

Σήματα-Συστήματα Εργασία Matlab (2021-2022)

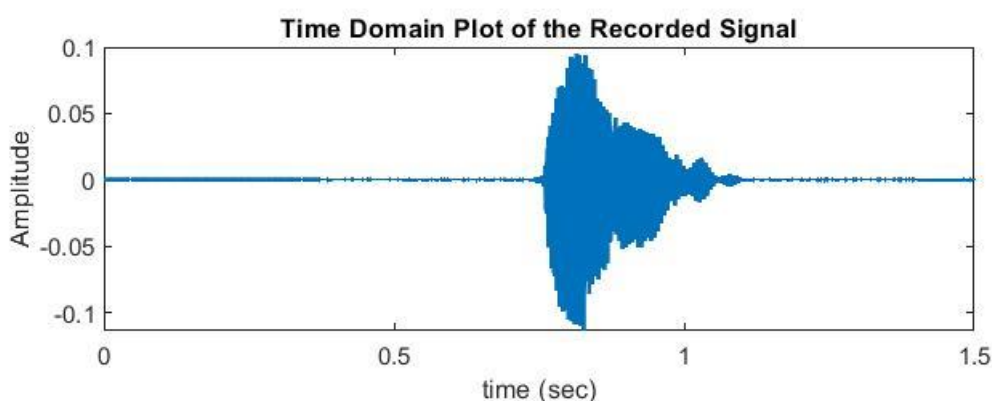
Ονοματεπώνυμο: Αλεξοπούλου Γεωργία

AM: 03120164

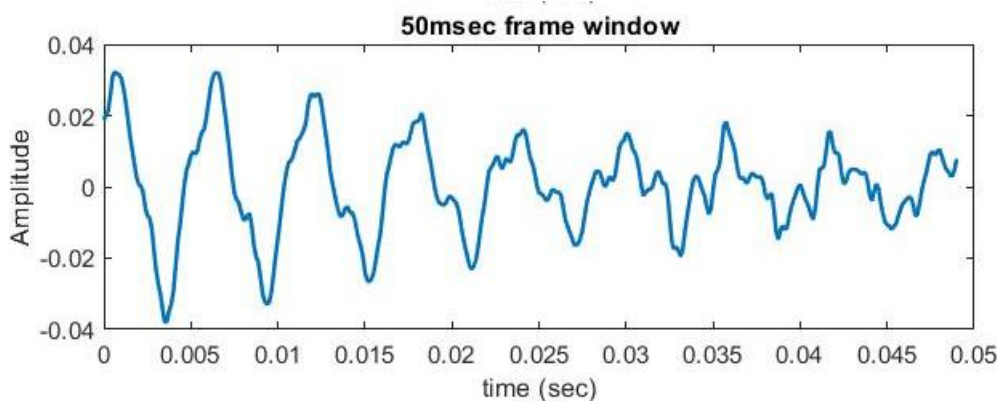
1. Ανάλυση Σήματος Φωνής στα Πεδία Χρόνου και Συχνότητας

α) Αρχικά, χρησιμοποιούμε τις εντολές `audiorecord` και `recordblocking`, προκειμένου να ηχογραφήσουμε το όνομα «Τζίνα». Έπειτα, με την εντολή `getaudiodata` δημιουργείται το αντίστοιχο array. Η διάρκεια της ηχογράφησης είναι 1.5 δευτερόλεπτα, επομένως το array περιέχει 12000 δείγματα. Αυτό συμβαίνει διότι η συχνότητα δειγματοληψίας ισούται με 8000 Hz, άρα κάθε ηχογραφούμενο array περιλαμβάνει 8000 δείγματα ανά δευτερόλεπτο.

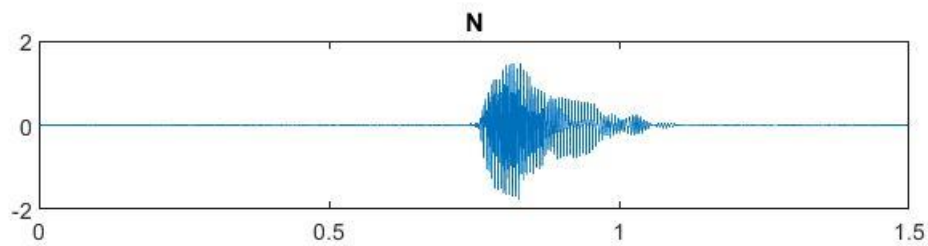
β) Αξιοποιώντας τις εντολές `plot` και `subplot`, εμφανίζουμε το σήμα του ηχογραφημένου αρχείου:



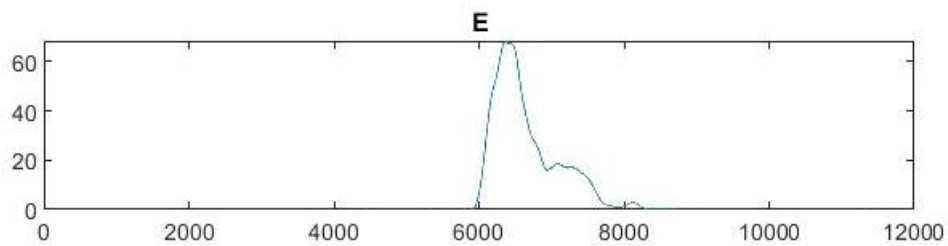
Εν συνεχεία, προκειμένου να έχουμε περίπου περιοδικό σήμα, επιλέγονται 400 διαδοχικά δείγματα του σήματος, τα οποία ταυτίζονται με τα δείγματα 951-1000. Όπως φαίνεται παρακάτω, το σχηματικό μας διαρκεί 0.05 sec (δηλαδή 50ms), όπως ήταν και το ζητούμενο. Ακούγοντας το παρακάτω αρχείο, μέσω της εντολής `sound`, παρατηρούμε πως αντιστοιχεί στη συλλαβή «να», με το γράμμα α να ακούγεται ελάχιστα.



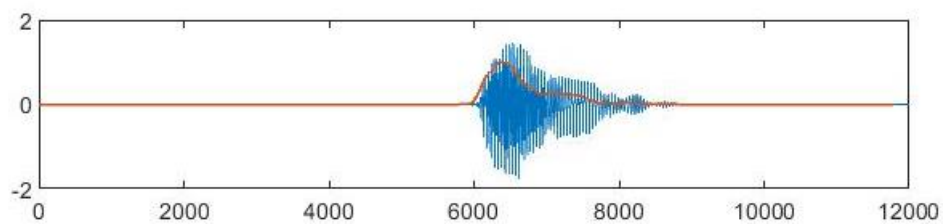
γ) Κανονικοποιούμε το σήμα στο ζητούμενο διάστημα, μέσω της εντολής `normalize` και το απεικονίζουμε, αξιοποιώντας τις εντολές `plot`, `subplot`:



Στη συνέχεια, υπολογίζουμε και απεικονίζουμε την ενέργεια του κανονικοποιημένου σήματος, με τη χρήση των εντολών `conv`, `plot` και `subplot`.



Όμοια λαμβάνουμε και την απεικόνιση του σήματος $w[n]$.

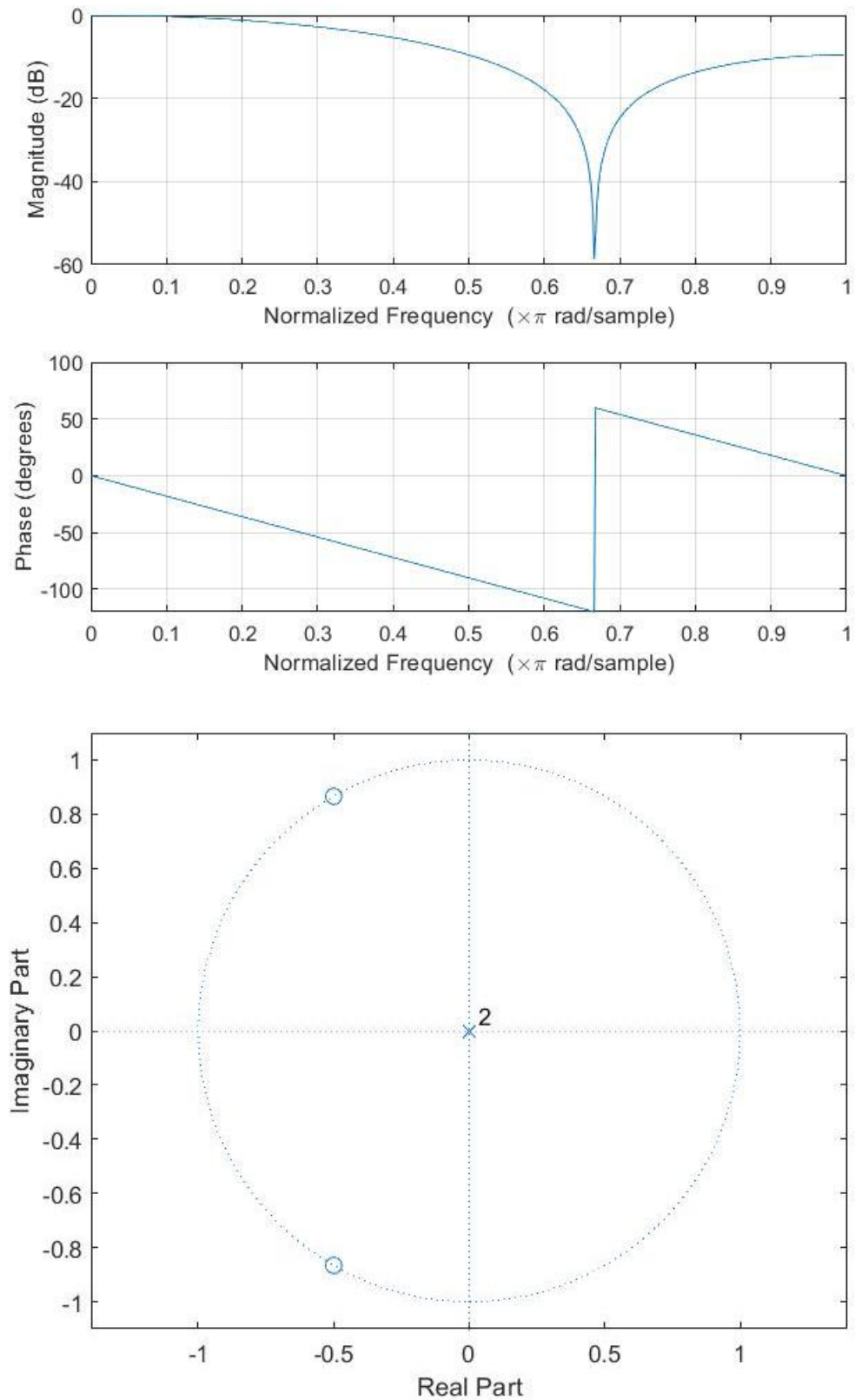


δ) Χρησιμοποιούμε την εντολή `fft` προκειμένου να υπολογίσουμε τον διακριτό μετασχηματισμό Fourier του σήματος διάρκειας 50ms του ερωτήματος **(β)**. Σύμφωνα με την εκφώνηση, ο DFT υπολογίστηκε για $N = 1024$ δείγματα. Παρακάτω έχουμε την απεικόνιση του μετασχηματισμού, τόσο σε κανονική, όσο και σε λογαριθμική κλίμακα, για το εύρος συχνοτήτων 0-8000 Hz.

