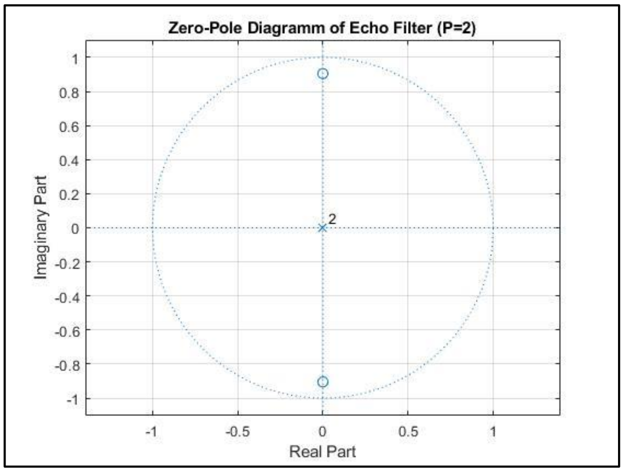
β) Με την εντολή freqz() σχεδιάζουμε την απόκριση συχνότητας των δύο φίλτρων:

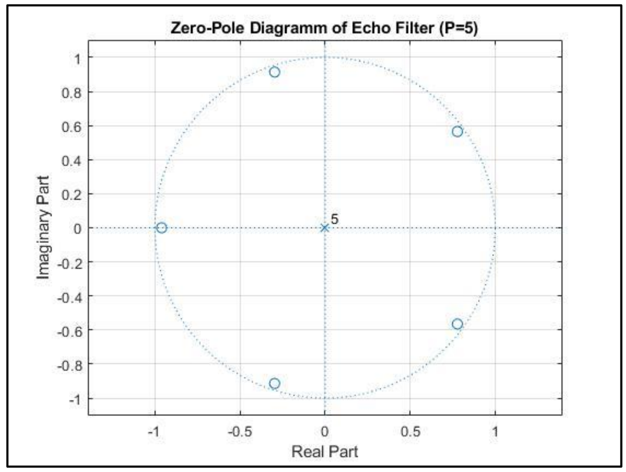




Παρατηρούμε ότι η απόκριση του φίλτρου με P=2 έχει περίοδο π, ενώ του φίλτρου με P=5 έχει περίοδο 0.4π, δηλαδή τα 2/5 του πρώτου φίλτρου.

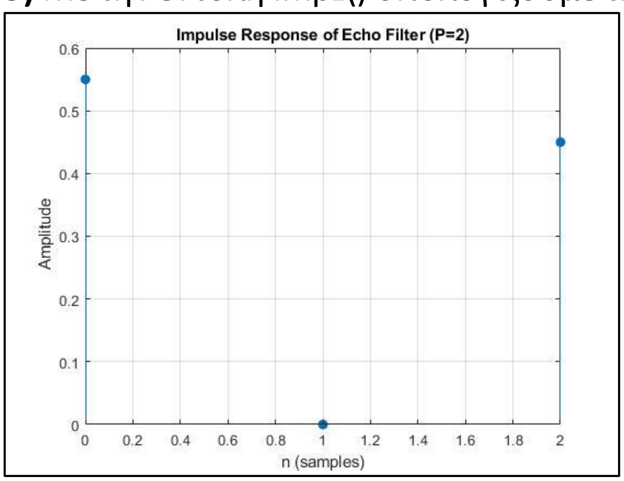
γ) Με την εντολή zplane() σχεδιάζουμε τα διαγράμματα πόλων και μηδενικών των δύο φίλτρων:

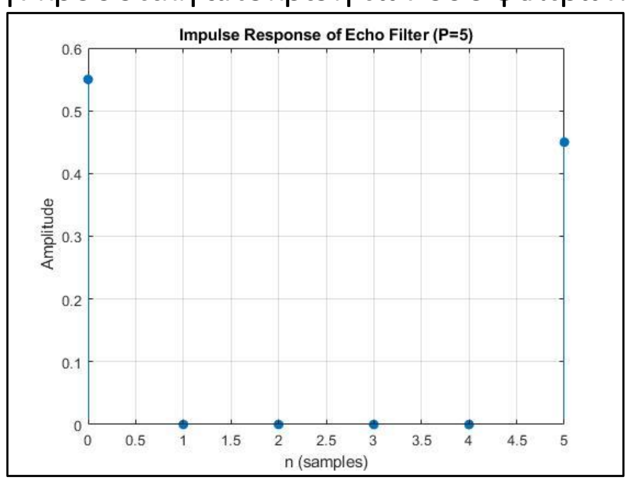




Παρατηρούμε ότι όσο είναι το P, τόση είναι και η πολλαπλότητα του κεντρικού πόλου, όπως επίσης τόσα είναι και τα μηδενικά, τα οποία είναι ομοιόμορφα κατανεμημένα γύρω από τον πόλο.

δ) Με την εντολή impz() υπολογίζουμε την κρουστική απόκριση των δύο φίλτρων:





Τα αποτελέσματα ήταν αναμενόμενα, αφού από την εξίσωση των φίλτρων καταλαβαίνουμε ότι θα είναι αυτή η μορφή της κρουστικής απόκρισης (μετατοπισμένη είσοδος στην εξίσωση διαφορών του φίλτρου σε αυτή την περίπτωση προκύπτει από συνέλιξη με μετατοπισμένο κρουστικό παλμό).

ε) Για να δημιουργήσουμε ένα φίλτρο αντήχησης τρίτου βαθμού, μπορούμε να χρησιμοποιήσουμε τρία διαδοχικά φίλτρα ηχούς, και αφού P=2, c=0.55 έχουμε:

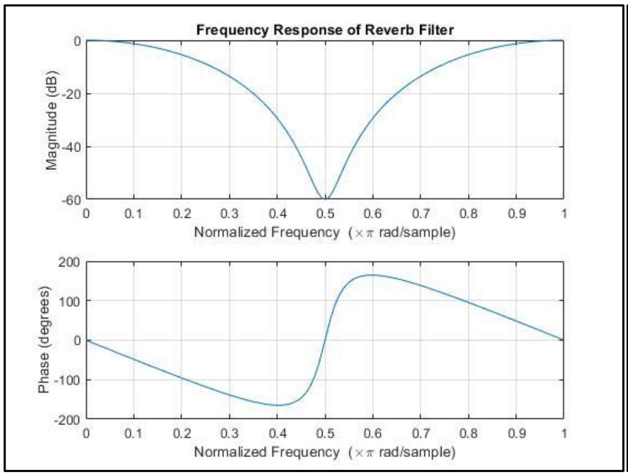
$$y_1[n] = 0.55x[n] + 0.45x[n-2], \qquad y_2 = 0.55y_1[n] + 0.45y_1[n-2],$$

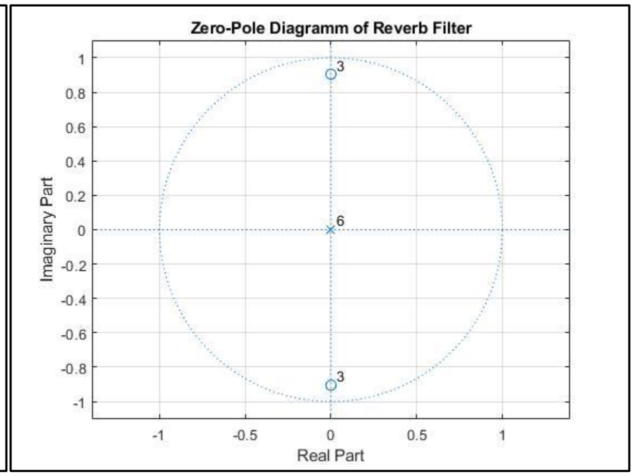
$$y[n] = 0.55y_2[n] + 0.45y_2[n-2] = \cdots$$

Τελικά καταλήγουμε στο ότι:

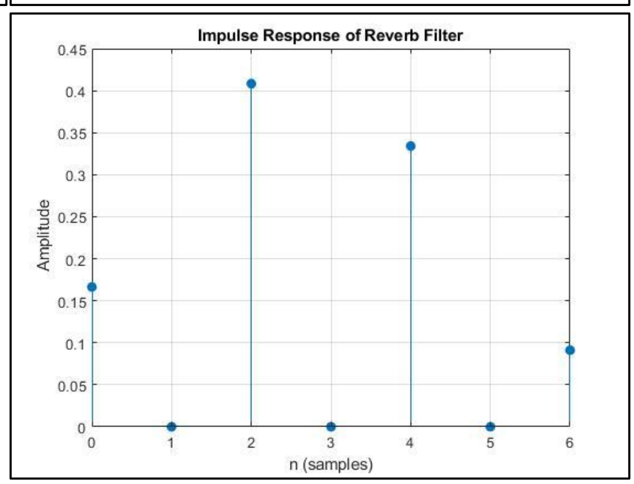
$$y[n] = 0.166375x[n] + 0.408375x[n\text{-}2] + 0.334125x[n\text{-}4] + 0.091125x[n\text{-}6]$$

Άρα από την εξίσωση διαφορών έχουμε ότι οι συντελεστές είναι: $a = [a_0] = [1]$, $b = [b_0\ b_1\ b_2\ b_3\ b_4\ b_5\ b_6] = [0.166375\ 0\ 0.408375\ 0\ 0.334125\ 0\ 0.091125]$. Στη συνέχεια σχεδιάζουμε την απόκριση συχνοτητας, το διάγραμμα πόλων και μηδενικών, και υπολογίζουμε την κρουστική απόκριση του φίλτρου:





Παρατηρούμε ότι η απόκριση συχνότητας επίσης και οι πολλαπλότητες των μηδενικων και των πόλων είναι τριπλάσιες από την απόκριση συχνότητας και των πολλαπλοτήτων των μηδενικών και των πόλων του φίλτρου ηχούς (με P=2). Αυτό εξηγείται από το ότι το φίλτρο αντληχησης είναι 3 διαδοχικά φίλτρα ηχούς.

