MATLAB PROJECT (ΣΗΜΑΤΑ και ΣΥΣΤΗΜΑΤΑ)

–

ΕΥΑΓΓΕΛΟΣ ΓΑΛΑΝΗΣ

el23156

(1) Σχεδίαση φίλτρων

1.1 Σχεδίαση φίλτρων ηχούς και αντήχησης (Echo, Reverb)

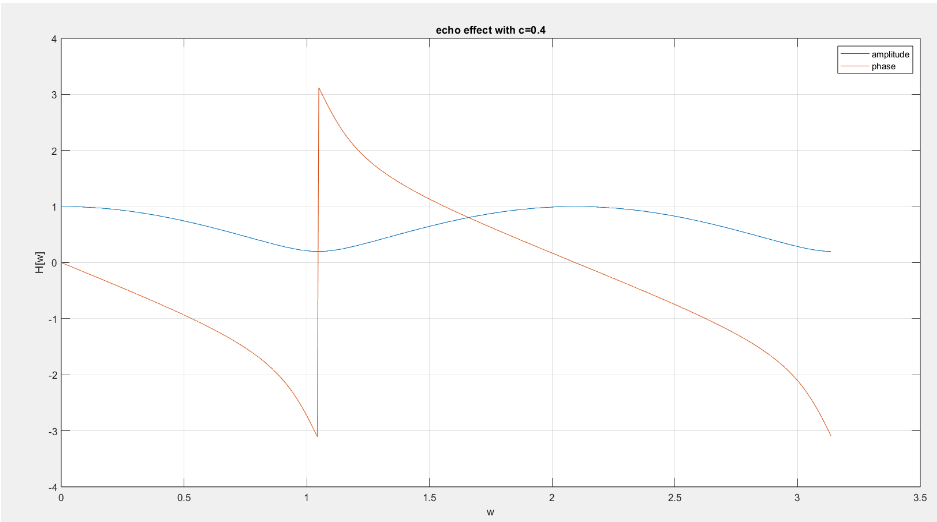
Αρχικά η εξίσωση διαφορών του φιλτρού της ηχούς είναι: $y[n] = cx[n] + (1 - c)x[n-P]$.
Οπου απο την εκφώνηση έχουμε $P = 3$ και $c=\{0.4\ ,\ 0.8\}$.
Αρα εχουμε δύο φίλτρα με αντίστοιχες εξισώσεις διαφορών:
y1[n]=0.4*x[n]+0.6*x[n-3] και y2[n]=0.8*x[n]+0.2*x[n-3].

(α) Αρα με τη χρήση διανυσμάτων του matlab ορίζουμε δύο φίλτρα για αυτές τις εξισώσεις.
Τα διανύσματα (συντελεστές αριθμητή):

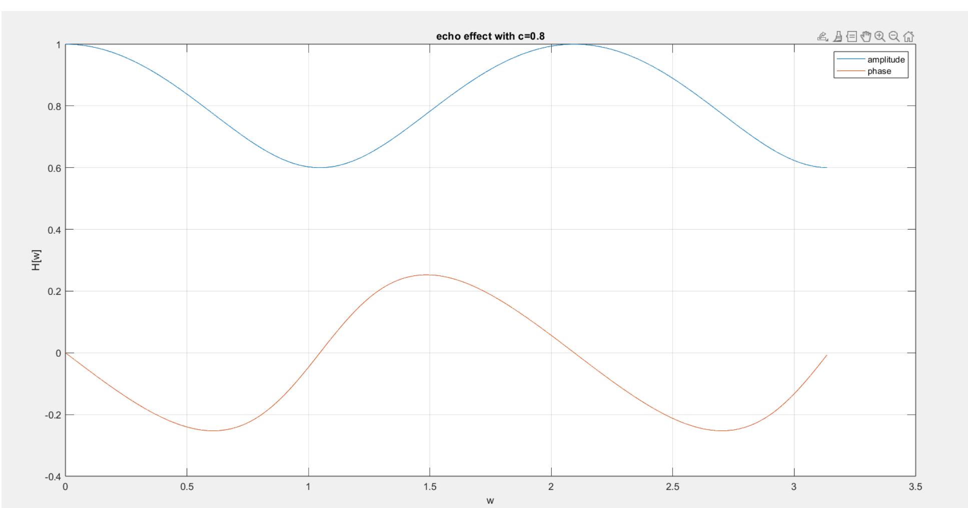
```
b1=[0.4 0 0 0.6] (c=0.4)
b2=[0.8 0 0 0.2] (c=0.8)
```

Διάνυσμα (συντελεστες παρονομαστή): `a=[1 0 0 0] (κοινό και για τα δύο φίλτρα).`

(β)Έπειτα σχεδιάζουμε την απόκριση πλάτους και φάσης των δύο συστημάτων ηχούς με χρήση της εντολής freqz(b,a):



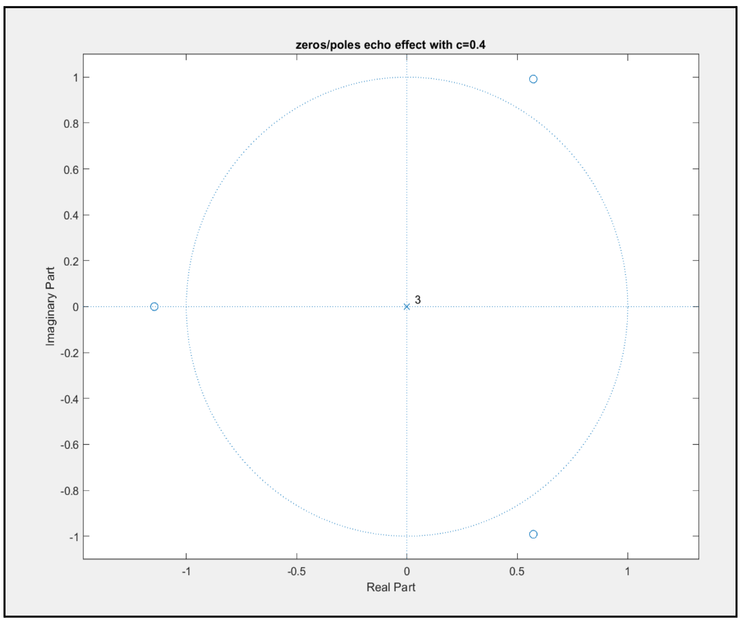
Γνωρίζοντας οτι: (1) ο DTFT είναι περιοδική συνάρτηση,(2) το πλάτος της απόκρισης συχνότητας είναι άρτιο,(3) η φάση της απόκρισης συχνότητας είναι περιττή. Μπορούμε να επεκτείνουμε τις παρατηρήσεις μας απο το φασμα στο διάστημα [0,π] (που φαίνεται στο σχήμα) για κάθε ωeR.
Αρα για το φάσμα του συστήματος με c=0.4 παρατηρούμε ότι:
Το πλάτος παίρνει μορφή “περίπου ημιτονοειδές” με μέγιστη τιμή το 1 για ω=2κπ/P
Και ελάχιστη τιμή για (2κ+1)π/P,όπου P=3,ενώ είναι πάντα θετικό (αναμενόμενο).
Η φάση μηδενίζεται οταν το πλάτος μεγιστοποιείτε (ω=2κπ/P), ενω απειρίζεται (κατακορυφη γραμμή) όταν το πλάτος ελαχιστοποιείται(ω=(2κ+1)π/P),και μπορούμε να πούμε πως έχει μορφή “περίπου εφαπτομένης”.

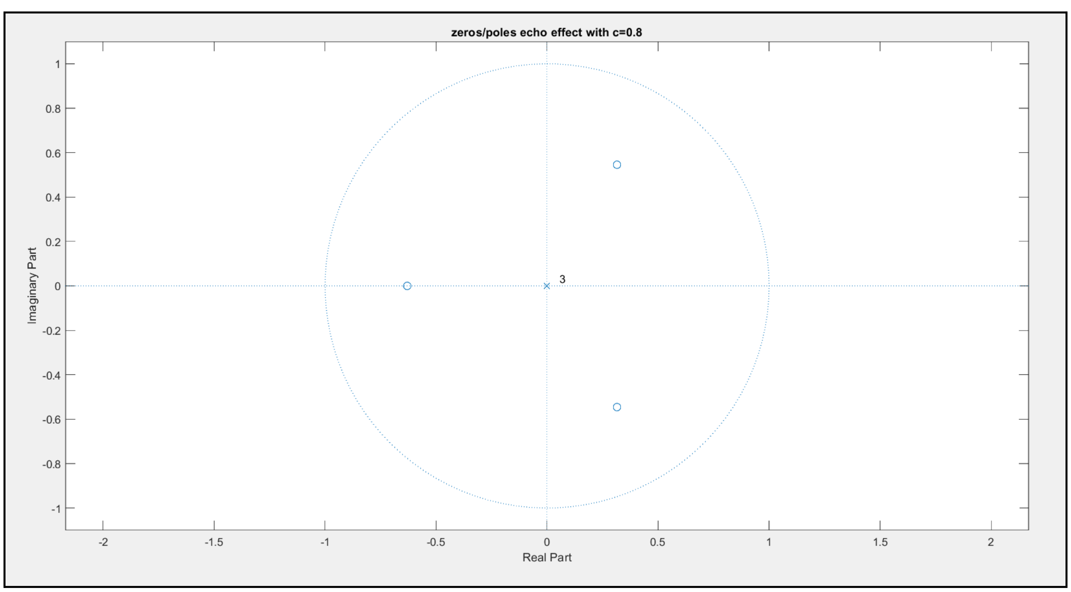


Ομοια με το c=0.4 μπορουμε να επεκτείνουμε τις παρατηρήσεις μας απο το διαστημα [0,π] σε όλο το R.

Αρα για το φάσμα του συστήματος με c=0.8 παρατηρούμε ότι:
Το πλάτος παίρνει μορφή “περίπου ημιτονοειδές” με μέγιστη τιμή το 1 για ω=2κπ/P
Και ελάχιστη τιμή για (2κ+1)π/P, ενώ είναι πάντα θετικό (αναμενόμενο).
Η φάση μηδενίζεται οταν το πλάτος μεγιστοποιείτε ή ελαχιστοποιειται (ω=κπ/P).Η φάση μεγιστοποιειται με τιμή≈0.25 κ’ ελαχιστοποιείται με τιμή≈-0.25 ,αρα η φάση μεγιστοποιείται κατα απόλυτη τιμή μεταξυ δυο διαδοχικων μηδενισμων με τιμή περίπου 0.25 ,και μπορούμε να πούμε πως έχει μορφή “περίπου ημιτονοειδες”.

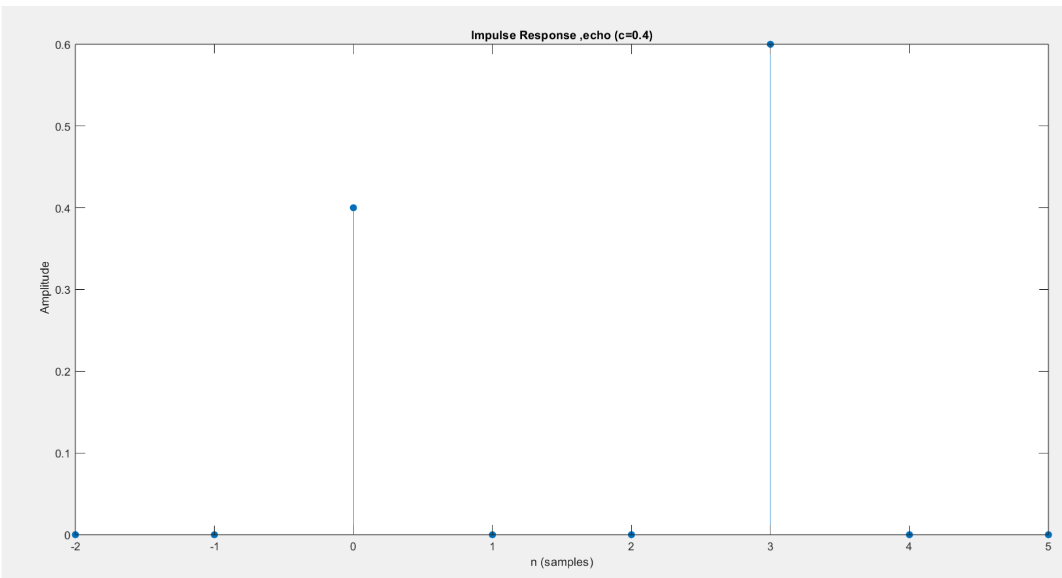
(γ) Τώρα θα σχεδιάζουμε τα διαγράμματα πόλων και μηδενικών των δύο φίλτρων, με χρήση της συνάρτησης zplane(), αφου τα μετατρέψουμε πρώτα στην απαιτούμενη μορφή με χρήση της συνάρτησης tf2zp().

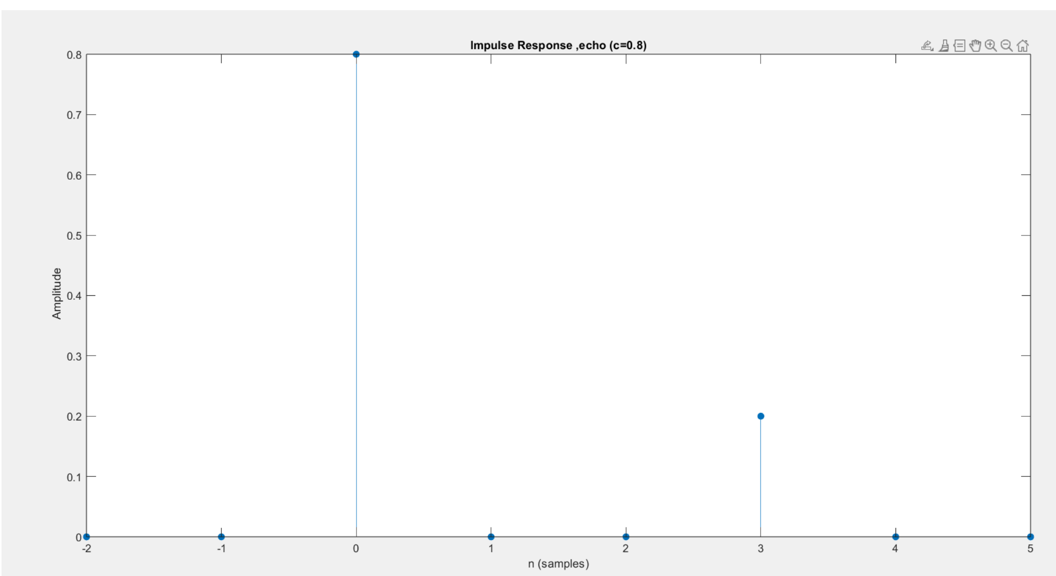




Και τα δύο φίλτρα έχουν πόλους στο σημείο (0,0) με πολλαπλότητα 3(=P).ενώ και τα δύο εμφανίζουν μηδενικά που βρίσκονται σε σημεία με γωνίες 60°,180°,300°.
Οι αποστάσεις των μηδενικών είναι η μόνη διαφορα των δύο φίλτρων ,όπου στο φίλτρο με c=0.4 τα μηδενικα εχουν απόσταση από το (0,0): d=1.14471>1 , ενώ στο φίλτρο με c=0.8 τα μηδενικα εχουν απόσταση από το (0,0): d=0.629961<1.
Επειδή τα μηδενικά για c=0.8 είναι μέσα στο μοναδιαίο κύκλο, ενώ για c=0.4 τα μηδενικά είναι εκτός του μοναδιαιου κύκλου, καταλαβαίνουμε οτι για c=0.8 το σήμα είναι ευσταθές ενώ για c=0.4 και για συγκεκριμένες συχνότητες το σήμα είναι ασταθές, το οποίο επιβεβαιώνεται και με τον απειρισμό της φάσης στην απόκριση συχνότητας.

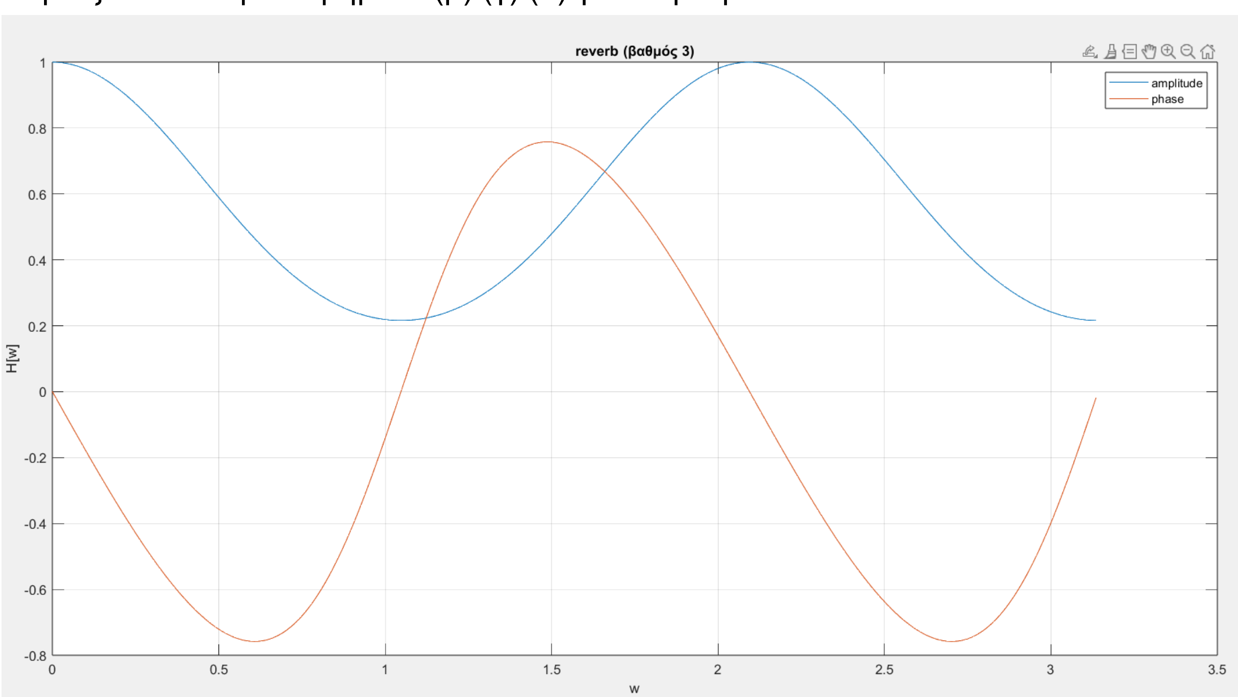
(δ)Έπειτα με χρήση της συνάρτησης impz(b,a) θα δούμε την κρουστική απόκριση των δύο φίλτρων:

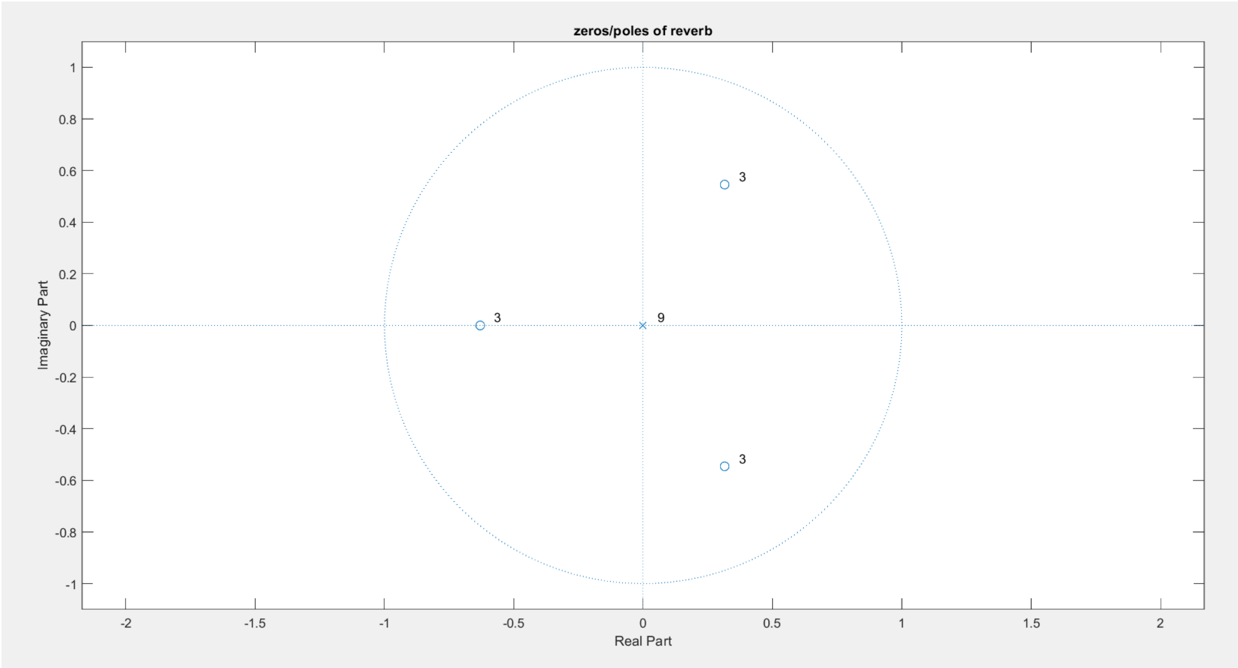


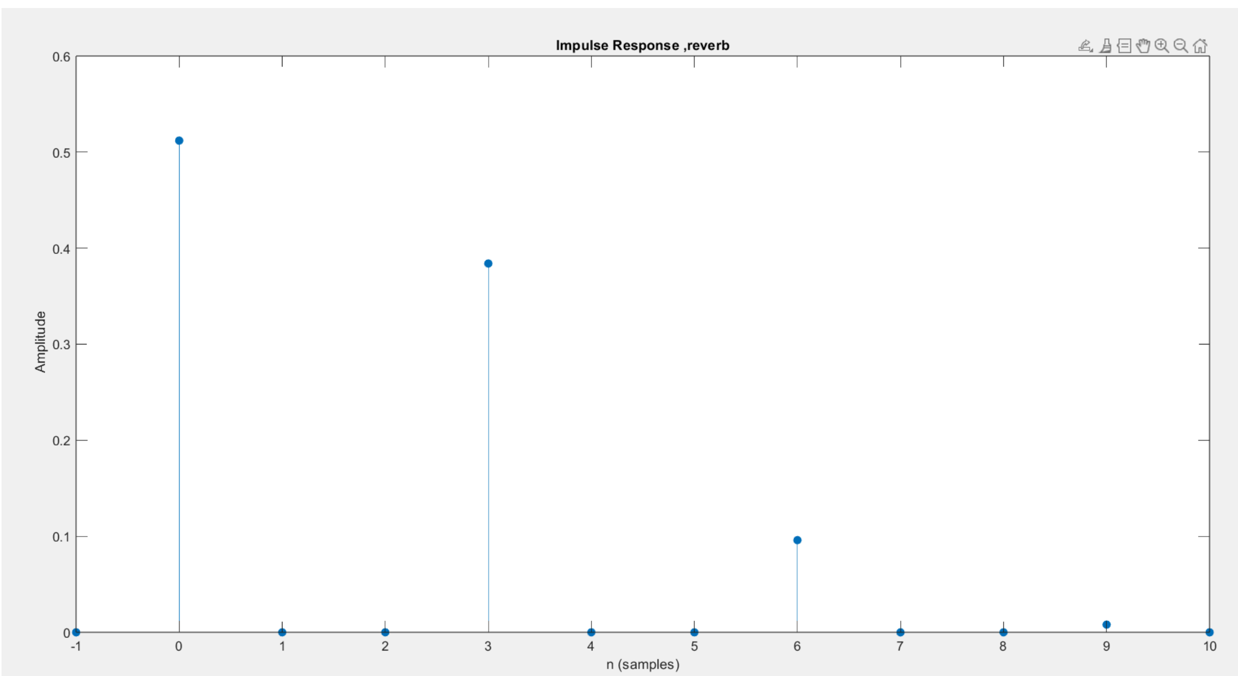


Ξέρουμε πως τα φίλτρα έχουν συναρτήσεις
y1[n]=0.4*x[n]+0.6*x[n-3] και y2[n]=0.8*x[n]+0.2*x[n-3]. Αρα για κρουστική απόκριση=δ[n],
θα έχουμε h1[n]=0.4*δ[n]+0.6*δ[n-3]={0.4 , 0 , 0 , 0.6 , 0} για c=0.4 και h2[n]=0.8*δ[n]+0.2*δ[n-3]={0.8 , 0 , 0 , 0.2} για c=0.8.
Που είναι ακριβώς αυτό που βλέπουμε (και για τα δύο φίλτρα) αφου amplitude≠0 μόνο για n=0 κ’ n=3, με τα αντίστοιχα αναμενόμενα πλάτη.

(ε)Για να βρούμε τους συντελεστες ενος φίλτρου αντήχησης 3ου βαθμού δηλαδή τρία διαδοχικά φίλτρα ήχούς με c=0.8 και P=3 ,αρκεί να κάνουμε 2 συνελιξεις στους συντελεστες του φιλτρου ηχούς με τον εαυτό του (η δευτερη με το αρχικο πάλι).Αυτο λειτουργει γιατι αν εχουμε ενα σύστημα με απόκριση h[n] στο πεδίο του χρόνου, και Η(Ω) στο πεδίο της για κάθε έξοδο ισχύει y[n]=conv(h[n],x[n]) ⇔ Y(Ω)=X(Ω)H(Ω)
Αρα για reverb (βαθμός 3):Y1(Ω)=X(Ω)H(Ω) , Y2(Ω)=Y1(Ω)H(Ω)=X(Ω)H(Ω)H(Ω)
Y3(Ω)=Y2(Ω)H(Ω)=X(Ω)H(Ω)H(Ω)H(Ω)=X(Ω)(H(Ω)^3).
H’(Ω)=H(Ω)^3 κρουστική απόκριση του reverb
Και επειδη οι συντελεστες του παρανομαστη ειναι [1 0 0 0…] το H(Ω)^3 είναι πολ/σμος “πολυωνυμων” αρα οι συντελεστες του Η’ είναι, ο παρανομαστής πάλι a’=[1 0 0 0…]
Και ο αριθμητής b’=conv(b,b) και ξανά b’=conv(b’,b) όπου b=[0.8 0 0 0.2]
Αυτό εντέλη δίνει b=[0.5120 0 0 0.3840 0 0 0.0960 0 0 0.0080]
Αρα σύστημα:y[n]= 0.5120x[n] + 0.3840x[n-3] + 0.0960x[n-6] + 0.008x[n-9].
Τώρα ξανακάνουμε τα βήματα (β) (γ) (δ) για το φίλτρο reverb:







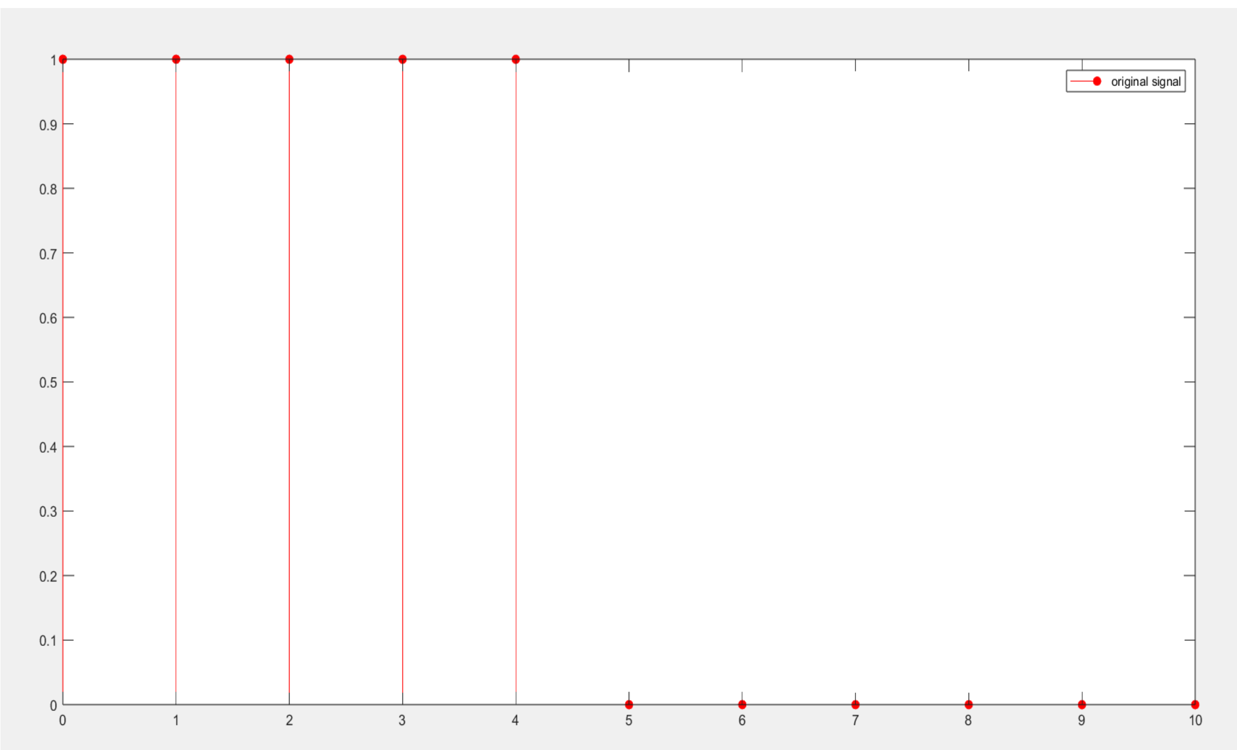
Όλες οι παρατηρησεις που κάναμε στα προηγούμενα ερωτήματα ισχύουν και εδώ.Αφού τα μηδενικά ειναι μεσα στο μοναδιαίο κύκλο (με ίδια απόσταση και γωνίες) αρα έχουμε ευστάθεια που φαίνεται και απο τις αποκρίσεις συχνότητας φάσης και πλάτους, απλα τώρα έχουμε τα τρία μηδενικά τα οποία έχουν πολλαπλότητα 3, και ο πόλος στο (0,0) έχει πολλαπλότητα 9.

Η απόκριση συχνότητας έχει τους μηδενισμούς,μεγιστοποιησεις και ελαχιστοποιησεις στα ίδια σημεία απλά η απόλυτη τιμή του πλάτους και της φάσης της απόκρισης έχει μεγαλώσει δηλαδή έχει μεγαλύτερα μέγιστα και μικρότερα ελάχιστα ,οι τιμές των οποίων φαίνεται στο γράφημα. Τελος, η κρουστική απόκριση είναι ακριβώς όπως την περιμέναμε, γιατί ομοίως με τα προηγούμενα ερωτήματα έχει, amplitude≠0 μόνο για n={0,3,6,9} με τις σωστές τιμές πλάτους.

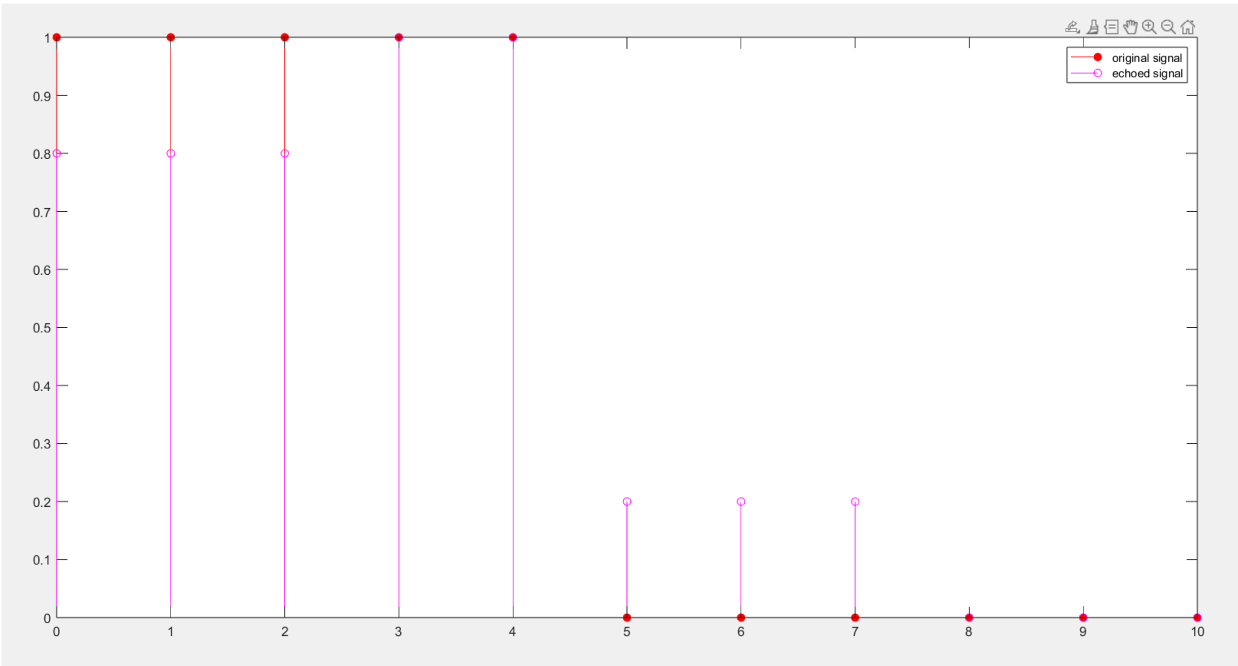
(στ) Έστω Y1(Ω)=X(Ω)H(Ω) η έξοδος του reverbed σήματος, ψάχνουμε H1(Ω) τετοιο ώστε H1(Ω)Y1(Ω)=X(Ω) (de-reverb) ⇔ H1(Ω)X(Ω)H(Ω)=X(Ω) ⇔ H1(Ω)=1/H(Ω).Άρα αν χρησιμοποιήσω την εντολή zp2tf(zeroes, poles) και στη θέση των zeroes βάλω τα poles του reverb σήματος, ενω στη θέση των poles βάλω τα zeroes του reverb σήματος (και κερδος 1/κ)του προηγούμενου ερωτήματος. Παιρνω τους συντελεστες [b,a] του αριθμητή και του παρανομαστη αντιστοιχα του φιλτρου απαλοιφης της αντηχησης, οι οποιοι προκύπτουν να είναι:
b=[1 0 0 0 0 0 0 0 0 0] (αριθμητής)
a=[0.5120 0 0 0.3840 0 0 0.0960 0 0 0.0080] (παρανομαστής)

Τώρα χρησιμοποιώντας την εντολή filter() εφαρμόζουμε τα αντίστοιχα φίλτρα για το κάθε παρακάτω ερώτημα με αρχική συνάρτηση εφαρμογής την x[n]=u[n]-u[n-5]=[1 1 1 1 1 0 0...], έχουμε:

-α-

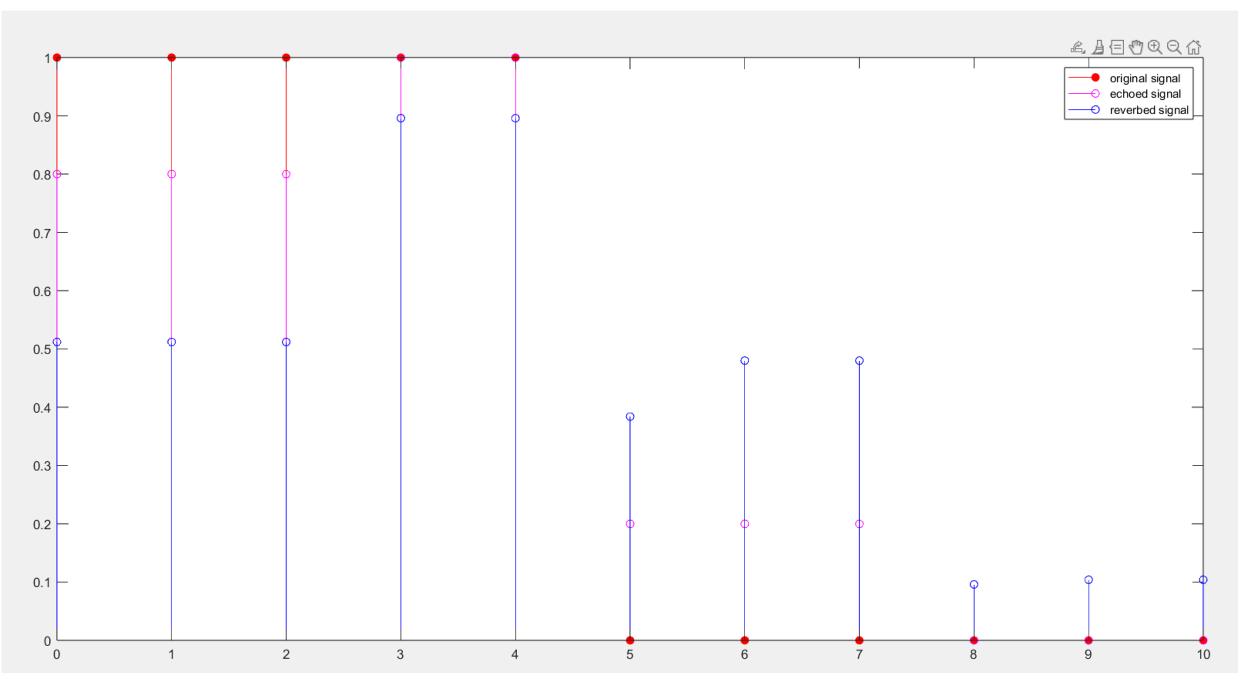


-β-



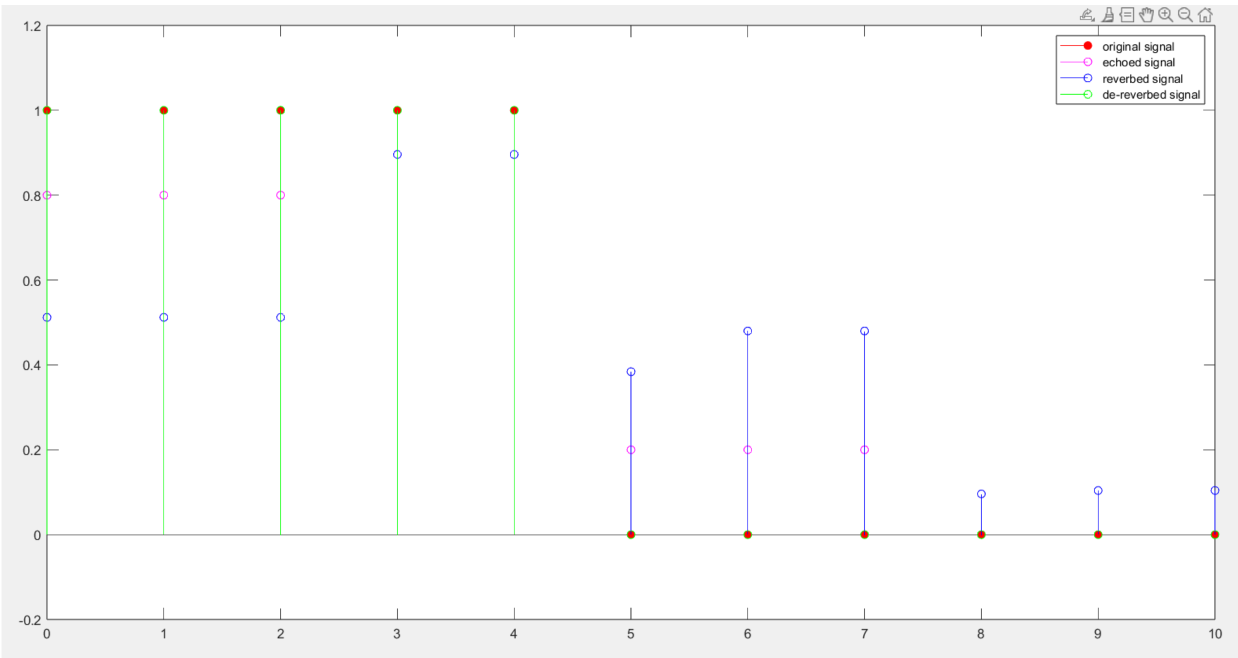
original signal
echoed signal

-γ-



original signal
echoed signal
reverbed signal

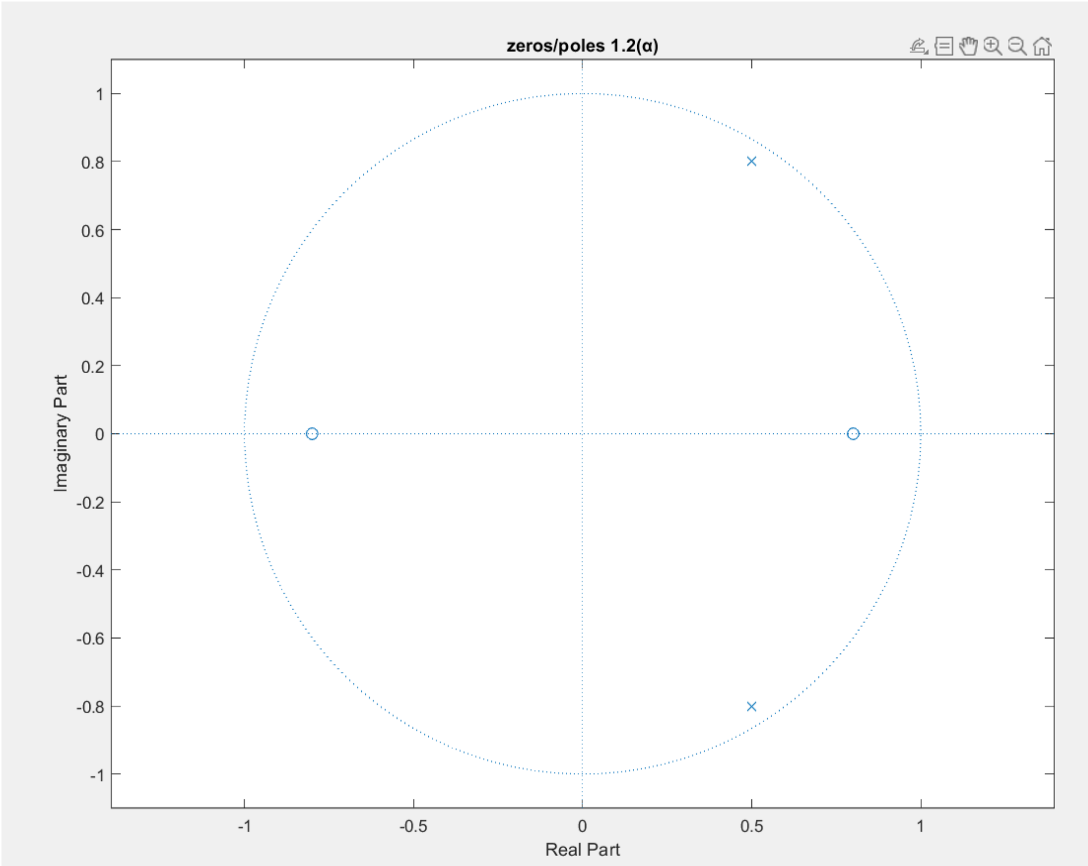
-δ-



Τα φίλτρα λειτουργούν ορθά αφου απο το διαγραμμα: original signal=dereverbed signal