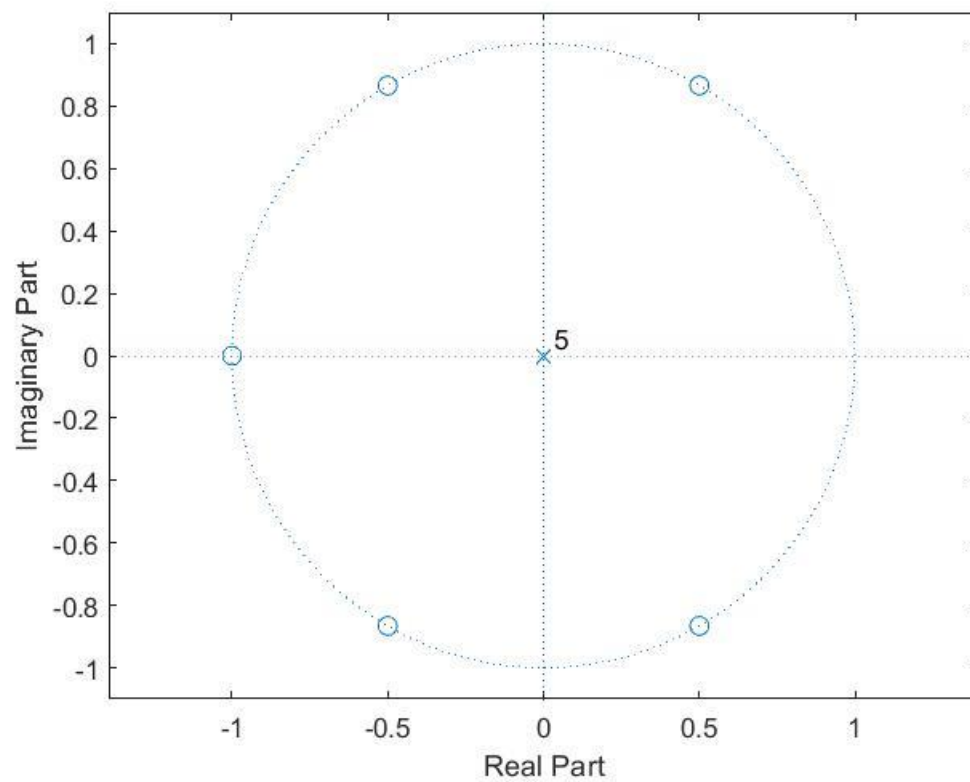
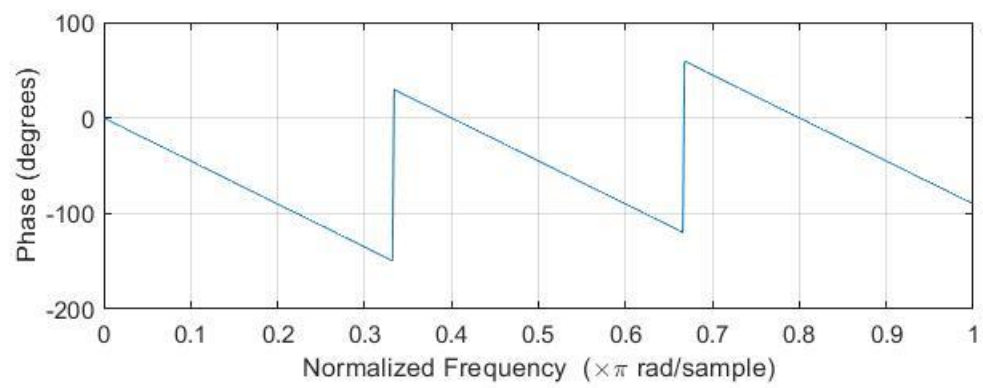
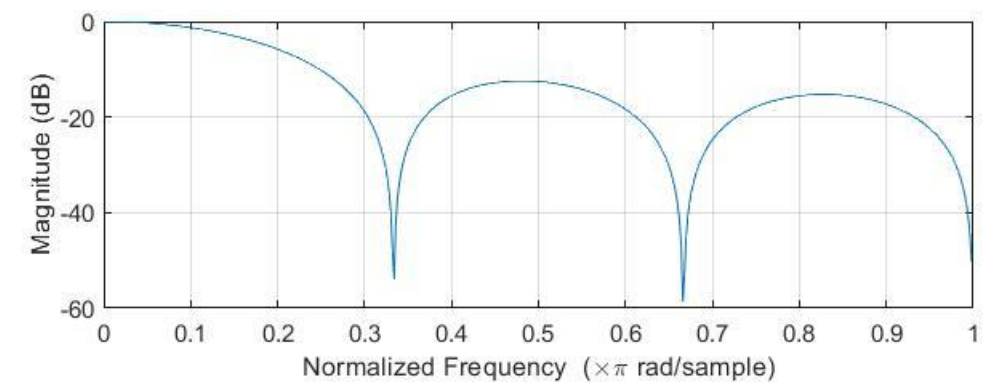
- [illegible]

Για τους παραπάνω όρους σχεδιάζουμε τόσο την απόκριση πλάτους και φάσης, όσο και το διάγραμμα πόλων και μηδενικών, αξιοποιώντας τις εντολές `freqz`, `tzpz` και `zplane` αντίστοιχα.

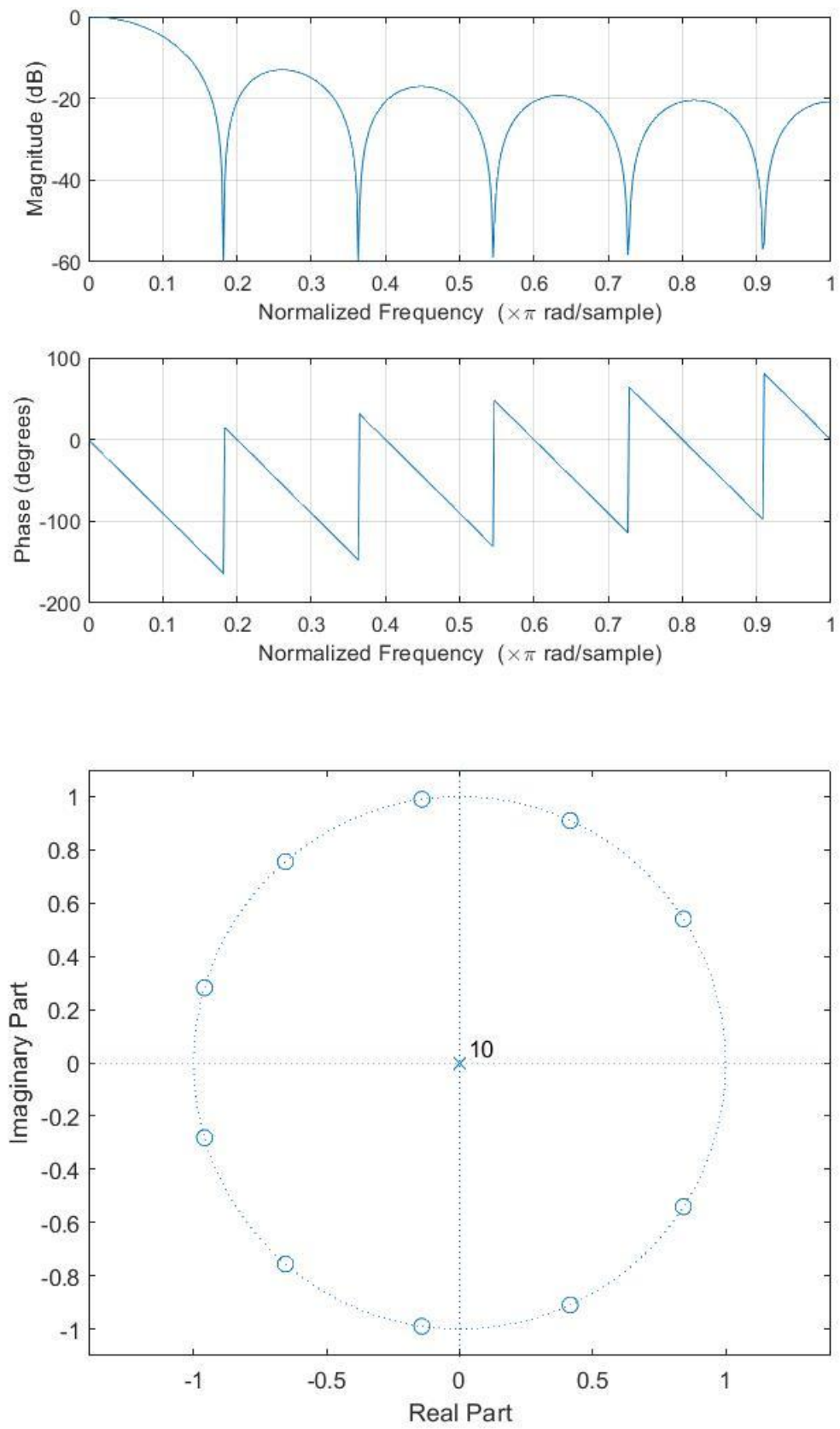
- $N=2$:



- N=5:



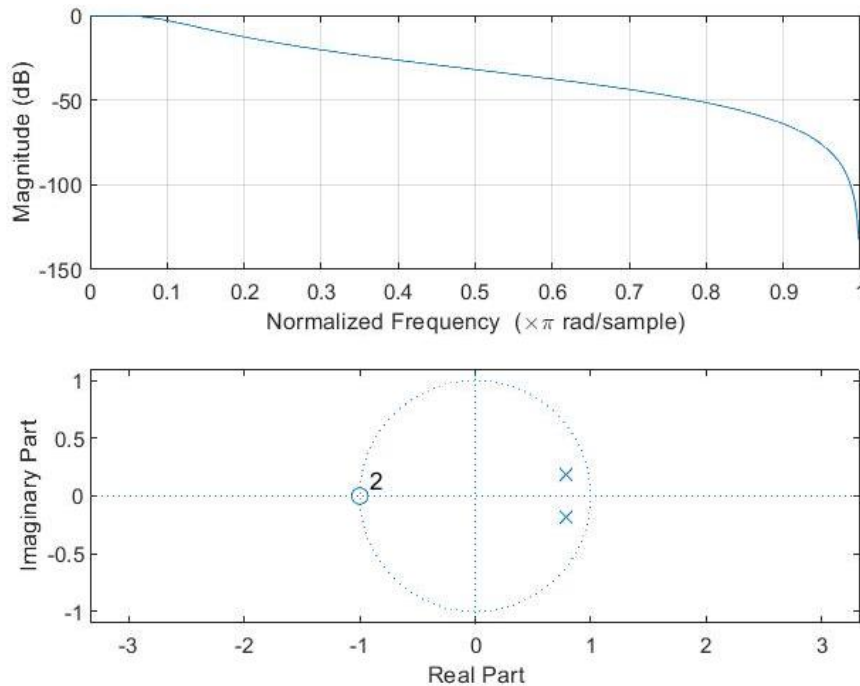
- N=10:



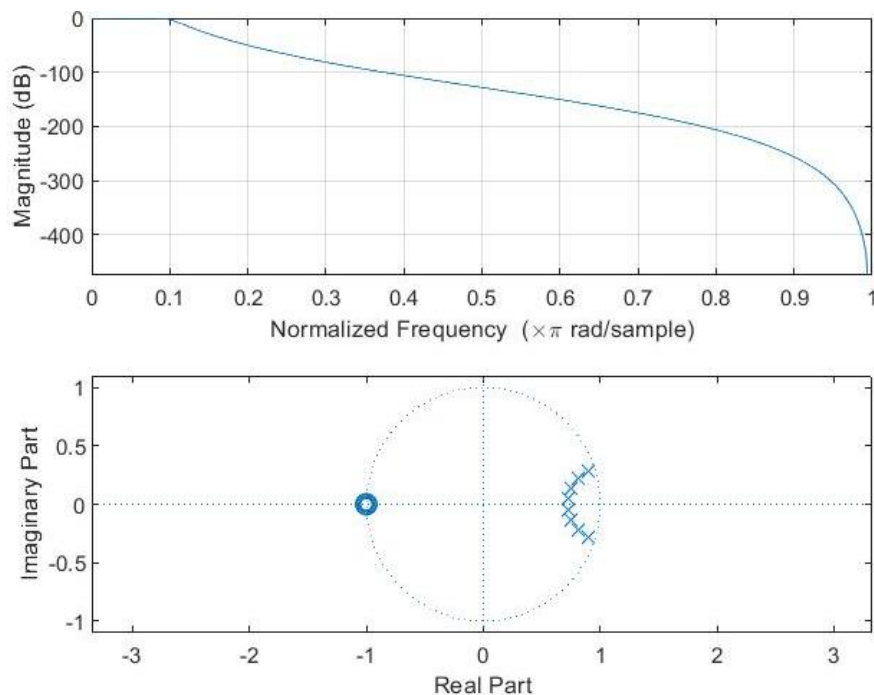
Παρατηρούμε πως, όσο αυξάνεται το N , λαμβάνουμε περισσότερα δείγματα απόκρισης πλάτους και φάσης στο ίδιο εύρος συχνοτήτων, ενώ το πλήθος των μηδενικών ισούται κάθε φορά με N . Όσον αφορά τον μοναδικό πόλο που εντοπίζεται κάθε φορά, βρίσκεται στο σημείο $(N, 0)$.

δ) Για τάξη φίλτρου $n = 2$ και $n = 8$, ορίζουμε τα διανύσματα $A1/A2$ και $B1/B2$ αντίστοιχα, χρησιμοποιώντας την εντολή `butter`. Στη συνέχεια, απεικονίζουμε, ξανά, την απόκριση πλάτους και φάσης, αλλά και τα διαγράμματα πόλων και μηδενικών για την εκάστοτε τάξη του βαθυπερατού φίλτρου Butterworth.

- $n=2$:



- $n=8$:

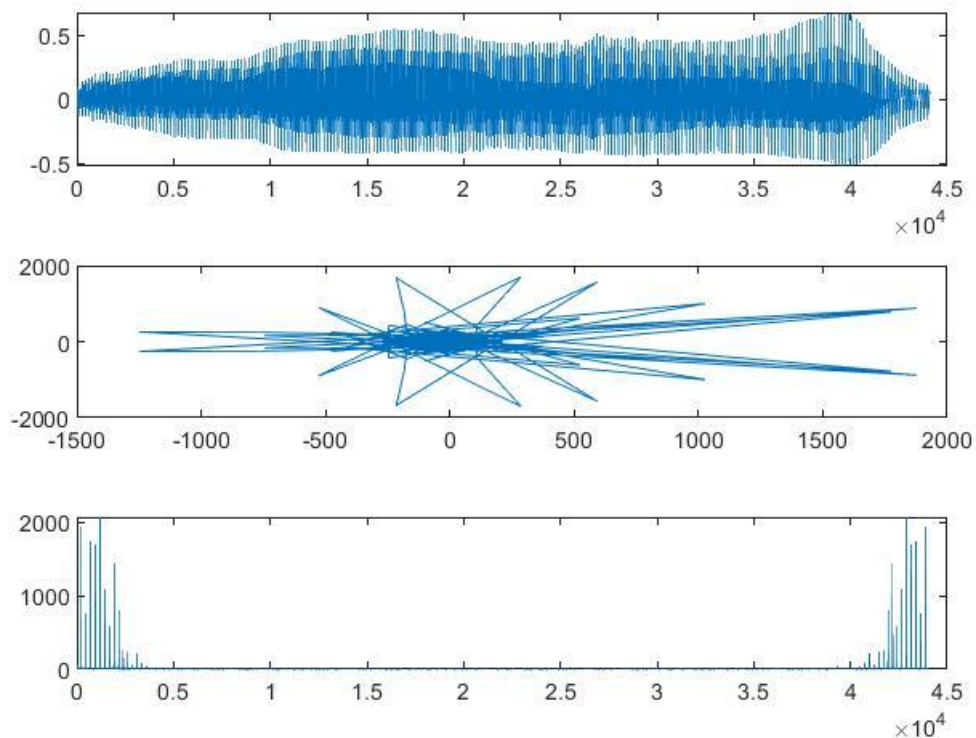


Παρατηρούμε, και πάλι, πως αυξάνοντας την τάξη του φίλτρου αυξάνονται και τα δείγματα απόκρισης πλάτους και φάσης. Αντίθετα με το ερώτημα (γ), ωστόσο, για την εκάστοτε τάξη του φίλτρου, το διάγραμμα μηδενικών και πόλων του περιέχει ένα μηδενικό σημείο (περισσότερο συγκεντρωμένο, όσο αυξάνεται η τάξη n) και n πλήθος πόλων.