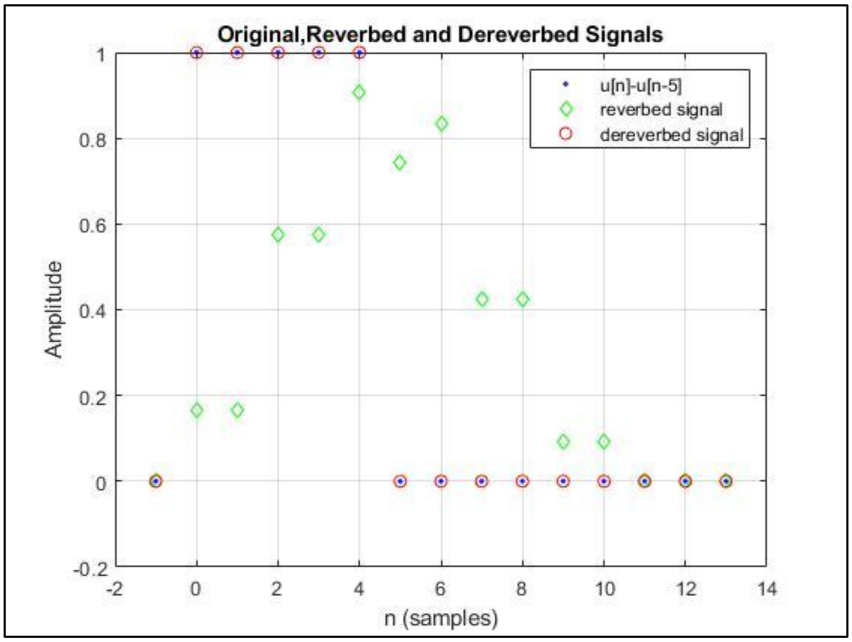


στ) Ζητούμε $h_1[n] * h_2[n] = \delta[n]$ και άρα $H_1(\Omega)H_2(\Omega) = 1$. Αφού ξέρουμε ότι

$$H(\Omega) = \frac{\sum_{i=0}^{M} b_i e^{-j\Omega i}}{\sum_{i=0}^{N} a_i e^{-j\Omega i}}$$

$H_1(\Omega)$=0.166375+0.408375$e^{-j2\Omega}$+0.334125$e^{-j4\Omega}$+0.091125$e^{-j6\Omega}$

$H_2(\Omega)$=1/$H_1(\Omega)$ και άρα τα διανύσματα συντελεστών a, b είναι: $b = [b_0] = [1]$, $a = [a_0\ a_1\ a_2\ a_3\ a_4\ a_5\ a_6] = [0.166375\ 0\ 0.408375\ 0\ 0.334125\ 0\ 0.091125]$. Πιο κάτω φαίνεται η χρήση εισόδου x[n]=u[n]-u[n-5] για επαλήθευση:

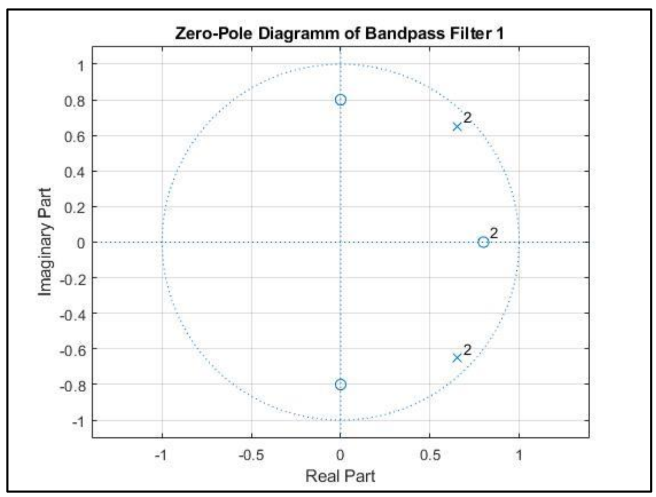


Παρατηρούμε ότι το αρχικό μας σήμα (μπλε κουκίδες), είναι ίδιο με το τελικό μας σήμα (κόκκινα κυκλάκια). Άρα οι πράξεις μας είναι σωστές, και το φίλτρο που δημιουργήσαμε κατάλληλο να απαλοίψει την αντήχηση που δημιουργείται σε ένα σήμα από το φίλτρο του ερωτήματος (ε).

1.2 Σχεδίαση Ζωνοπερατών Φίλτρων

α) Δημιουργούμε ένα φίλτρο με:

- Συζυγείς διπλούς πόλους στις θέσεις 0.65±0.65i
- Διπλό μηδενικό στη θέση 0.8
- Συζυγή μηδενικά στις θέσεις ±0.8i

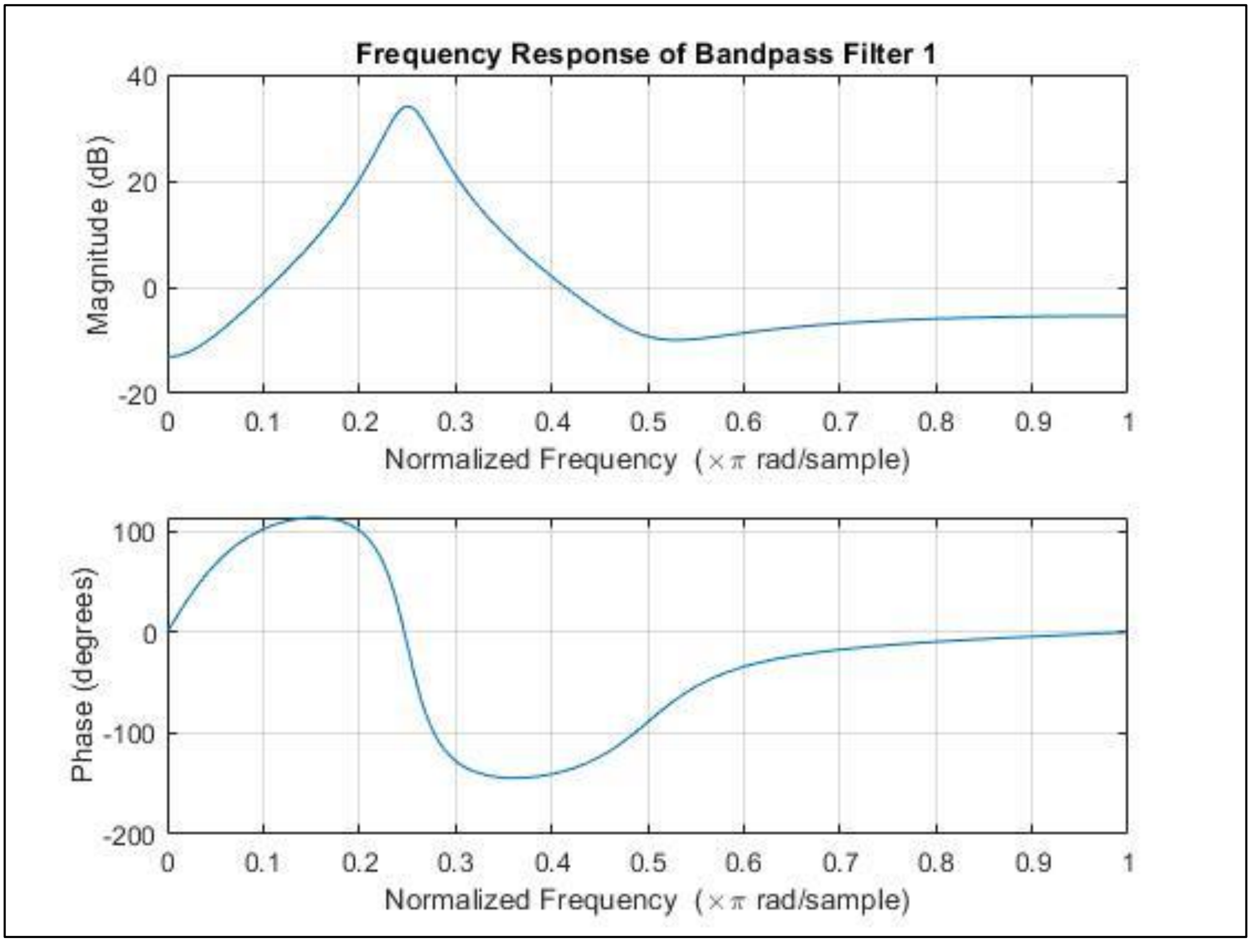


Με την εντολή zp2tf() υπολογίζουμε τους συντελεστές της εξίσωσης διαφορών:

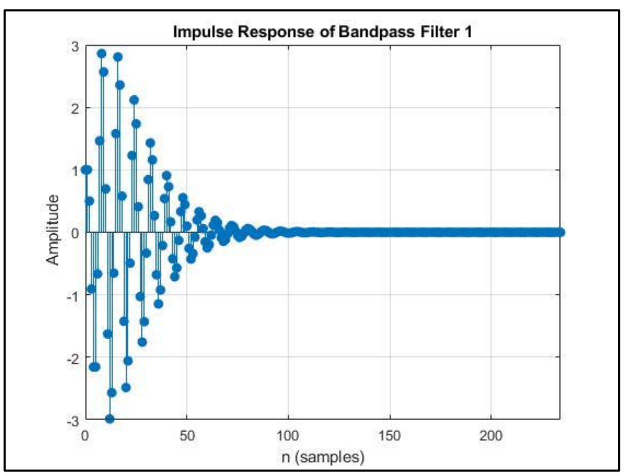
$a = [a_0\ a_1\ a_2\ a_3\ a_4]$ = [1 -2.6 3.38 -2.197 0.714]

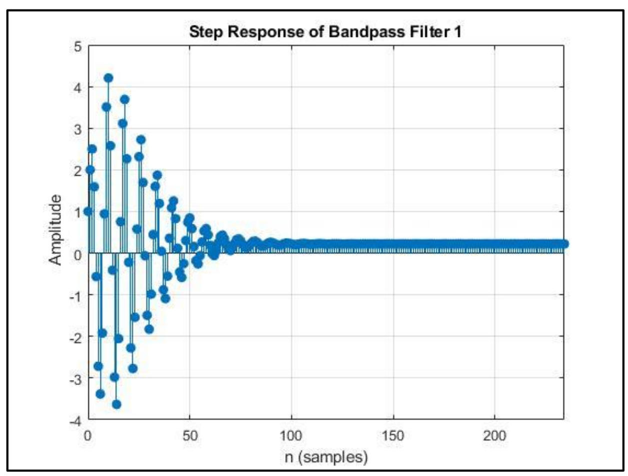
$b = [b_0\ b_1\ b_2\ b_3\ b_4]$ = [1 -1.6 1.28 -1.024 0.4096]