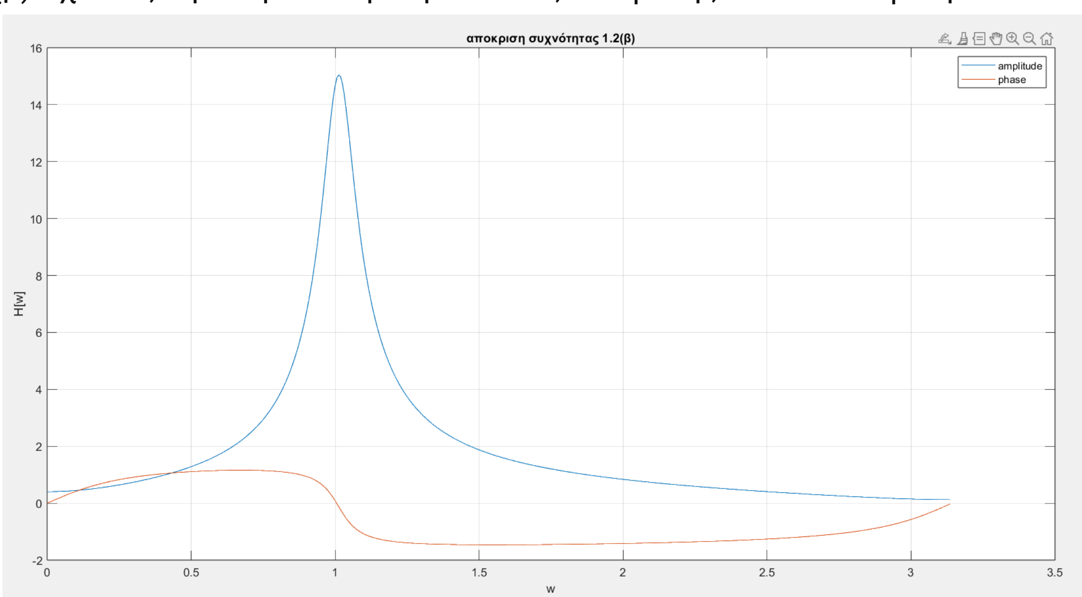
<u>1.2 Σχεδίαση Ζωνοπερατών Φίλτρων:</u>

(α) για να σχεδιάσω ένα φίλτρο με ζεύγος συζυγών μιγαδικών πόλων στις θέσεις 0.5±0.8i, και μηδενικά στις θέσεις ±0.8.
Θεωρώ διανύσματα z,p με (p,z)=({0.5±0.8i},{±0.8}) Το αντιστοιχο zeroes/poles γράφημα αυτου του φίλτρου είναι:



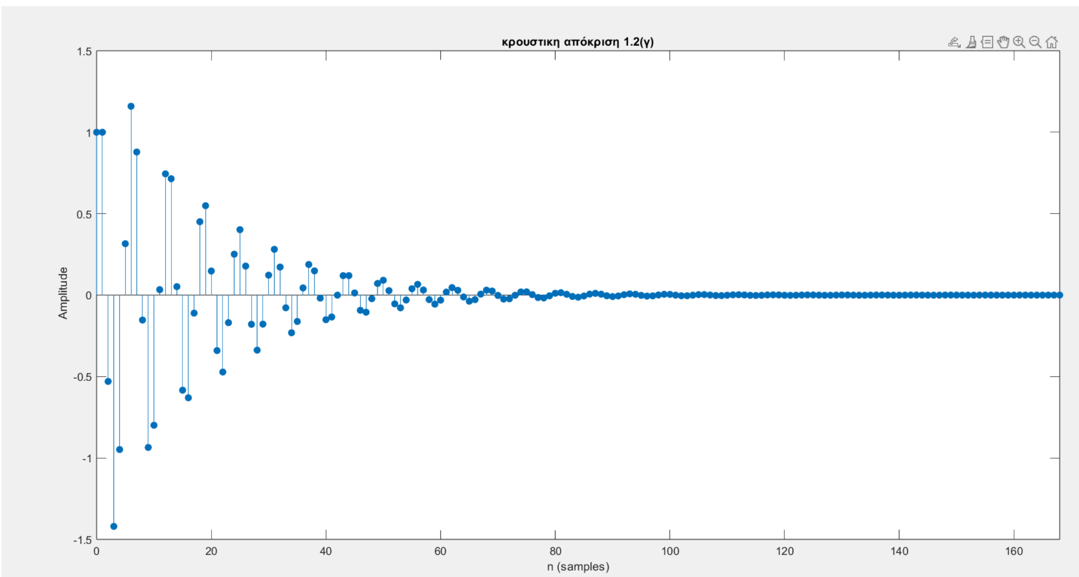
Και βρίσκω τα διανύσματα συντελεστών αυτού με χρήση [b,a]=zp2tf(z,p,1), οπου:
b=[1.0000 0 -0.6400](αριθμητής)
a=[1.0000 -1.0000 0.8900](παρανομαστής)

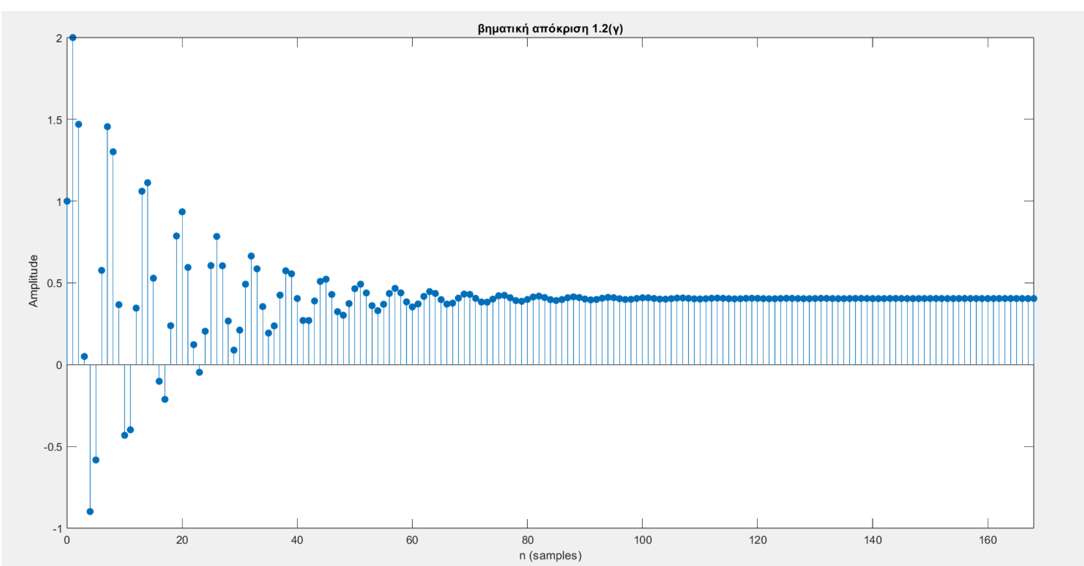
(β) Σχεδιάζουμε την απόκριση πλάτους και φάσης αυτού του φίλτρου:



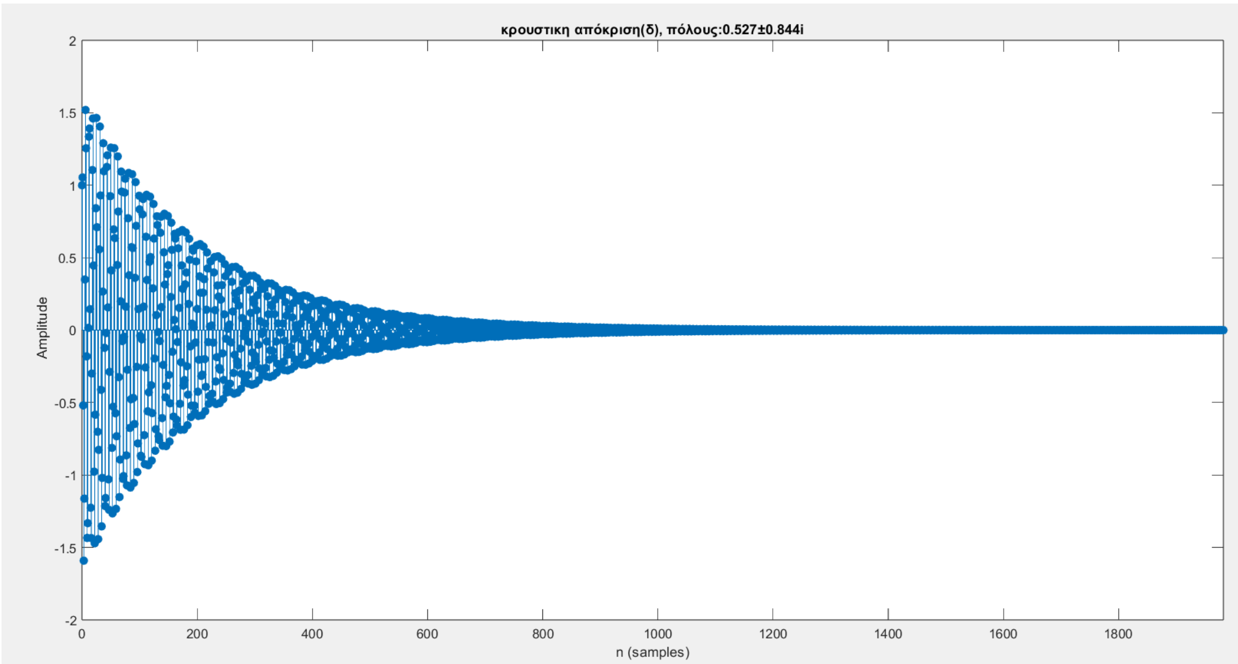
Η περιοδικότητα, άρτια/περιττή κλπ. Ισχύει ακριβώς οπως στα προηγούμενα ερωτήματα.Βλεπουμε το πλάτος να μεγιστοποιειται οταν η φάση μηδενίζεται χωρίς να απειρίζεται άρα το φίλτρο είναι ευσταθές, που ισχυει αφου και οι πόλοι και τα μηδενικά είναι εντός του μοναδιαίου κύκλου. Η μεγιστοποίηση αυτή γίνεται με ”έντονο τρόπο” γύρω από το 1.012=arctan(0.8/0.5) . Αρα η μορφή της απόκρισης συχνότητας είναι συγκεντρωμένη γύρω από 1.012, άρα μπορεί να θεωρηθεί πως το φίλτρο λειτουργεί κυρίως ως ζωνοπερατο φίλτρο γύρω από την περιοχή 1.012 ,γιατί σε αυτή τη περιοχή το σήμα θα ενισχυεται ,ενώ μακριά απο τη συχνότητα 1.012 το σήμα σχεδόν μηδενίζεται.

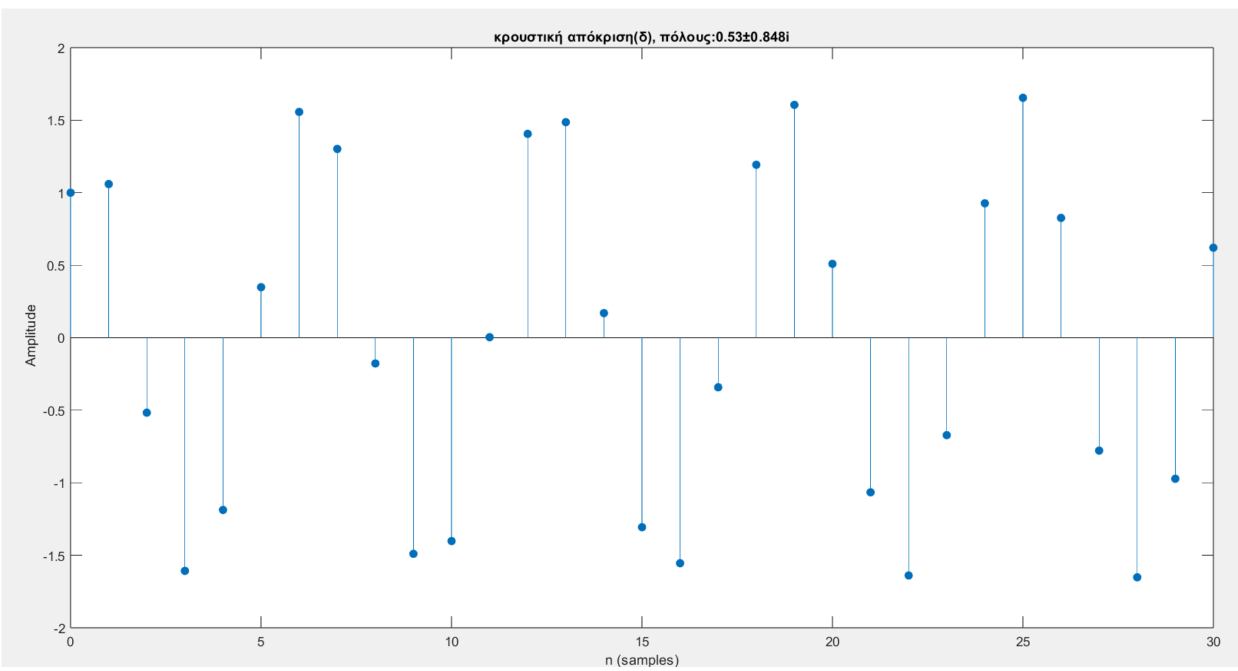
(γ)Τώρα σχεδιάζουμε την κρουστική απόκριση του συστήματος (με χρήση της συνάρτησης impz()), καθώς και τη βηματική απόκριση (με χρήση της stepz()).

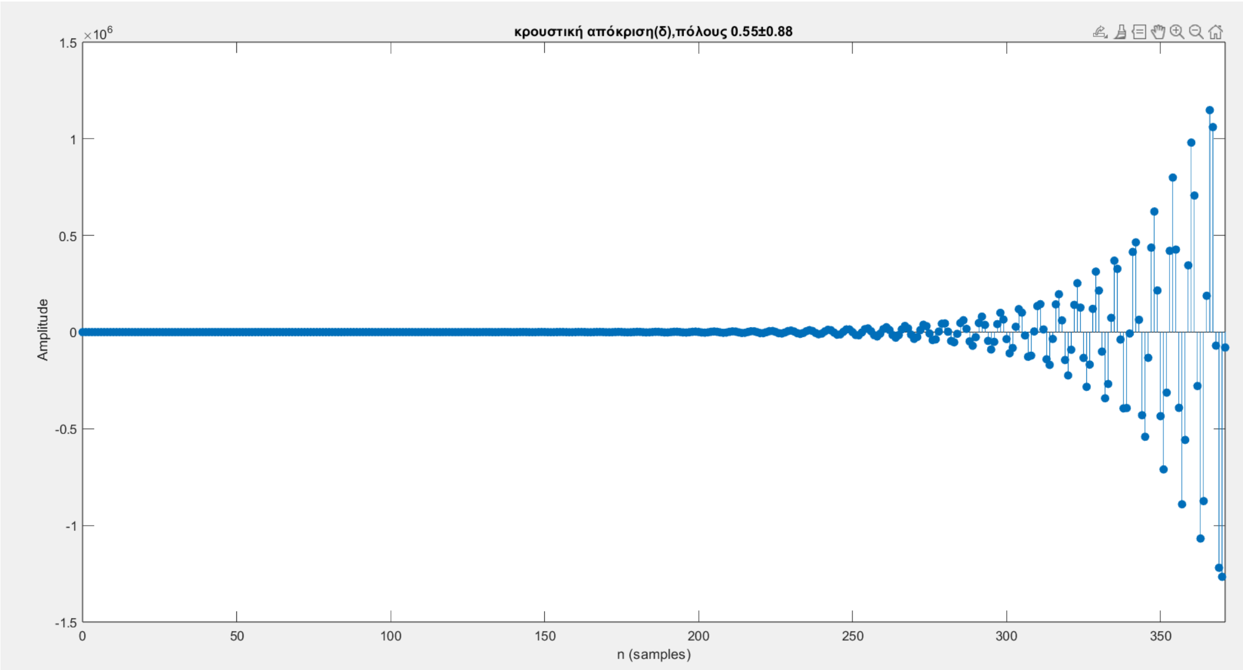




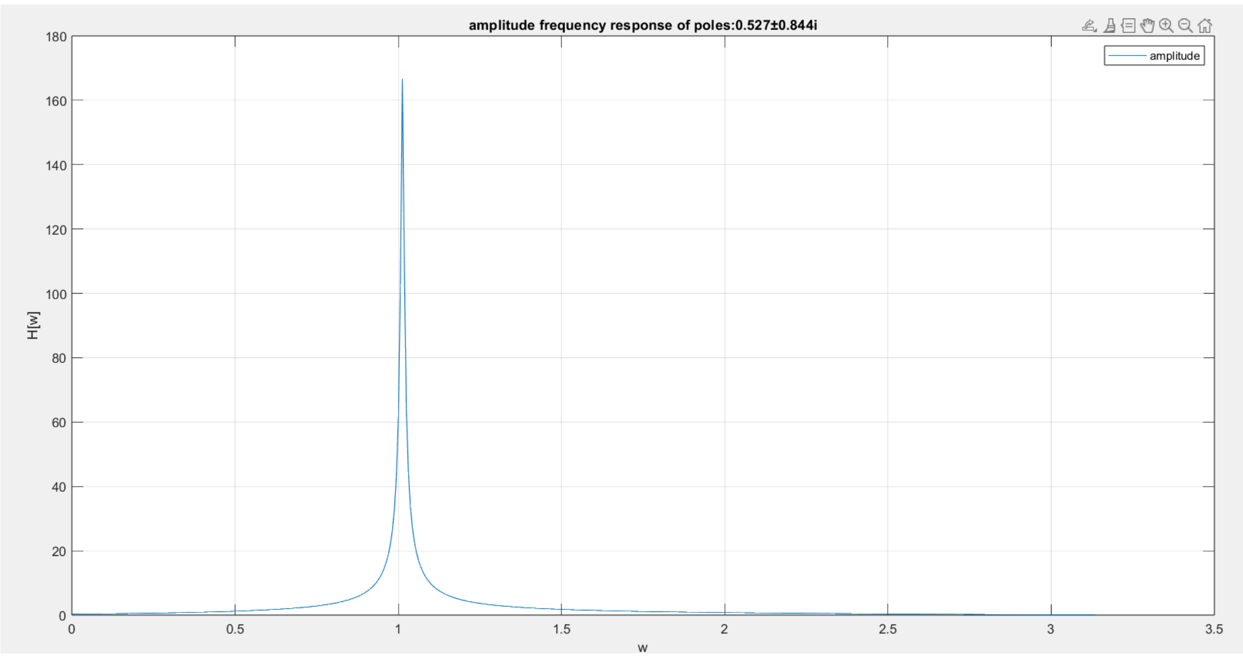
(δ)Σε αυτό το ερώτημα μετακινούμε τους πόλους του συστήματος στις θέσεις: {0.527±0.844i}, {0.53±0.848i}, {0.55 ± 0.88i} ,με τα μηδενικά να παραμένουν ίδια.Και σχεδιάζουμε τις αντίστοιχες κρουστικές αποκρίσεις