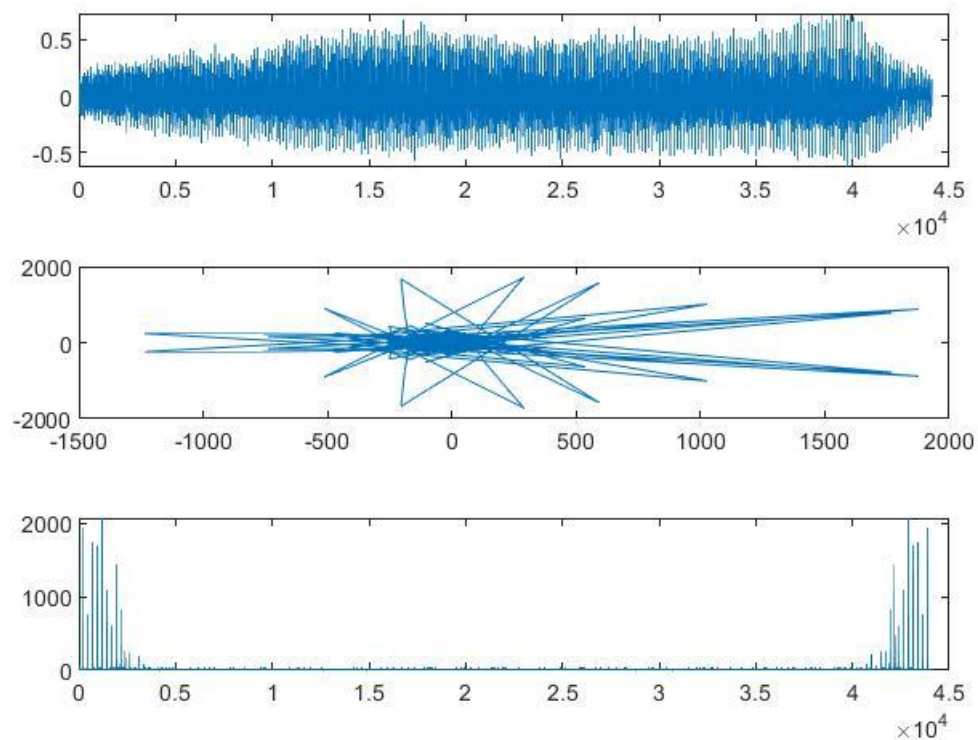
ε) Συγκρίνοντας τα διαγράμματα του φίλτρου κινούμενου μέσου τάξης 10 και του βαθυπερατού φίλτρου Butterworth τάξης 8, παρατηρούμε αρκετές διαφορές. Αρχικά, η απόκριση πλάτους του πρώτου έχει σχήμα που παραπέμπει στο άνω μισό ημιτονοειδούς παλμού και περιέχει σημαντικά περισσότερα δείγματα, σε σχέση με το δεύτερο φίλτρο, το οποίο έχει τη μορφή αντίστροφης εκθετικής συνάρτησης. Επιπλέον, συγκρίνοντας το διάγραμμα πόλων-μηδενικών, παρατηρούμε πως το φίλτρο κινούμενου μέσου τάξης 10 διαθέτει έναν μόνο πόλο στο σημείο $(10, 0)$ και 10 μηδενικά σημεία, εξίσου συγκεντρωμένα μεταξύ τους, που βρίσκονται στην περιφέρεια του μοναδιαίου κύκλου. Αντίθετα, το φίλτρο Butterworth τάξης 8 περιέχει ένα μόνο μηδενικό σημείο, περισσότερο συγκεντρωμένο, το οποίο βρίσκεται στο $(0, -1)$, δηλαδή στην περιφέρεια του μοναδιαίου κύκλου, και 8 πόλους εντός του μοναδιαίου κύκλου.

2.2 Εφαρμογή Φίλτρων σε Μουσικά Σήματα

α) Φορτώνουμε στο Matlab το αρχείο `cello_note.wav` και το ακούμε, μέσω των εντολών `audioread` και `sound` αντίστοιχα. Απεικονίζουμε το φάσμα του σήματος μέσω των εντολών `fft`, `plot`, `subplot` και το μέτρο του μέσω των εντολών `plot(abs(y))`, `subplot`.

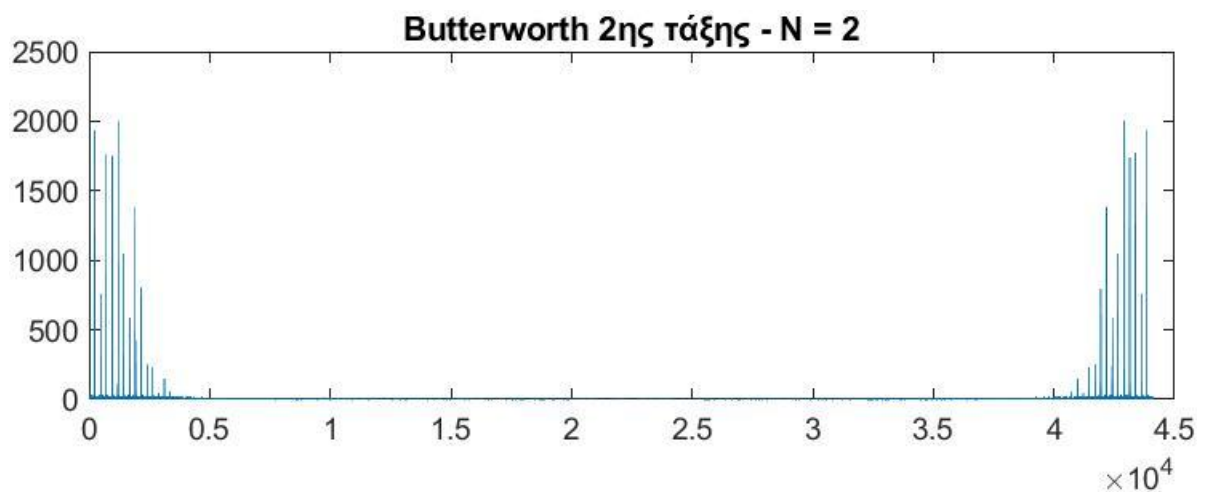


β) Με αντίστοιχο τρόπο, φορτώνουμε στο Matlab και ακούμε το αρχείο `cello_note_noisy.wav`, το οποίο και απεικονίζουμε.

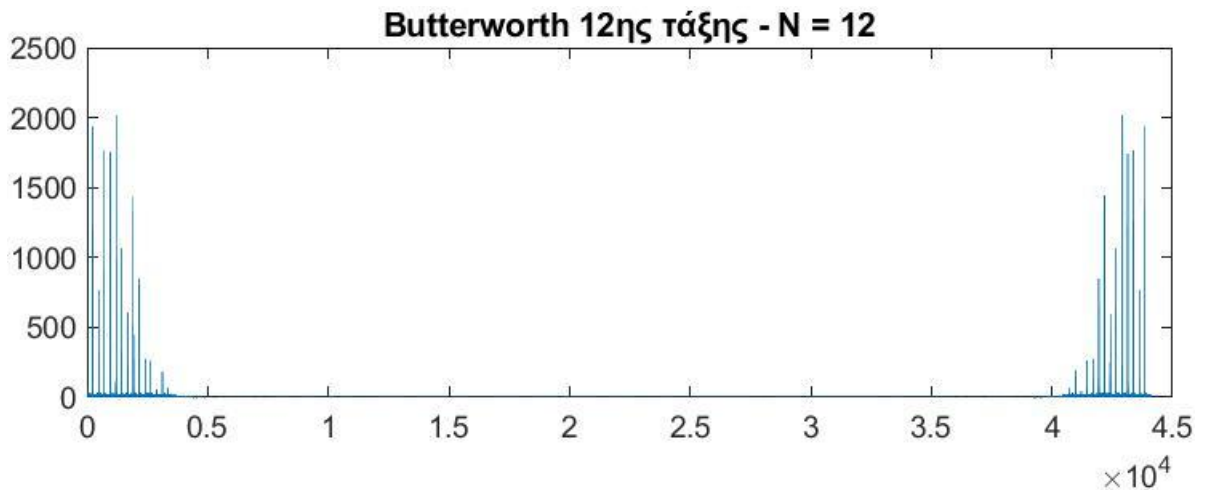


γ) Για την αποθορυβοποίηση του αρχείου cello_note_noisy.wav χρησιμοποιούμε τις εντολές butter(), abs(), fft() και plot(). Επιλέγουμε να εφαρμόσουμε το φίλτρο Butterworth τόσο για τάξη $N = 2$, όσο και για τάξη $N = 12$, προκειμένου να συγκρίνουμε τους δύο ήχους που προκύπτουν και ως προς τον αριθμό δειγμάτων απόκρισης πλάτους και φάσης. Έτσι, λαμβάνουμε τα εξής αποτελέσματα:

- $N = 2$:

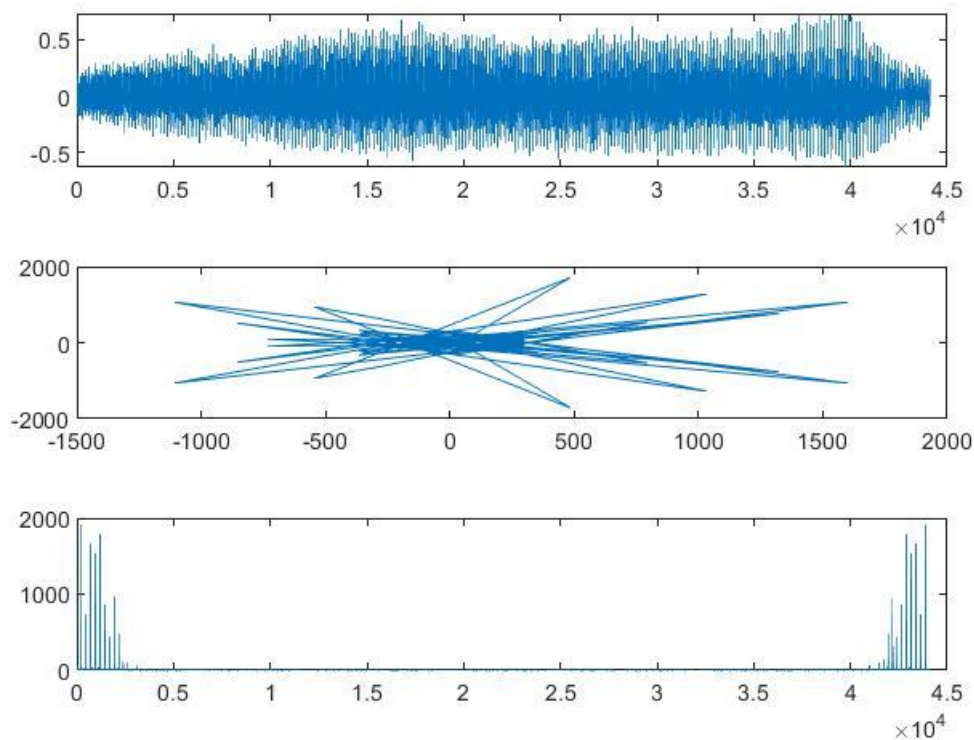


- $N = 12$:



Ακούγοντας τα δύο νέα αρχεία ήχου, παρατηρούμε πως κανένα από αυτά δεν είναι πανομοιότυπο με το αρχικό, ωστόσο το προσομοιάζουν ικανοποιητικά. Η απουσία θορύβου είναι αισθητή και στις δύο περιπτώσεις. Για $N = 2$, ο ήχος είναι πιο οξύς και δυνατός, ενώ για $N = 12$, ο ήχος είναι πιο υπόκωφος. Αυτό, άλλωστε, απεικονίζεται και στα διαγράμματα των μέτρων κάθε αποθορυβοποιημένου φίλτρου: για $N = 2$, το πλάτος μεγιστοποιείται σε συχνότητα μεγαλύτερη, σε σχέση με την αντίστοιχη συχνότητα για $N = 12$, κάτι που δικαιολογεί την αυξημένη οξύτητα του πρώτου.

δ) Για την αποθορυβοποίηση του θορυβώδους σήματος με τη χρήση φίλτρου κινούμενου μέσου, θα επιλέξουμε τάξη φίλτρου $N = 10$, προκειμένου να έχουμε επαρκή δείγματα απόκρισης πλάτους και φάσης, όπως αναλύσαμε στην άσκηση **2.1**. Για την αποθορυβοποίηση, αξιοποιούμε την εντολή `filter`. Στη συνέχεια, ακούμε διαδοχικά το αρχείο `cello_note_noisy.wav`, το αποθορυβοποιημένο σήμα και το αρχείο `cello_note.wav`. Παρατηρούμε πως τα δύο τελευταία είναι εξαιρετικά παρόμοια, με το δεύτερο να είναι ελάχιστα χαμηλότερο σε ένταση από το τρίτο (συνεπώς η επιλογή τάξης $N=10$ του φίλτρου είναι αρκετά πετυχημένη). Η διαφορά οξύτητας των δύο ήχων είναι αναμενόμενη και επεξηγείται μέσω της γραφικής παράστασης του μέτρου των δύο φίλτρων. Συγκρίνοντάς τες, βλέπουμε πως το πλάτος του μέτρου του αποθορυβοποιημένου φίλτρου μεγιστοποιείται σε συχνότητα μικρότερη, σε σχέση με αυτή του αρχικού φίλτρου (`cello_note.wav`), γι' αυτό και το δεύτερο ακούγεται περισσότερο οξύ από το πρώτο. Παρακάτω απεικονίζονται το φάσμα και το μέτρο του αποθορυβοποιημένου σήματος.