β) Με την εντολή freqz() σχεδιάζουμε την απόκριση συχνότητας του φίλτρου. Όπως φαίνεται, η απόκριση πλάτους είναι μεγαλύτερη από 20db, για μια μικρή ζώνη (0.2π-0.3π), ενώ αλλού είναι μικρότερη, και ιδιαίτερα όσο μεγαλώνει η συχνότητα, η απόκριση φαίνεται να τείνει ασυμπτωτικά σε ένα μικρό αριθμό (μπορεί και 0). Το γράφημα φαίνεται παρακάτω:

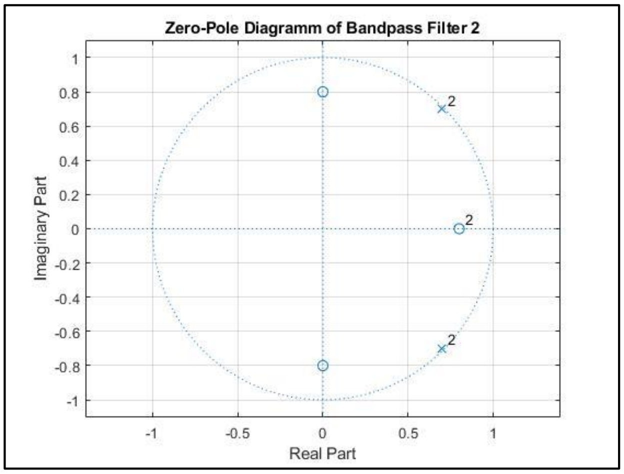


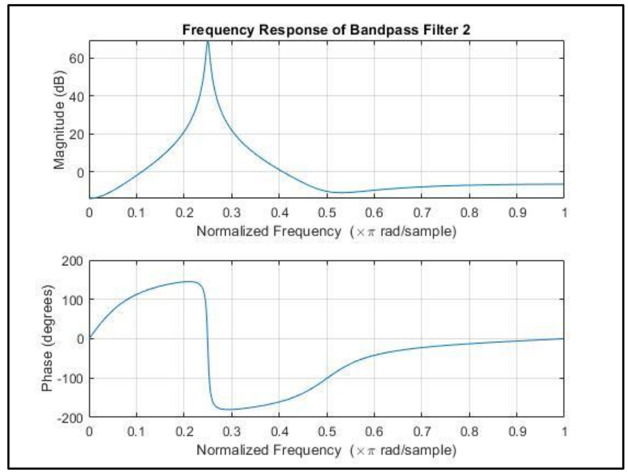
γ) Με τις εντολές impz() και stepz() υπολογίζουμε την κρουστική και την βηματική απόκριση του φίλτρου:

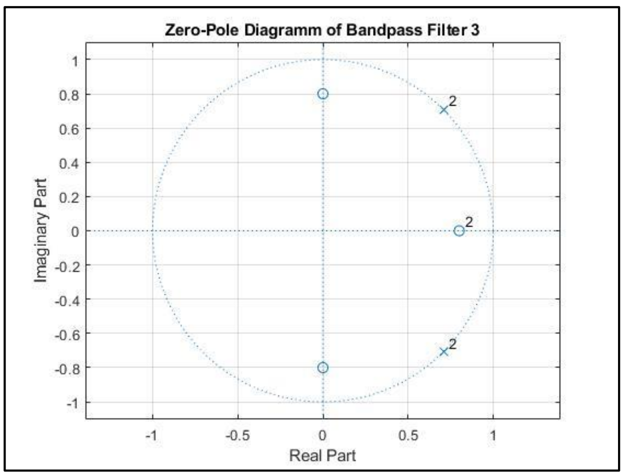


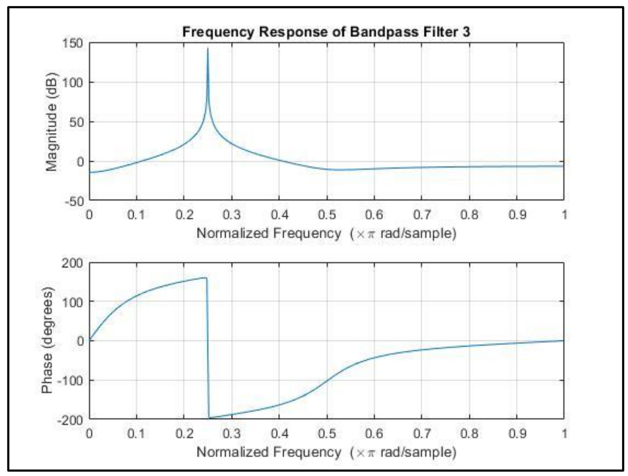


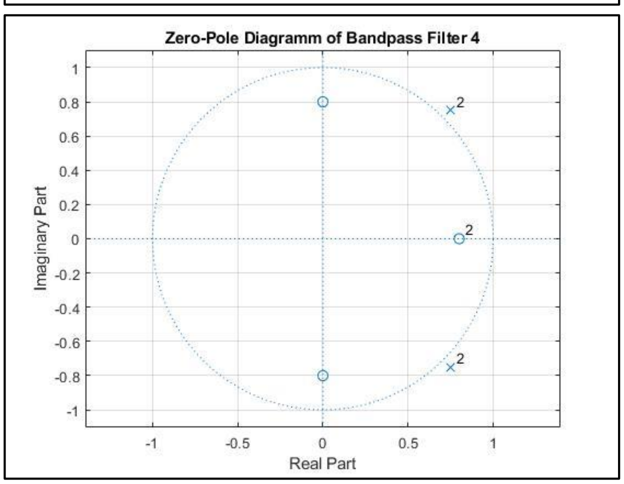
δ) Στη συνέχεια, διατηρώντας τα μηδενικά ως έχουν, μετακινούμε τους πόλους αρχικά στις θέσεις {0.7±0.7i},{0.707±0.707i}, και τέλος στις {0.75±0.75i} (πολλαπλότητα 2). Για κάθε φίλτρο που προκύπτει σχεδιάζουμε το διάγραμμα μηδενικών και πόλων, την απόκριση συχνότητας, και την κουστική απόκριση:

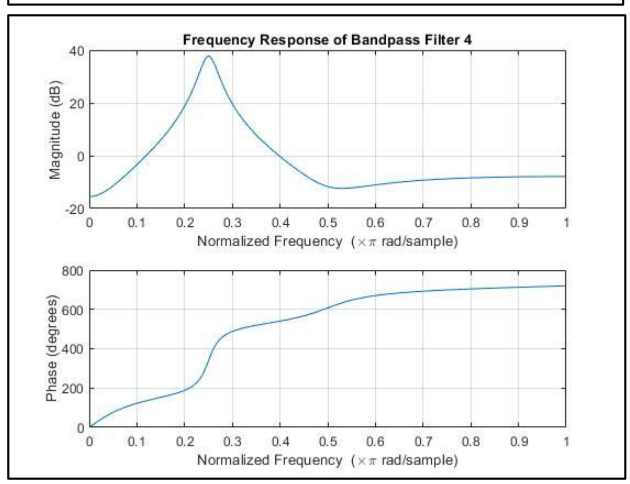




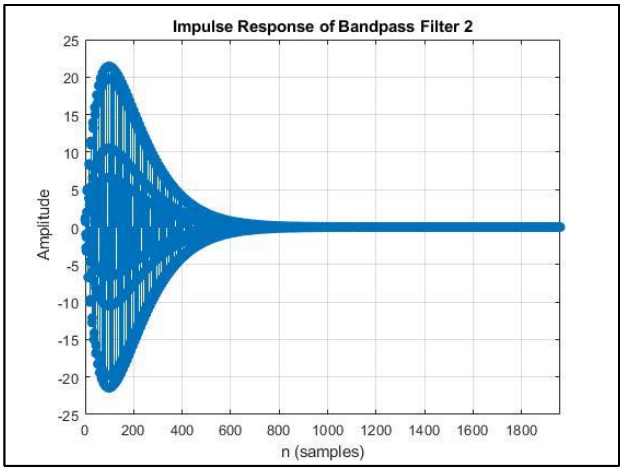


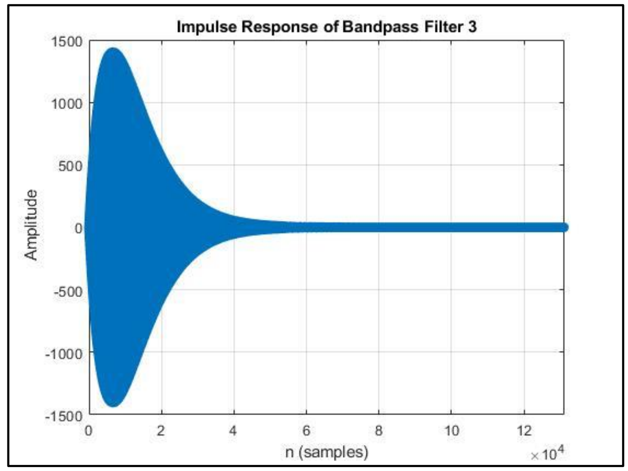




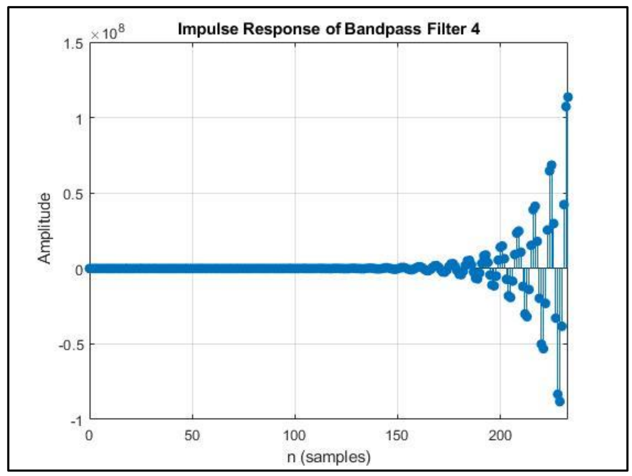


Συμπεραίνουμε ότι όσο οι πόλοι κοντεύουν στον μοναδιαίο κύκλο (και είναι εντός του κύκλου), τόσο πιο απότομη είναι η απόκριση πλάτους γύρω από τη ζώνη όπου μεγαλώνει, ενώ όταν οι πόλοι βγουν έξω από τον μοναδιαίο κύκλο, η απόκριση πλάτους αρχίζει να γίνεται πιο ομαλή. Επίσης, η κεντρική συχνότητα του φίλτρου (κανονικοποιημένη), ισούται με τη φάση των πόλων (σε rad) διά π (πχ πιο πάνω η φάση είναι γύρω στα π/4, και η κεντρική συχνότητα είναι 1/4=0.25 περίπου).