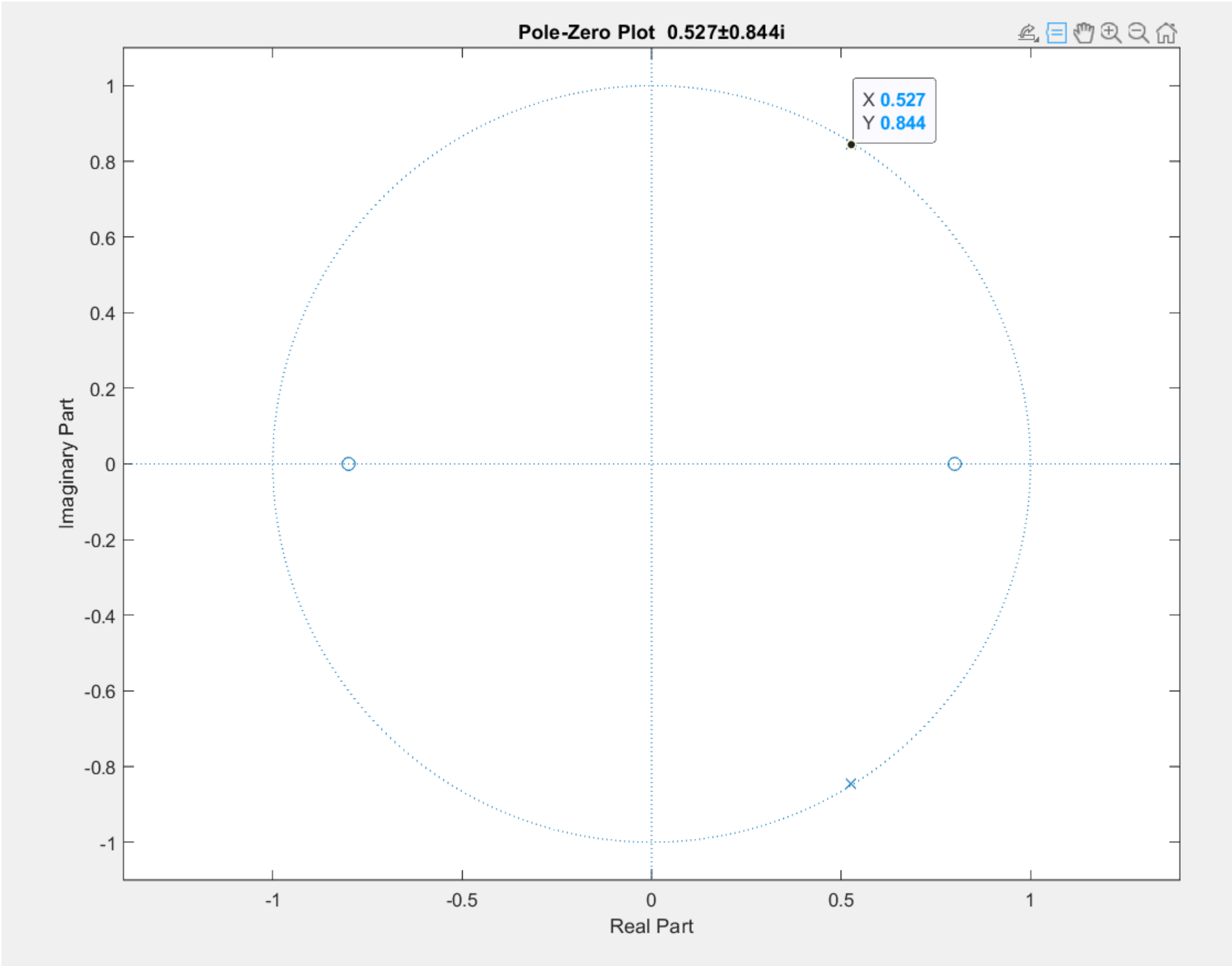


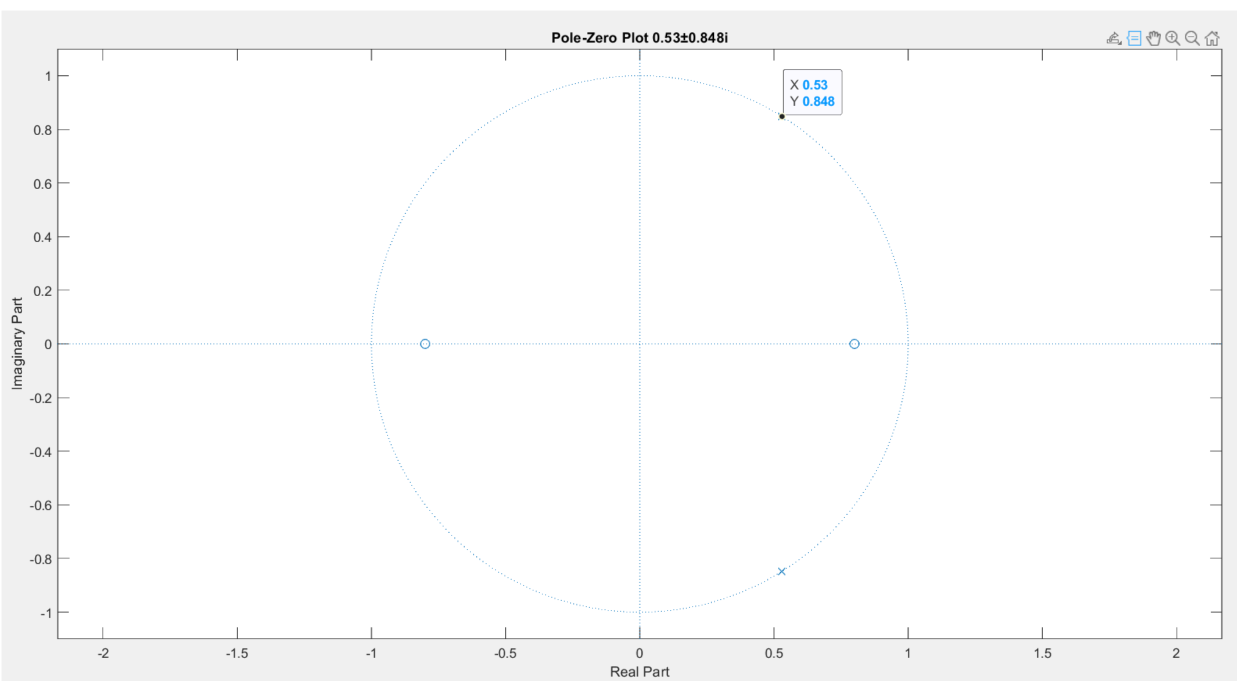


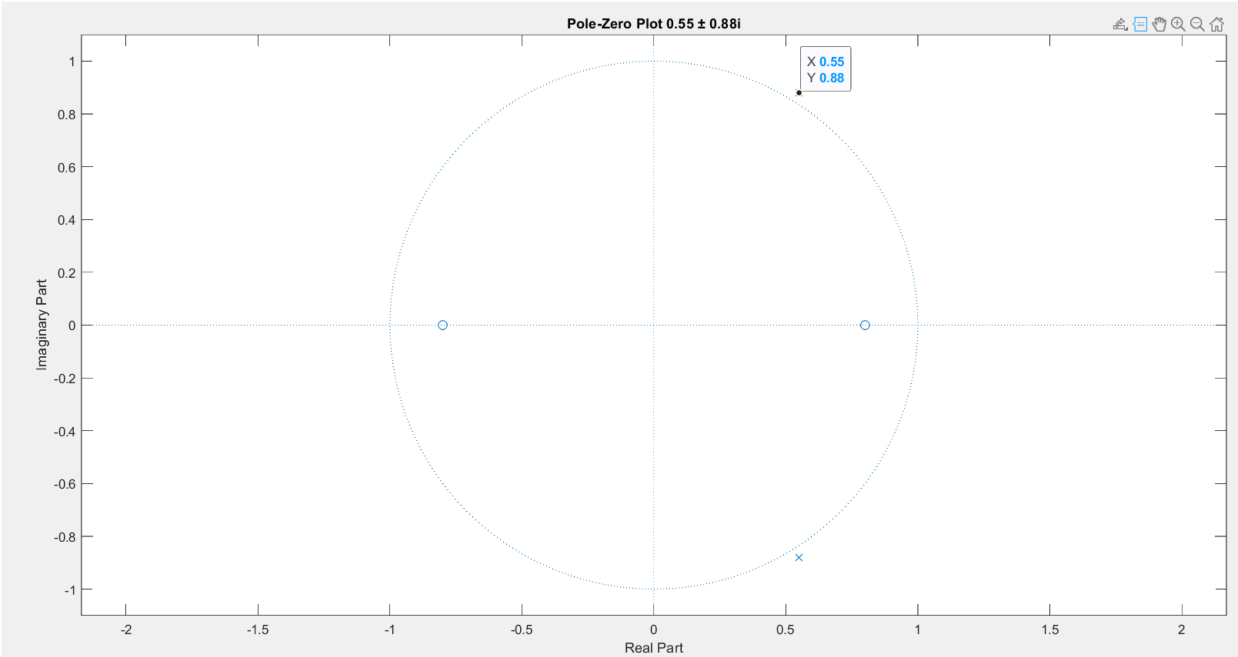
Και η απόκριση πλάτους της περίπτωσης όπου poles={0.527±0.844i}



Τέλος σχεδιάζω τα διαγράμματα μηδενικών/πόλων για την καλύτερη κατανόηση των αποκρίσεων των νέων αυτών φίλτρων:

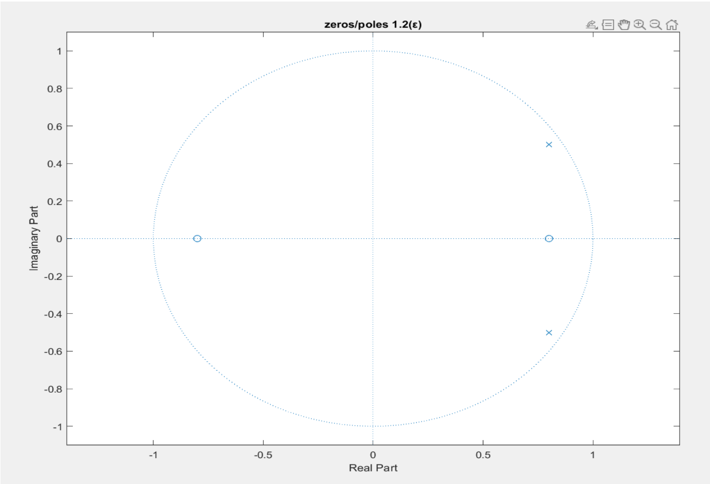

Pole-Zero Plot 0.527±0.844i
X 0.527
Y 0.844
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1


Pole-Zero Plot 0.53±0.848i
X 0.53
Y 0.848
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2



Απο το zplane() παρατηρούμε πως η γωνία των πόλων παραμένει ιδια απλά αυξανεται η απόσταση από το κέντρο και μειώνεται η αποσταση απο το μοναδιαίο κύκλο μέχρι την στιγμή που τον περνά με προφανή τρόπο distance(poles,(0,0))>1
Αφου αντιστοιχα με τη παραπάνω σειρά εμφάνισης είναι
dist1=0.995<1
dist2=1.00002≈1
dist3=1.0378>1
Αρα με βάση τις παρατηρήσεις μας απο το zplane() και τα γραφήματα του impulse response μπορούμε να πούμε πως ,για {0.527±0.844i} το σύστημα είναι ευσταθές αφου αρχίζει την έντονη ταλάντωση αλλά με τη πάροδο του χρόνου το σήμα ελαττώνεται, αυτο φαίνεται και απο το frequency response, όπου το amplitude τίνει να απειρειστεί (μεγαλώνει πολύ έντονα),αλλα δεν απειρίζεται, αυτο γίνεται γιατι το distance τείνει στο 1 άλλα distance<1.
Για {0.53±0.848i} οπου dist2=1.00002≈1 το σήμα είναι οριακά ασταθες αφου δε σταθεροποιείται πότε αλλα δεν εκτινάζεται σε αστάθεια ,αλλά πάλλεται και ταλαντώνεται με σχετικά σταθερό τρόπο χωρίς να σταθεροποιηθεί ποτέ επομένως το σύστημα μπορεί να θεωρηθεί πώς είναι στη μετάβαση μεταξύ αστάθειας και ευστάθειας, αλλα τείνει προς την οριακή αστάθεια.
Τέλος για {0.55 ± 0.88i} όπου dist3=1.0378 είναι εμφανώς μεγαλύτερο του 1 (εκτός μοναδιαίου κύκλου).Αρα το σύστημα είναι ασταθές, το οποίο διαπιστώνουμε και απο το impulse response ,αφου σταθεροποιήτε πρακτικά απευθείας και μετα απο καποιο χρονικο διάστημα η ταλάντωση ξανα-ξεκινάει με έντονο ρυθμό αρα βλέπουμε αστάθεια.

(ε)Τώρα θέτουμε πόλους p=0.8 ± 0.5i ,και διατηρούμε τα ίδια μηδενικά z={+0.8,-0.8}. Αρχικά φτιάχνουμε διάγραμμα πόλων/μηδενικών για το νέο αυτό φίλτρο:

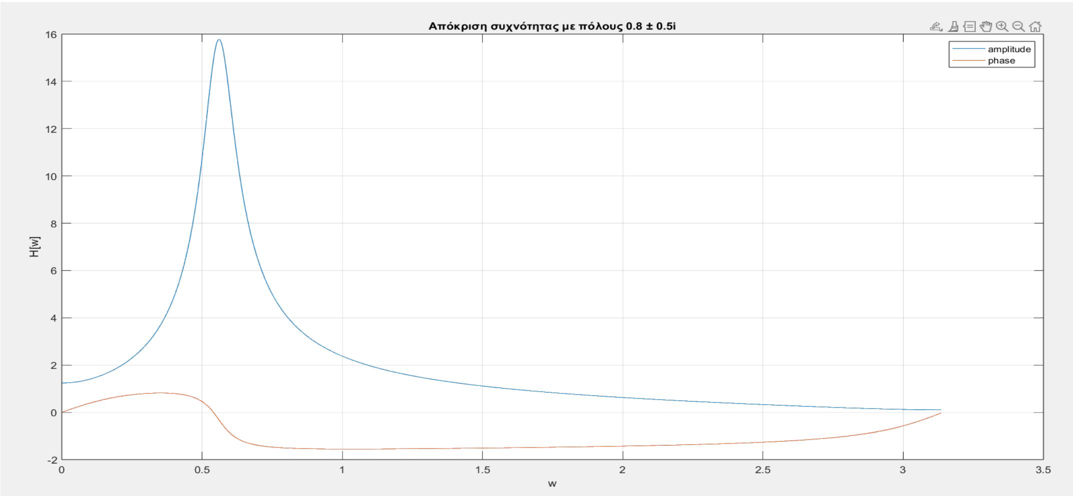


Και με την zp2tf(), βρίσκουμε τους συντελεστές:

b=[1.0000   0   -0.6400] (αριθμητή)

a=[1.0000   -1.6000   0.8900] (παρονομαστή)

Και τωρα φτιαχνουμε το amplitude/phase frequency response αυτού του φίλτρου, με την εντολή freqz():

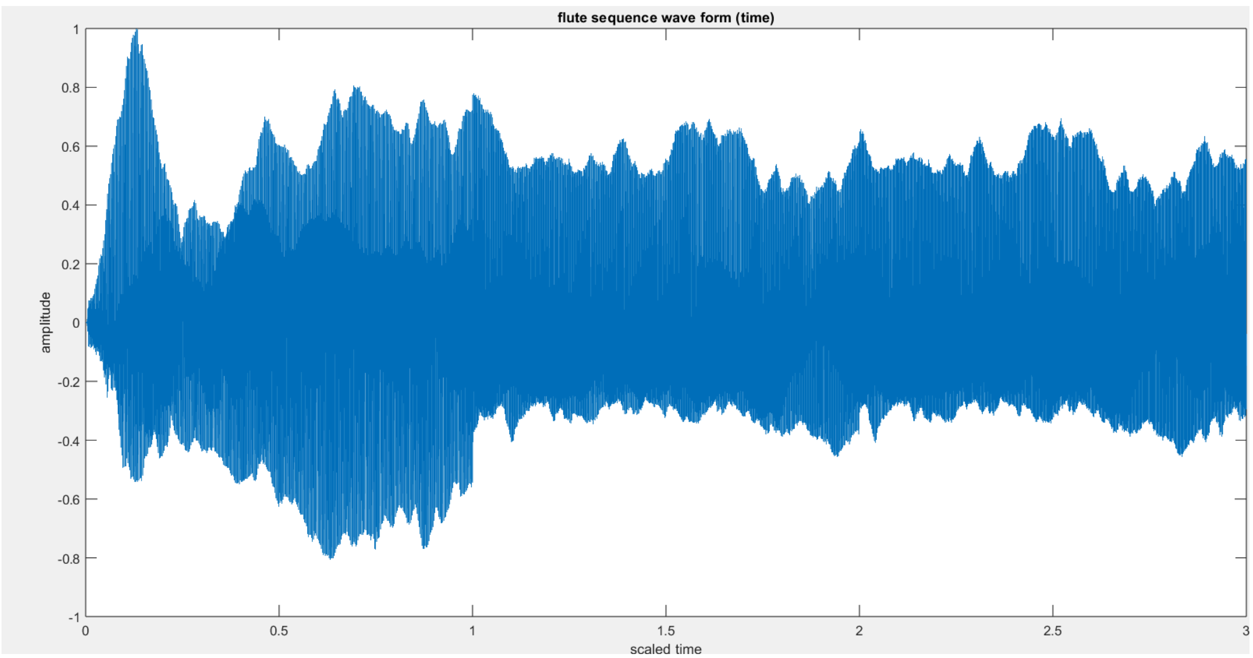


Παρατηρούμε πως η μορφή του frequency response είναι σχεδόν πανομοιότυπο με το frequency response του πρώτου ερωτήματος, ομως τώρα ο μηδενισμός της φάσης και η μεγιστοποιηση του amplitude είναι στο 0.5645=arctan(0.5/0.8) , αρα οι παρατηρήσεις του αρχικού σήματος ισχύουν και σε αυτο το σήμα με τη διαφορά ότι πλέον το σήμα είναι συγκεντρωμένο στο 0.5645 ,αρα έχουμε πάλι ένα φίλτρο που είναι όμοιο με ζωνοπερατό που αποκόπτει συχνότητες εκτός της περιοχης της συχνότητας γύρω από το 0.5645. Παρατηρούμε πως η ζώνη διέλευσης μετακινήθηκε κατα την γωνία μεταβολής των πόλων δηλαδή κατα arctan(0.8/0.5)-arctan(0.5/0.8) = = 1.012-0.5645 = 0.4475.

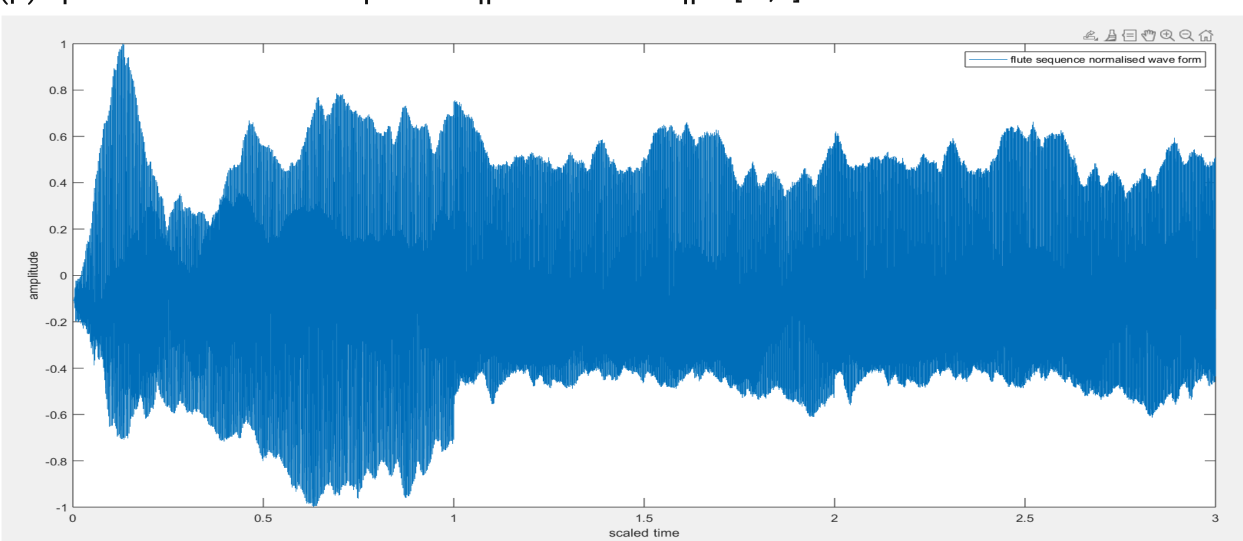
(2) Ανάλυση Μουσικών Σημάτων και Εφαρμογή Φίλτρων

2.1 Ανάλυση Μουσικών Σημάτων

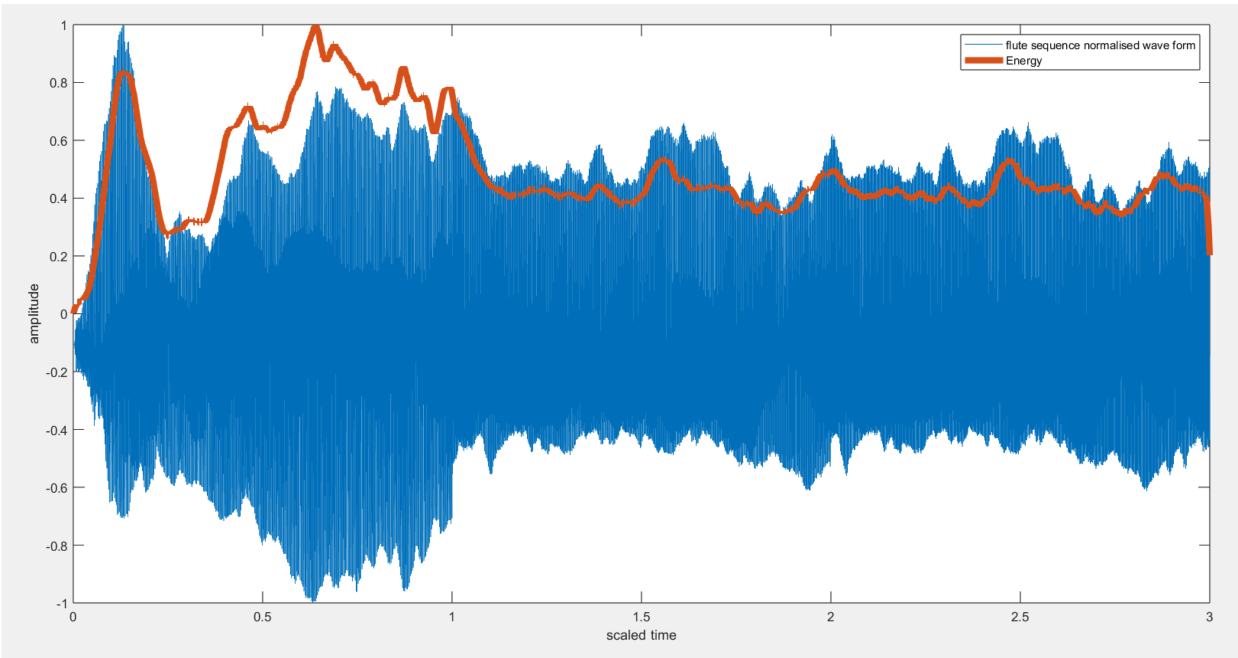
(α) Φορτώνουμε στο Matlab, με χρήση της εντολής audioread() , το αρχείο flute sequence, το ακούμε με την εντολή sound(), ακούμε τρεις νότες, και το σχεδιάζουμε με την εντολή plot():