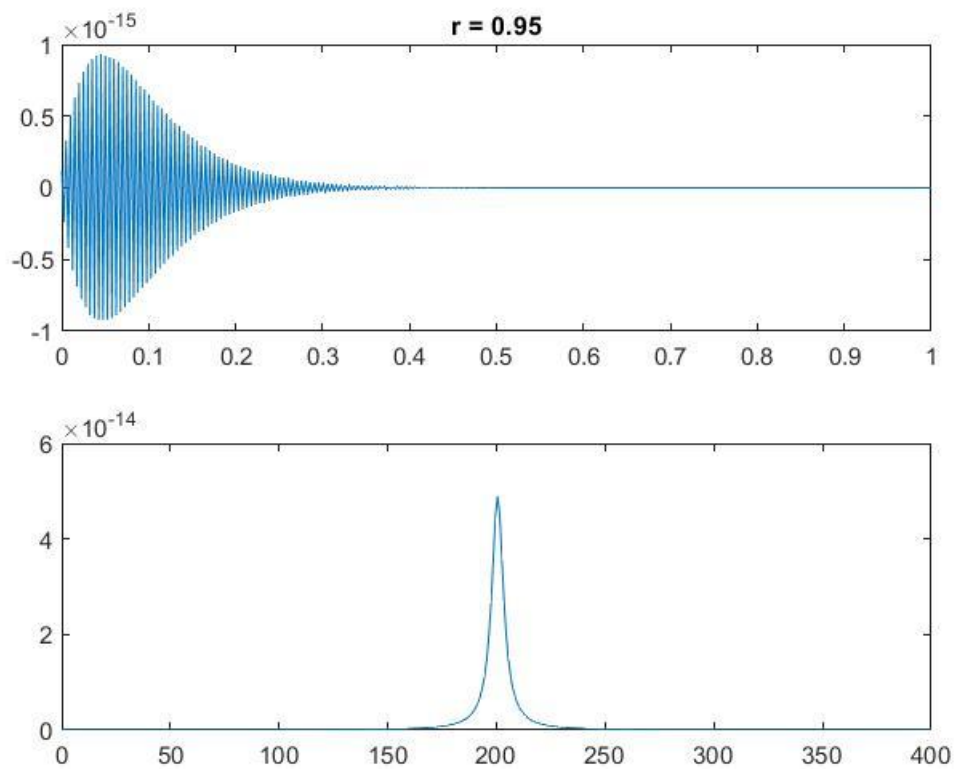
3. Υλοποίηση Ταλαντωτών για την Παραγωγή Φωνήματος

α) Η συνάρτηση του ταλαντωτή βρίσκεται, σύμφωνα με τις οδηγίες, σε ξεχωριστό αρχείο resonator.m. Η υλοποίηση επιτεύχθηκε με βάση τον τύπο της σελίδας 281 του βιβλίου:

$$H(z) = \frac{b_0 * z^2}{z^2 - 2r_0 * \cos(\Omega_0) * z + r_0^2}, \text{ με } \Omega_0 = 2\pi * \frac{F_r}{F_s}$$

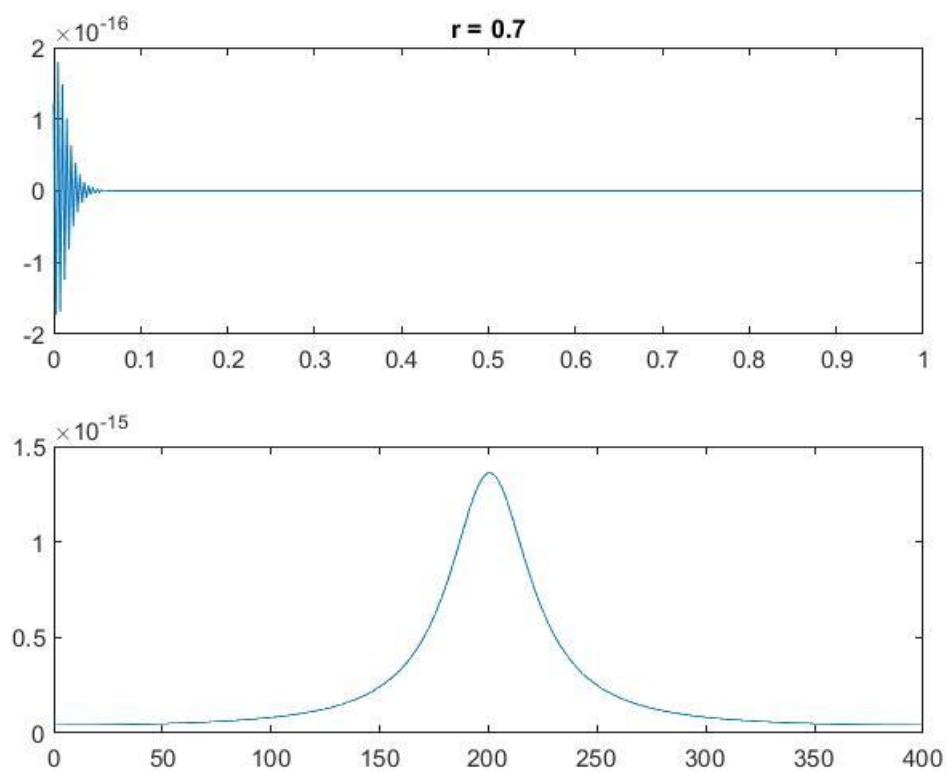
Ορίζουμε στο Matlab τη συχνότητα $W = \Omega_0$ σύμφωνα με τον παραπάνω τύπο, τον όρο b_0 ως $b_0 = \sin(\Omega_0)$. Στη συνέχεια, ορίζουμε τον αριθμητή num = [sin(W), 0, 0]. Αντίστοιχα, ορίζουμε τον παρονομαστή ως den = [1, 2r₀*cos(W), r₀²].

β) Επιλέγουμε τυχαίες συχνότητες $F_r = 200$ Hz και $F_s = 400$ Hz, άρα $\Omega_0 = \pi$. Για τιμή $r = 0.95$, λαμβάνουμε τα παρακάτω διαγράμματα κρουστικής απόκρισης και απόκρισης συχνότητας του συστήματος. Σύμφωνα με αυτά, είναι εύκολο να αποφανθούμε πως η κρουστικής απόκριση του συστήματος πρόκειται για φθίνουσα ταλάντωση, ενώ το πλάτος της εξόδου του συστήματος μεγιστοποιείται για συχνότητα ίση με 200 Hz (αυτό, άλλωστε, είναι αναμενόμενο, καθώς η ιδιοσυχνότητα ταλάντωσης ισούται με 200 Hz).



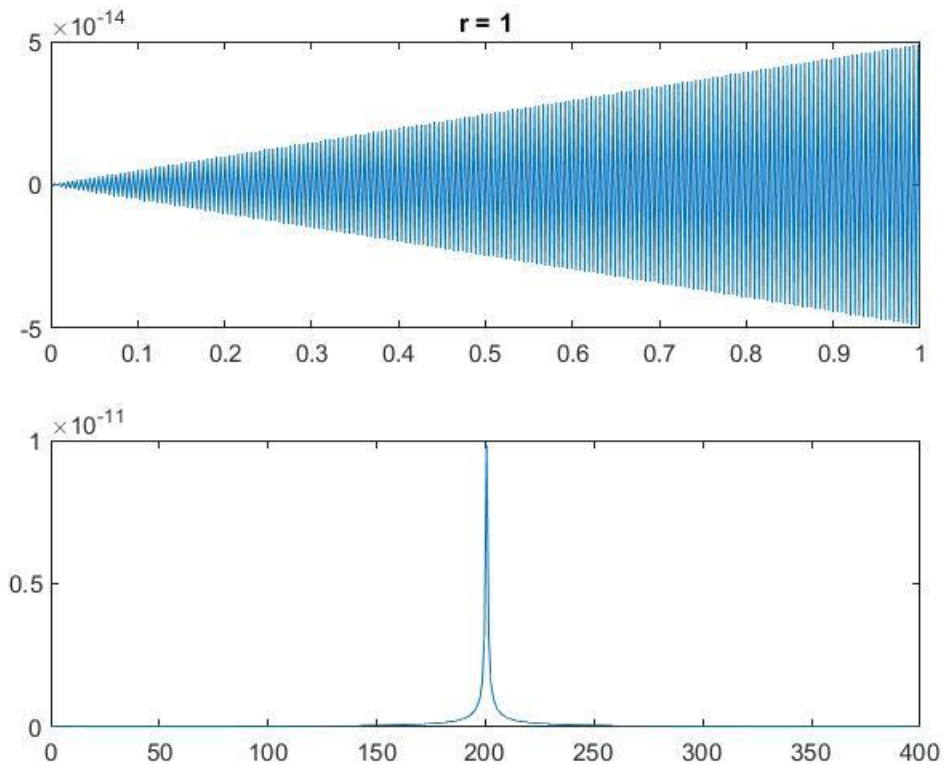
Ακολουθούμε την ίδια διαδικασία για διαφορετικές τιμές των r .

- Για $r = 0.7$:



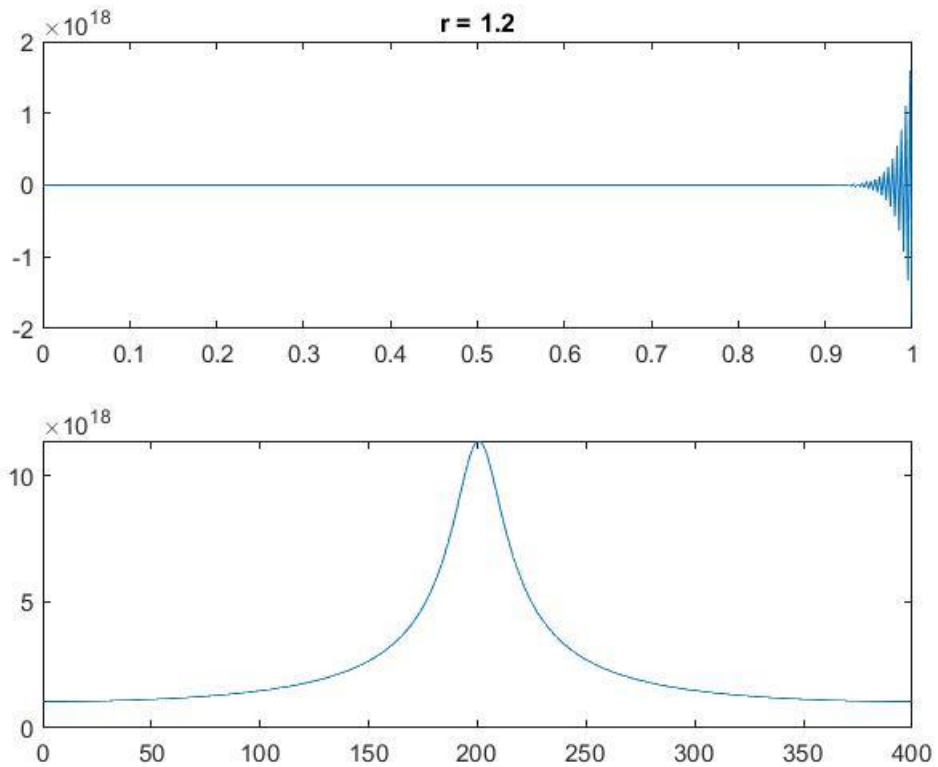
Παρατηρούμε πως η μορφή της κρουστικής απόκρισης δεν μεταβάλλεται (πρόκειται, και πάλι, για φθίνουσα ταλάντωση), ωστόσο μεταβάλλεται ο συντελεστής απόσβεσης της ταλάντωσης, καθώς παρατηρούμε πως μηδενίζεται γρηγορότερα, σε σχέση με παραπάνω. Όσον αφορά στην απόκριση συχνότητας, παρατηρούμε πως, ενώ προηγουμένως προσομοιάζε τριγωνικό παλμό, πλέον προσομοιάζει περισσότερο ημιτονοειδή παλμό. Το πλάτος της έχει υποστεί αρκετά μεγάλη μείωση, καθώς είναι λιγότερο από υποδεκαπλάσιο σε σχέση με προηγουμένως.