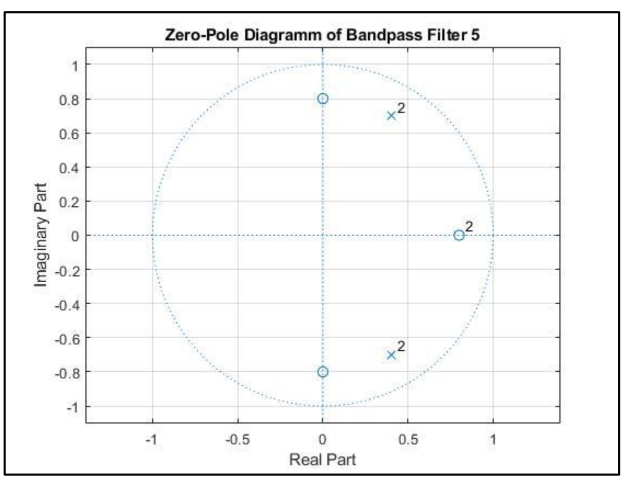


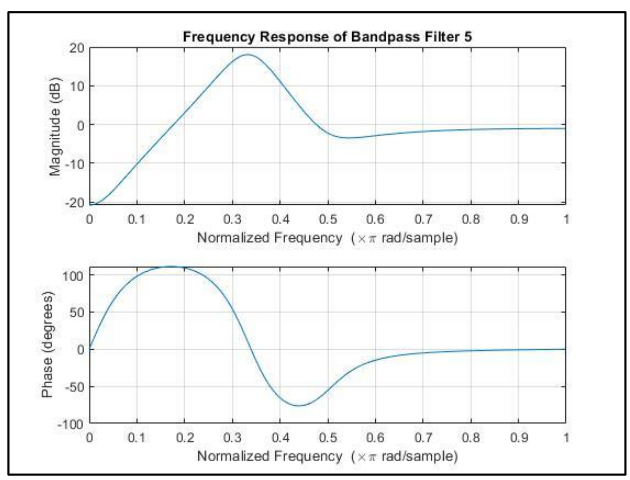


Παρατηρούμε ότι στην τρίτη περίπτωση η κρουστική απόκριση φαίνεται να είναι ασταθής, διότι αντί να μειωθεί σιγά σιγά, αυξάνεται με τον χρόνο. Αυτο εξηγείται από το διάγραμμα μηδενικών και πόλων, το οποίο είναι το μόνο όπου οι πόλοι είναι εκτός του μοναδιαίου κύκλου.



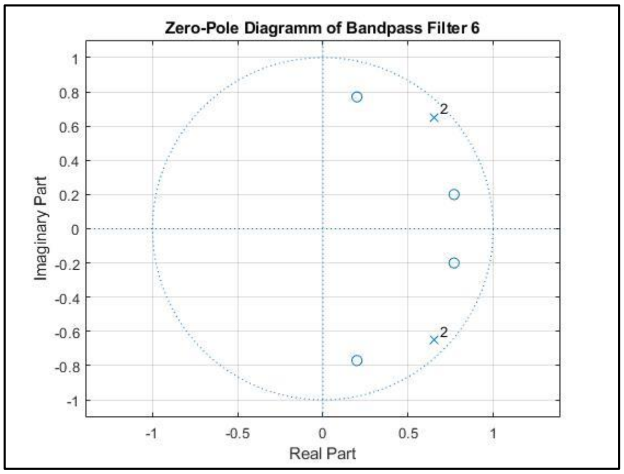
ε) Μετακινούμε πάλι του πόλους στις θέσεις {0.4±0.7i}, και σχεδιάζουμε το διάγραμμα πόλων μηδενικών και την απόκριση συχνότητας:

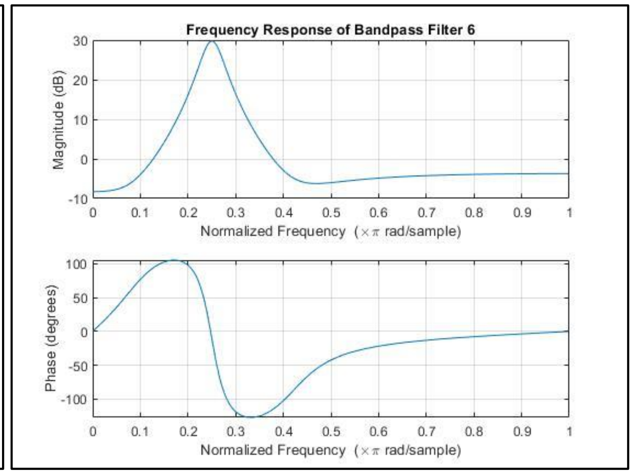


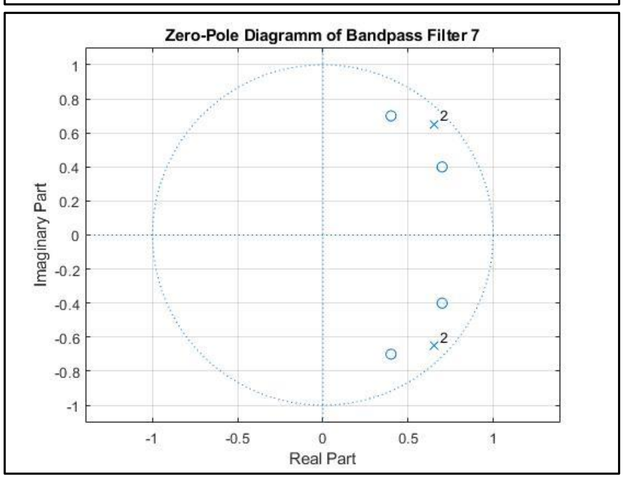


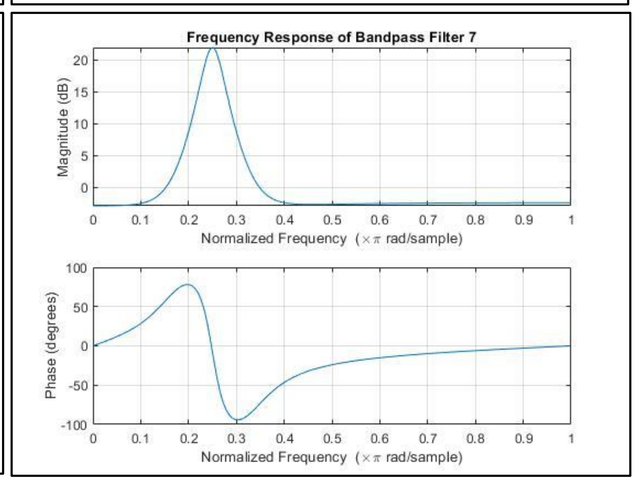
Παρατηρούμε ότι η απόκριση πλάτους αυτού του φίλτρου είναι μεγαλύτερη σε μέτρο στις αρνητικές τιμές (στις χαμηλές συχνότητες), και ότι δεν είναι πολύ απότομη.

στ) Βάζουμε τους πόλους στις αρχικές τους θέσεις, και μετακινούμε τα μηδενικά στις θέσεις {0.77±0.2i,0.2±0.77i} και {0.4±0.7i,0.7±0.4i}. Για κάθε περίπτωση σχεδιάζουμε το διάγραμμα πόλων μηδενικών, και την απόκριση συχνότητας:





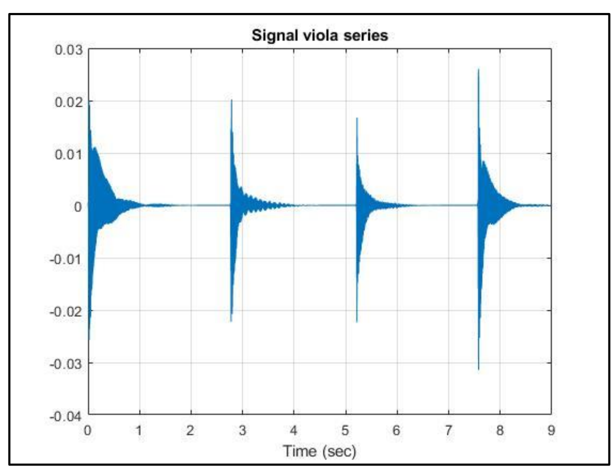




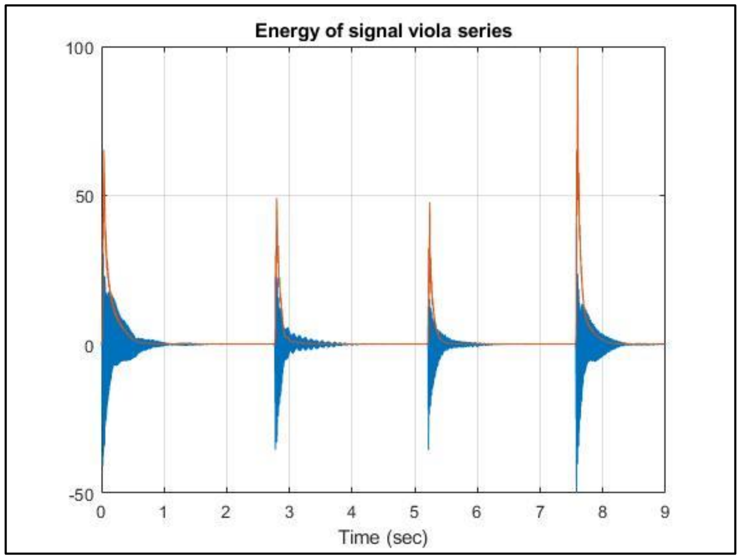
Παρατηρούμε ότι όσο πλησιάζουν τα μηδενικά στους πόλους, τόσο η απόκριση πλάτους περιορίζεται στις θετικές τιμές, και άρα η ζώνη διέλευσης παίρνει μόνο θετικές τιμές.