(β)Πρώτα κανονικοποιούμε το σήμα στο διάστημα [-1,1]:

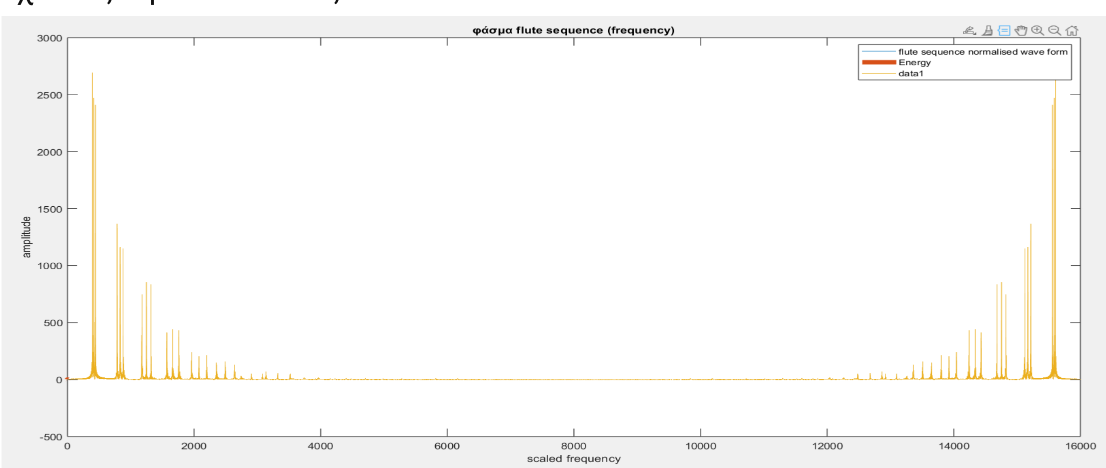


Έπειτα σύμφωνα με τη σχέση: $$E[n] = \sum_{m=0}^{M} x^2[m]w[n-m],$$ υπολογίζουμε την ενέργεια του σήματος σε κυλιόμενα παράθυρα, όπου x[m] το σήμα,
και w[n] = 0.54−0.46cos(2πn/N) ,0≤n≤N. Με N=400, δηλαδή E=conv(x^2,w):



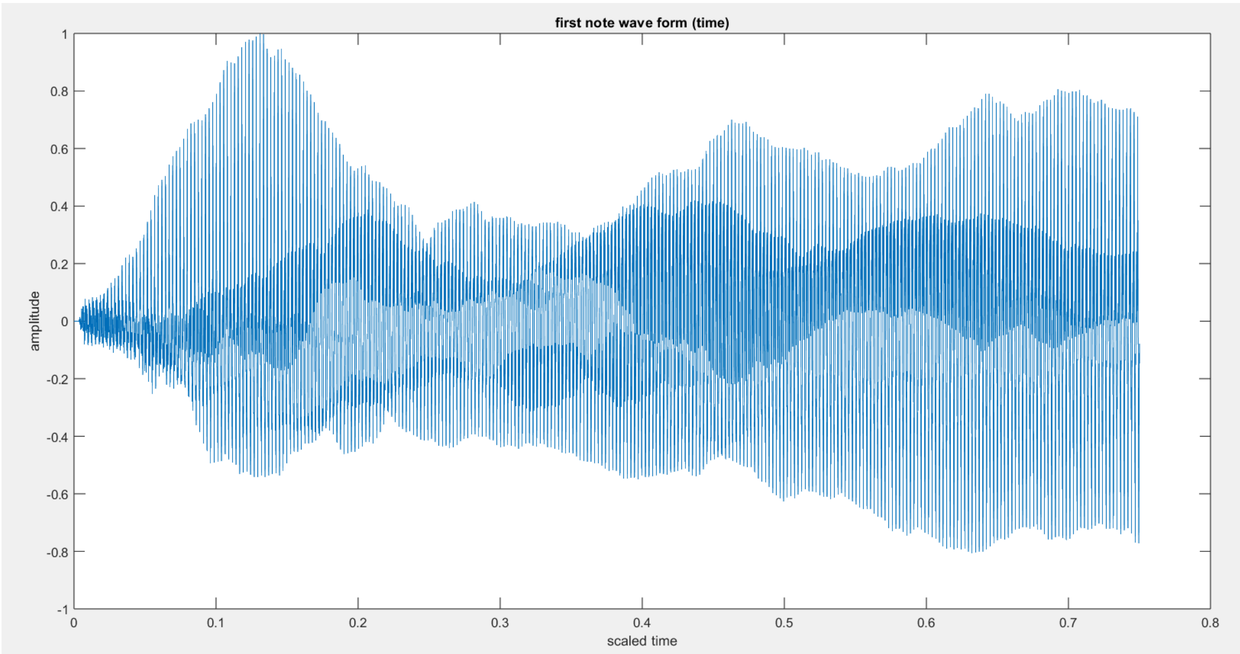
Παρατηρούμε πως η ενέργεια του σήματος σε κυλιόμενα παράθυρα είναι σχετικά ανάλογη με το πλάτος τους σήματος, εφόσον παρουσιάζει μέγιστα και ελάχιστα στα ίδια σημεία που το πλάτος παρουσιάζει μέγιστα και ελάχιστα, με σχετική αναλογία στα μεγέθη αυτών. Αφου βλέπουμε πως σε ψηλότερα μέγιστα του πλάτους και η ενέργεια παρουσιάζει μεγαλύτερα μέγιστα (αντίστοιχα για τα ελάχιστα).

(γ)Τώρα με την εντολή fft() υπολογίζουμε το φάσμα (DTFT) του σήματος και σχεδιάζουμε το πλάτος του:

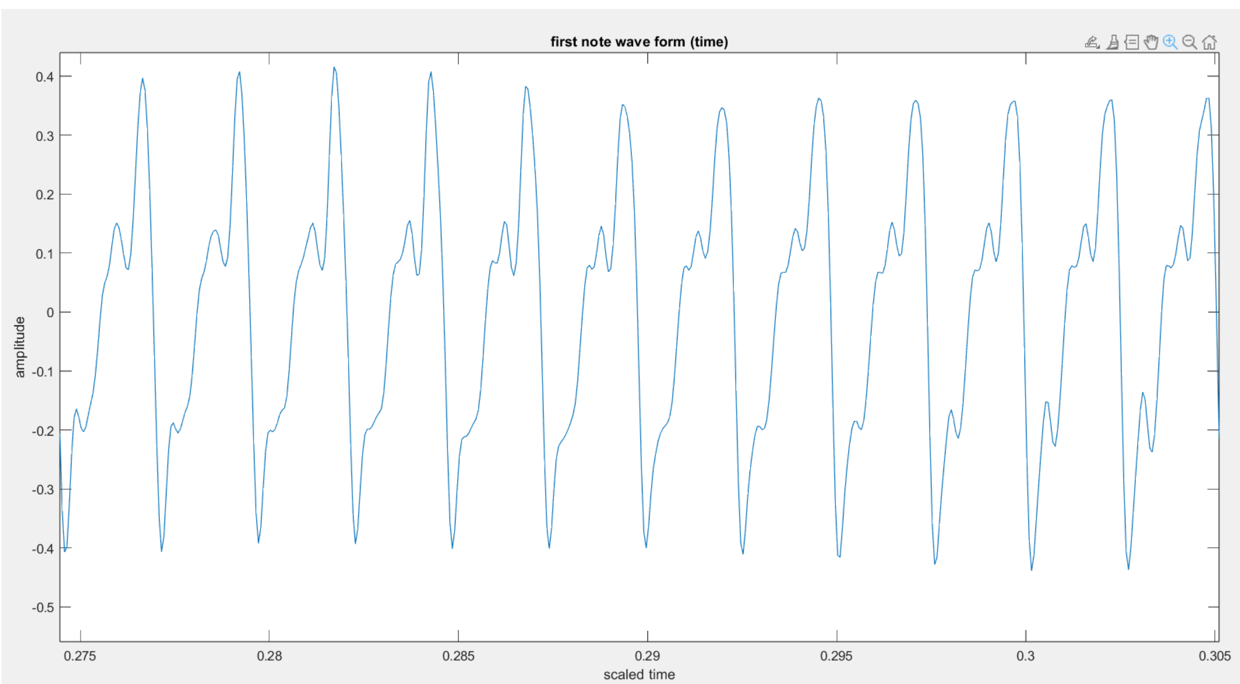


Βλέπουμε τις κεντρικές συχνότητες (τα μεγαλύτερα πλάτη) και τις αντίστοιχες αρμονικές (μικρότερα πλάτη),όπου η κάθε αρμονική περιέχει τρεις συχνότητες για τις αντίστοιχες 3 νότες του σήματος.

(δ)Τώρα απομονώνουμε μια νότα (την πρώτη), σχεδιάζοντας μόνο το πρώτο ⅓ μέρος του σήματος με την εντολή plot():

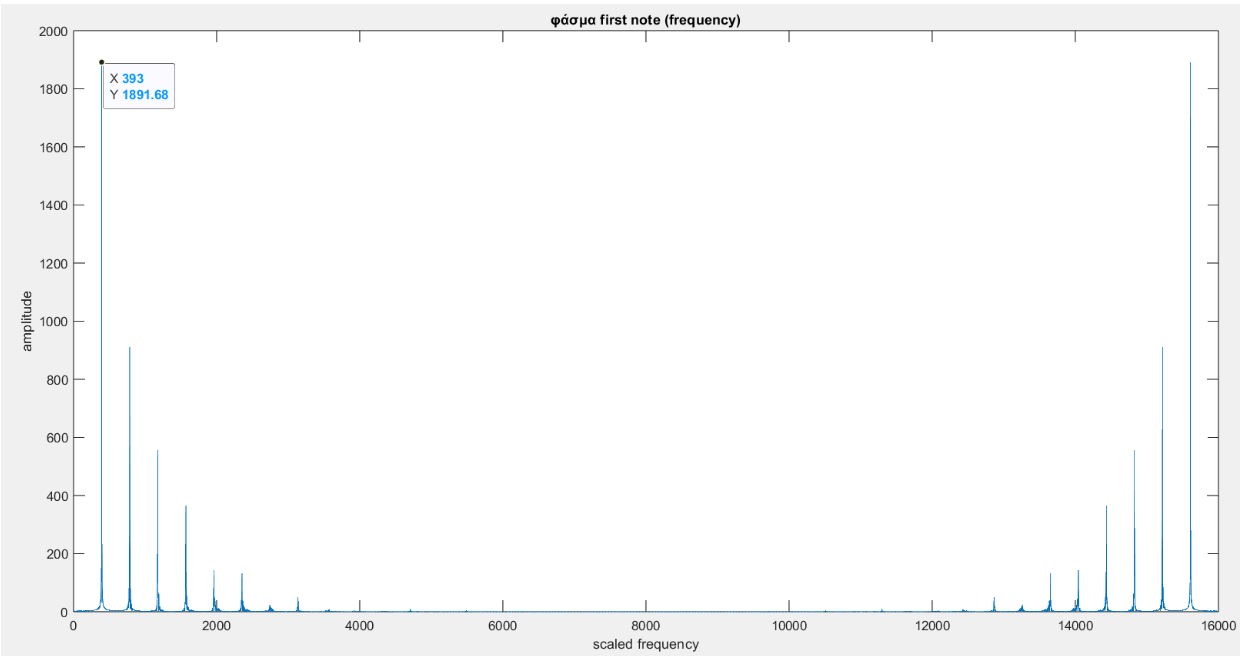


Και το σχεδιασμένο μεγεθυμένο σήμα:



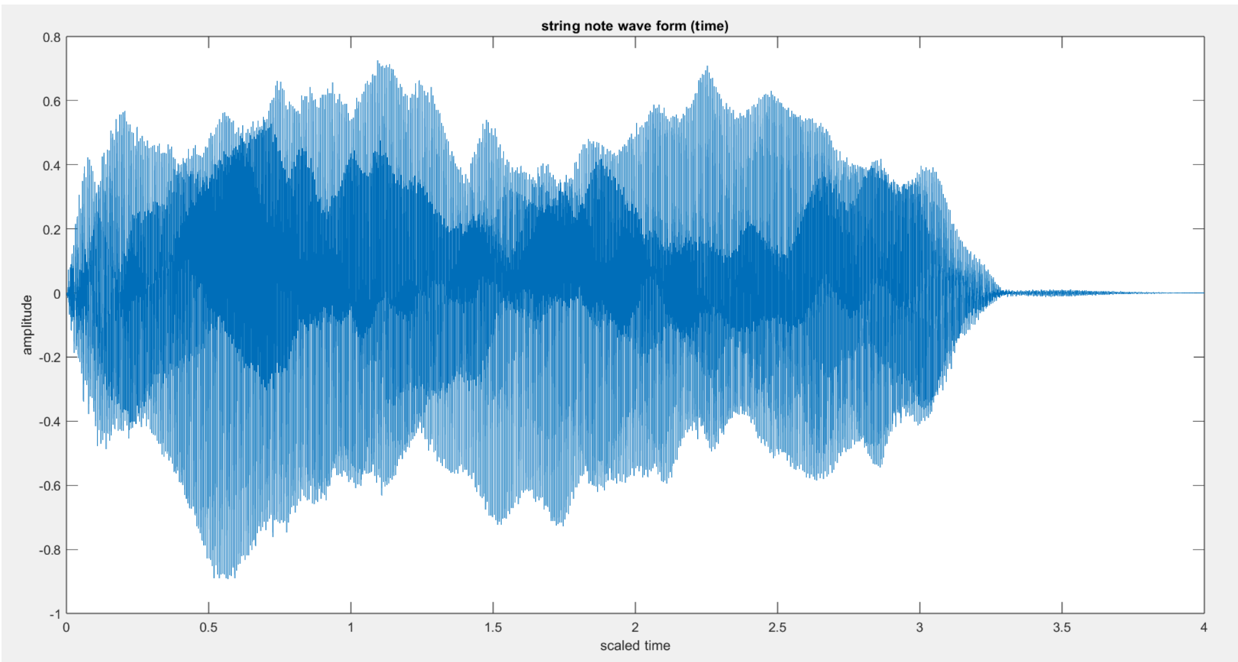
Αρα βλέπουμε πως όντως το σήμα της απομονωμένης (πρώτης) νότας είναι περιοδικό με περίοδο περίπου:T=0.289962-0.287399=0.002563=2.563 ms

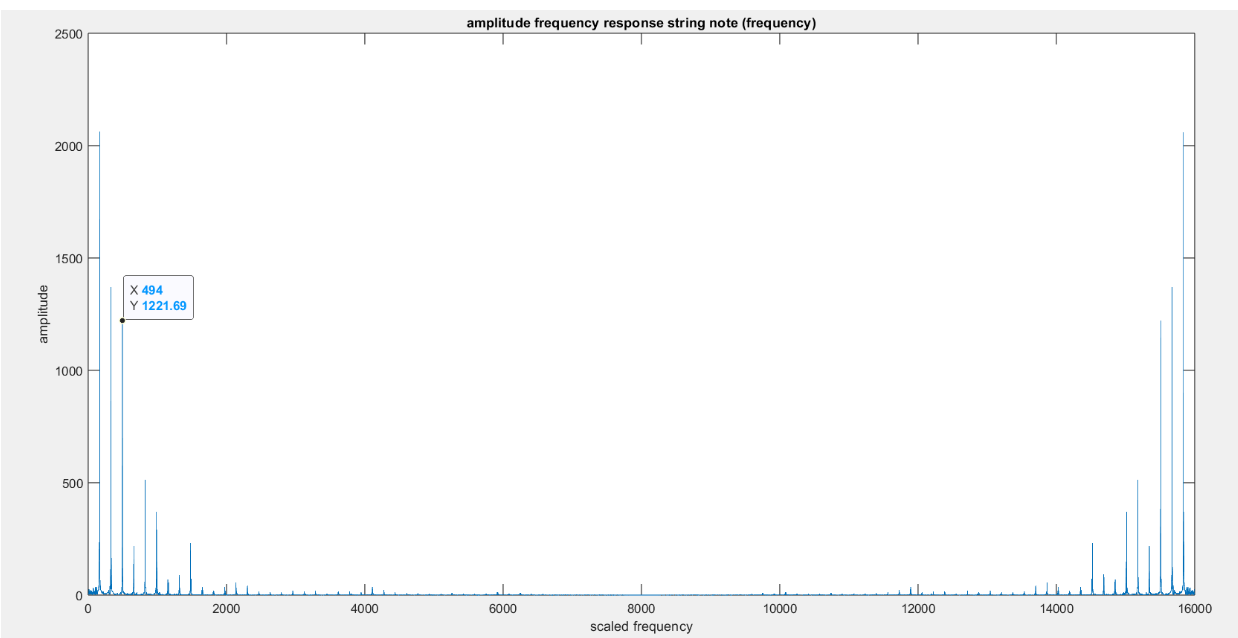
(ε)Με χρήση της εντολής fft() σχεδιάζουμε το μέτρο του φάσματός της νότας που απομονώσαμε:

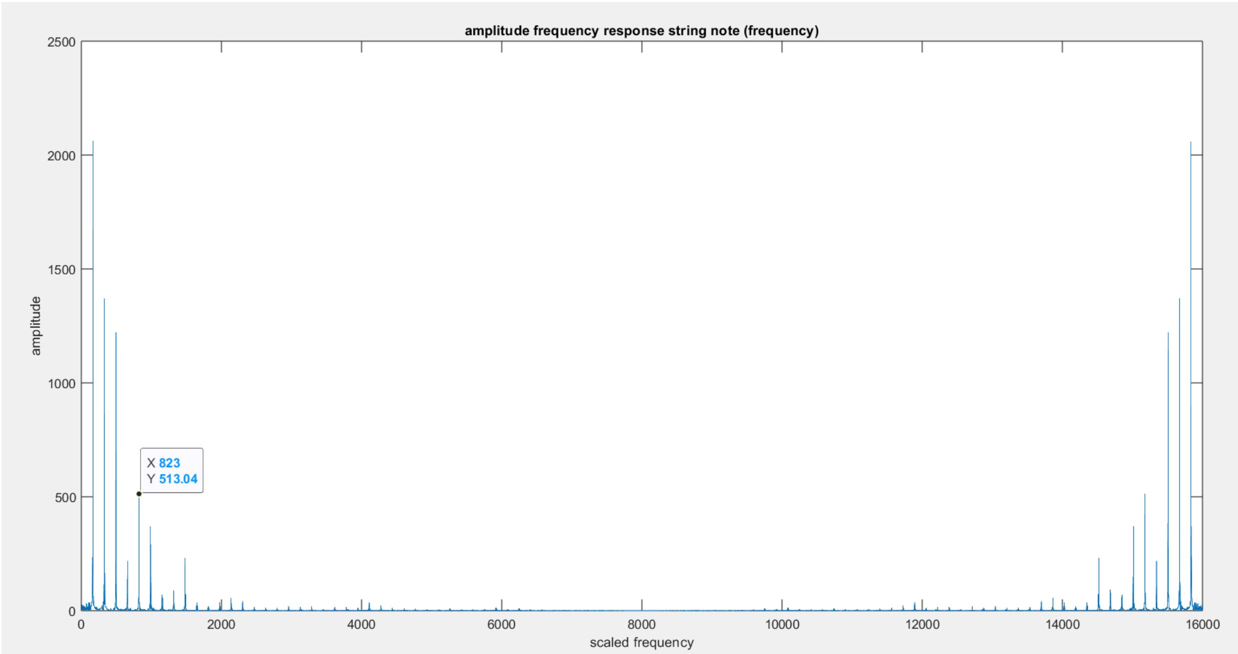


Η κεντρική συχνότητα (το μεγαλύτερο πλάτος στο φάσμα) βλέπουμε πως έχει συχνότητα f=393Hz, και πριν βρήκαμε ότι η περίοδος είναι T=2.563 ms, όπου: 1/T(=f) =390.1 Hz ≈ 393 Hz. Αρα η σχέση μεταξύ θεμελιώδους συχνότητας και περιόδου ισχύει.
Στο νέο αυτό φάσμα βλέπουμε τις αρμονικές μεγαλύτερης συχνότητας ,να είναι όντως ακέραια πολλαπλάσια της κεντρικής συχνότητας, με το πλάτος αυτών να μειώνεται καθώς αυξάνεται η συχνότητα .Συγκριτικά με το προηγούμενο φάσμα βλέπουμε πως έχουν αποκοπεί οι άλλες δύο νότες ,αφού πλέον η κάθε αρμονική και η κεντρική συχνότητα περιέχουν μόνο μία συχνότητα (τη συχνότητα της πρώτης νότας) και οχι τρεις όπως προηγουμένως.Αρα η μορφή αυτού του φάσματος μοιάζει με τη μορφή του συνολικού/αρχικού φάσματος, απλά αντι για σετ τριων συχνοτήτων γυρω απο κάθε αρμονική, έχουμε μόνο μια συχνότητα.
(Τα δειγματα προσαρμόστηκαν απο 4*16000 σε 16000 με σκοπό να βρούμε τη κεντρική συχνότητα χωρίς περαιτέρω βήματα, αλλά και με την μεγαλύτερη ακρίβεια των 64000 δειγμάτων προκύπτουν τα ίδια αποτελέσματα).