- Για $r = 1$:



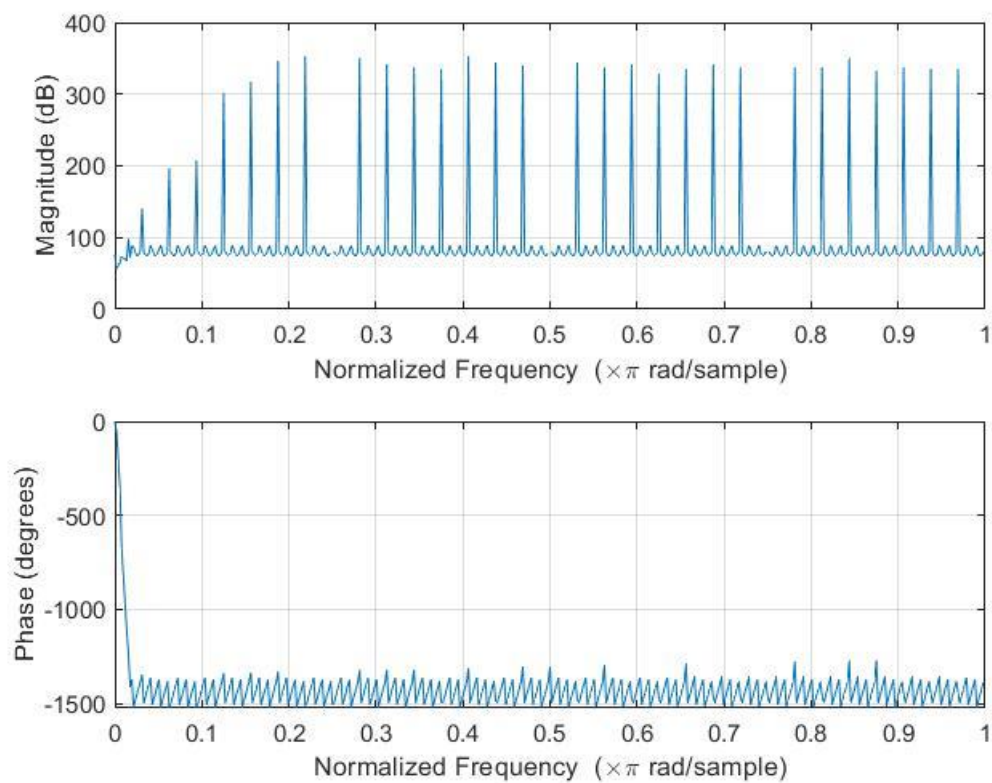
Καθώς θέτουμε $r = 1$, η μορφή της απόκρισης πλάτους αλλάζει, από φθίνουσα σε αύξουσα ταλάντωση, με αύξηση σταδιακή (γραμμική). Η απόκριση συχνότητας έχει κι αυτή υποστεί αλλαγές. Πιο συγκεκριμένα, το πλάτος της έχει αυξηθεί κατά πολύ, και από την τάξη των 10^{-14} (για $r = 9.5$), έχει μετατοπιστεί στην τάξη των 10^{-11} (για $r = 1$). Ταυτόχρονα, η ίδια η μορφή του γραφήματος είναι πολύ λιγότερο εξομαλυσμένη σε σχέση με την αρχική. Μάλιστα, το εύρος μεταβολής του πλάτους της απόκρισης συχνότητας είναι τόσο μικρό, ώστε για τη συχνότητα των 200 Hz το γράφημα προσομοιάζει ευθεία γραμμή.

- Για $r = 1.2$:

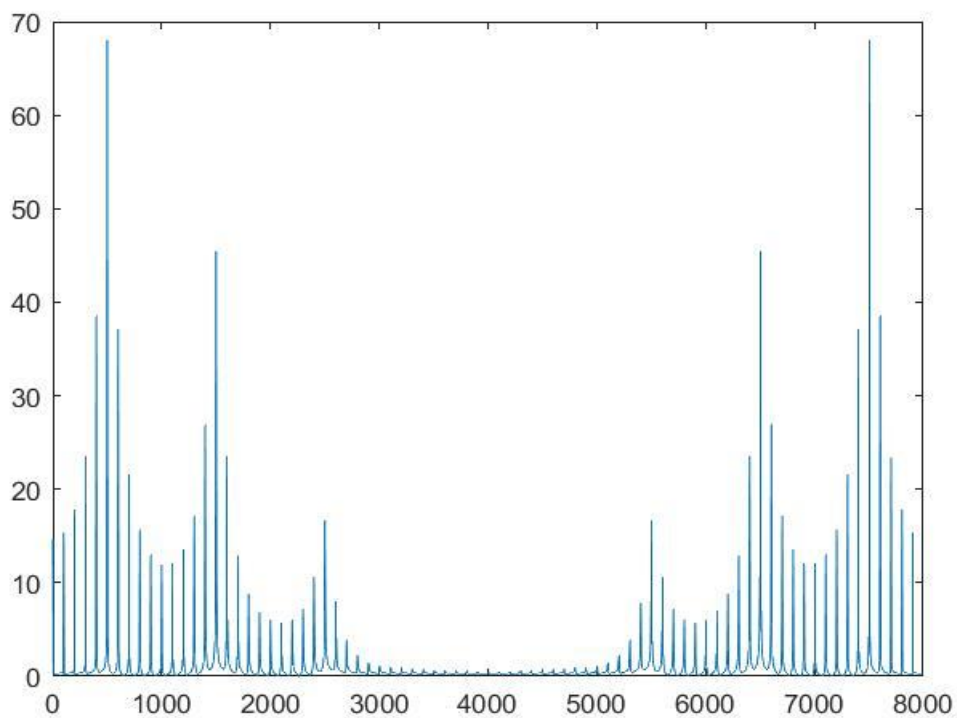


Και σε αυτή την περίπτωση, η κρουστική απόκριση πρόκειται για αύξουσα ταλάντωση, και μάλιστα με ραγδαία αύξηση (καθώς αρχικά πρόκειται για ευθεία γραμμή και στη συνέχεια παρουσιάζει αύξηση). Η απόκριση συχνότητας, όμοια με την περίπτωση $r = 0.7$, προσομοιάζει περισσότερο ημιτονοειδή παλμό, ωστόσο είναι μετατοπισμένη κατά πολύ προς τα πάνω (τα πλάτη της είναι της τάξης των 10^{18}). Τέλος, βλέπουμε πως, στο αρχικό παράδειγμα, εκτός του εύρους συχνοτήτων στο οποίο το πλάτος της απόκρισης συχνότητας μεγιστοποιείται, το πλάτος έχει τόσο μικρές τιμές που θεωρούνται πρακτικά μηδενικές. Κάτι τέτοιο, προφανώς, δεν ισχύει εδώ.

γ) Ακολουθώντας τις οδηγίες της εκφώνησης, θέτουμε $F_s = 8$ kHz και $r = 0.95$. Η απεικόνιση της συνολικής απόκρισης συχνότητας του συστήματος φαίνεται παρακάτω:

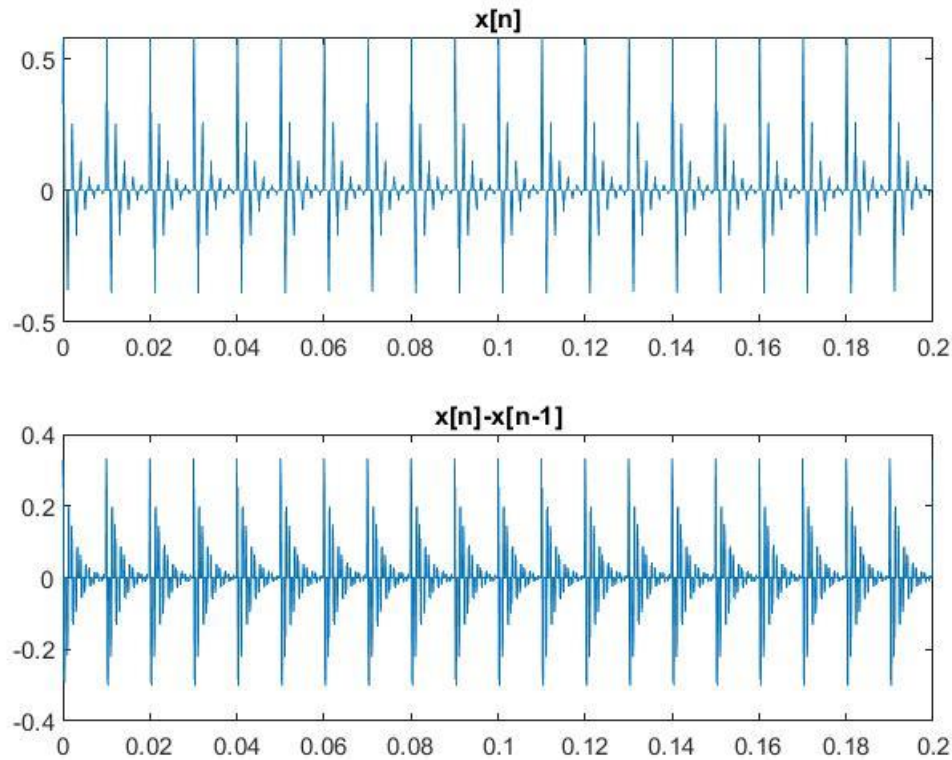


δ) Σύμφωνα με τις οδηγίες της εκφώνησης, τοποθετούμε ως είσοδο στον ταλαντωτή #1 μια σειρά από κρουστικές παλμοσειρές. Ο μετασχηματισμός Fourier που προκύπτει από την έξοδο του ταλαντωτή #3 φαίνεται παρακάτω:

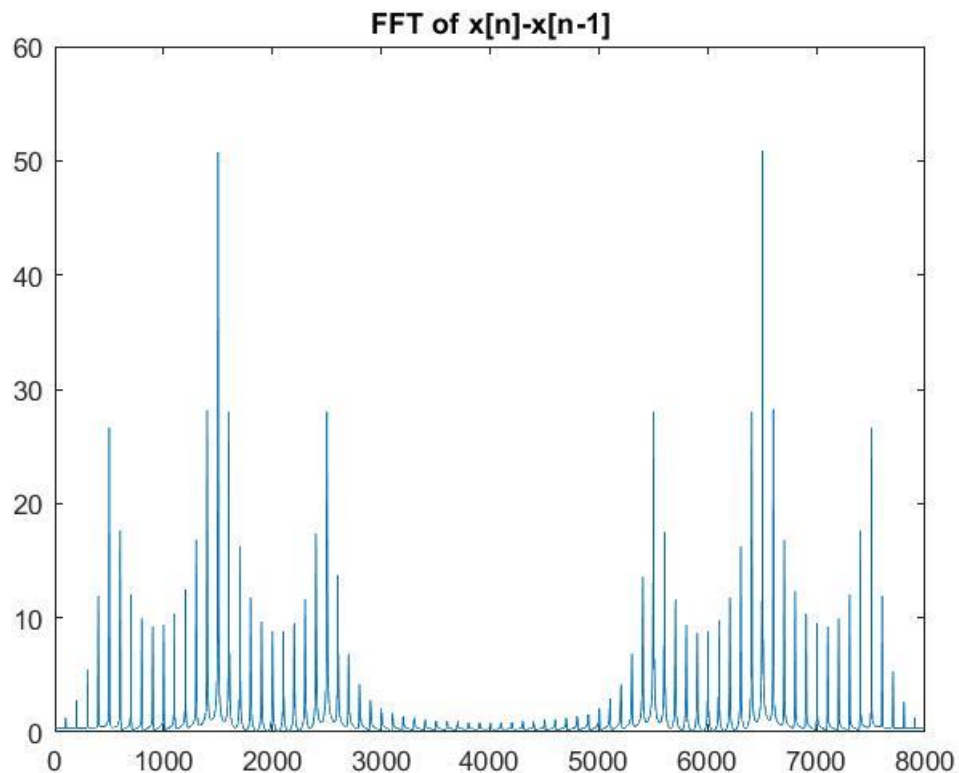


Παρατηρούμε πως στις τιμές των ιδιοσυχνοτήτων συντονισμού των τριών ταλαντωτών (500 Hz, 1500Hz, 2500Hz) η αύξηση πλάτους είναι ραγδαία. Ωστόσο, το πλάτος του σήματος αυξομειώνεται διαρκώς, λιγότερο ραγδαία. Το γράφημα είναι συμμετρικό ως προς τα 4000Hz.

Στη συνέχεια, μέσω των εντολών plot, subplot και filter υπολογίζονται αρχικά το σήμα $x[n]$ και στη συνέχεια η διαφορά μεταξύ των διαδοχικών σημάτων $x[n]-x[n-1]$.

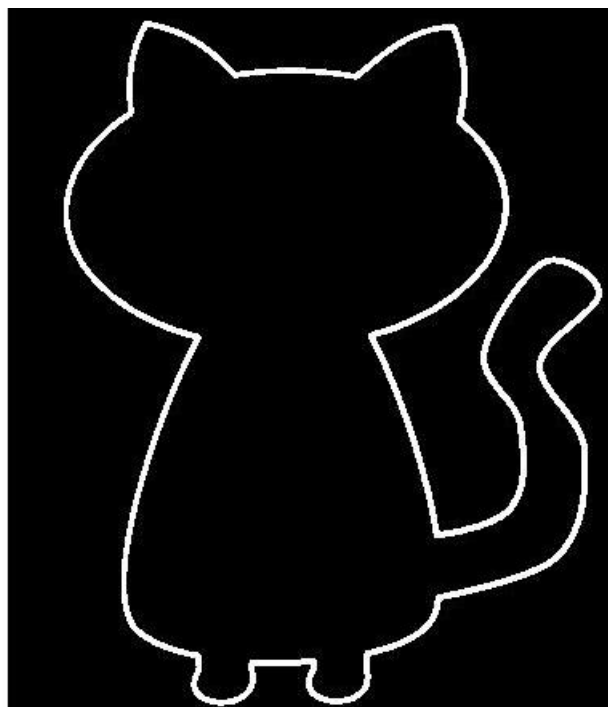


Μέσω τις εντολής sound ακούμε τα δύο παραπάνω σήματα, τα οποία αντιστοιχούν σε ένα φώνημα που δεν είναι εντελώς ξεκάθαρο, αλλά προσομοιάζει το φωνήεν «ε». Έπειτα, παρατηρούμε πως η διαφορά $x[n]-x[n-1]$ ακούγεται περισσότερο οξεία από το σήμα $x[n]$. Παρακάτω φαίνεται το μέτρο του μετασχηματισμού Fourier της διαφοράς $x[n]-x[n-1]$. Συγκρίνοντας το παρακάτω γράφημα, με το άνωθεν DTFT διάγραμμα του σήματος $x[n]$, παρατηρούμε πως το πλάτος του πρώτου μεγιστοποιείται σε μικρότερη συχνότητα σε σχέση με το πλάτος του δεύτερου, γεγονός που εξηγεί επαρκώς τη διαφορά οξύτητας.