2.1 Ανάλυση Μουσικών Σημάτων

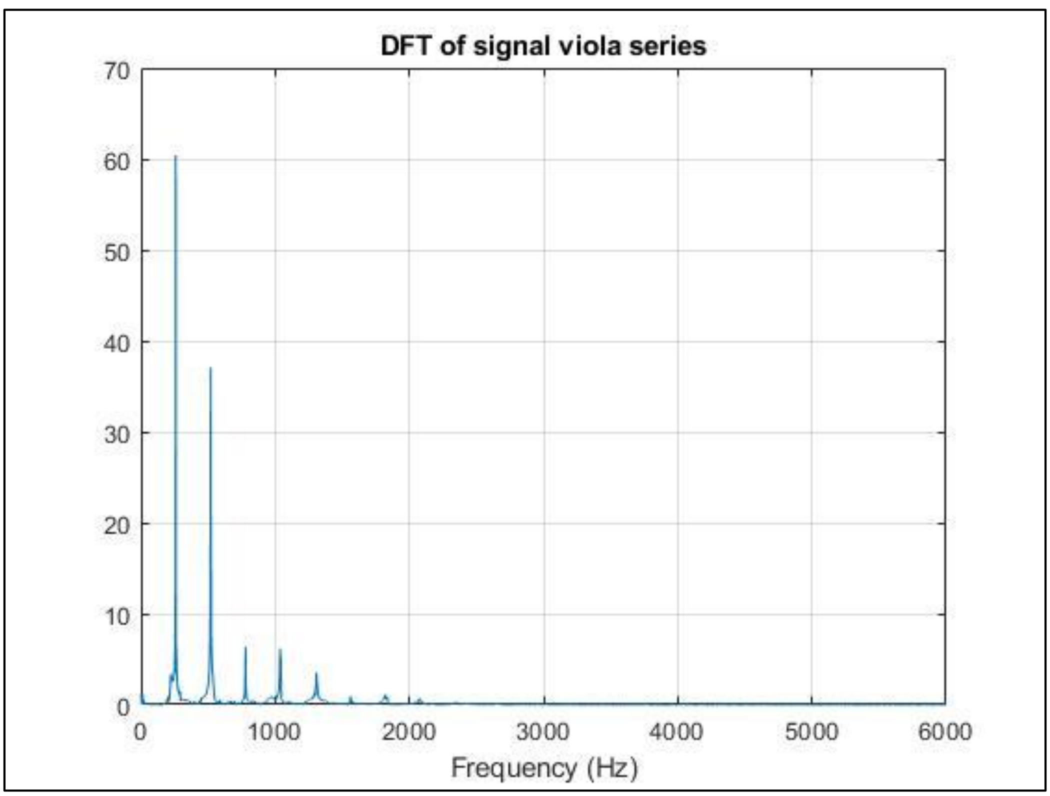
α) Με την εντολή audioread() φοτώνουμε το αρχείο viola_series.wav με συχνότητα δειγματολειψίας 44.1kHz, και στη συνέχεια το σχεδιάζουμε με την εντολή plot(), και το ακούμε με την εντολή sound(). Η γραφική που προκύπτει φαίνεται δίπλα:



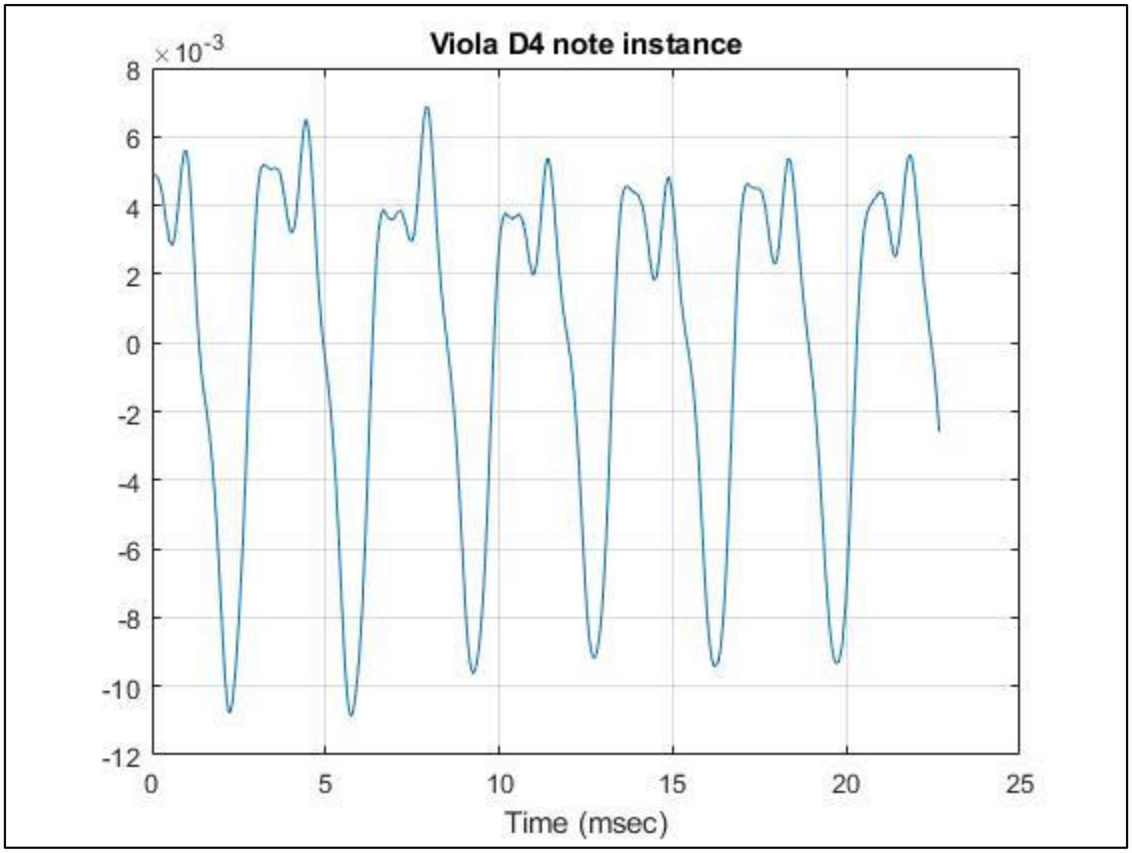
β) Στη συνέχεια κανονικοποιούμε το σήμα (βρίσκοντας τη μέγιστη απόλυτη τιμή και διαιρώντας το με αυτή), και υπολογίζουμε την ενέργεια του στο κυλιόμενο παράθυρο $w[n] = 0.54 + 0.46\cos(\frac{2\pi n}{N})$, χρησιμοποιώντας τον τύπο, ο οποίος εκφράζεται και ως συνέλιξη, $E[n] = \sum_{m=0}^{M} x^2[m]w[n-m] = x^2[n] * w[n]$. Έπειτα, σχεδιάζουμε την ενέργεια του σήματος στο ίδιο γράφημα με το σήμα, και παρατηρούμε ότι μεγαλύτερη ενέργεια εμφανίζεται στην αρχή της κάθε νότας. Το γράφημα αυτό φαίνεται δίπλα. Σημειώνεται ότι το σήμα εμφανίζεται 50 φορές μεγαλύτερο από το κανονικοποιημένο.



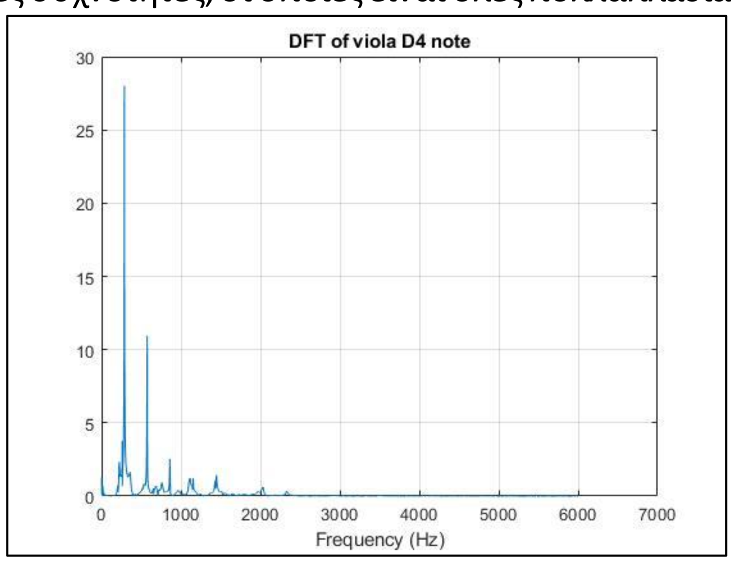
γ) Με την εντολή fft(), υπολογίζουμε τον Διακριτό Μετασχηματισμό Fourier (DFT) του σήματος (δηλαδή το φάσμα του), και απεικονίζουμε το μέτρο του.



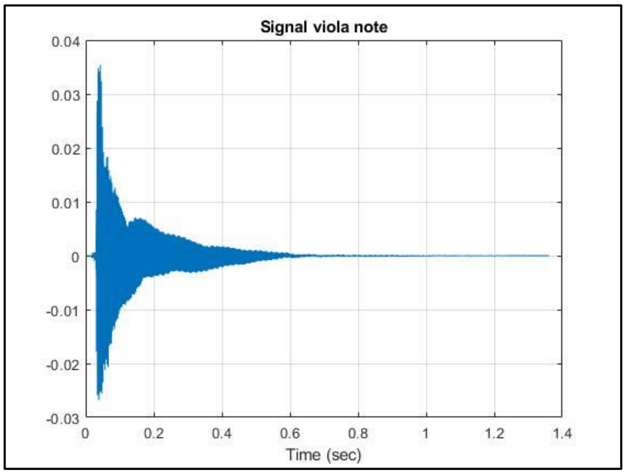
δ) Απομονώνουμε μέρος από το σήμα μας, σε μία νότα, και το απεικονίζουμε. Παρατηρούμε ότι είναι περιοδικό με περίοδο N=155 δείγματα ή T=3.51474ms.