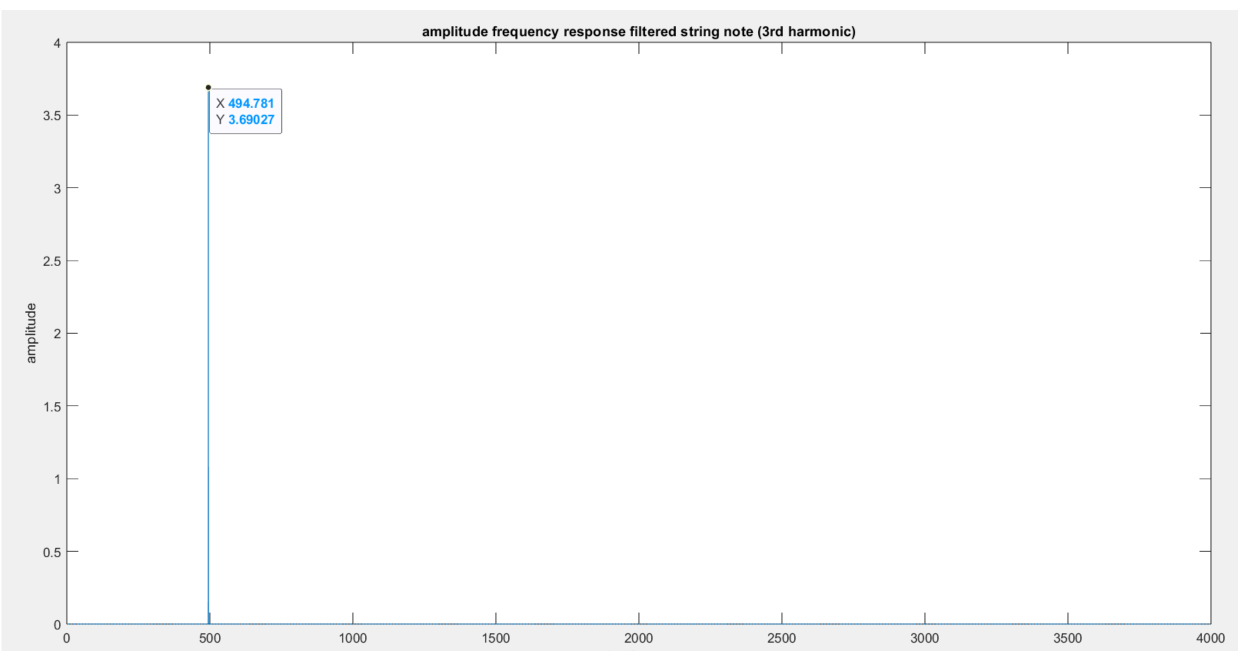
(στ) Τώρα Χρησιμοποιώντας το αρχείο string_note.wav (συχνότητα δειγματοληψίας: 16 kHz), υλοποιούμε ένα ζωνοπερατό φίλτρο, με στόχο να απομονώσουμε την 3η αρμονική του σήματος.
Αρχικά μερικά γραφήματα του string_note.wav για τη καλύτερη απεικόνιση του σήματος:



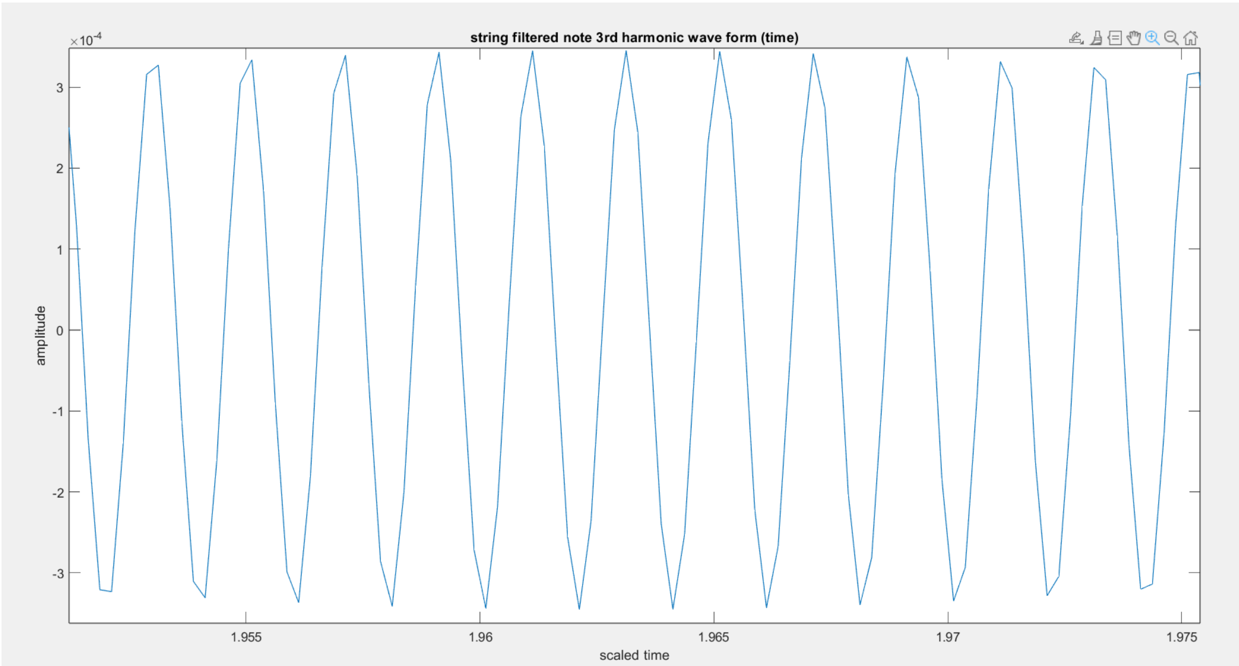




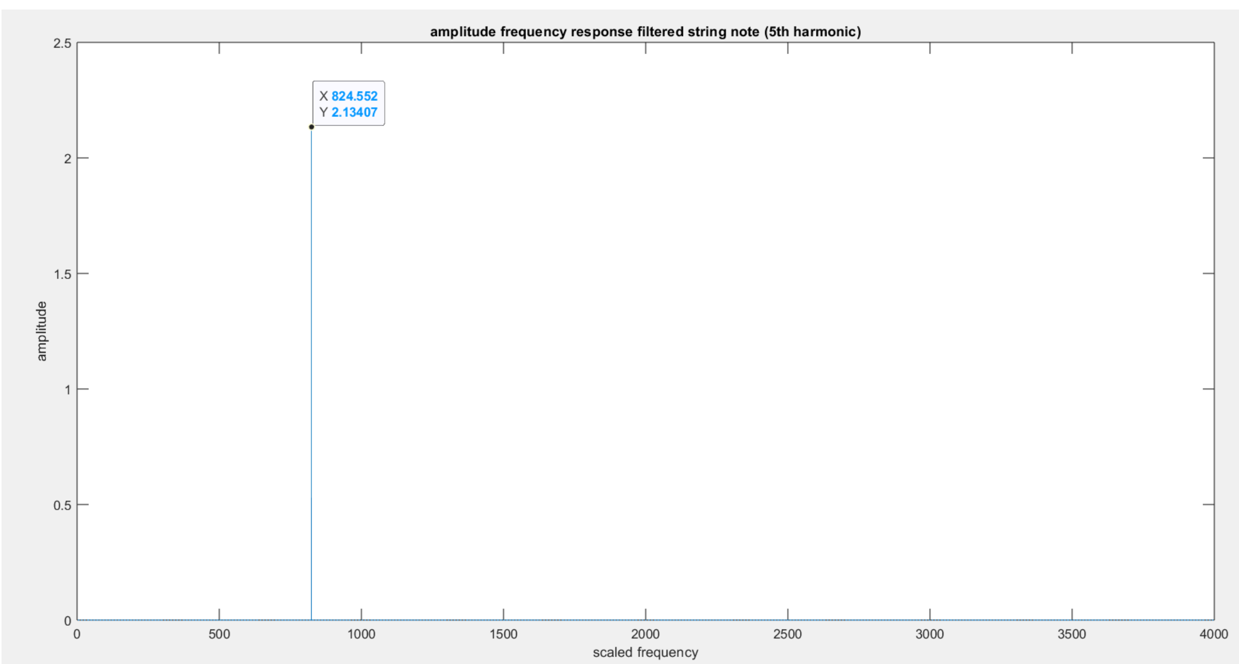
Το φίλτρο το φτιάχνουμε στο πεδίο της συχνότητας που θα είναι μια συνάρτηση: H(F)={1 για |F-Fs|<F1, 0 αλλιώς} όπου Fs η συχνοτητα που θέλουμε να απομονώσουμε και F1 ενα επαρκώς μικρο παράθυρο γύρω από αυτή.
Αρα τώρα θα έιναι Fs=”συχνότητα 3ής αρμονικής”=494 Hz και F1=”small value”=1Hz.
Και πολλαπλασιάζουμε το σήμα στο πεδίο της συχνότητας με αυτό το φίλτρο για να πάρουμε το αναμενόμενο αποτέλεσμα.



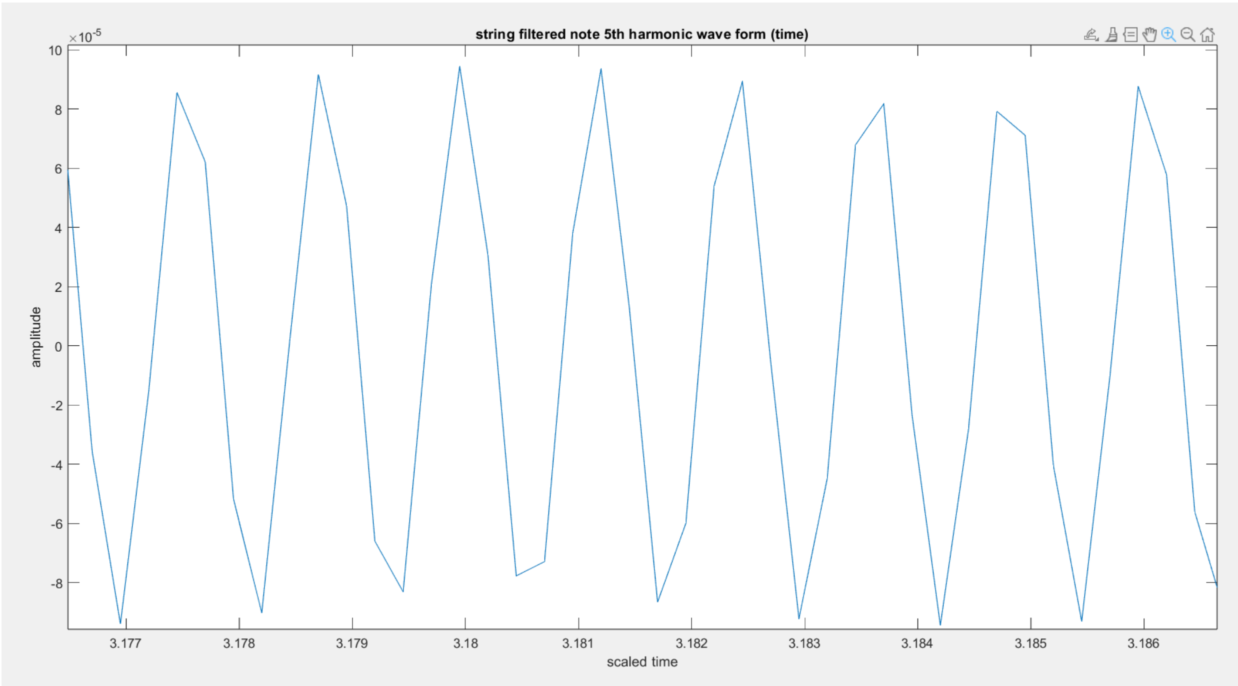
Απόσπασμα στο χρόνο:



Τώρα με την ίδια ακριβώς διαδικασία απομονώνουμε την 5ή αρμονική, απλά στο φίλτρο που φτιάξαμε αλλάζουμε τη συχνότητα που θέλουμε να απομωνοσουμε απο Fs=494Hz σε Fs=”συχνότητα 5ης αρμονικής”=823Hz, όλα τα άλλα παραμένουν τα ίδια.



Απόσπασμα στο χρόνο:

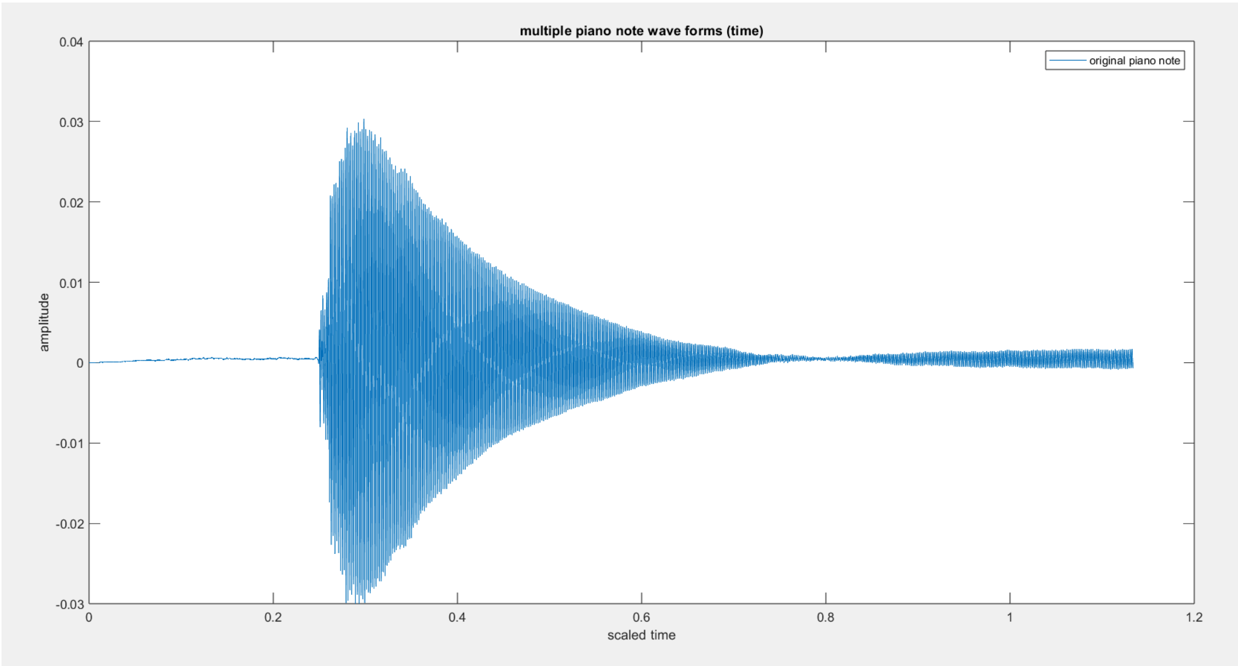


Και στις δύο περιπτώσεις μπορούμε να κάνουμε τις ίδιες παρατηρήσεις, όπου με τη χρήση του φίλτρου όντως παραμένει μόνο μία συχνότητα (η 3ή και η 5ή αρμονική αντίστοιχα), γι αυτο και βλέπουμε μόνο μια κατακόρυφη γραμμή στο φάσμα στην αναμενόμενη συχνότητα (494 Hz και 824 Hz αντίστοιχα).Έπειτα στα αποσπάσματα στο χρόνο βλέπουμε πως τα σήματα ειναι ημιτονοειδους μορφης (η απόκλιση από αυτή τη μορφή δηλαδή: οι γωνίες, τα ευθύγραμμα τμήματα και γενικά η “μη-ομαλότητα” οφείλονται στον μικρό αριθμό δειγμάτων),αναμενόμενο αφού έχουμε μόνο μία συχνότητα . Όπου η περίοδος κάθε σήματος (μετρώντας την όπως στο 2.1(δ)) είναι η αναμενόμενη δηλαδή ισχύει για αυτή: T=1/f={1/494 για 3η αρμονική, 1/824 για 5η αρμονική}.

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

(α) Φορτώνουμε στο Matlab, με χρήση της εντολής audioread(), το αρχείο piano_note.wav

Ακούμε το αρχείο με την εντολή sound() (είναι μια νότα πιάνου), και το σχεδιάζουμε με την εντολή plot():



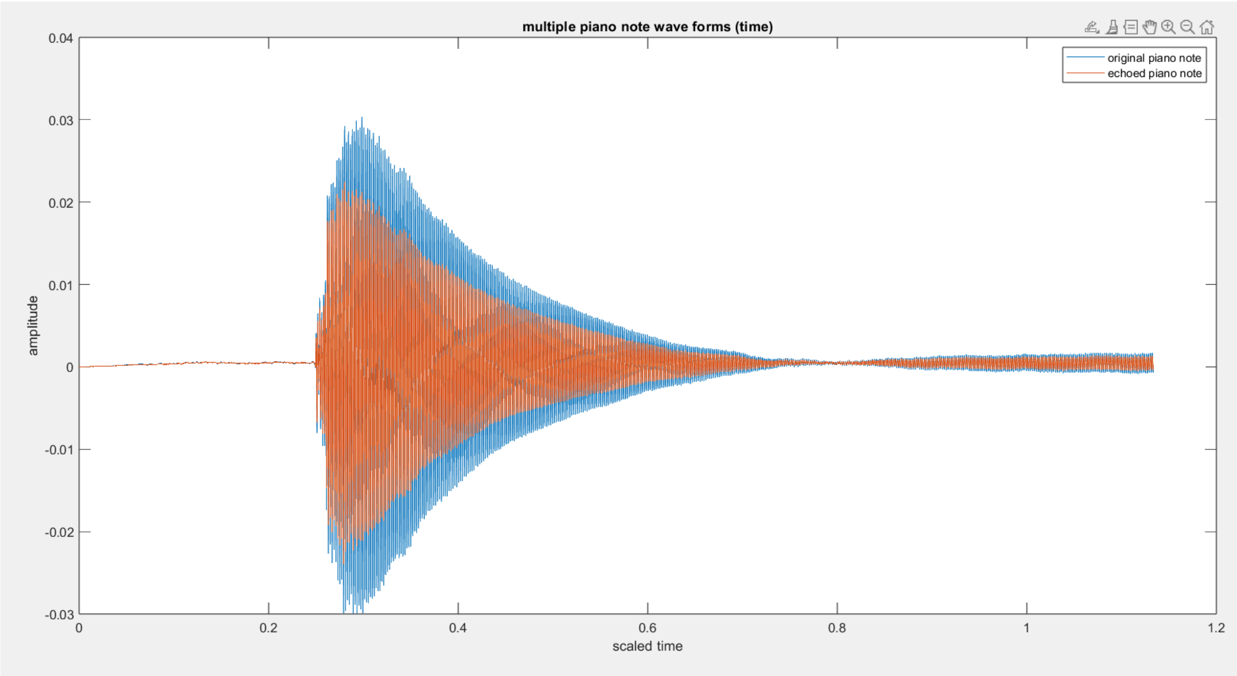
(β) Θα υλοποιήσουμε ένα φίλτρο ηχούς και ένα φίλτρο αντήχησης, με τα οποίο θα φιλτράρουμε το μουσικό σήμα, οπου η εξισώσεις διαφορών των φίλτρων(για c=0.85 P=850) είναι: y[n]=0.85*x[n]+0.15*x[n-850] ,το φίλτρο ηχούς. Για το φίλτρο αντήχησης εφαρμόζουμε 12 διαδοχικές φορές το φίλτρο ηχούς, και τους συντελεστες τους παίρνουμε (όμοια με το αρχικό φίλτρο αντήχησης) εκτελώντας 11 διαδοχικές συνελίξεις. Έτσι έχουμε τα εξής φίλτρα με τους εξής συντελεστές:

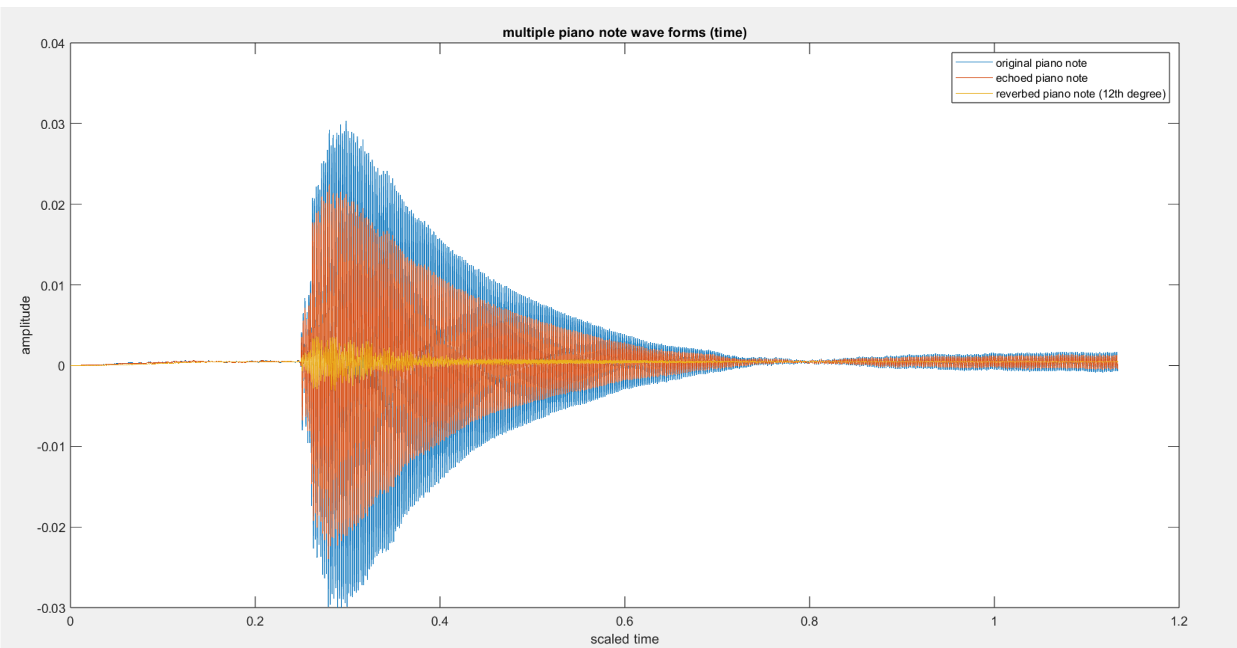
Φίλτρο ηχούς: `be=[0.85 ,[zeros(1,849)], 0.15] και ae=[1, [zeros(1,p)]]`
Φίλτρο αντήχησης: (θα βγει διάνυσμα με πολλά μη-μηδενικά στοιχεία αρα δε το γράφω αλλα δίνω την εντολή υπολογισμού του στο matlab)->

```
br=be;
for i=1:11
   br=conv(br,be);              %reverb filter
end
```

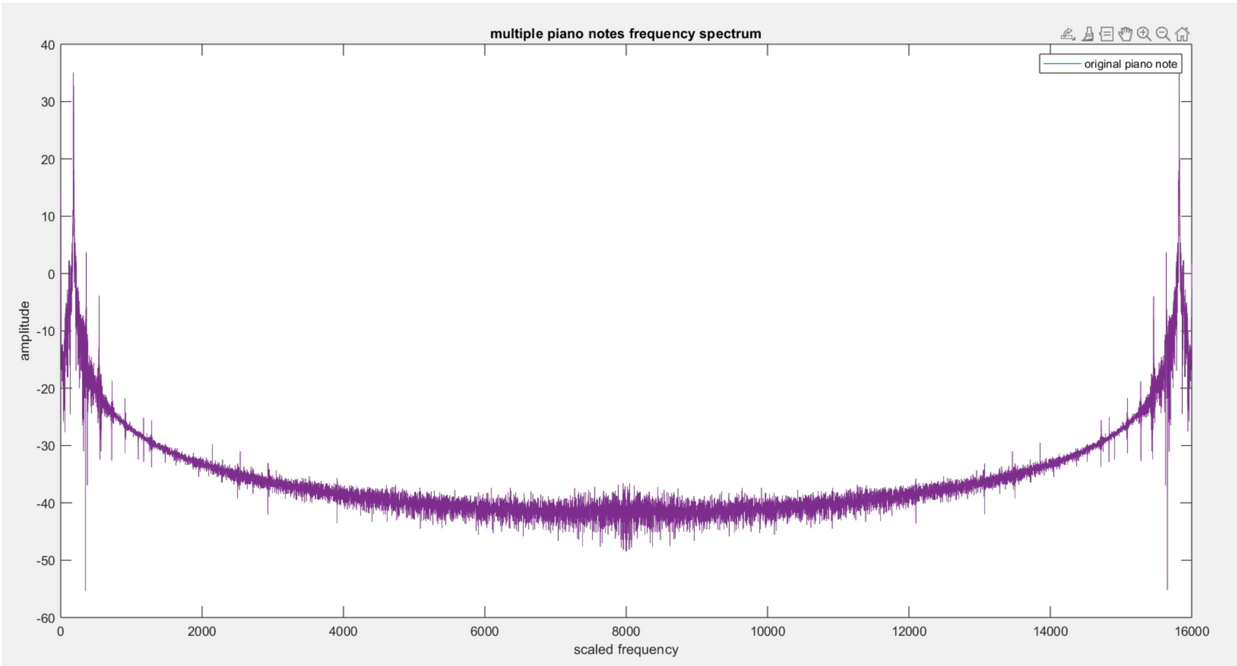
Και `ar=[1,[zeros(1,length(br)-1)]]`

Και τώρα ακούμε με το sound() της δύο εξόδους των φίλτρων (παραμορφώθηκε με αναμενόμενο τρόπο το αρχικό σήμα), και σχεδιάζουμε με την plot() τα δύο παραμορφωμένα σήματα(στο ίδιο γράφημα με το αρχικό για καλύτερη σύγκριση):

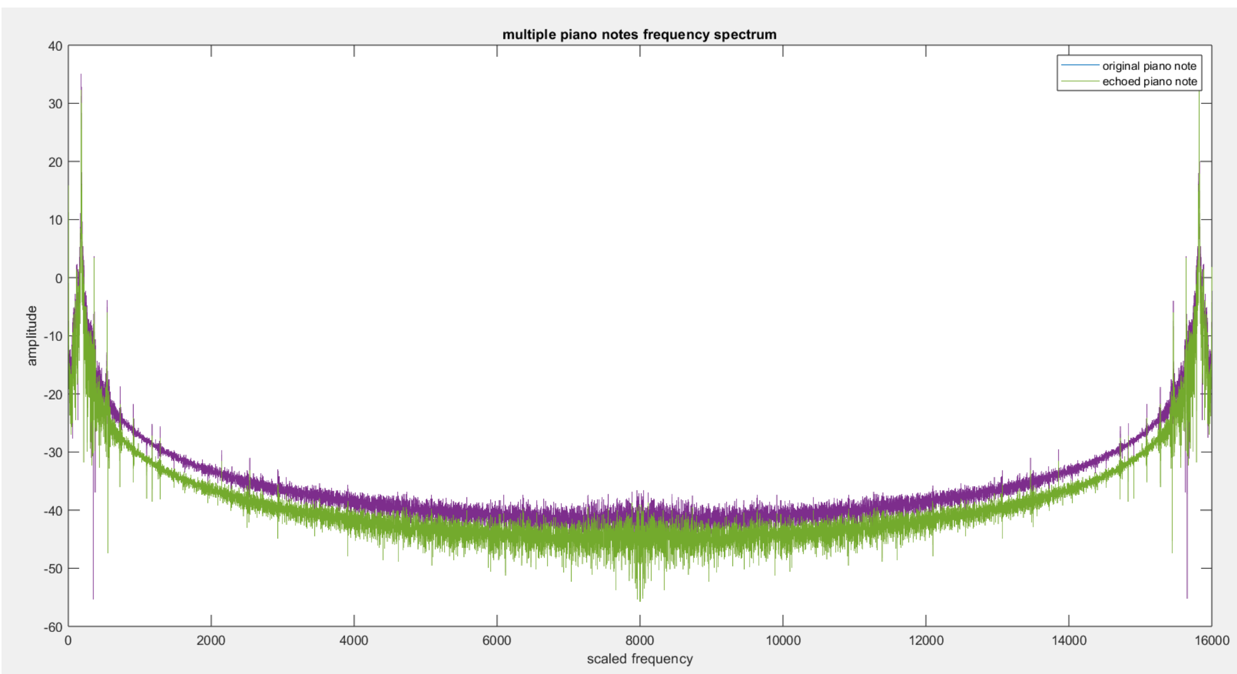




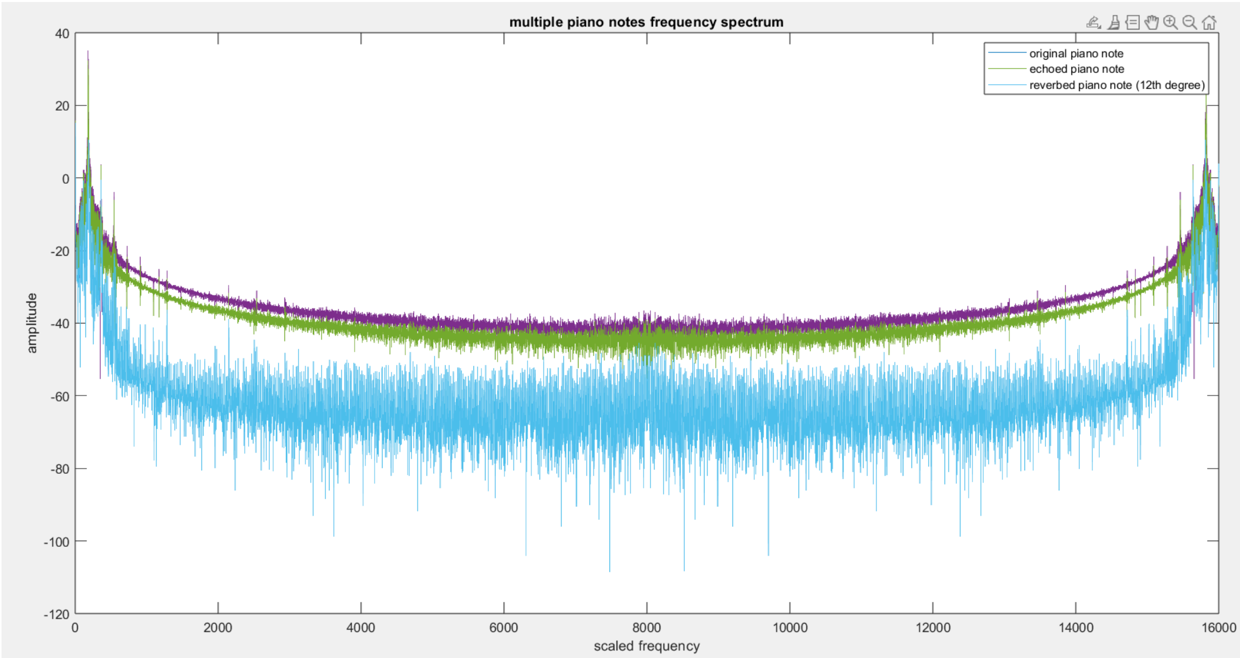
(γ)Με χρήση της εντολής fft() και για τα τρία σήματα (αρχικό και δύο φιλτραρισμένα, δηλαδή echoed και reverbed) σχεδιάζουμε το μέτρο του φάσματος τους σε λογαριθμική κλίμακα (20log10(| · |)) (dB) (με την εντολή mag2db(), σε κοινό γράφημα):



multiple piano notes frequency spectrum
original piano note
amplitude
scaled frequency
40
30
20
10
0
-10
-20
-30
-40
-50
-60
0
2000
4000
6000
8000
10000
12000
14000
16000



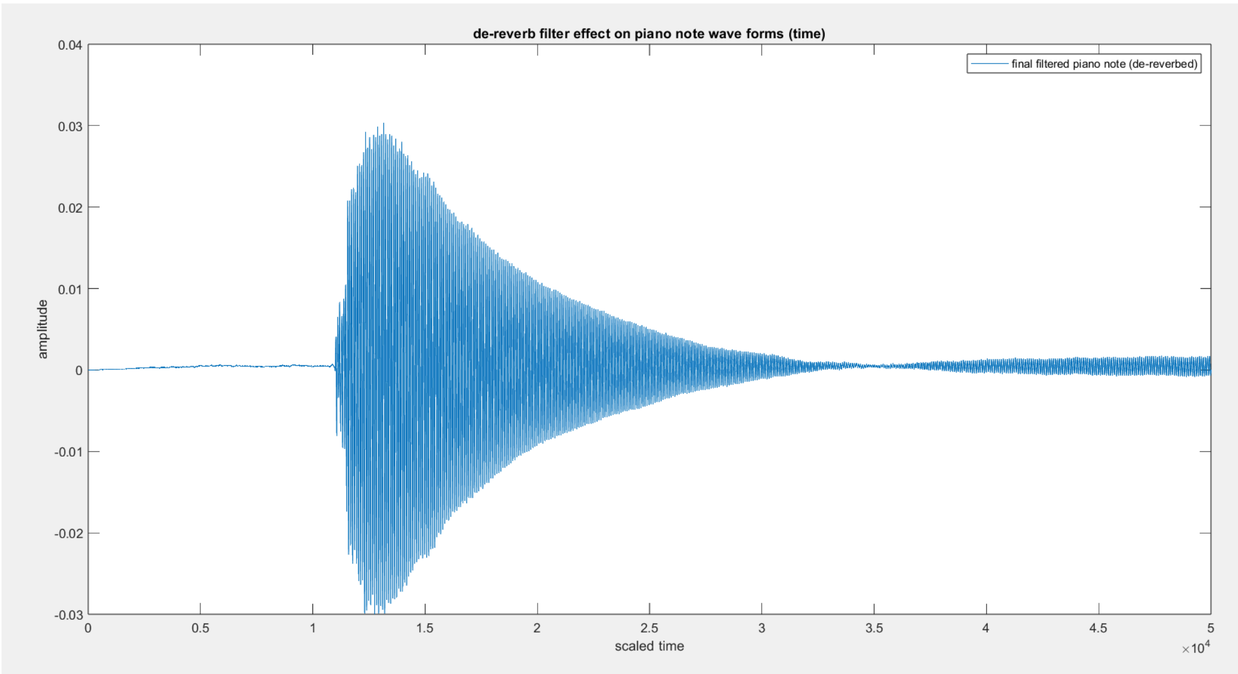
multiple piano notes frequency spectrum
original piano note
echoed piano note
amplitude
scaled frequency
40
30
20
10
0
-10
-20
-30
-40
-50
-60
0
2000
4000
6000
8000
10000
12000
14000
16000

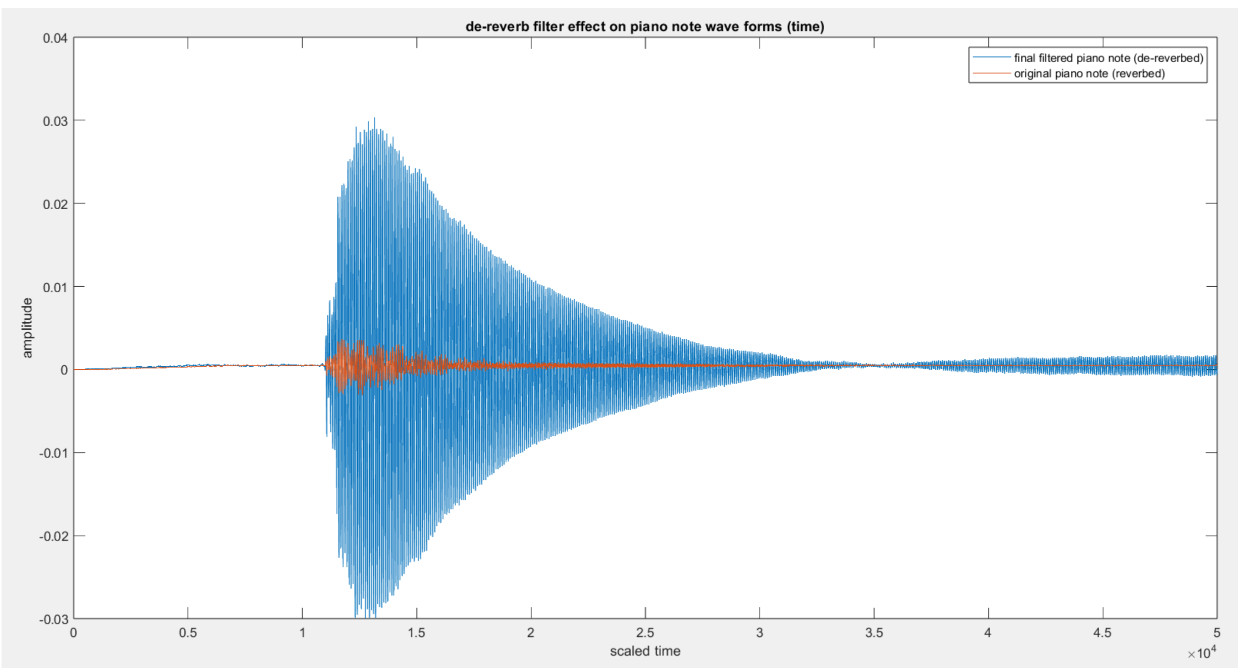


Αρχικά παρατηρούμε πως το αρχικό σήμα έχει πιο προφανή ,απότομα και μοναδικά μέγιστα, έτσι καταλαβαίνουμε καλύτερα τις συχνότητες, μετά το echoed είναι αρκετά παρόμοιο με το αρχικό αφου επαναλαμβάνει το σήμα μία ακόμα φόρα μετά από ένα χρονικό διάστημα, όμως παρατηρούμε τα μεγαλύτερα μέγιστα να μειώνουν το πλάτος αφου η κεντρική συχνότητα επικρατεί σε μικρότερο βαθμό, και το φάσμα γίνεται ελάχιστα πιο ομοιόμορφο εφόσον έχουμε ενισχυτική και καταστροφική συμβολή κυμάτων που "διασπείρουν" τις είδη υπάρχουσες συχνότητες. Όλα αυτά τα φαινόμενα ενισχύονται ακόμη περισσότερο στο reverbed φίλτρο,αφού το σήμα επαναλαμβάνεται 12 φορές, όπου βλέπουμε μια ακόμη πιο ομοιόμορφη κατανομή με τα μέγιστα να έχουν μειωθεί ακόμα παραπάνω. Μια τελική παρατήρηση είναι ότι γενικά η μορφή του φάσματος και στις τρεις περιπτώσεις είναι παρόμοια (εχουν ιδια μέγιστα/επικρατούσες συχνότητες), που είναι αναμενόμενο αφού τα φίλτρα που φτίαξαμε απλα επαναλαμβάνουν/προσθέτουν το ίδιο σήμα (με καθυστέρηση) άρα οι συχνότητες που υπάρχουν στο σήμα παραμένουν ίδιες.

(δ)Τώρα αποθηκεύουμε τα δύο φιλτραρισμένα σήματα σε δύο αρχεία τύπου .wav, με χρήση της εντολής audiowrite(), έτσι φτιάχνουμε ένα αρχείο echoed.wav για το echoed σήμα και ένα αρχείο reverbed.wav για το reverbed σήμα.

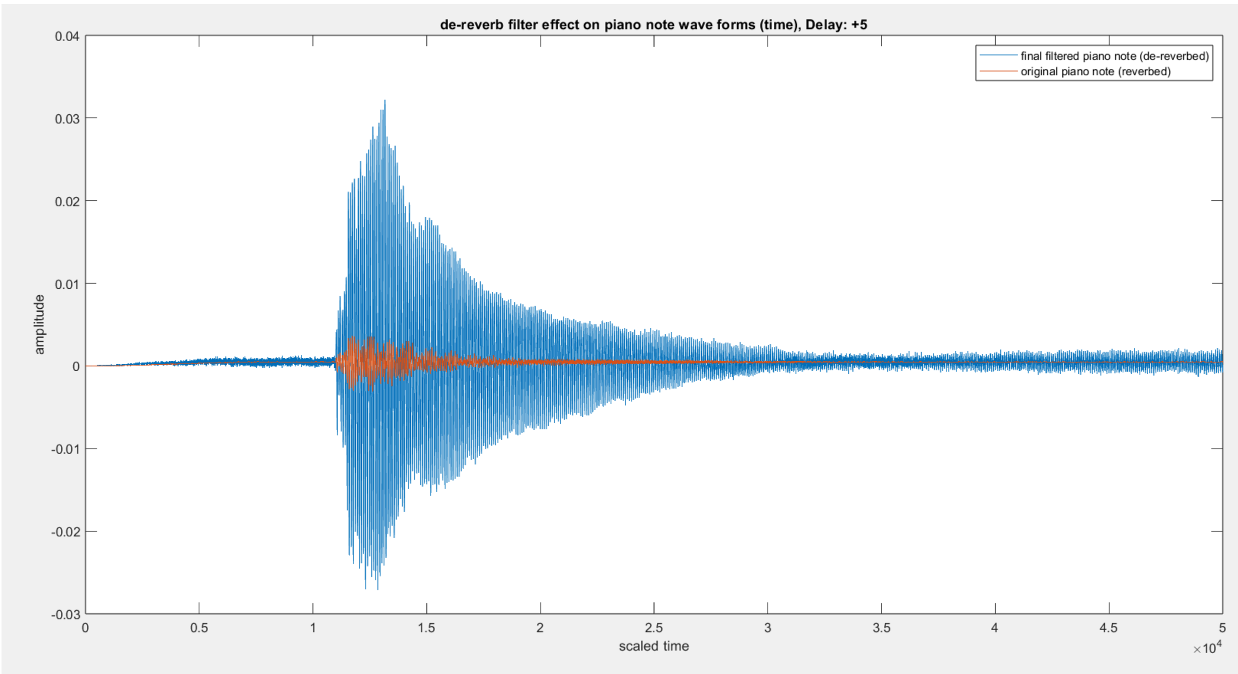
(ε) Με βάση το ερώτημα 1.1 στ), υπολογίζουμε τις παραμέτρους ενός κατάλληλου φίλτρου με στόχο την απαλοιφή της αντήχησης, όπου για να το υλοποιήσουμε απλά θέτουμε τους συντελεστές του αριθμητή του reverbed φίλτρου ως συντελεστες του παρονομαστή του dereverbed φίλτρου και αντίστοιχα για τους συντελεστές του αριθμητή, και σχεδιάζουμε με την plot() το προκύπτον σήμα, μαζί με το (reverbed) σήμα του ερωτήματος (β):

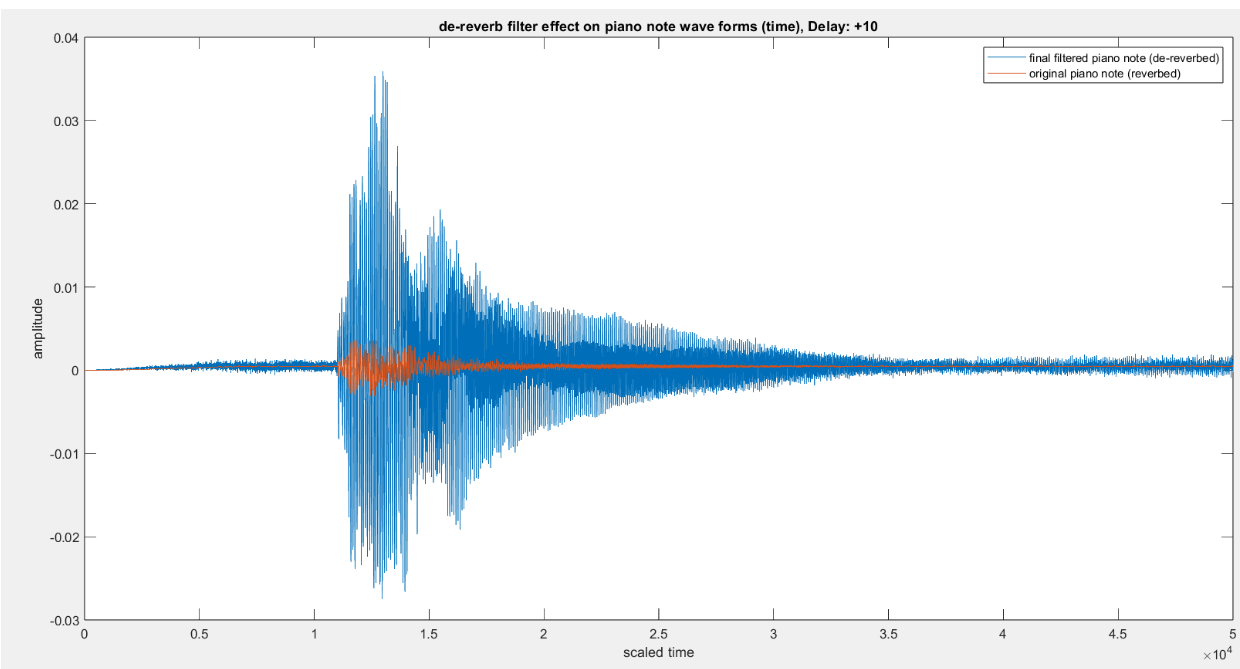


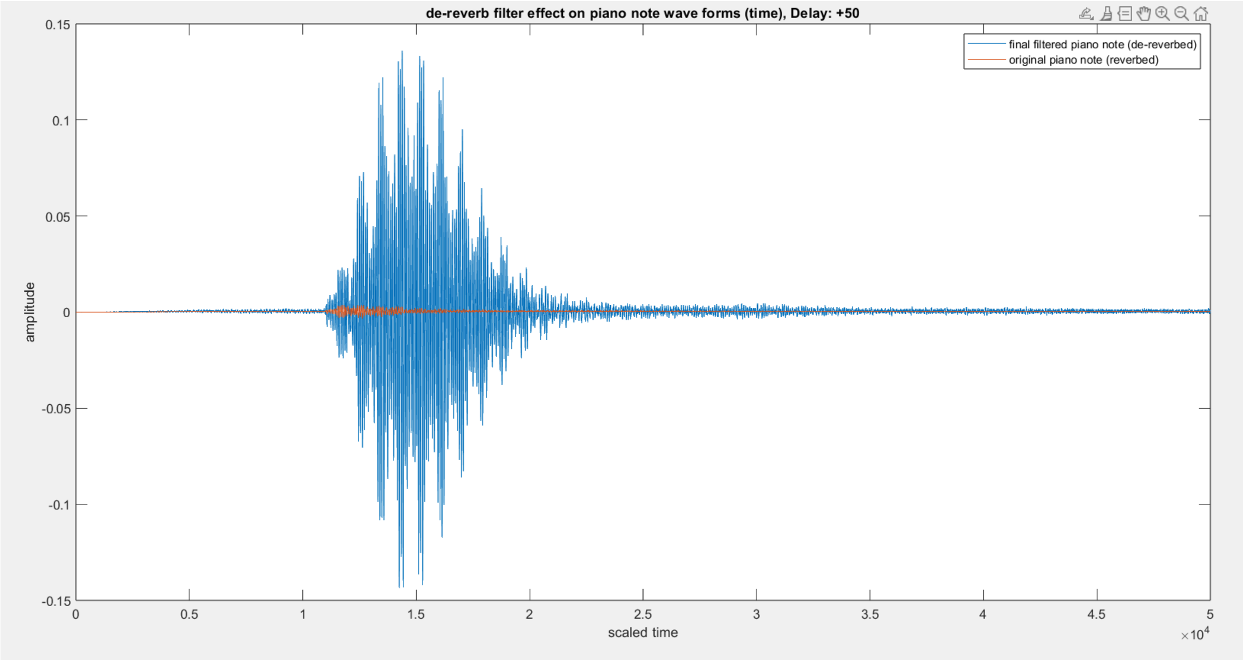


Με αυτή τη διαδικασία παρατηρούμε πως όντως το σήμα επιστρέφει στο αρχικό δηλαδή το νέο φίλτρο που σχεδιάσαμε όντως κάνε de-reverb και εξαλείφει την επιρροή του reverb φίλτρου στο σήμα, αρα λειτουργεί ως αντίστροφο φίλτρο απ’το reverb.

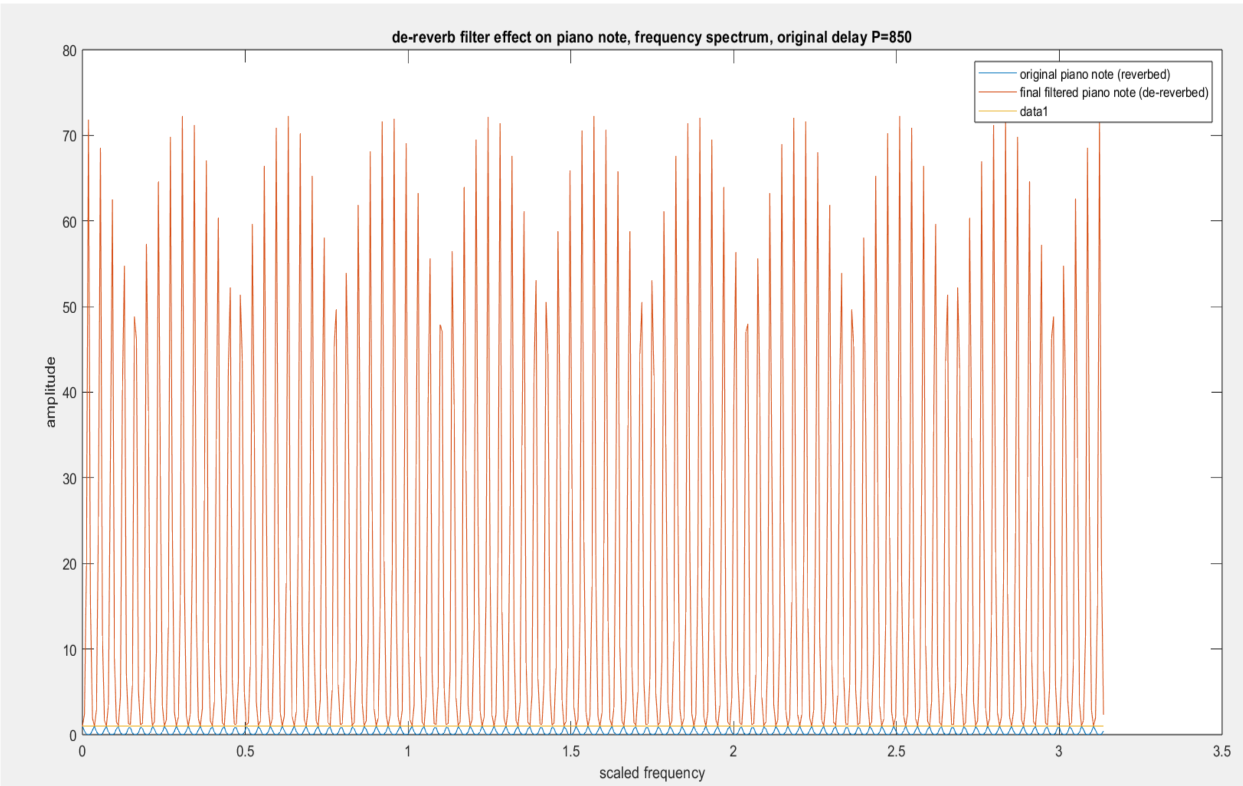
(στ) Τροποποιούμε την παράμετρο καθυστέρησης του φίλτρου που σχεδιάσαμε στο ερώτημα (ε), ώστε να αντιστοιχεί σε καθυστέρηση 5, 10 και 50 δειγμάτων μεγαλύτερη του P που μας δόθηκε στο ερώτημα (β), και οπτικοποιούμε εκ νέου τα dereverbed σήματα μαζί με το αρχικό (το αρχικο δε το μεταβάλλουμε, δηλαδή στο reverbed του (β) εφαρμόζουμε τα νέα φίλτρα).Σχεδιάζουμε τα νέα σήματα (de-reverbed) μαζί με το αρχικό (reverbed):

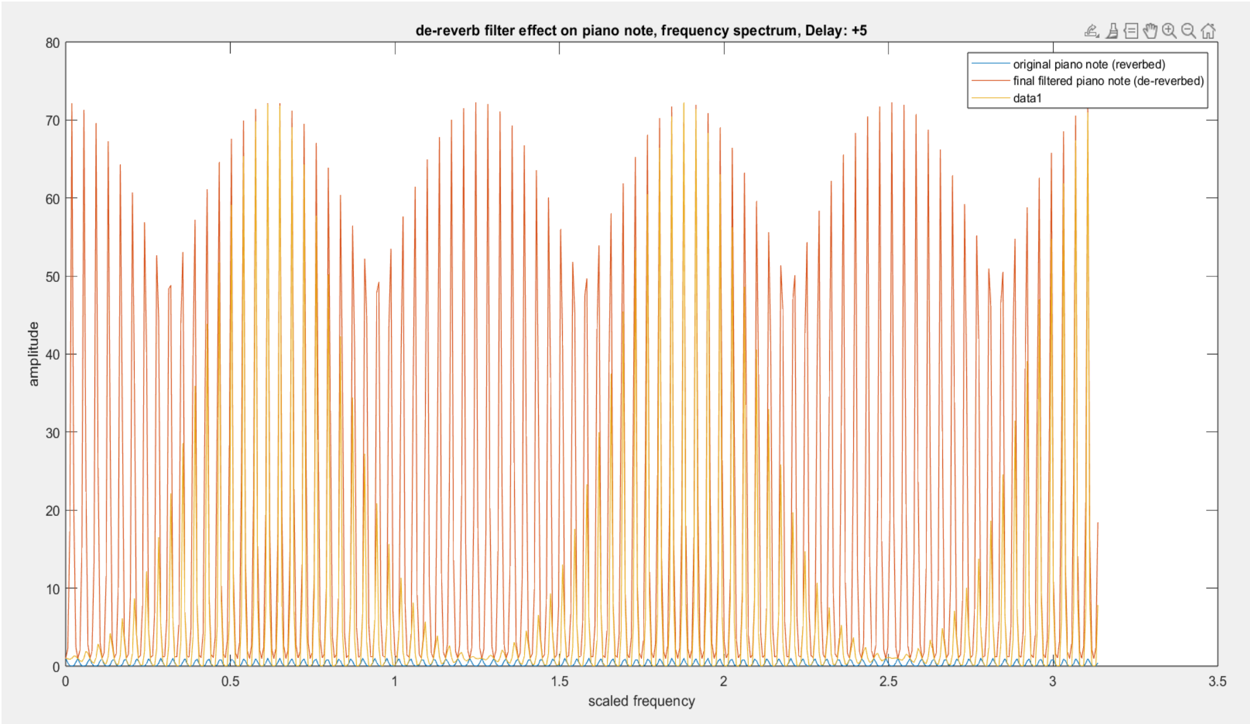


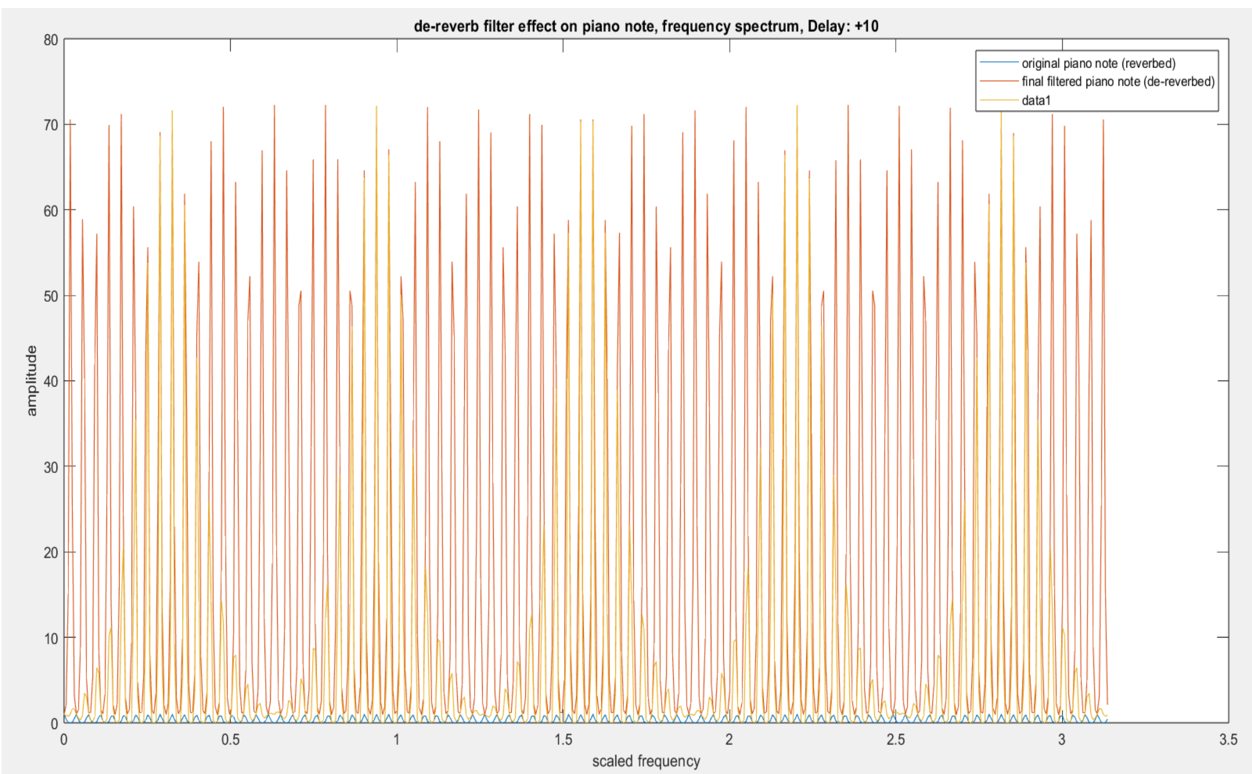


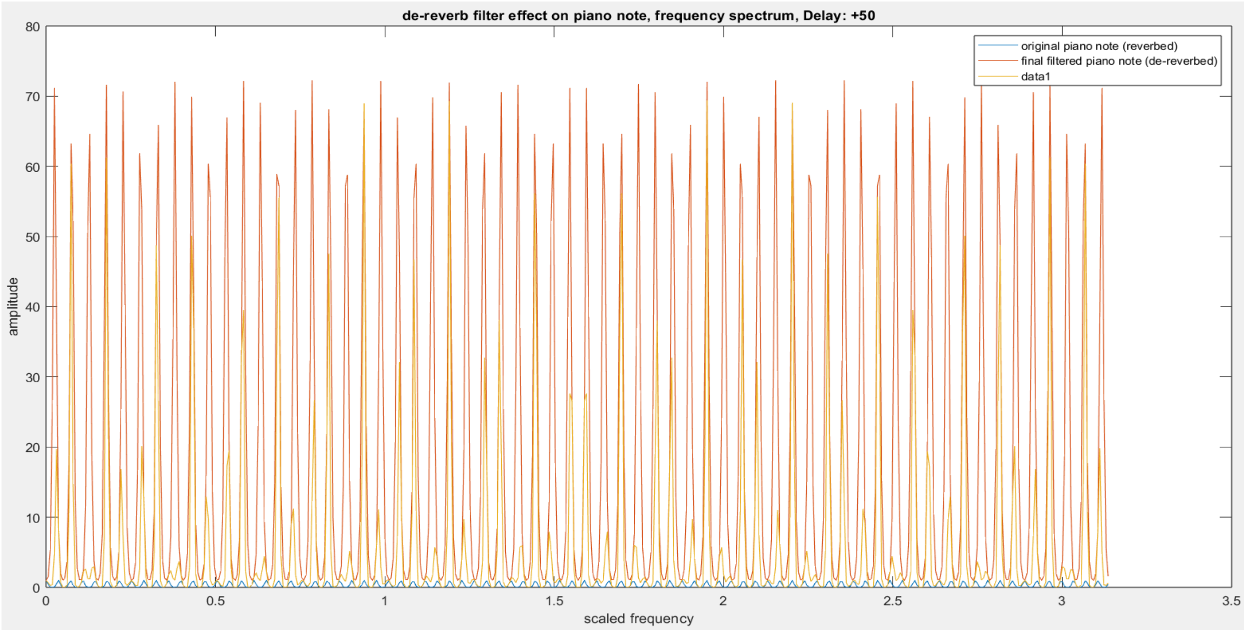


Τώρα για να μας βοηθήσει για τις παρατηρήσεις σχεδιάζουμε και τις αποκρίσεις πλάτους του φάσματος του συνολικού συστήματος:




de-reverb filter effect on piano note, frequency spectrum, Delay: +5
original piano note (reverbed)
final filtered piano note (de-reverbed)
data1
amplitude
scaled frequency
0
10
20
30
40
50
60
70
80
0.5
1
1.5
2
2.5
3
3.5


de-reverb filter effect on piano note, frequency spectrum, Delay: +10
original piano note (reverbed)
final filtered piano note (de-reverbed)
data1
amplitude
scaled frequency
0
10
20
30
40
50
60
70
80
0.5
1
1.5
2
2.5
3
3.5



Τα data1 είναι ο πολ/σμος φασμα(reverb)*φασμα(dereverb) (εσωτερικο γινόμενο δύο διανυσμάτων).
Γενικά παρατηρούμε πως η συχνότητα του φάσματος διατηρειται, και στις τρεις περιπτώσεις, που βγάζει νόημα αφού όπως είπαμε και πρίν τα φίλτρα που φτιάχνουμε απλά επαναλαμβάνουν το ίδιο σήμα (σε συγκεκριμένες καθυστερήσεις).Όμως καθώς αυξάνεται η καθυστέρηση του dereverb φίλτρου τα σημεία του φάσματος στα οποία δεν γίνεται σωστά η απαλοιφή του reverb φίλτρου αυξάνονται, αυτο το καταλαβαίνουμε από το data1 που δίνει το γινόμενο μεταξύ των δύο φασμάτων, οπου στην ιδανικη περιπτωση (χωρίς delay) είναι data1=1=σταθ. Ενώ έπειτα βλέπουμε να δημιουργούνται μέγιστα της τιμης με data1>>1 αυτο δείχνει τη διαφορά του αρχικού σήματος από το τελικό dereverbed σήμα.Αυτες οι διαφορές είναι περιοδική συνάρτηση (στο πεδίο της συχνότητας) με όλο και ελαττώμενη περίοδο ,που σημαίνει “εμφανίζονται” πιο συχνά καθώς αυξάνεται η καθυστέρηση του dereverbed φίλτρου (+5 -> +10 -> +50) (ετσι απέχει περισσότερο απ΄το ιδανικο dereverb φίλτρο).
Τελος απο τα σχεδιασμένα γραφήματα καταλαβαίνουμε πως η πληροφορία της συχνότητας διατηρείται όμως η πληροφορία του πλάτους του σήματος χάνεται γι αυτό όσο αυξάνεται η καθυστέρηση ,το σημα απο συνεχώς ελαττωμένο ημιτονοειδες (αρχικη περίπτωση) πάει να γίνει ημιτονοειδες με συνεχώς μεταβαλλόμενο πλάτος, το οποίο έχει ακόμα μεγαλύτερα μέγιστα και ακόμα μικρότερα ελάχιστα, που εντέλη πάει να γίνει διακριτό λόγω της περιοδικότητας του σφάλματος που αναφέραμε προηγουμένως στο πεδίο της συχνότητας, δηλαδή βλέπουμε το σήμα να αλλάζει πλάτος με τη μορφή περιοδικών παλμών οι οποιοι έχουν πολύ χαμηλά ελάχιστα και πολύ μεγάλα μέγιστα.