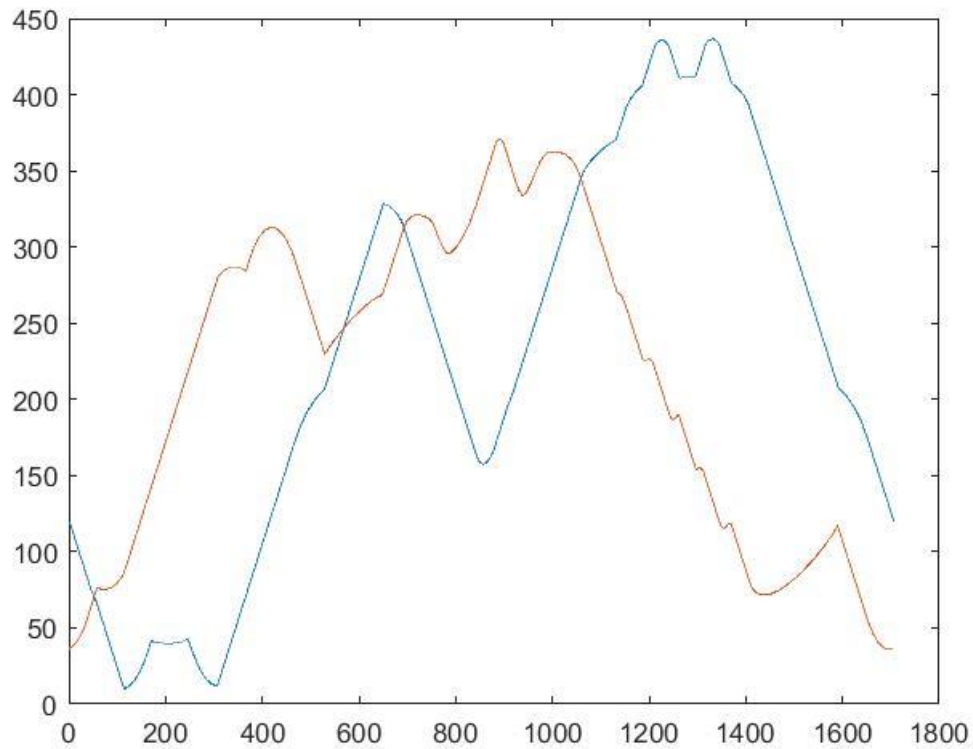


4. Ανακατασκευή Περιγράμματος Σχήματος (Shape Tracing) Μέσω Σειρών Fourier

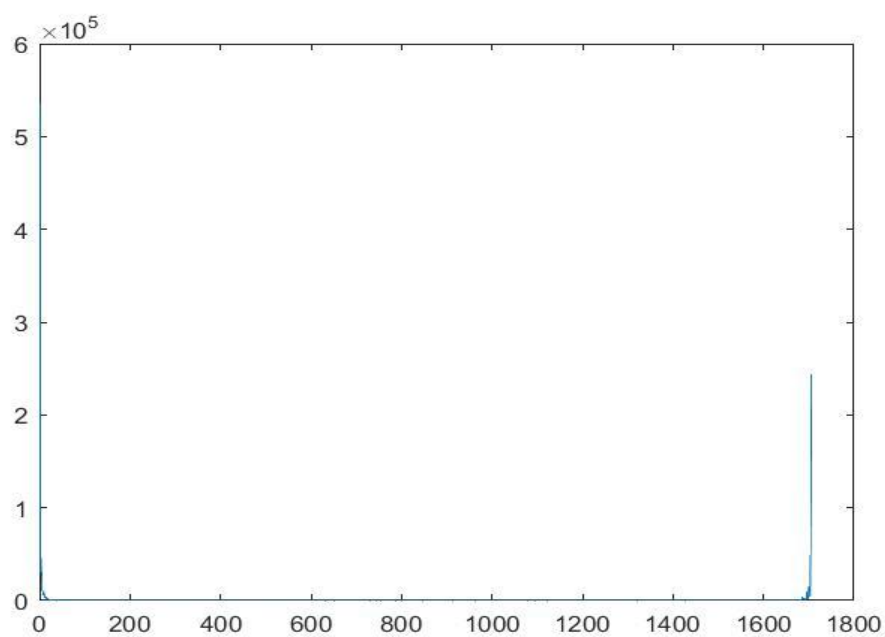
α) Φορτώνουμε στο Matlab την εικόνα image.png μέσω της εντολής `imread` και την απεικονίζουμε με την εντολή `imshow`.



β) Μέσω της εντολής `find` απομονώνουμε τα μονοδιάστατα σήματα $x[n]$ και $y[n]$, που αντιστοιχούν στις συντεταγμένες της παραμετροποιημένης καμπύλης $(x[n], y[n])$. Στη συνέχεια, με τις εντολές `bwtraceboundary` και `plot` απεικονίζουμε τα δύο σήματα συναρτήσει της μεταβλητής n .



γ) Σύμφωνα με τα παραπάνω, ορίζουμε τα διανύσματα x και y στο φάσμα των πραγματικών αριθμών και ορίζουμε, επίσης, τον μιγαδικό αριθμό $z = x + iy$. Υπολογίζουμε τον DTFT του z μέσω της εντολής `fft` και απεικονίζουμε το μέτρο του μέσω της εντολής `plot(abs())`.

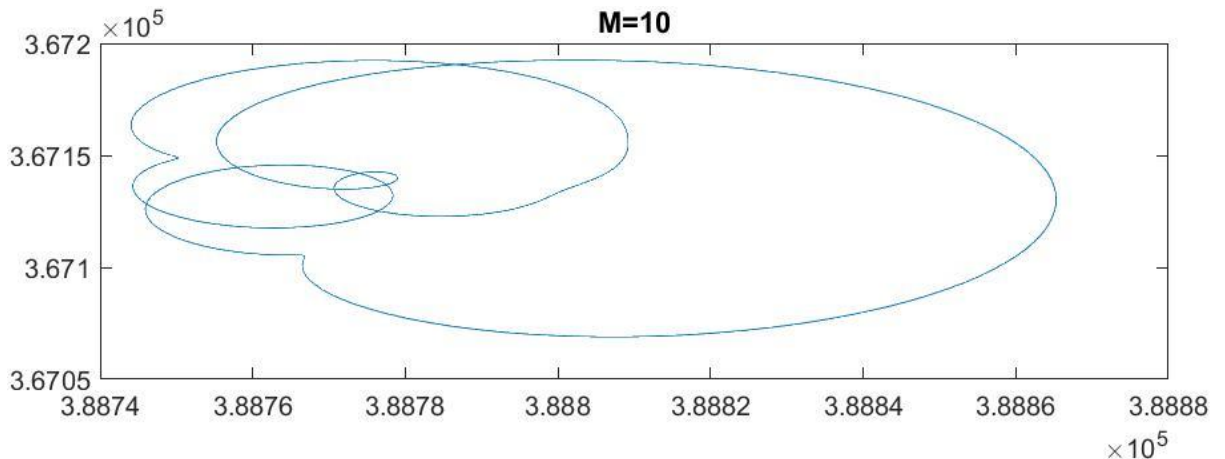


δ) Αρχικά, θέτουμε το N ίσο με το μήκος της σειράς $z_M[n]$. Για καθεμία από τις ζητούμενες τιμές του M , απομονώνουμε τις συντεταγμένες $x_M[n]$ και $y_M[n]$ αξιοποιώντας τον μετασχηματισμό Fourier του σήματος $z[n]$ του προηγούμενου ερωτήματος και κατασκευάζοντας ένα διπλό for loop (for loop μήκους M (για το εκάστοτε M) εντός for loop μήκους N), που υπολογίζει αναδρομικά τη δοθείσα σχέση:

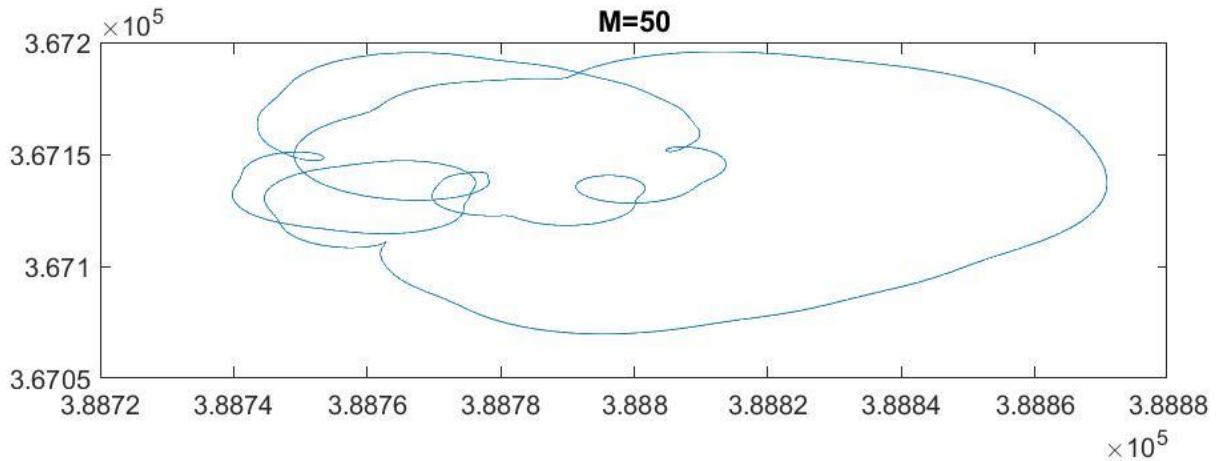
$$z_M[n] = \frac{1}{N} * \sum_{k=0}^M Z[k] * e^{j2\pi kn/N}$$

Θέτοντας το $x_M[n] = \text{Re}\{z_M[n]\}$ και το $y_M[n] = \text{Im}\{z_M[n]\}$ απεικονίζουμε τα περιγράμματα που προκύπτουν, με την εντολή `plot(x_M, y_M)`.

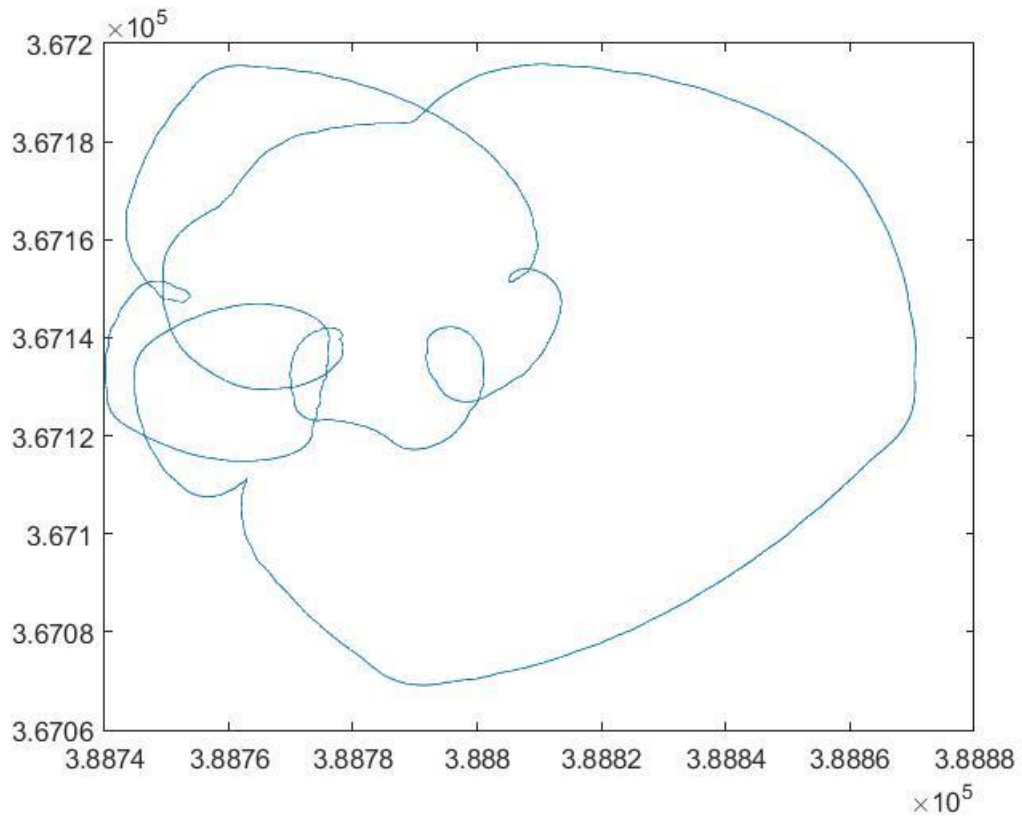
- $M = 10$:



- $M = 50$:



- M = 200:



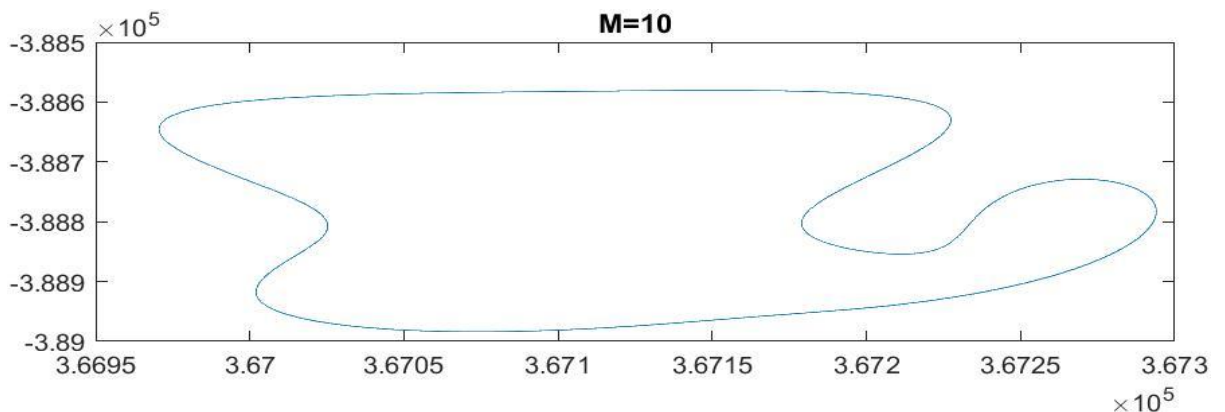
Παρατηρούμε πως όσο η τιμή του M μεγαλώνει, δηλαδή όσο περισσότερα δείγματα λαμβάνουμε από τη σειρά $z_M[n]$, το περίγραμμα που προκύπτει τείνει όλο και περισσότερο να μοιάζει στο πρωτότυπο.