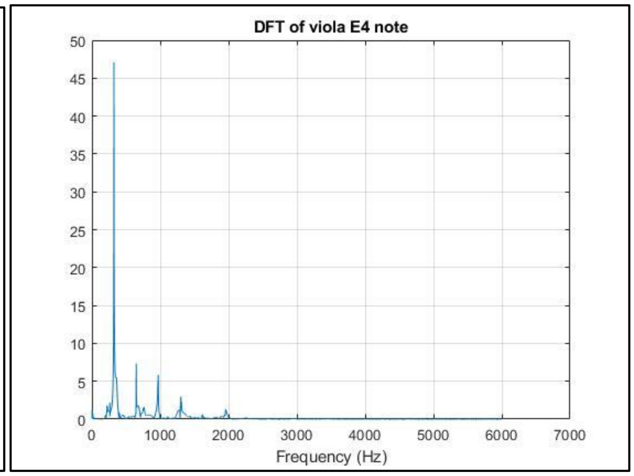


ε) Ακολούθως βρίσκουμε τον DFT της νότας αυτής, και βρίσκουμε ότι η θεμελιώδης συχνότητα της είναι f=288Hz, γεγονός που επαληθεύει τη σχέση περιόδου θεμελιώδους συχνότητας. Επίσης, παρατηρούμε ότι το φάσμα αποτελείται βασικά από ορισμένες συχνότητες, οι οποίες είναι όλες πολλαπλάσια της θεμελιώδους. Αυτές ονομάζονται αρμονικές. Σε σχέση με το φάσμα του αρχικού σήματος, παρατηρούμε ότι το μέτρο αυτών των συχνοτήτων είναι μικρότερο, γεγονός που μπορεί να οφείλεται στο ότι στο αρχικό σήμα υπάρχουν 4 νότες, με κοντινές θεμελιώδη συχνότητες.

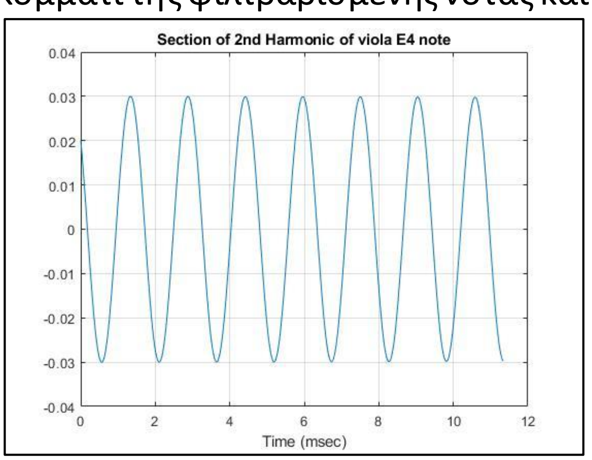


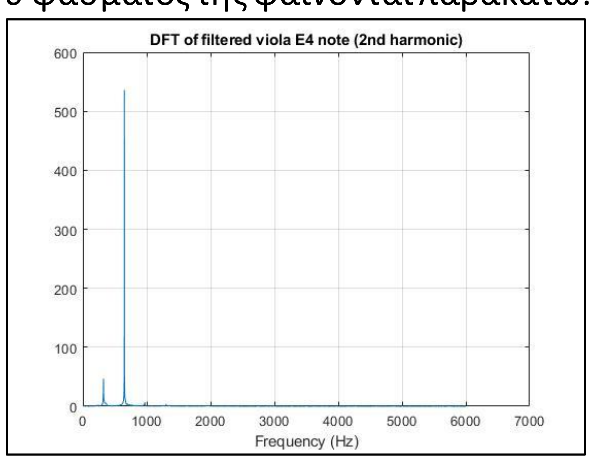
στ) Φορτώνουμε το αρχείο viola_note.wav. Αυτό φαίνεται πιο κάτω. Στη συνέχεια θα σχεδιάσουμε ζωνοπερατά φίλτρα (συμβουλευόμενοι το πρώτο μέρος της άσκησης) για να απομονώσουμε την 2η και την 4η αρμονική του (ξεχωριστά την κάθε μία). Αρχικά βρίσκουμε τον DFT της νότας, για να βρούμε τη συχνότητα της κάθε αρμονικής. Η διαδικασία για κάθε αρμονική αναλύεται παρακάτω.





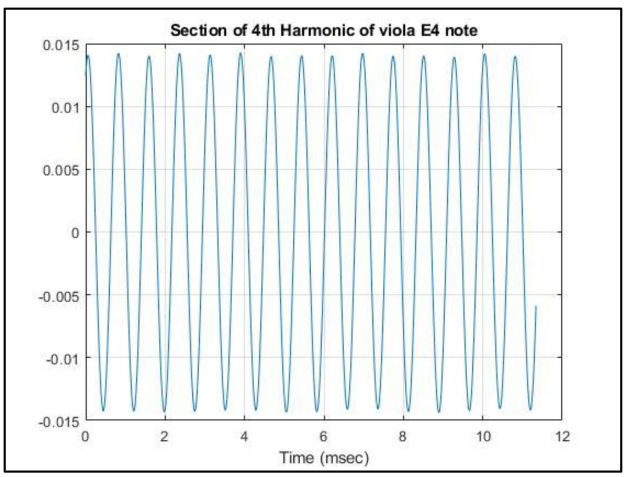
<u>2η Αρμονική:</u> Η συχνότητα της αρμονικής, όπως υπολογίστηκε από το γράφημα του DFT είναι f2=649Hz. Άρα η κανονικοποιημένη συχνότητα της αρμονικής θα είναι $f = \frac{f_2}{f_{max}} = \frac{f_2}{f_s/2} = \frac{649}{22050} \approx 0.02943$ (fmax προκύπτει από θεώρημα δειγματολειψίας, σχέση με Nyquist). Άρα η φάση φ των διπλών πόλων που θα έχει το φίλτρο πρέπει να είναι περίπου ±5.29796, η οποία έχει cos(φ)=0.995728 και sin(φ)=0.092335. Επιλέγουμε να τοποθετήσουμε τους πόλους όσο πιο κοντά γίνεται στον μοναδιαίο κύκλο, έτσι ώστε η ζώνη διέλευσης του φίλτρου να είναι όσο πιο περιορισμένη γίνεται. Άρα τοποθετούμε διπλούς πόλους στις θέσεις {0.9957±0.0923i}. Επιπλέον, τοποθετούμε τα μηδενικά πολύ κοντά στους πόλους, έτσι ώστε το μέτρο της απόκρισης συχνότητας να περιορίζεται στις θετικές τιμές. Κομμάτι της φιλτραρισμένης νότας και του φάσματός της φαίνονται παρακάτω.

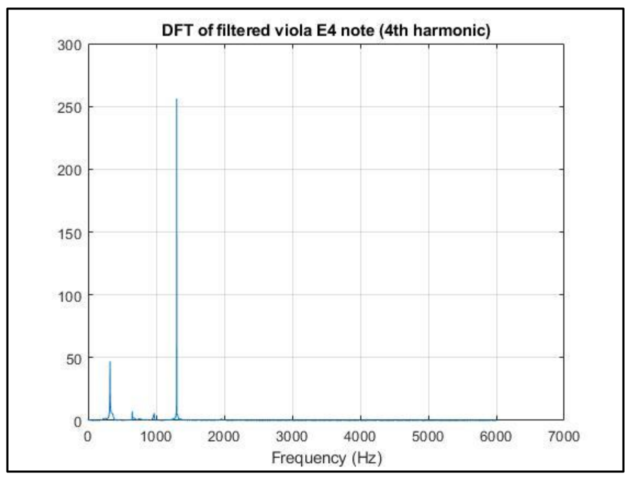




Αν υπολογίσουμε την περίοδο στο πιο πάνω σήμα, βρίσκουμε ότι T≈1.56ms, και άρα η συχνότητα είναι f≈641Hz, γεγονός που επαληθεύει ότι απομονώσαμε τη $2^η$ αρμονική, αφού 641≈649. Αν παρατηρήσουμε το φάσμα, καταλαβαίνουμε ότι το φίλτρο μας έκοψε σχεδόν όλες τις άλλες συχνότητες (εκτός από λίγο την θεμελιώδη), αλλά άφησε να περάσει, και μάλιστα ενισχυμένα, τη $2^η$ αρμονική.

<u>$4^η$ Αρμονική:</u> Η συχνότητα της αρμονικής, όπως υπολογίστηκε από το γράφημα του DFT είναι f4=1302Hz. Άρα η κανονικοποιημένη συχνότητα της αρμονικής θα είναι $f = \frac{f_4}{f_{max}} = \frac{f_4}{f_s/2} = \frac{1302}{22050} \approx 0.05905$ (fmax προκύπτει από θεώρημα δειγματοληψίας, σχέση με Nyquist). Άρα η φάση φ των διπλών πόλων που θα έχει το φίλτρο πρέπει να είναι περίπου ±10.62857, η οποία έχει cos(φ)=0.982843 και sin(φ)=0.184441. Επιλέγουμε να τοποθετήσουμε τους πόλους όσο πιο κοντά γίνεται στον μοναδιαίο κύκλο, έτσι ώστε η ζώνη διέλευσης του φίλτρου να είναι όσο πιο περιορισμένη γίνεται. Άρα τοποθετούμε διπλούς πόλους στις θέσεις {0.9828±0.1844i}. Επιπλέον, τοποθετούμε τα μηδενικά πολύ κοντά στους πόλους, έτσι ώστε το μέτρο της απόκρισης συχνότητας να περιορίζεται στις θετικές τιμές. Κομμάτι της φιλτραρισμένης νότας και του φάσματός της φαίνονται παρακάτω.

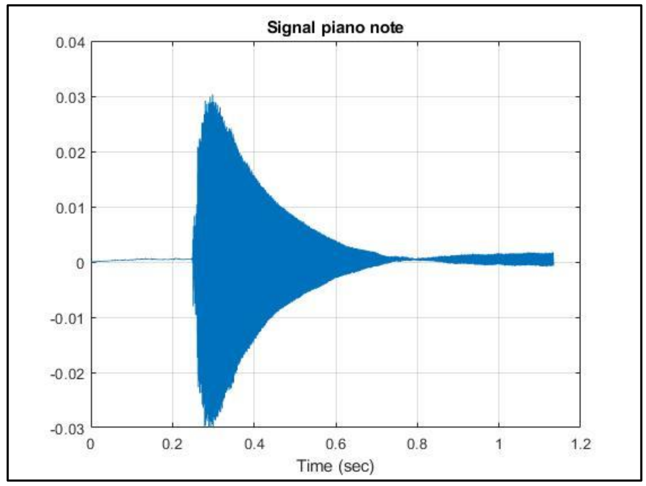




Αν υπολογίσουμε την περίοδο στο πιο πάνω σήμα, βρίσκουμε ότι T≈0.77ms, και άρα η συχνότητα είναι f≈1297Hz, γεγονός που επαληθεύει ότι απομονώσαμε την $4^η$ αρμονική, αφού 1297≈1302. Αν παρατηρήσουμε το φάσμα, καταλαβαίνουμε ότι το φίλτρο μας έκοψε σχεδόν όλες τις άλλες συχνότητες (εκτός από λίγο την θεμελιώδη), αλλά άφησε να περάσει, και μάλιστα ενισχυμένα, την $4^η$ αρμονική.

2.2 Εφαρμογή Φίλτρων για τη Δημιουργία Ηχούς και Αντήχησης Εφέ σε Μουσικά Σήματα

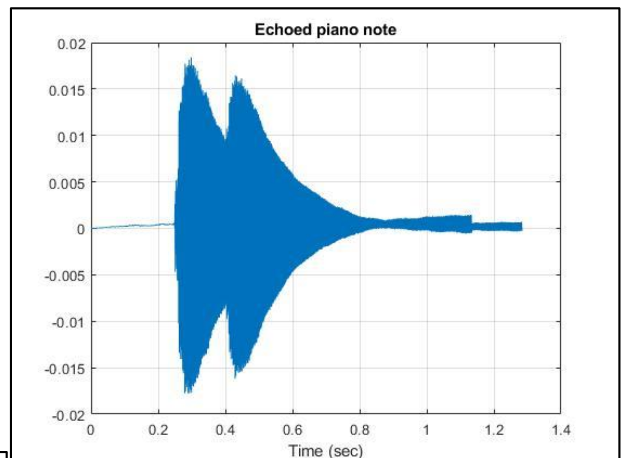
α) Αρχικά φορτώνουμε στη Matlab το αρχείο piano_note.wav, το οποίο ακούμε και απεικονίζουμε, όπως φαίνεται δίπλα. Με αυτό το σήμα θα ασχοληθούμε σε όλο το τελευταίο μέρος της εργασίας.

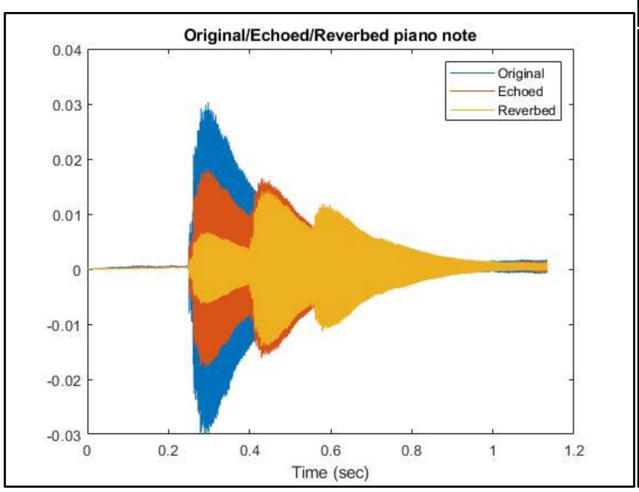


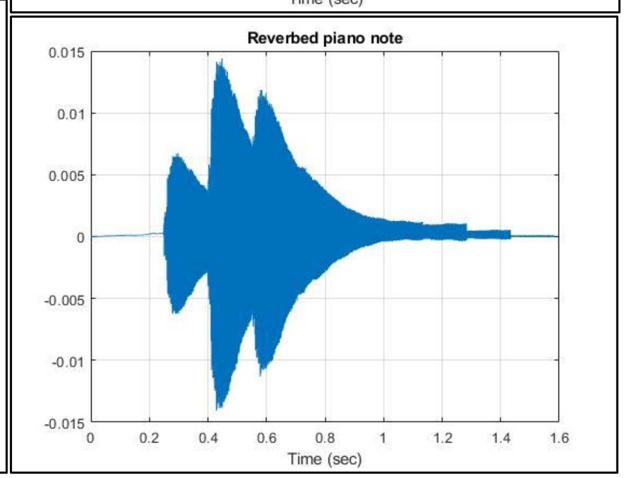
β) Ακολούθως, σχεδιάζουμε φίλτρα ηχούς και αντήχησης, με c=0.6, και P τέτοιο ώστε να αντιστοιχεί σε καθυστέρηση 0.15 δευτερόλεπτα. Αφού fs=44.1kHz και Δt=P*Ts=0.15, άρα P=0.15*fs=6615. Άρα η εξισώσεις των φίλτρων, σύμφωνα με τη θεωρία του μέρους 1.1 είναι:

- Echo: y[n]=0.6x[n]+0.4x[n-6615]
- Reverb: y[n]=0.216x[n]+0.432x[n-6615]+0.288x[n-13230]+0.064x[n-19845]

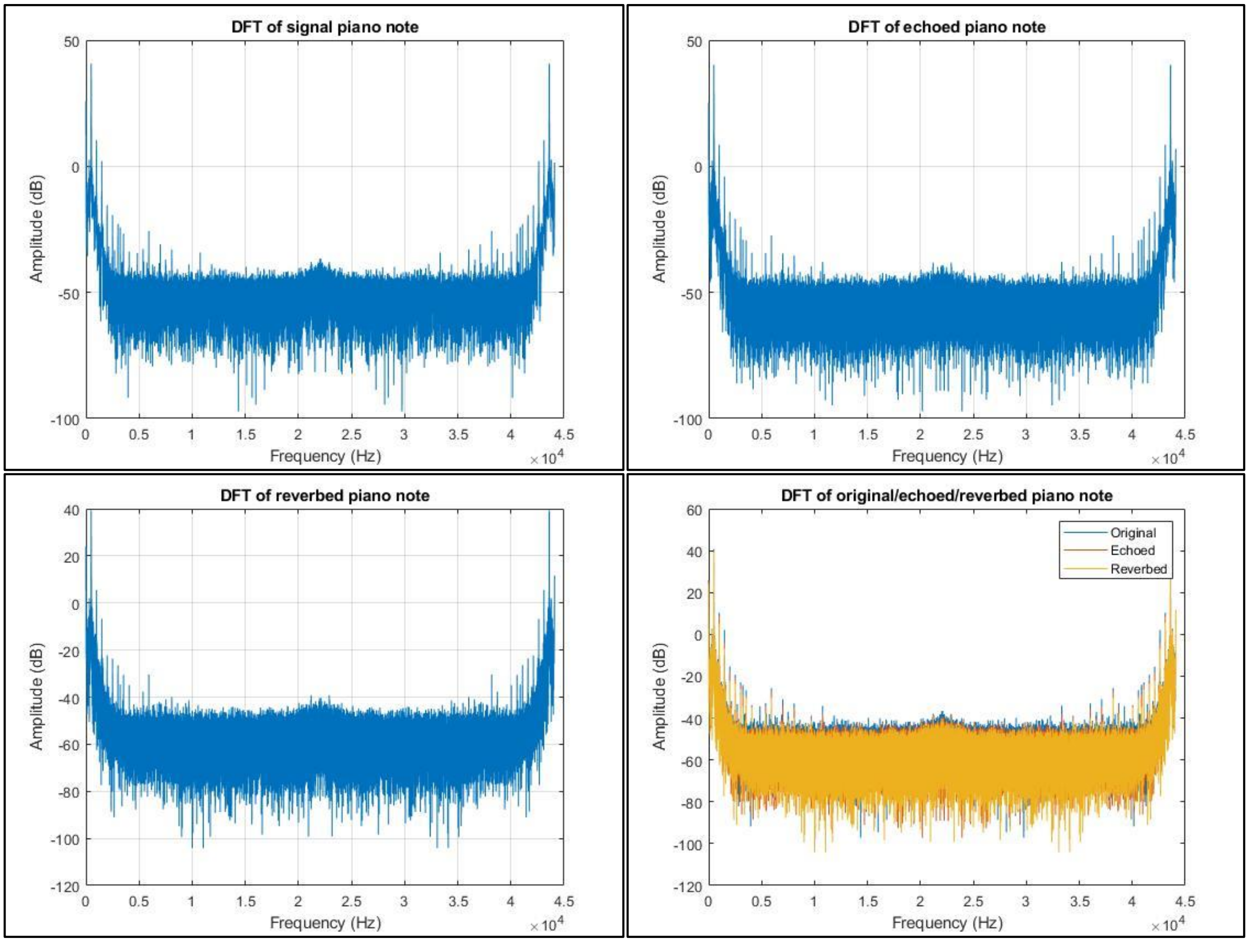
Στη συνέχεια, φιλτράρουμε το σήμα, το ακούμε, και το απεικονίζουμε για κάθε περίπτωση. Επίσης, απεικονίζουμε το αρχικό και τα φιλτραρισμένα σήματα σε ένα γράφημα. Τα γραφήματα αυτά φαίνονται δίπλα:







γ) Στη συνέχεια σχεδιάζουμε το μέτρο των φασμάτων σε λογαριθμική κλίμακα $\left(20log_{10}(|\cdot|)\right)(dB)$, αρχικά μόνα τους και στη συνέχεια όλα μαζί.



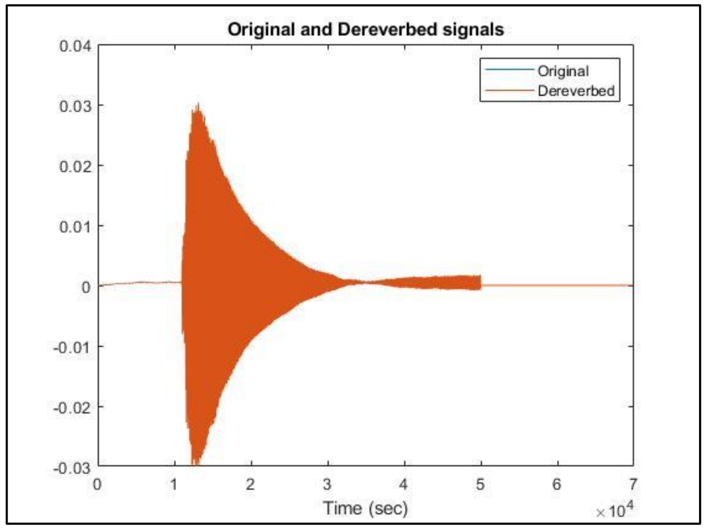
Παρατηρούμε ότι τα φάσματα δεν έχουν ιδιαίτερες διαφορές, αλλά μοιάζουν πολύ το ένα στο άλλο. Αυτό εξηγείται από το γεγονός ότι τα παραπάνω φίλτρα επιφέρουν καθυστέρυση στο σήμα, και δεν έχουν επίδραση στις συχνότητες του, αφού οι πολλαπλοί πόλοι είναι στην αρχή των αξόνων του μιγαδικού επιπέδου, ενώ τα μηδενικά είναι ομοιόμορφα κατανεμημένα γύρω από αυτούς.

δ) Μετά από πειραματισμούς, καταλήξαμε στο ότι όταν το P πέσει γύρω στο 1200, δεν γίνεται πλέον αντιληπτό το εφέ ηχούς. Αυτό μπορούμε να το ακούσουμε τρέχωντας τον κώδικα στη matlab.

ε) Ακολούθως, χρησημοποιώντας την εντολή audiowrite(), αποθηκεύουμε τα δύο φιλτραρισμένα σήματα σε δύο αρχεία τύπου .wav.

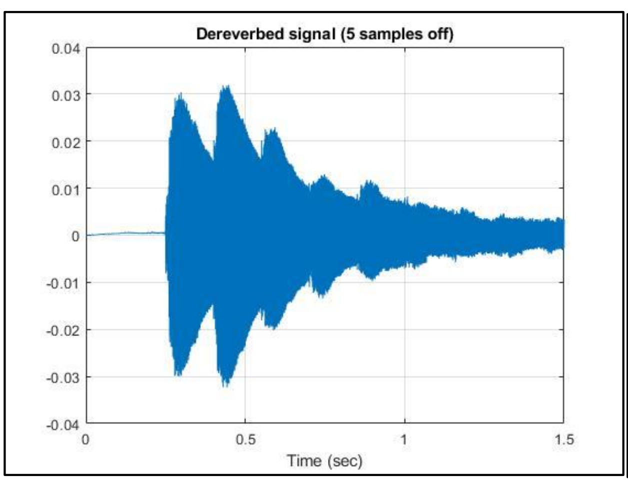
στ) Έπειτα, σχεδιάζουμε ένα φίλτρο απαλοιφής της αντήχησης που δημιουργείται από το φίλτρο του ερωτήματος (β). Σύμφωνα με τη θεωρία του μέρους 1.1, η εξίσωση που θα περιγράφει το φίλτρο είναι η εξής:

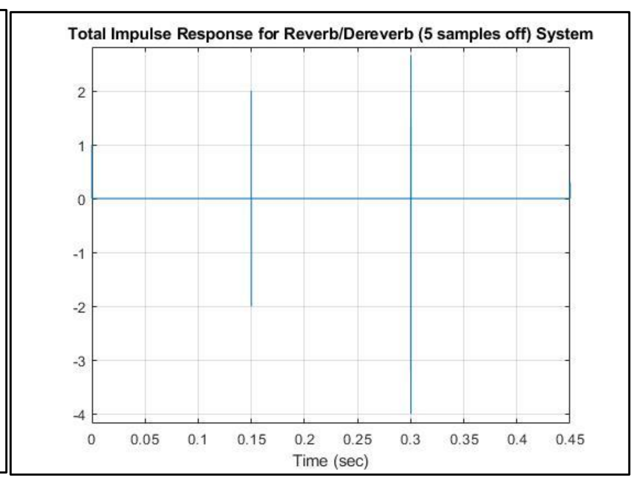
- Dereverb: 0.216y[n]+0.432y[n-6615]+0.288y[n-13230]+0.064y[n-19845]=x[n]

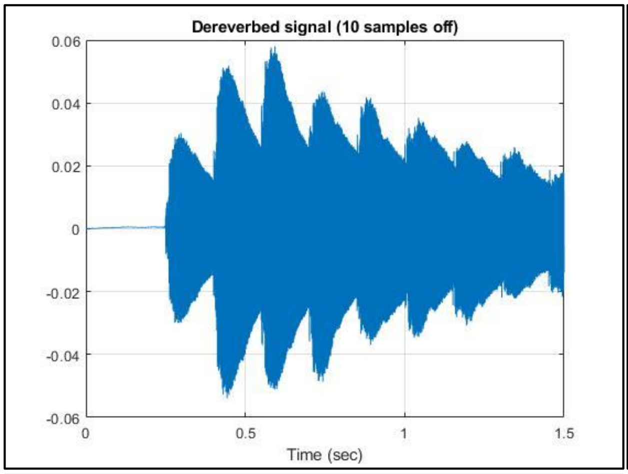


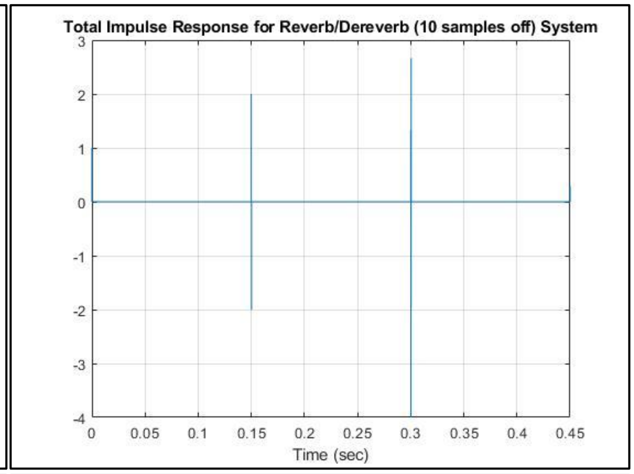
Περνούμε το reverbed σήμα από το φίλτρο απαλοιφής αντήχησης με την εντολή filter() – δεν μπορούμε να υπολογίσουμε την κρουστική απόκριση του φίλτρου και στη συνέχεια να εφαρμόσουμε συνέλιξη λόγω μνήμης – και το απεικονίζουμε στο ίδιο διάγραμμα με το αρχικό, παρατηρώντας έτσι, ότι το dereverbed και το αρχικό σήμα είναι ακριβώς τα ίδια, αφού το dereverbed σήμα πέφτει και καλύπτει ακριβώς το αρχικό.

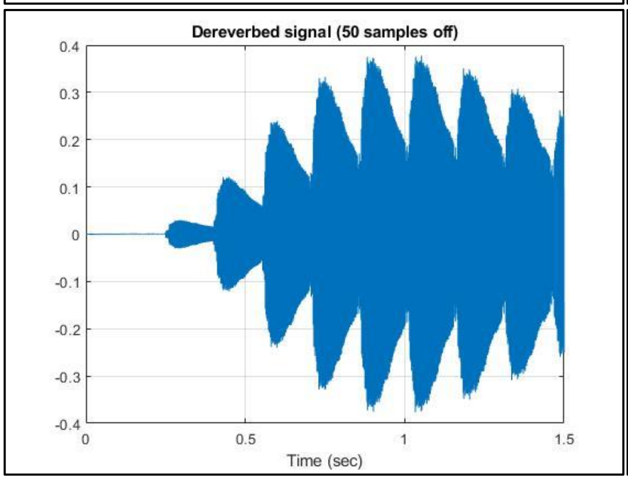
ζ) Τέλος, τροποποιούμε την παράμετρο P του φίλτρου απαλοιφής αντήχησης, έτσι ώστε να αντιστοιχεί σε καθυστέρηση 5, 10 και 50 δειγμάτων μεγαλύτερη του 6615 (δηλαδή την παράμετρο P του φίλτρου αντήχησης). Για κάθε περίπτωση, βρίσκουμε το derevebed σήμα, και την συνολική κρουστική απόκριση του συστήματος αντήχησης/απαλοιφής. Τα οποία και απεικονίζουμε παρακάτω:

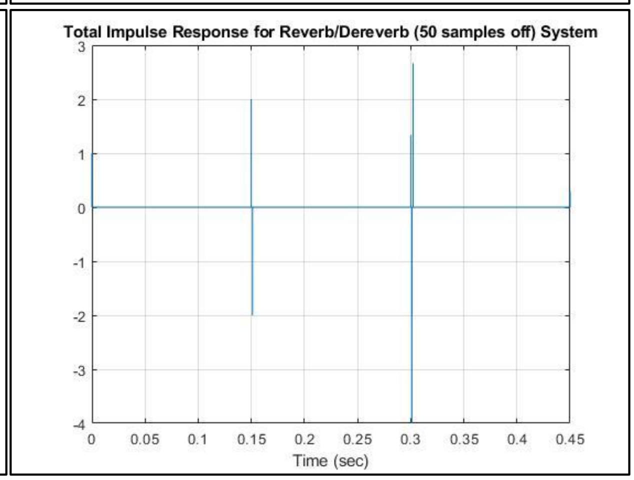












Παρατηρούμε ότι όταν το φίλτρο απαλοιφής της αντήχησης δεν είναι εντελώς συγχρονισμένο με το φίλτρο δημιουργίας της, αυτό όχι απλά δεν την απαλοίφει, αλλα την ενισχύει. Αυτό φαίνεται από το γεγονός ότι η συνολική κρουστική απόκριση, αντί να είναι ένας κρουστικός παλμός στο 0 (όπως φαίνεται πιο κάτω, όταν τα δύο φίλτρα είναι απόλυτα συγχρονισμένα), αποτελείται από περισσότερους κρουστικούς παλμούς, οι οποίοι εμφανίζονται 2 και 3 μαζί κάθε 0.15sec, και παίρνουν τόσο θετικές όσο και αρνητικές τιμές. Επίσης, αυτή την ενισχυμένη αντήχηση στα dereverbed σήματα μπορούμε να την ακούσουμε χρησιμοποιώντας την εντολή sound().