ε) Όμοια με παραπάνω, υπολογίζουμε αναδρομικά, κατασκευάζοντας και πάλι ένα διπλό for loop (ένα for loop μήκους M/2 κι ένα δεύτερο for loop (N-M)/2 (για το εκάστοτε M) εντός ενός τρίτου for loop, μήκους N), υπολογίζουμε αναδρομικά τη δοθείσα σχέση:

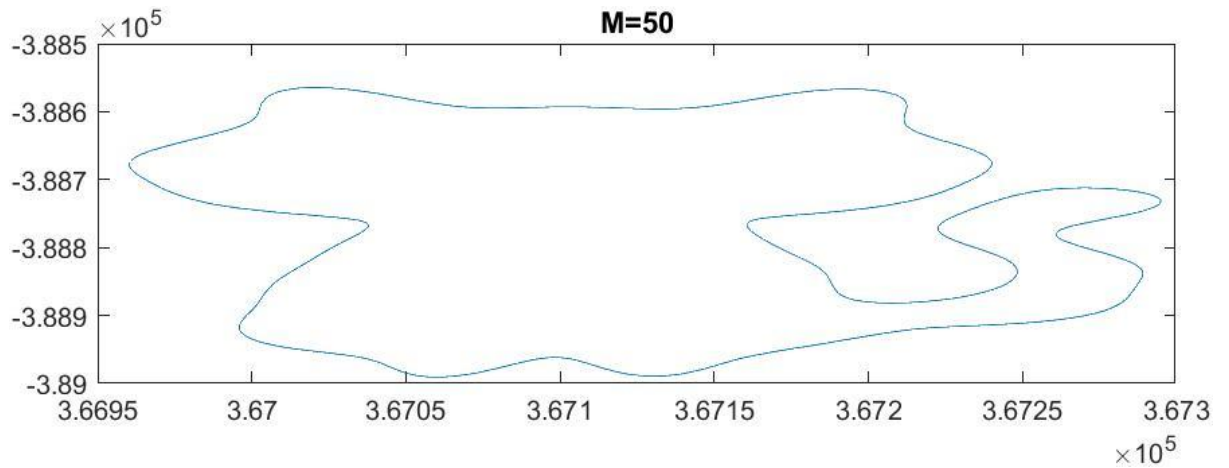
$$z_M[n] = \frac{1}{N} * \sum_{k=0}^{M/2} Z[k] * e^{j2\pi kn/N} + \frac{1}{N} * \sum_{k=N-M/2}^{N-1} Z[k] * e^{j2\pi kn/N}, \quad K = N - \frac{M}{2}$$

Θέτοντας το $x_M[n] = \text{Re}\{z_M[n]\}$ και το $y_M[n] = \text{Im}\{z_M[n]\}$ απεικονίζουμε τα περιγράμματα που προκύπτουν, με την εντολή `plot(x_M, y_M)`.

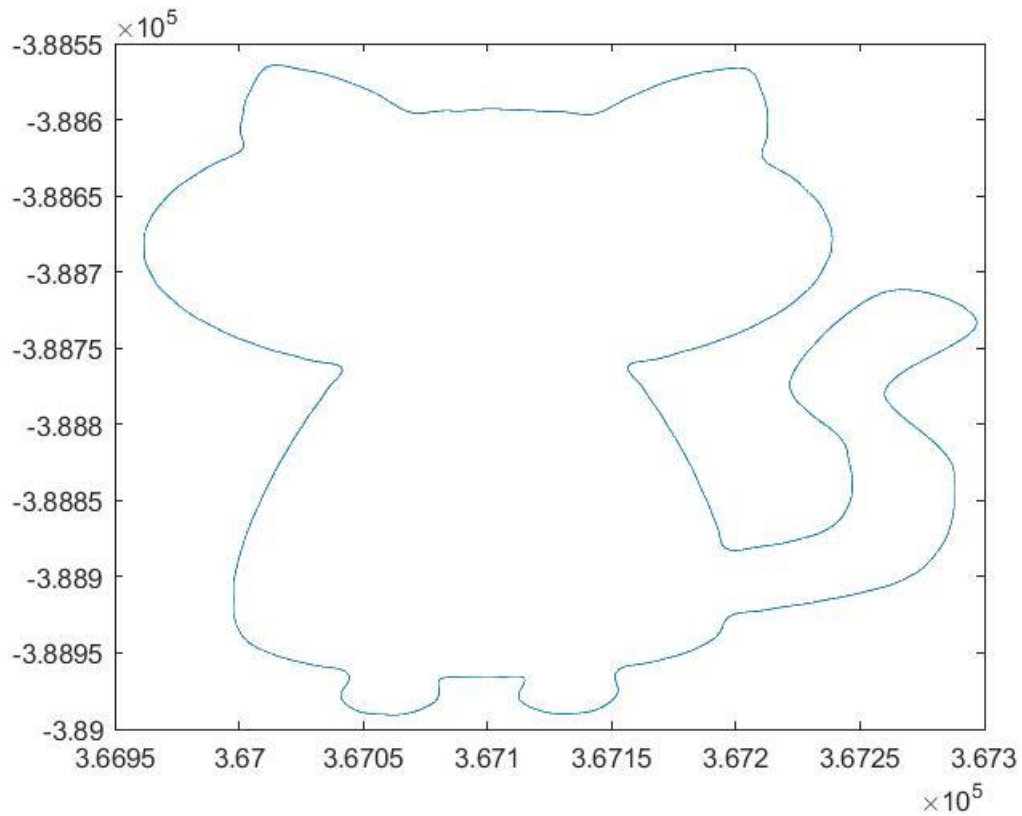
- M = 10:



- $M = 50$:

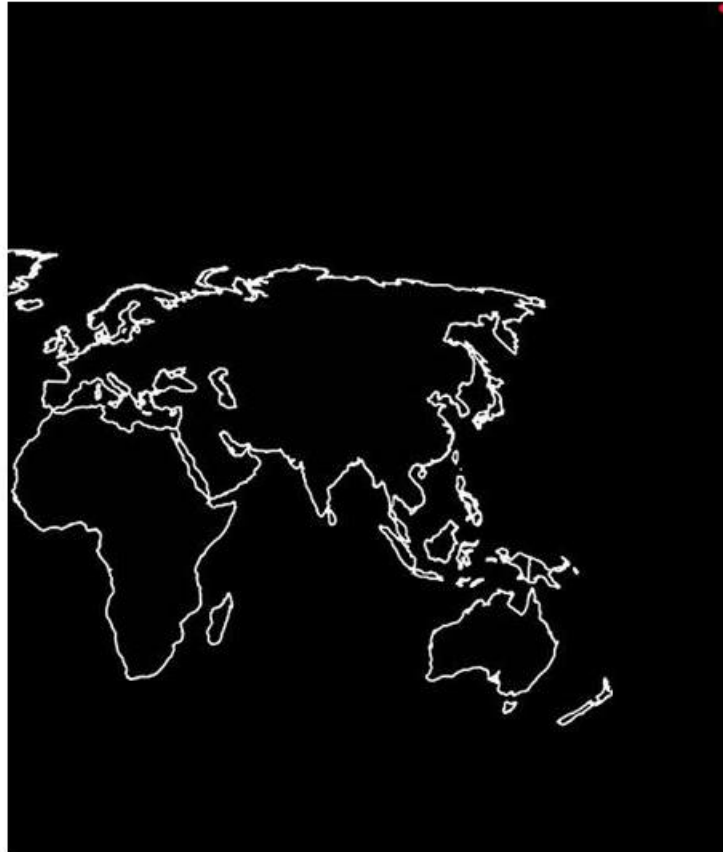


- $M = 200$:

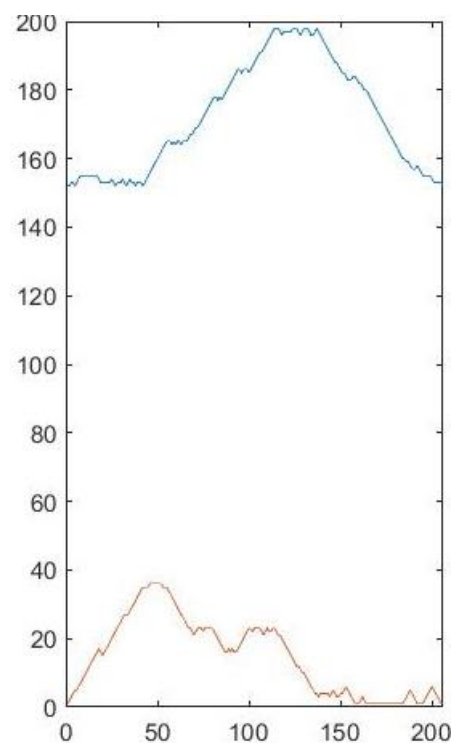
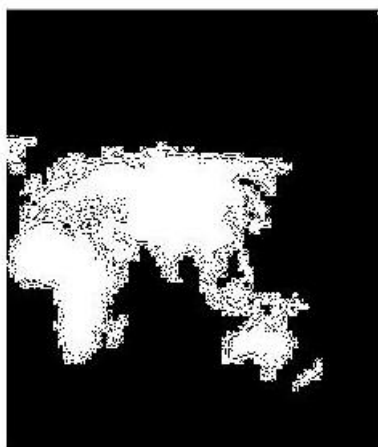


Παρατηρούμε ότι για $M = 200$ το περίγραμμα που προκύπτει είναι πανομοιότυπο με το αρχικό. Συνεπώς, 200 συντελεστές της παραμέτρου 200 επαρκούν για την ακριβή επανακατασκευή του περιγράμματος της εικόνας `image.png`.

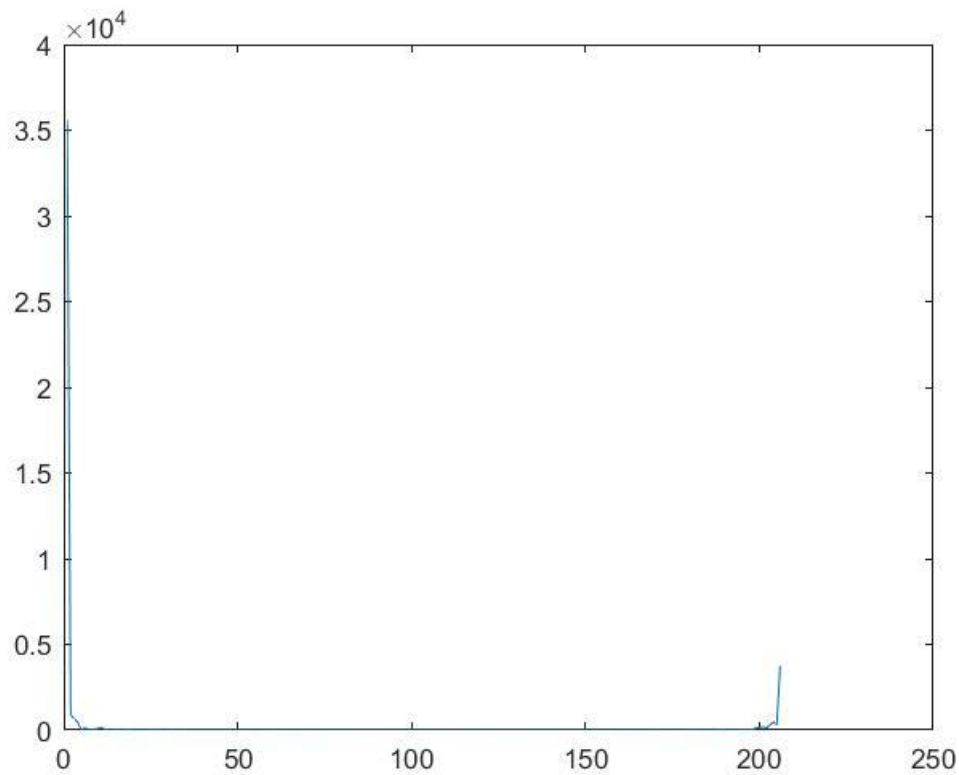
στ) Στη συνέχεια θα επαναλάβουμε όλα τα παραπάνω ερωτήματα, για διαφορετική εικόνα (η οποία δεν είναι binary). Για την αρχική της απεικόνιση, ακολουθούμε την ίδια διαδικασία με το ερώτημα α, και λαμβάνουμε το εξής διάγραμμα:



Για να ακολουθήσουμε τα βήματα του ερωτήματος β, είναι απαραίτητο πρωτίστως να μετατρέψουμε την παραπάνω εικόνα σε binary image, κάτι που επιτυγχάνεται μέσω της εντολής `im2bw`. Έπειτα, επιτελούμε την ίδια ακριβώς διαδικασία με το ερώτημα β, ενώ απεικονίζουμε και την binary εικόνα, και απεικονίζουμε τα δύο σήματα μέσω της μεταβλητής `n`:

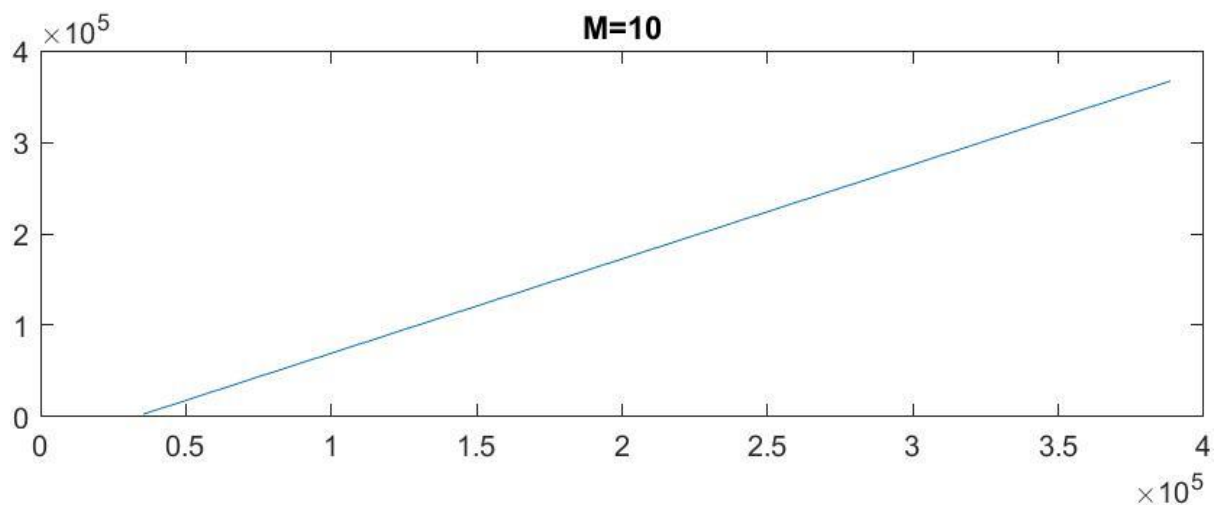


Όμοια με προηγουμένως, ορίζουμε τα διανύσματα x και y στο φάσμα των πραγματικών αριθμών και ορίζουμε, επίσης, τον μιγαδικό αριθμό $z = x + iy$. Υπολογίζουμε τον DTFT του z μέσω της εντολής `fft` και απεικονίζουμε το μέτρο του μέσω της εντολής `plot(abs())`.

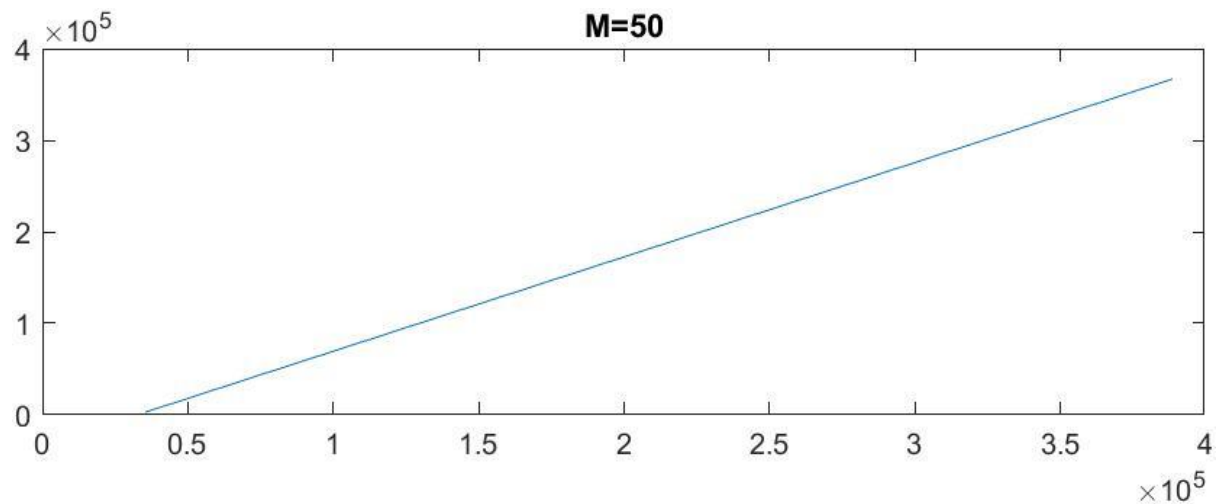


Ως εδώ, οι ομοιότητες, τόσο στον αλγόριθμο όσο και στην αποτελεσματικότητα της εικόνας και των διανυσμάτων της, είναι αρκετές. Εκτελώντας, ωστόσο, το ερώτημα **δ)** για τη νέα εικόνα, οι διαφορές είναι οφθαλμοφανείς. Πιο συγκεκριμένα, ακολουθώντας τους τύπους του ερωτήματος και κάνοντας δοκιμές για διάφορες τιμές του M , λαμβάνουμε τα εξής αποτελέσματα:

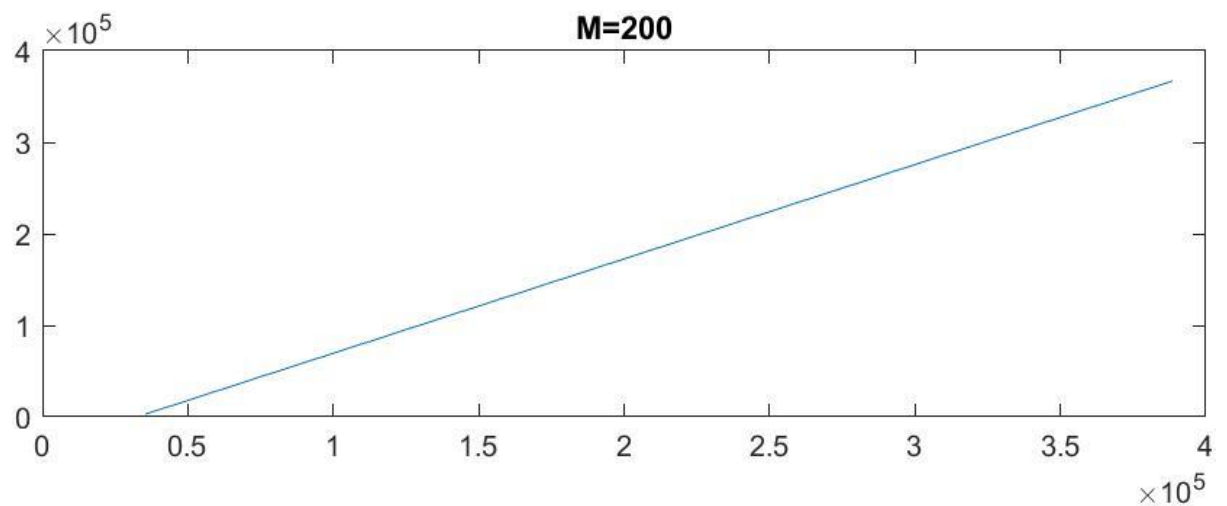
- $M = 10$:



- $M = 50$:

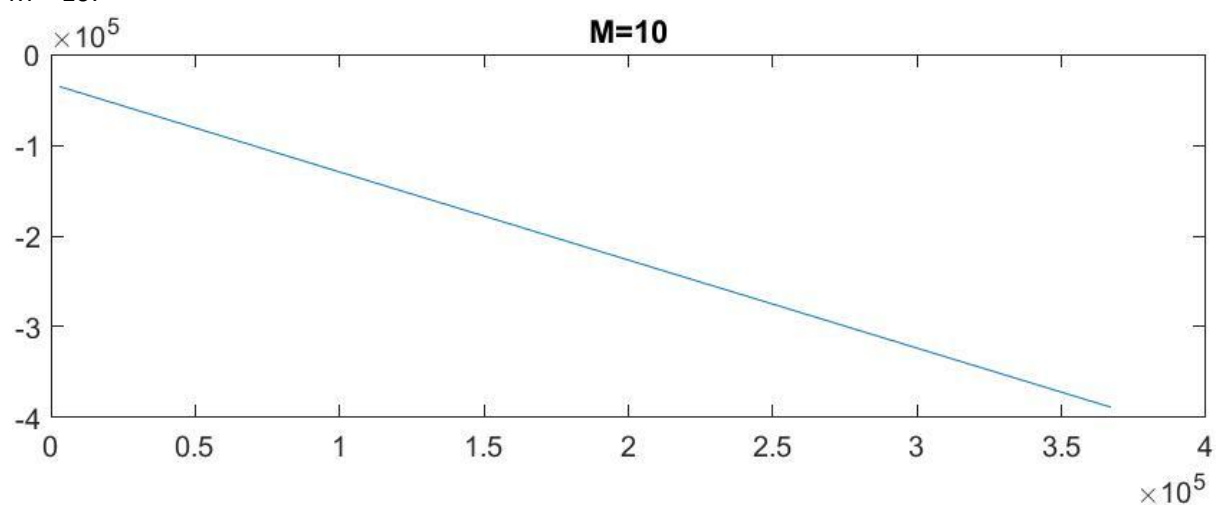


- $M = 200$:

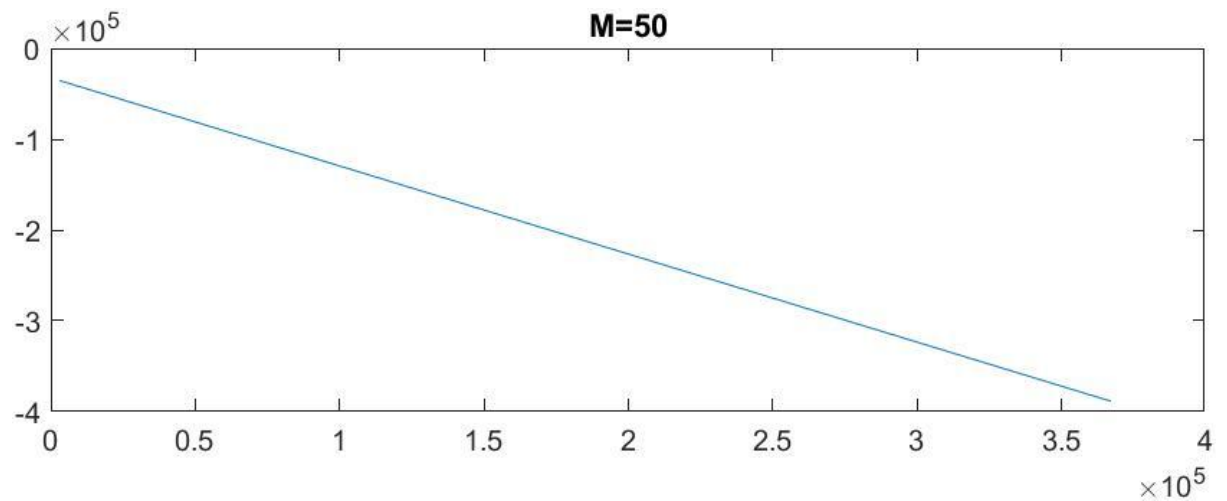


Όσο κι αν αυξάνεται, δηλαδή, η τιμή του M , το διάγραμμα δεν προσεγγίζει το περίγραμμα της εικόνας. Κάτι τέτοιο οφείλεται στην αυξημένη ανάλυση της δεύτερης εικόνας, έναντι της πρώτης. Όμοια, προχωρώντας στο ερώτημα **ε)**, παρατηρούμε αντίστοιχα αποτελέσματα:

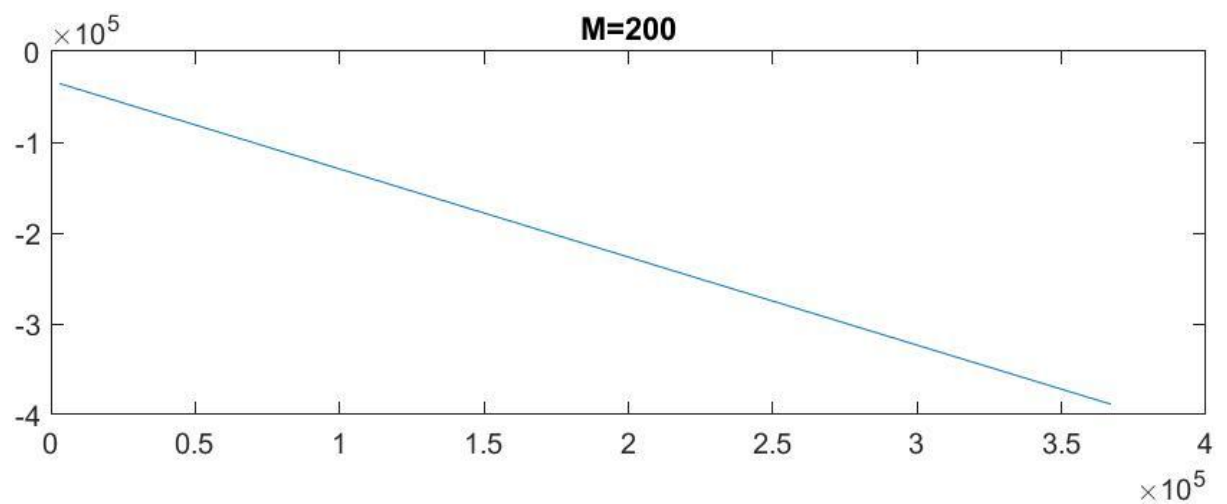
- $M = 10$:



- $M = 50$:



- $M = 200$:



Οι παραπάνω τεχνικές απεικόνισης, λοιπόν, του περιγράμματος της νέας εικόνας δεν είναι αποτελεσματικές. Συνεπώς, απεικονίζουμε το περίγραμμα μέσω της εντολής `bwperim` και λαμβάνουμε το παρακάτω:

