

**Μάθημα: «Αρχές Εφαρμογές Σημάτων και  
Συστημάτων (4<sup>ο</sup> Εξάμηνο)»**

**ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ:**

**Αναστάσιος Κωνσταντινίδης Π17057**

**Φώτιος Σιούζιος Π17121**

**Χρήστος Δήμτσας Π17027**

# Γ1

1.

Με Βάση το σήμα  $x(t) = \cos(100\pi t) + \cos(200\pi t) + \sin(500\pi t)$

Και τον τύπο  $T = 2\pi/\omega$  .

Έχουμε, πως το  $\cos(100\pi t)$  έχει  $T_1 = 0,02\text{sec}$

Το  $\cos(200\pi t)$  έχει  $T_2 = 0,01\text{sec}$

Το  $\sin(500\pi t)$  έχει  $T_3 = 0,004\text{sec}$

και η περίοδος του σήματος ισούται με  $T_s = 2\pi/\omega = 0,02 \text{ sec}$ .

Απο το θεώρημα Shannon – Nyquist έχουμε :  $1/T \leq 2s(o)$  .

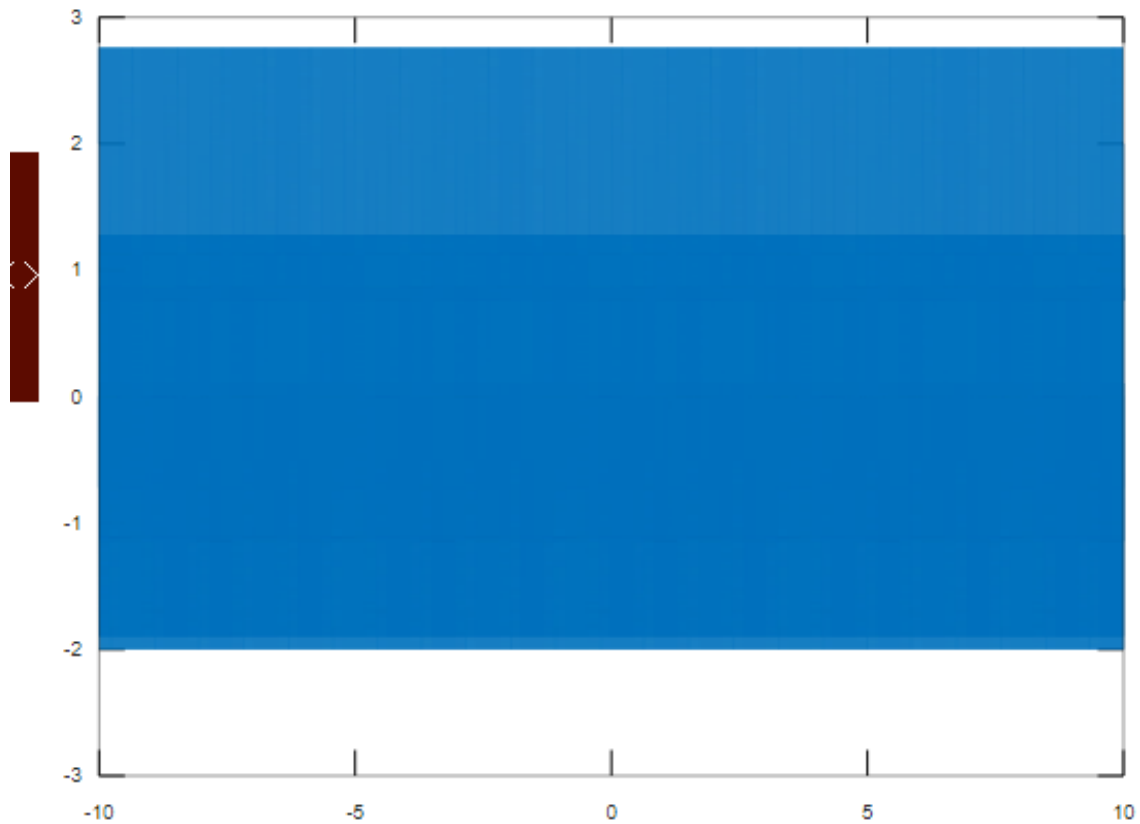
Άρα:  $s(o) = 1/2T = 1/0,04 = 25 \text{ Hz}$

Η ελάχιστη απαιτούμενη συχνότητα θα είναι  $s(o) = 25 \text{ Hz}$

2.

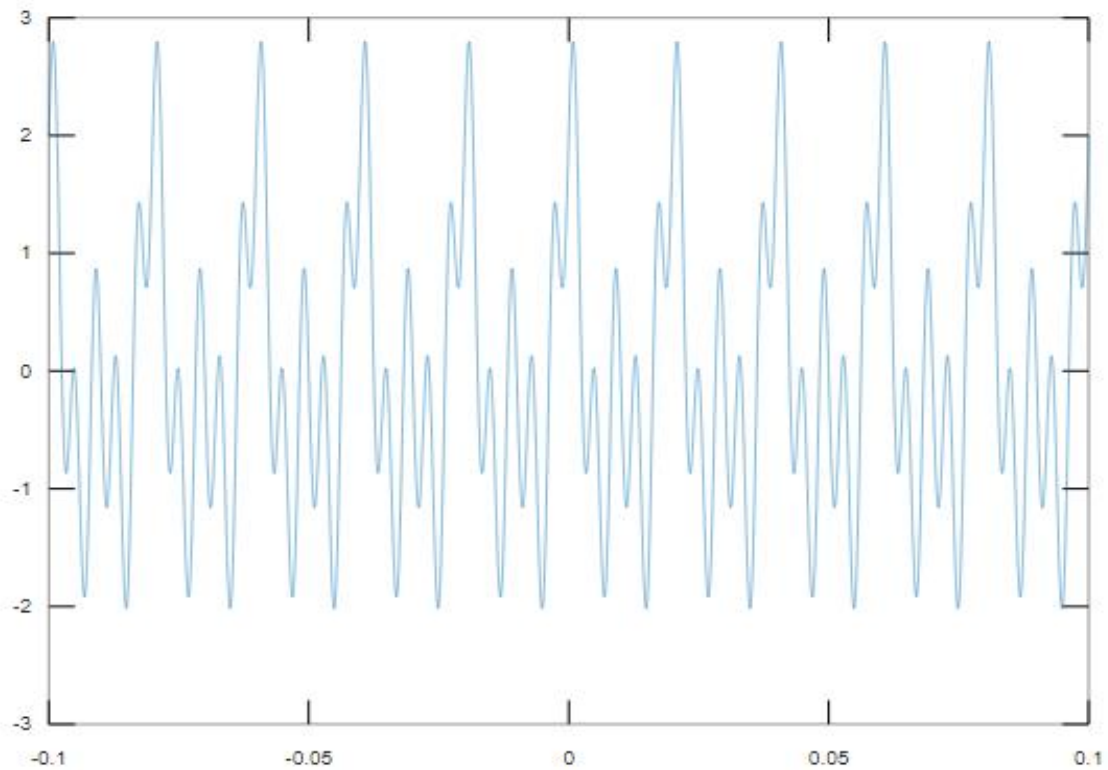
Έχουμε  $x(t)$  όπου  $t$  ανήκει  $[-10,10]$ , με βήμα  $\Delta t=0,001s$

```
octave:2> t=-10:0.001:10;  
octave:3> x= cos(100*pi*t) + cos(200*pi*t) + sin(500*pi*t);  
octave:4> plot(t,x);
```



Μειώνουμε την κλίμακα, με το να διαιρέσουμε με 100 τον χρόνο για να φανεί καλύτερα το σχήμα.

```
octave:2> t=-0.1:0.00001:0.1;  
x= cos(100*pi*t) + cos(200*pi*t) + sin(500*pi*t);  
plot(t,x);
```



3.

Με συχνότητα δειγματοληψίας  $S(o)=25\text{Hz}$  έχουμε ότι :

$$T=0,04\text{sec}$$

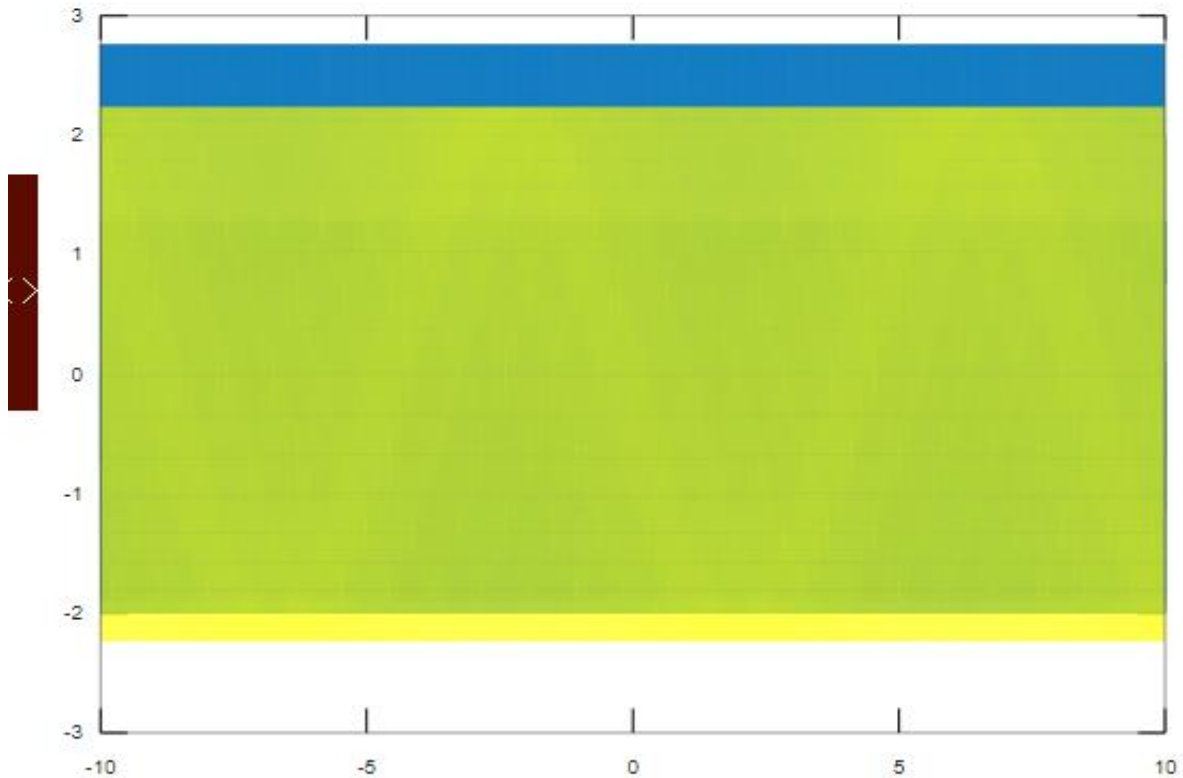
$$T=2\pi/\omega \Leftrightarrow \omega=2\pi/T \Leftrightarrow \omega=2\pi/0,04 \Leftrightarrow \underline{\omega=50\pi}$$

Έτσι λοιπόν το ανακατασκευασμένο σήμα είναι το ακόλουθο :

$$y(t)=\cos(50\pi t)+\cos(50\pi t)+\sin(50\pi t)$$

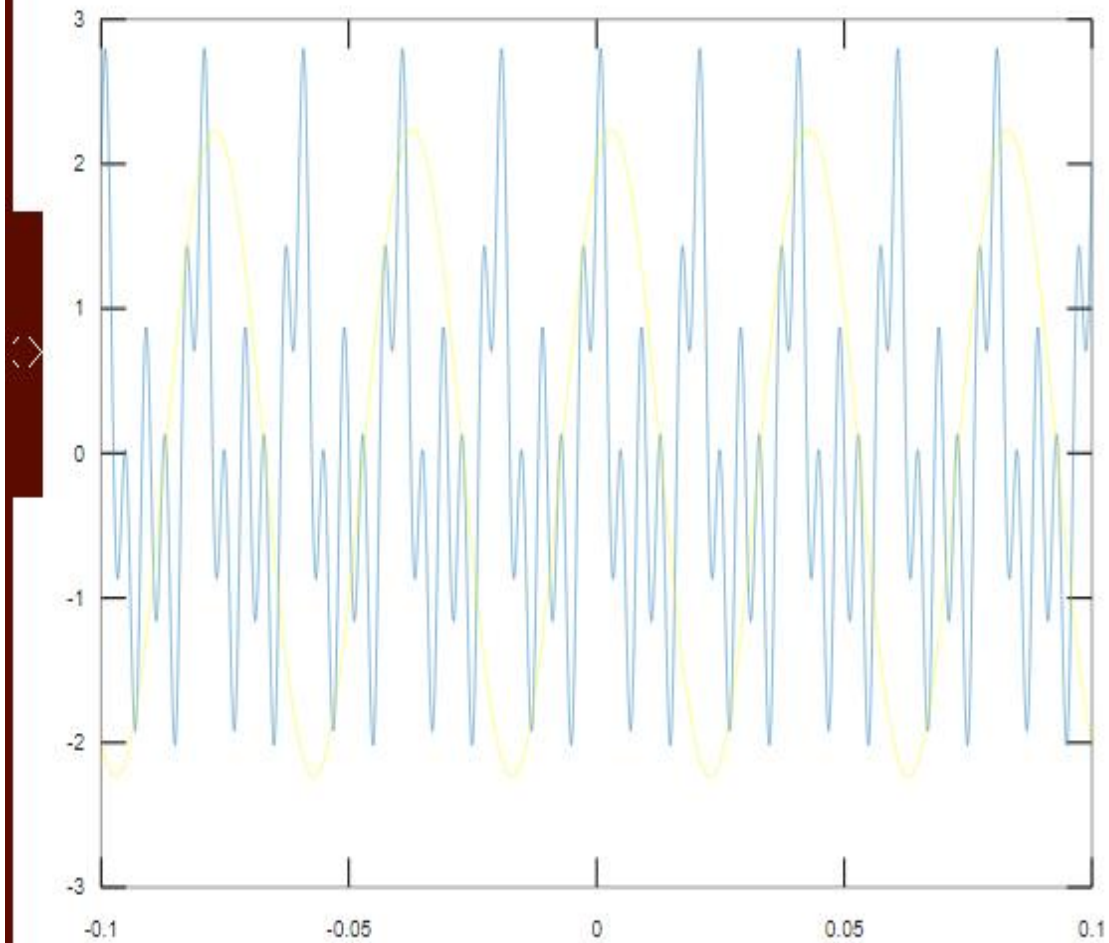
octave:15>

```
t=-10:0.001:10;  
x= cos(100*pi*t) + cos(200*pi*t) + sin(500*pi*t);  
y=cos(50*pi*t)+cos(50*pi*t)+sin(50*pi*t);  
plot(t,x,t,y,'y');
```



Μειώνουμε την κλίμακα, με το να διαιρέσουμε με 100 τον χρόνο για να φανεί καλύτερα το σχήμα.

```
octave:5> t=-0.1:0.00001:0.1;  
x= cos(100*pi*t) + cos(200*pi*t) + sin(500*pi*t);  
y=cos(50*pi*t)+cos(50*pi*t)+sin(50*pi*t);  
plot(t,x,t,y,'y');
```



#### 4.

Έστω νέα συχνότητα δειγματοληψίας είναι

$$S2 = 40 \text{ Hz.}$$

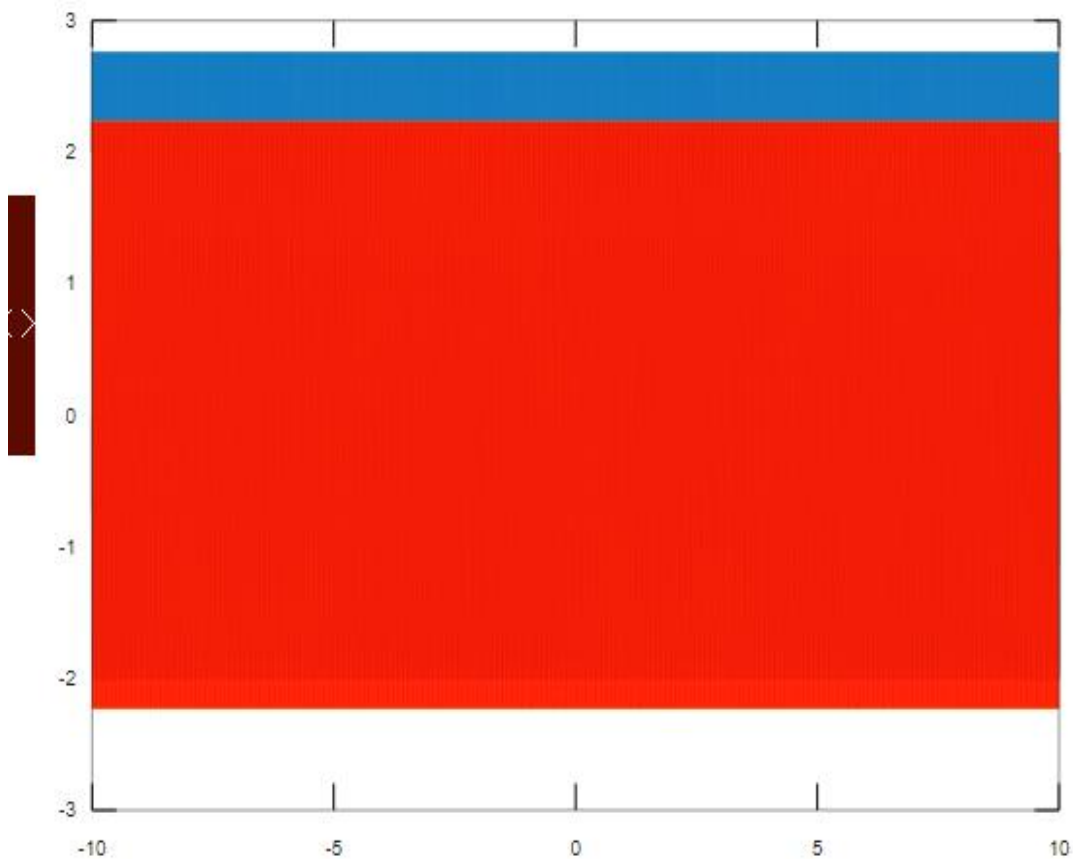
Επομένως  $T2 = 1/S2 = 0,025 \text{ sec}$  και  $\omega2 = 2\pi/T2 = 80 \pi$ .

Άρα το καινούργιο σήμα είναι το

$$y2(t) = \cos(80\pi t) + \cos(80\pi t) + \sin(80\pi t);$$

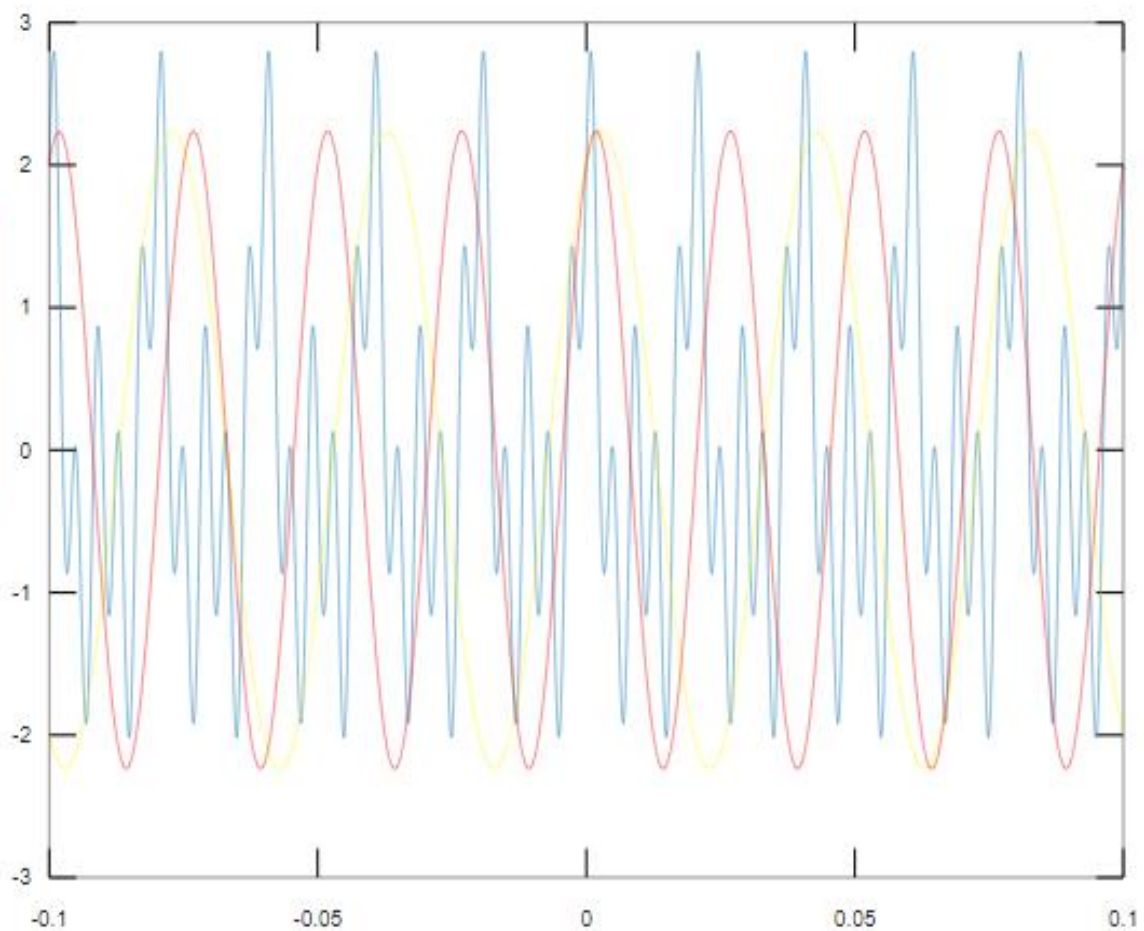
και έτσι προκύπτει το παρακάτω σχήμα.

```
octave:24> t=-10:0.001:10;  
x= cos(100*pi*t) + cos(200*pi*t) + sin(500*pi*t);  
y=cos(50*pi*t)+cos(50*pi*t)+sin(50*pi*t);  
y2= cos(80*pi*t) + cos(80*pi*t) + sin(80*pi*t);  
plot(t,x,t,y,'y',t,y2,'r');
```



Μειώνουμε την κλίμακα, με το να διαιρέσουμε με 100 τον χρόνο για να φανεί καλύτερα το σχήμα.

```
octave:9> t=-0.1:0.00001:0.1;  
x= cos(100*pi*t) + cos(200*pi*t) + sin(500*pi*t);  
y=cos(50*pi*t)+cos(50*pi*t)+sin(50*pi*t);  
y2= cos(80*pi*t) + cos(80*pi*t) + sin(80*pi*t);  
plot(t,x,t,y,'y',t,y2,'r');
```





## 5.

Επιλέξαμε συχνότητα δειγματοληψίας μικρότερη από 25Hz.

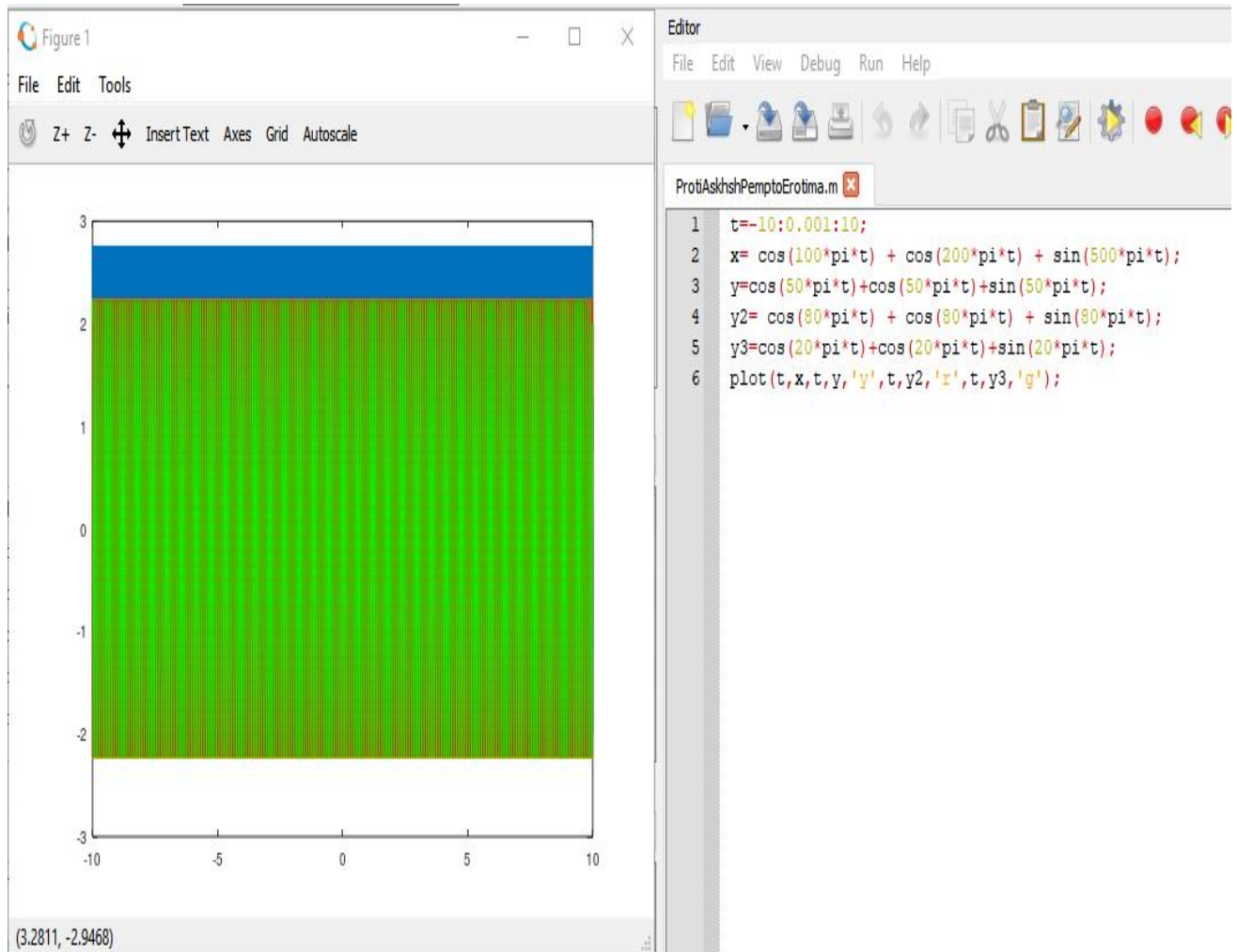
Θεωρούμε έτσι **S3=10Hz**.

$$S3=1/T \Leftrightarrow T=1/S3 \Leftrightarrow T=0,1\text{sec}$$

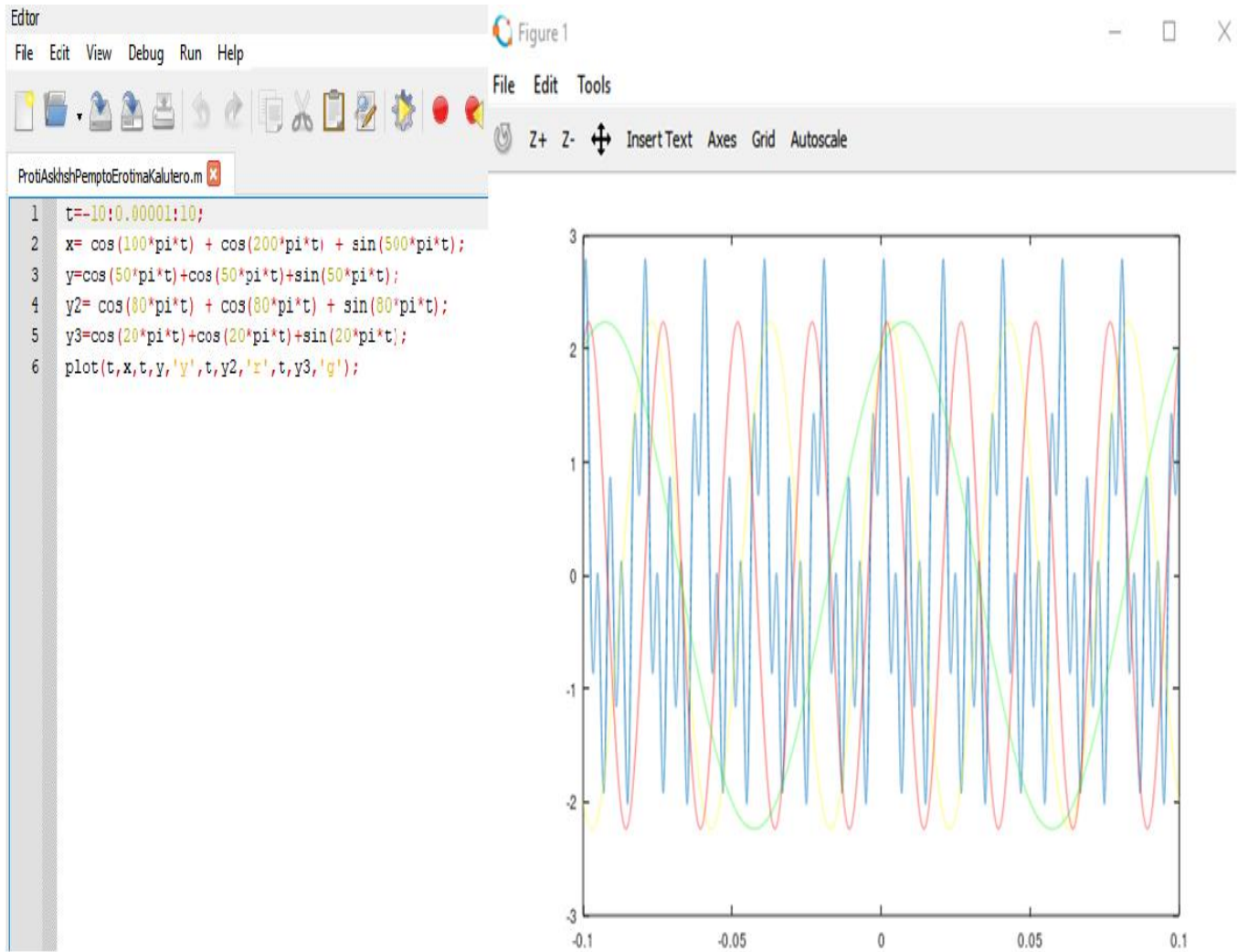
$$T=2\pi/\omega \Leftrightarrow \omega=2\pi/T \Leftrightarrow \omega=2\pi/0,1 \Leftrightarrow \omega=20\pi$$

Συνεπώς το νέο σήμα είναι :

$$y3(t)=\cos(20\pi t)+\cos(20\pi t)+\sin(20\pi t)$$



Μειώνουμε την κλίμακα, με το να διαιρέσουμε με 100 τον χρόνο για να φανεί καλύτερα το σχήμα.



## 6.

Παρατηρούμε ότι από τα τέσσερα σήματα το σήμα  $y_3(t)$  λόγω του ότι έχει τη μικρότερη συχνότητα( $f=10\text{Hz}$ ) έχει και το μεγαλύτερο μήκος κύματος ενώ το σήμα  $y_2(t)$  που έχει την μεγαλύτερη συχνότητα έχει το μικρότερο μήκος κύματος. Επίσης διακρίνουμε ότι, το σήμα  $y_2(t)$  , έχει μικρότερη απώλεια πληροφορίας από το σήμα  $y_3(t)$  σε σύγκριση με το αρχικό σήμα  $x(t)$  και αυτό οφείλεται στην μεγαλύτερη συχνότητα που επιλέξαμε. **Επιπλέον μπορούμε να πούμε ότι το πρώτο σήμα(το ανακατασκευασμένο)  $y(t)$  που ικανοποιεί το θεώρημα δειγματοληψίας κρατάει αναλλοίωτη(πιστή) την πληροφορία του αρχικού σήματος  $x(t)$ .**

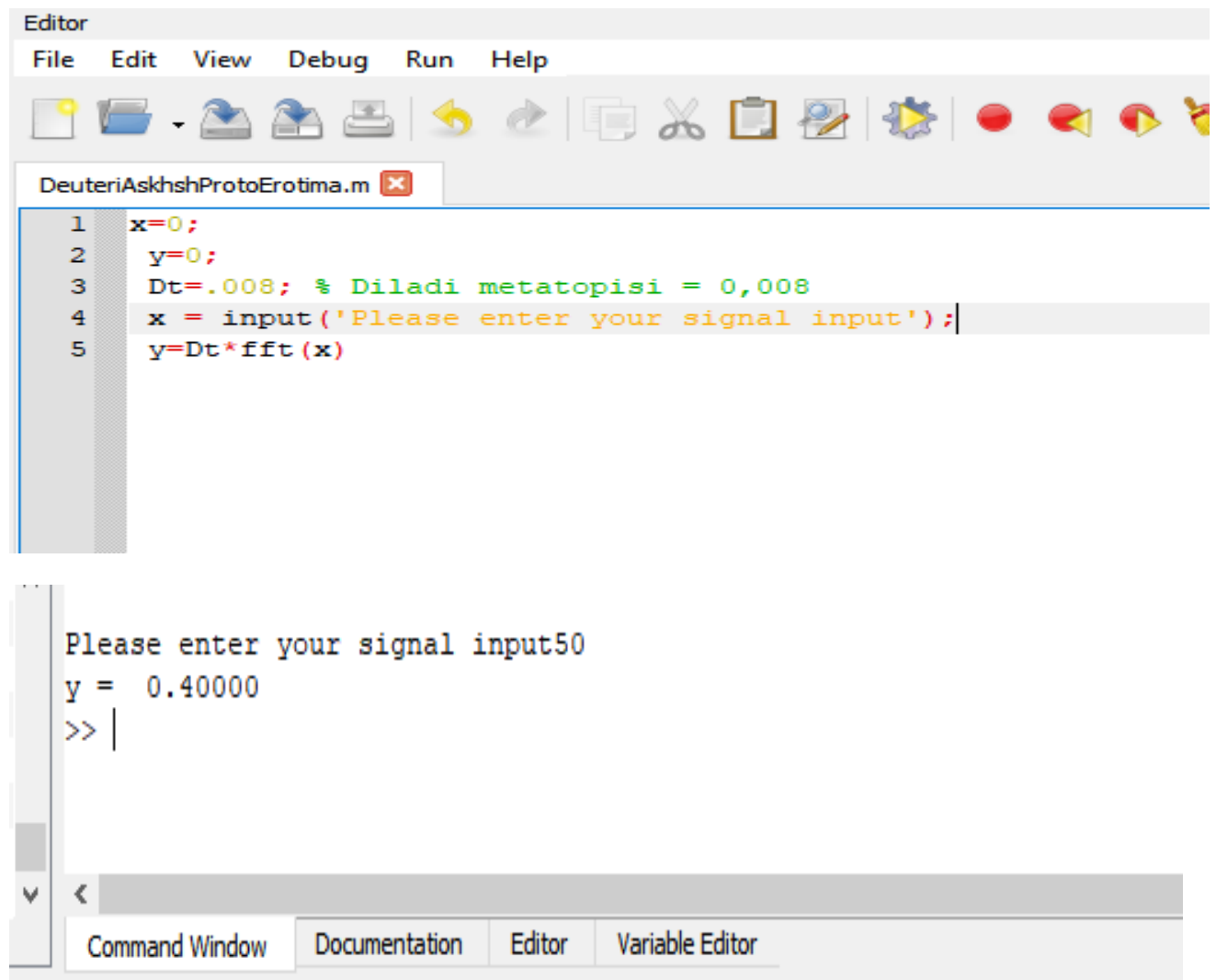
## Γ2

### 1.

Ο διακριτός μετασχηματισμός Fourier έχει τύπο:

$$x(m) = \sum_{n=1}^n x(n) \cdot \exp(-j \cdot 2 \cdot \pi \cdot (m-1) \cdot (n-1)/N)$$

(όπου  $1 \leq m \leq N$ ) με N να είναι το μήκος του διανύσματος X



The image shows two screenshots from a MATLAB environment. The top screenshot is of the MATLAB Editor window, displaying a script named 'DeuteriAskhshProtoErotima.m'. The script contains the following code:

```
1 x=0;  
2 y=0;  
3 Dt=.008; % Diladi metatopisi = 0,008  
4 x = input('Please enter your signal input');  
5 y=Dt*fft(x)
```


The bottom screenshot is of the MATLAB Command Window, showing the execution of the script. The prompt 'Please enter your signal input' is followed by the user input '50'. The output shows 'y = 0.40000' and the prompt '>>'.

## 2.

Ο αντίστροφος διακριτός μετασχηματισμός Fourier έχει τύπο:

$$x(n) = (1/N) \sum_{k=1}^N X(k) \exp(-j \cdot 2 \cdot \pi \cdot (k-1) \cdot (n-1)/N)$$

**(όπου  $1 \leq n \leq N$ )** με N να είναι το μήκος του διανύσματος X

\*DeuteriAskhshDeuteroErotima.m 

```
1 y=0;  
2 y=input('input the Vector ');  
3 n=0;  
4 n=input('input the fundamental frequency ');  
5 Dt=0;  
6 Dt=input('input the factors ');  
7 z=0;  
8 z=(1/Dt).*ifft(y,n)
```

### Command Window

```
input the Vector60
input the fundamental frequency 20
input the factors0.05
```

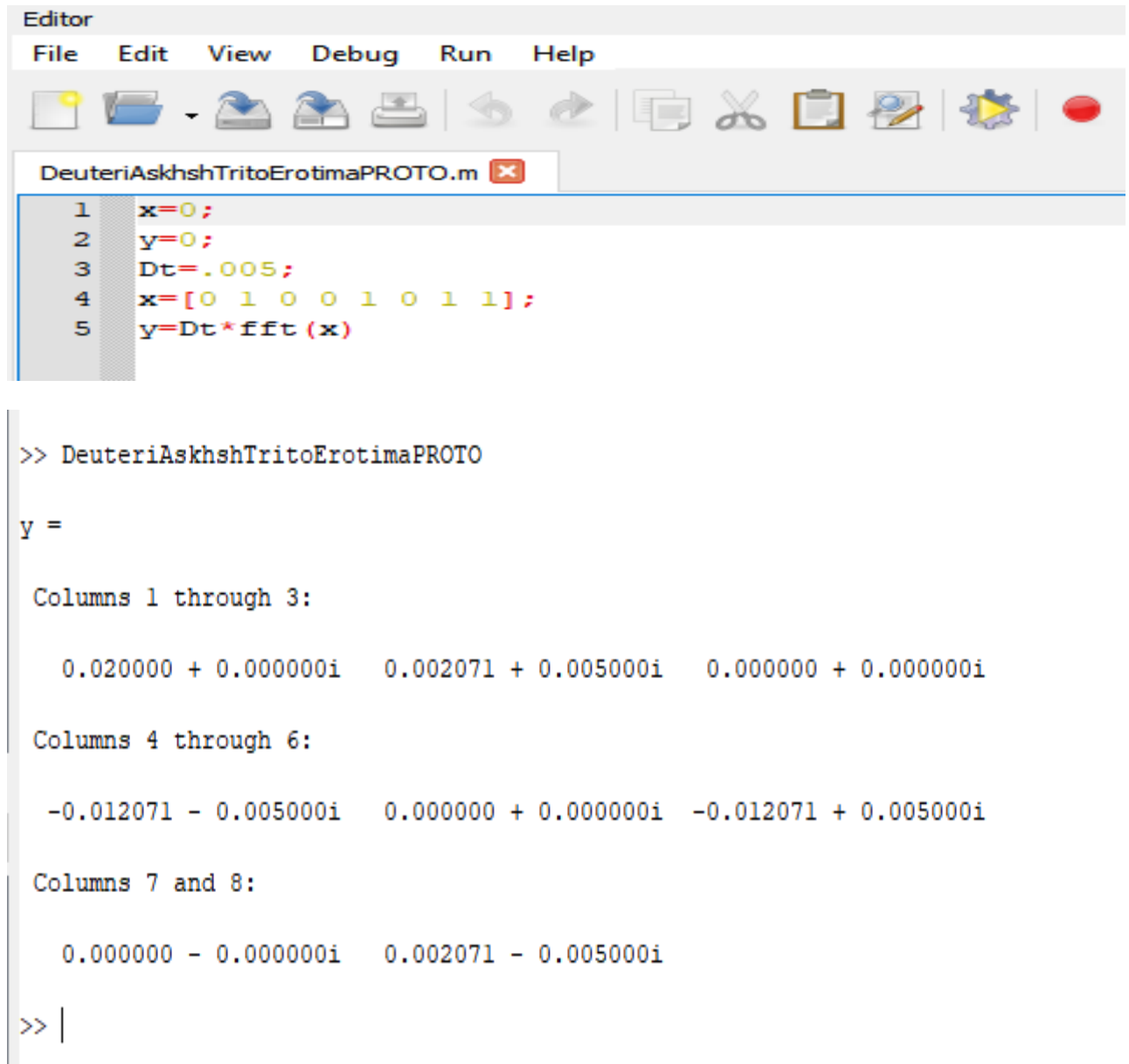
[illegible]

>> |

### 3.

Έχουμε επιλέξει αγγλικό αλφάβητο και το επώνυμό μου ξεκινάει από K( 0100 1011 ).

Για τον μετασχηματισμό Fourier έχουμε :



The image shows a MATLAB Editor window with a script named 'DeuteriAskhshTritoErotimaPROTO.m'. The script contains five lines of code: 1. `x=0;`, 2. `y=0;`, 3. `Dt=.005;`, 4. `x=[0 1 0 0 1 0 1 1];`, and 5. `y=Dt*fft(x)`. Below the editor, the Command Window shows the execution of the script, displaying the output for the FFT calculation. The output is organized into columns, with the first three columns showing the real and imaginary parts of the first three frequency components, the next three columns showing the real and imaginary parts of the next three frequency components, and the final two columns showing the real and imaginary parts of the last two frequency components.

```
Editor
File Edit View Debug Run Help
DeuteriAskhshTritoErotimaPROTO.m
1 x=0;
2 y=0;
3 Dt=.005;
4 x=[0 1 0 0 1 0 1 1];
5 y=Dt*fft(x)

>> DeuteriAskhshTritoErotimaPROTO

y =

Columns 1 through 3:

    0.020000 + 0.000000i    0.002071 + 0.005000i    0.000000 + 0.000000i

Columns 4 through 6:

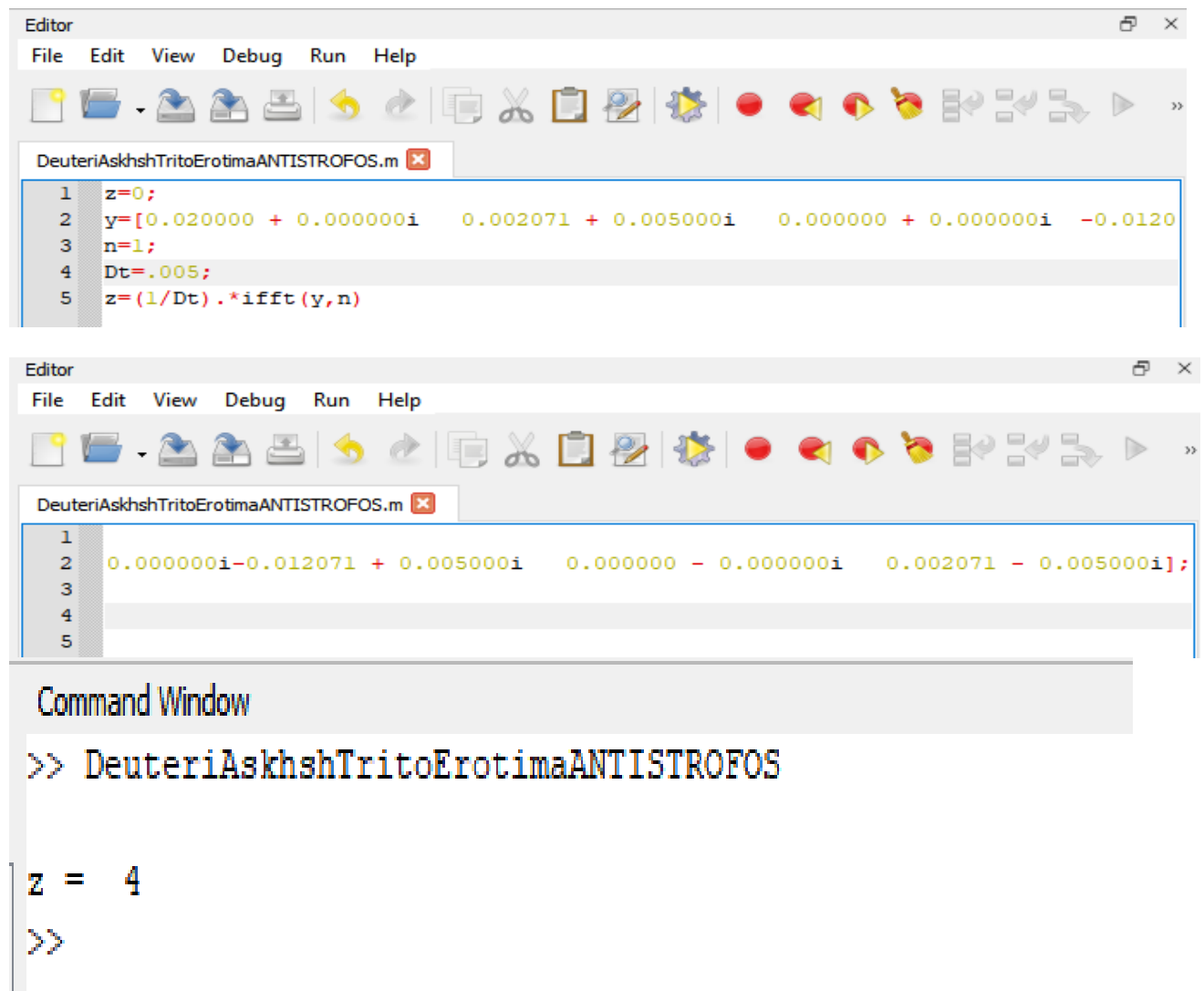
   -0.012071 - 0.005000i    0.000000 + 0.000000i   -0.012071 + 0.005000i

Columns 7 and 8:

    0.000000 - 0.000000i    0.002071 - 0.005000i

>> |
```

Για τον αντίστροφο μετασχηματισμό Fourier έχουμε :



The image displays two screenshots of the MATLAB environment. The top screenshot shows the Editor window with a script named 'DeuteriAskhshTritoErotimaANTISTROFOS.m'. The script contains five lines of code: 1. `z=0;`, 2. `y=[0.020000 + 0.000000i 0.002071 + 0.005000i 0.000000 + 0.000000i -0.0120`, 3. `n=1;`, 4. `Dt=.005;`, and 5. `z=(1/Dt).*ifft(y,n)`. The bottom screenshot shows the same Editor window with the script content partially visible, and the Command Window below it. The Command Window shows the command `>> DeuteriAskhshTritoErotimaANTISTROFOS` being executed, followed by the output `z = 4` and a prompt `>>`.

```
Editor
File Edit View Debug Run Help
DeuteriAskhshTritoErotimaANTISTROFOS.m
1 z=0;
2 y=[0.020000 + 0.000000i 0.002071 + 0.005000i 0.000000 + 0.000000i -0.0120
3 n=1;
4 Dt=.005;
5 z=(1/Dt).*ifft(y,n)

Editor
File Edit View Debug Run Help
DeuteriAskhshTritoErotimaANTISTROFOS.m
1
2 0.000000i-0.012071 + 0.005000i 0.000000 - 0.000000i 0.002071 - 0.005000i];
3
4
5

Command Window
>> DeuteriAskhshTritoErotimaANTISTROFOS

z = 4
>>
```



### Γ3

#### 1.

Για τις νότες:

Το 1/8 της νότας έχει διάρκεια 0,25 sec

το ¼=0,5 sec και

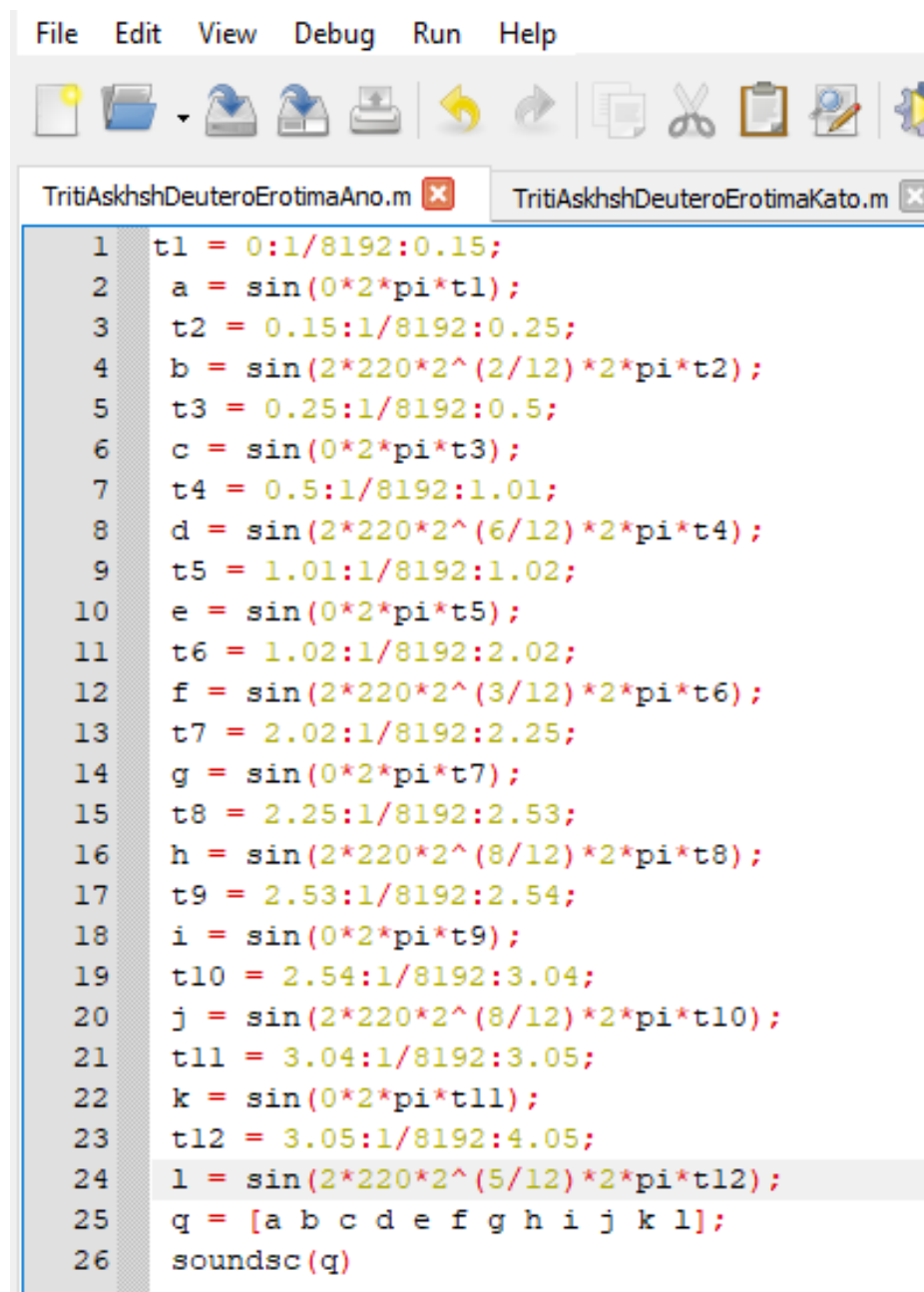
το ½ =1 sec.

Η συχνότητα δειγματοληψίας είναι στα 8000 δείγματα ανά δευτερόλεπτο. Άρα θα χρησιμοποιηθεί η  $\Delta t = (1/8) * 1024 = 1/8192$  sec (Μετατόπιση) και η συνάρτηση  $f(t) = \sin 2\pi n t$  η οποία θα μηδενίζεται για  $n=0$  Hz.

```
TritiAskshshProtoErotima.m TritiAskshshDeuteroErotima.m
1 t1 = 0:1/8192:0.15;
2 a = sin(0*2*pi*t1);
3 t2 = 0.15:1/8192:0.25;
4 b = sin(220*2^(2/12)*2*pi*t2);
5 t3 = 0.25:1/8192:0.5;
6 c = sin(0*2*pi*t3);
7 t4 = 0.5:1/8192:1.01;
8 d = sin(220*2^(6/12)*2*pi*t4);
9 t5 = 1.01:1/8192:1.02;
10 e = sin(0*2*pi*t5);
11 t6 = 1.02:1/8192:2.02;
12 f = sin(220*2^(3/12)*2*pi*t6);
13 t7 = 2.02:1/8192:2.25;
14 g = sin(0*2*pi*t7);
15 t8 = 2.25:1/8192:2.53;
16 h = sin(220*2^(8/12)*2*pi*t8);
17 t9 = 2.53:1/8192:2.54;
18 i = sin(0*2*pi*t9);
19 t10 = 2.54:1/8192:3.04;
20 j = sin(220*2^(8/12)*2*pi*t10);
21 t11 = 3.04:1/8192:3.05;
22 k = sin(0*2*pi*t11);
23 t12 = 3.05:1/8192:4.05;
24 l = sin(220*2^(5/12)*2*pi*t12);
25 q = [a b c d e f g h i j k l];
26 soundsc(q)
```

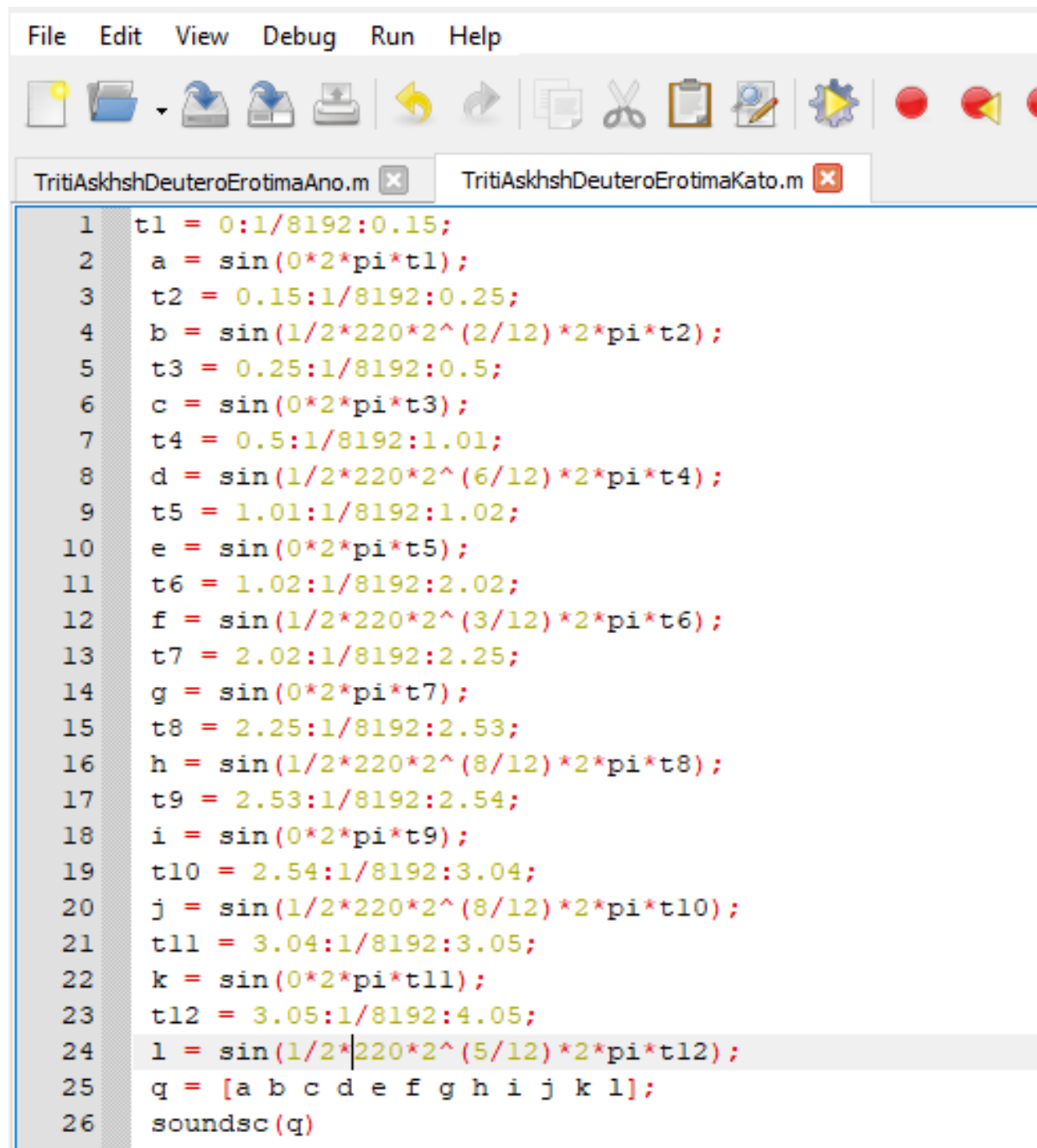
## 2

Για ολίσθηση προς τα άνω μιας οχτάδας θα διπλασιάσουμε την διάρκεια της νότας.



```
File Edit View Debug Run Help
TritiAskhshDeuteroErotimaAno.m TritiAskhshDeuteroErotimaKato.m
1 t1 = 0:1/8192:0.15;
2 a = sin(0*2*pi*t1);
3 t2 = 0.15:1/8192:0.25;
4 b = sin(2*220*2^(2/12)*2*pi*t2);
5 t3 = 0.25:1/8192:0.5;
6 c = sin(0*2*pi*t3);
7 t4 = 0.5:1/8192:1.01;
8 d = sin(2*220*2^(6/12)*2*pi*t4);
9 t5 = 1.01:1/8192:1.02;
10 e = sin(0*2*pi*t5);
11 t6 = 1.02:1/8192:2.02;
12 f = sin(2*220*2^(3/12)*2*pi*t6);
13 t7 = 2.02:1/8192:2.25;
14 g = sin(0*2*pi*t7);
15 t8 = 2.25:1/8192:2.53;
16 h = sin(2*220*2^(8/12)*2*pi*t8);
17 t9 = 2.53:1/8192:2.54;
18 i = sin(0*2*pi*t9);
19 t10 = 2.54:1/8192:3.04;
20 j = sin(2*220*2^(8/12)*2*pi*t10);
21 t11 = 3.04:1/8192:3.05;
22 k = sin(0*2*pi*t11);
23 t12 = 3.05:1/8192:4.05;
24 l = sin(2*220*2^(5/12)*2*pi*t12);
25 q = [a b c d e f g h i j k l];
26 soundsc(q)
```

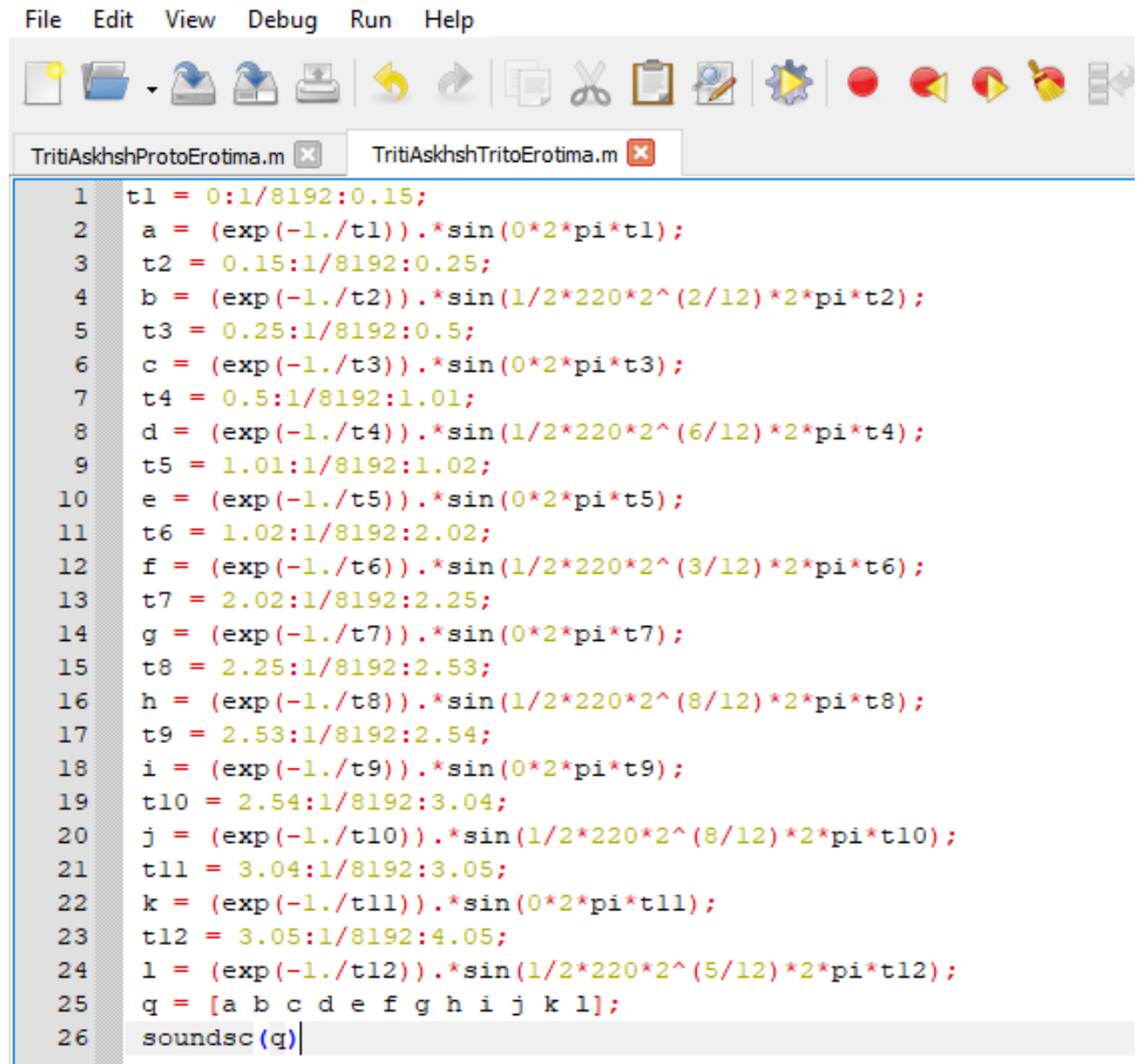
Για ολίσθηση προς τα κάτω μιας οχτάδας θα υποδιπλασιάσουμε την διάρκεια της νότας.



```
File Edit View Debug Run Help
TritiAskhshDeuteroErotimaAno.m TritiAskhshDeuteroErotimaKato.m
1 t1 = 0:1/8192:0.15;
2 a = sin(0*2*pi*t1);
3 t2 = 0.15:1/8192:0.25;
4 b = sin(1/2*220*2^(2/12)*2*pi*t2);
5 t3 = 0.25:1/8192:0.5;
6 c = sin(0*2*pi*t3);
7 t4 = 0.5:1/8192:1.01;
8 d = sin(1/2*220*2^(6/12)*2*pi*t4);
9 t5 = 1.01:1/8192:1.02;
10 e = sin(0*2*pi*t5);
11 t6 = 1.02:1/8192:2.02;
12 f = sin(1/2*220*2^(3/12)*2*pi*t6);
13 t7 = 2.02:1/8192:2.25;
14 g = sin(0*2*pi*t7);
15 t8 = 2.25:1/8192:2.53;
16 h = sin(1/2*220*2^(8/12)*2*pi*t8);
17 t9 = 2.53:1/8192:2.54;
18 i = sin(0*2*pi*t9);
19 t10 = 2.54:1/8192:3.04;
20 j = sin(1/2*220*2^(8/12)*2*pi*t10);
21 t11 = 3.04:1/8192:3.05;
22 k = sin(0*2*pi*t11);
23 t12 = 3.05:1/8192:4.05;
24 l = sin(1/2*220*2^(5/12)*2*pi*t12);
25 q = [a b c d e f g h i j k l];
26 soundsc(q)
```

### 3

Για να μετατραπεί η ένταση σε μειούμενη με τον χρόνο πολλαπλασιάσαμε την συνάντηση με την εκθετική συνάρτηση.



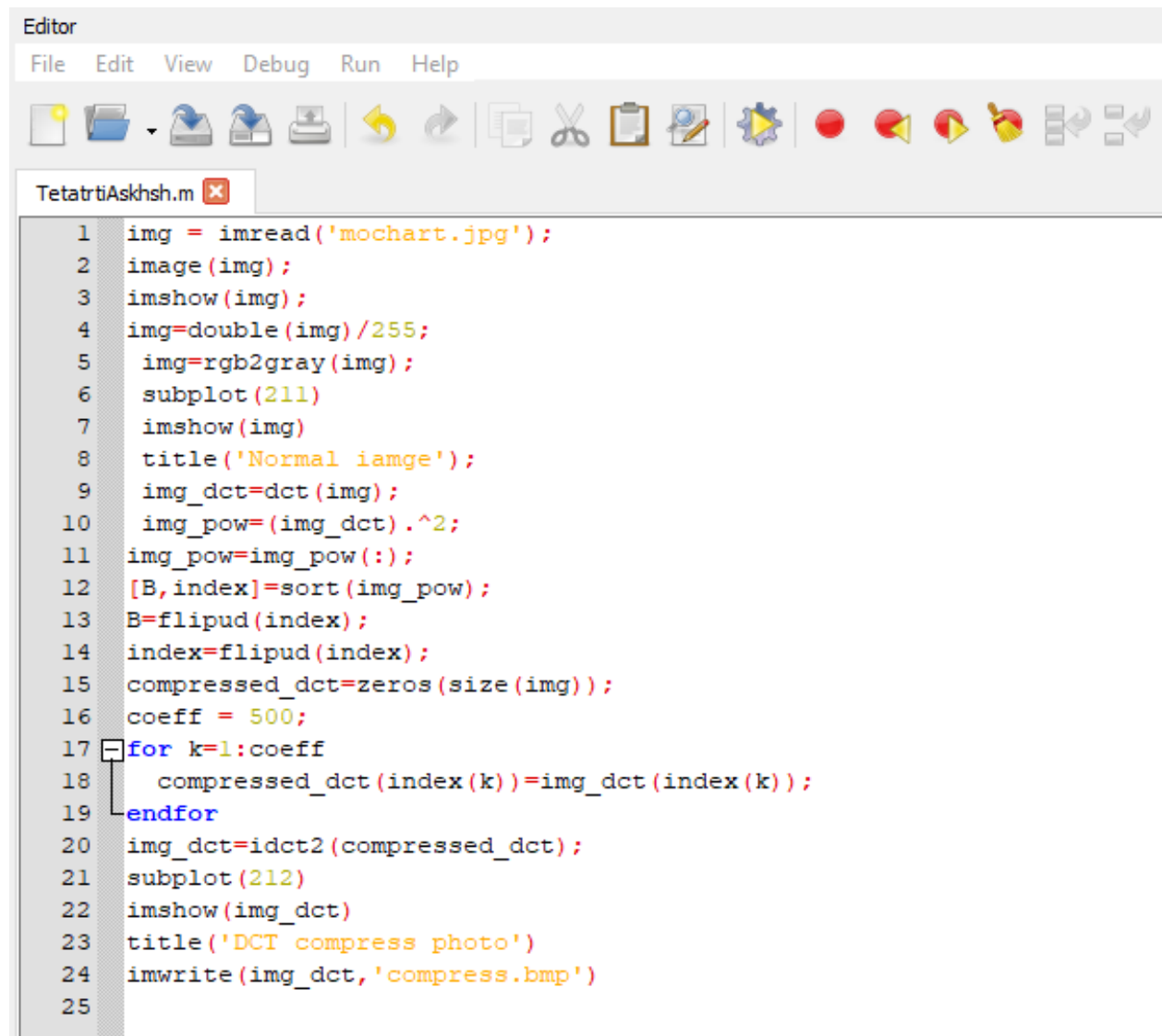
```
File Edit View Debug Run Help

TritiAskhshProtoErotima.m TritiAskhshTritoErotima.m

1 t1 = 0:1/8192:0.15;
2 a = (exp(-1./t1)).*sin(0*2*pi*t1);
3 t2 = 0.15:1/8192:0.25;
4 b = (exp(-1./t2)).*sin(1/2*220*2^(2/12)*2*pi*t2);
5 t3 = 0.25:1/8192:0.5;
6 c = (exp(-1./t3)).*sin(0*2*pi*t3);
7 t4 = 0.5:1/8192:1.01;
8 d = (exp(-1./t4)).*sin(1/2*220*2^(6/12)*2*pi*t4);
9 t5 = 1.01:1/8192:1.02;
10 e = (exp(-1./t5)).*sin(0*2*pi*t5);
11 t6 = 1.02:1/8192:2.02;
12 f = (exp(-1./t6)).*sin(1/2*220*2^(3/12)*2*pi*t6);
13 t7 = 2.02:1/8192:2.25;
14 g = (exp(-1./t7)).*sin(0*2*pi*t7);
15 t8 = 2.25:1/8192:2.53;
16 h = (exp(-1./t8)).*sin(1/2*220*2^(8/12)*2*pi*t8);
17 t9 = 2.53:1/8192:2.54;
18 i = (exp(-1./t9)).*sin(0*2*pi*t9);
19 t10 = 2.54:1/8192:3.04;
20 j = (exp(-1./t10)).*sin(1/2*220*2^(8/12)*2*pi*t10);
21 t11 = 3.04:1/8192:3.05;
22 k = (exp(-1./t11)).*sin(0*2*pi*t11);
23 t12 = 3.05:1/8192:4.05;
24 l = (exp(-1./t12)).*sin(1/2*220*2^(5/12)*2*pi*t12);
25 q = [a b c d e f g h i j k l];
26 soundsc(q)
```

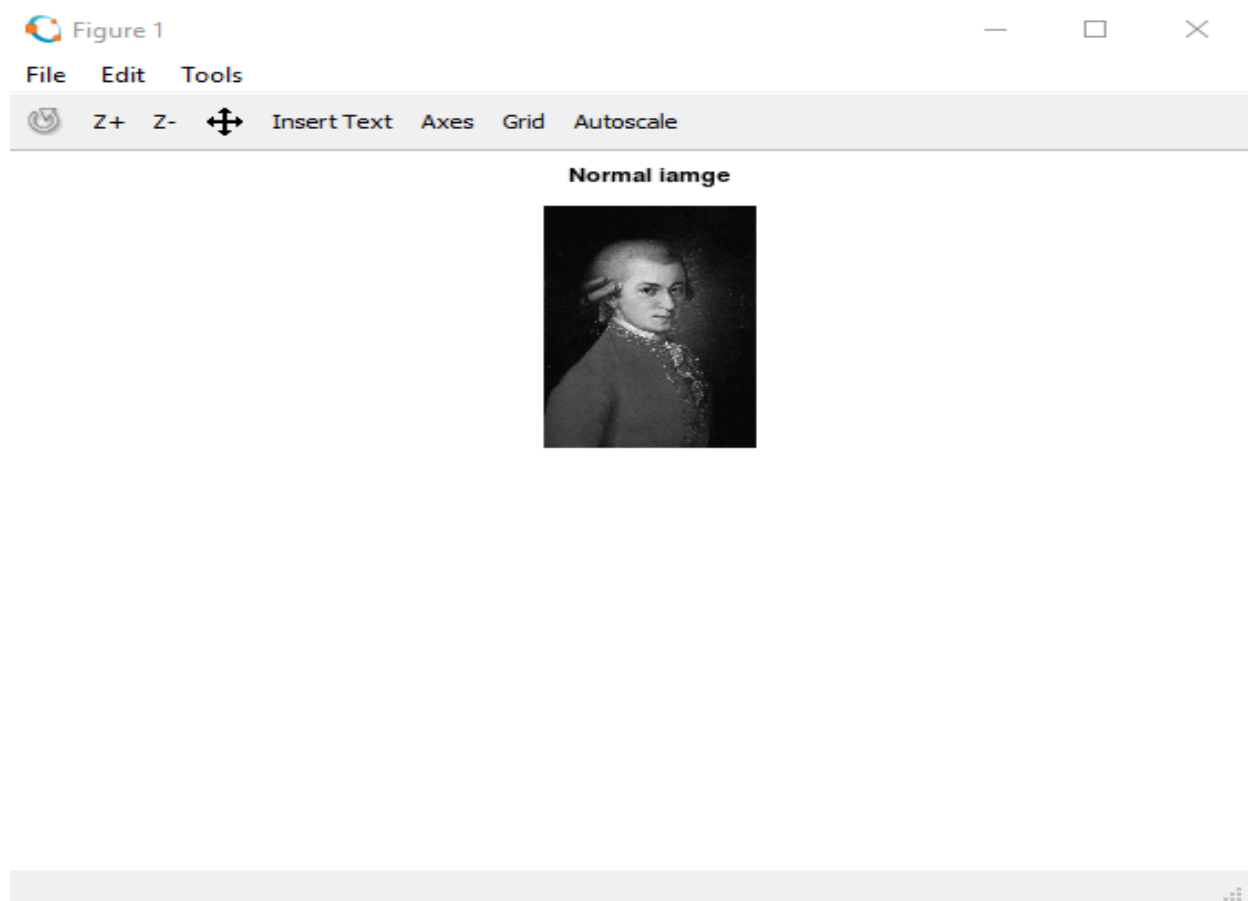
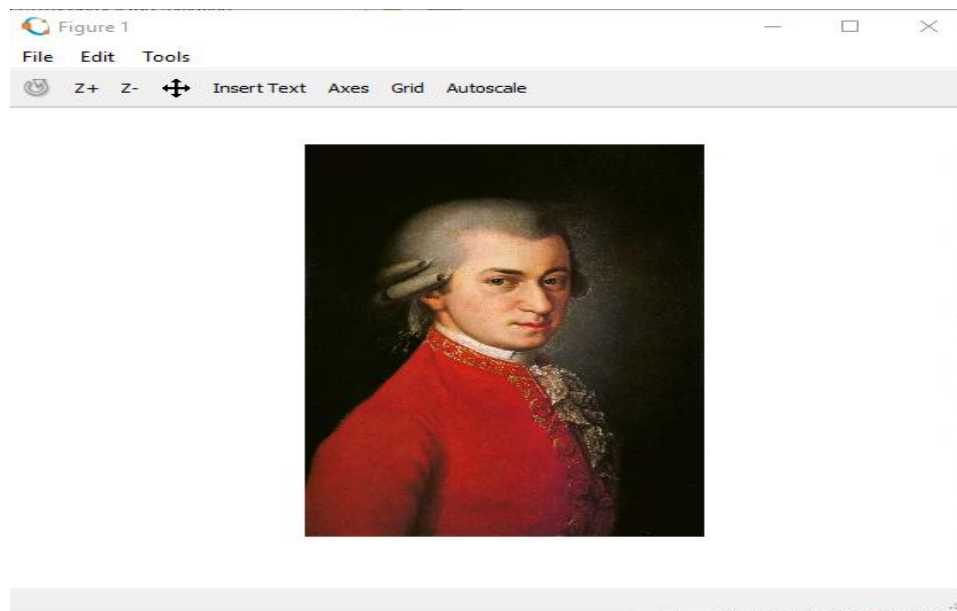
Γ4

Χρησιμοποιήσαμε το `image` package για τον discrete cosine transformation.



The image shows a MATLAB Editor window with a menu bar (File, Edit, View, Debug, Run, Help) and a toolbar. The script file is named 'TetatrtiAsksh.m'. The code performs the following steps: 1. Reads 'mochart.jpg' into 'img'. 2. Displays 'img' using 'image' and 'imshow'. 3. Converts 'img' to double and then to grayscale using 'rgb2gray'. 4. Displays the grayscale image in a subplot (2,1,1) and titles it 'Normal iamge'. 5. Computes the DCT of the grayscale image using 'dct'. 6. Squares the DCT coefficients and sorts them in descending order. 7. Flips the sorted indices vertically. 8. Creates a 'compressed\_dct' matrix of zeros with the same size as the original image. 9. Sets a coefficient threshold of 500. 10. Iterates through the sorted DCT coefficients, keeping only those greater than the threshold. 11. Reconstructs the image using 'idct2'. 12. Displays the reconstructed image in a subplot (2,1,2) and titles it 'DCT compress photo'. 13. Writes the reconstructed image to 'compress.bmp' using 'imwrite'.

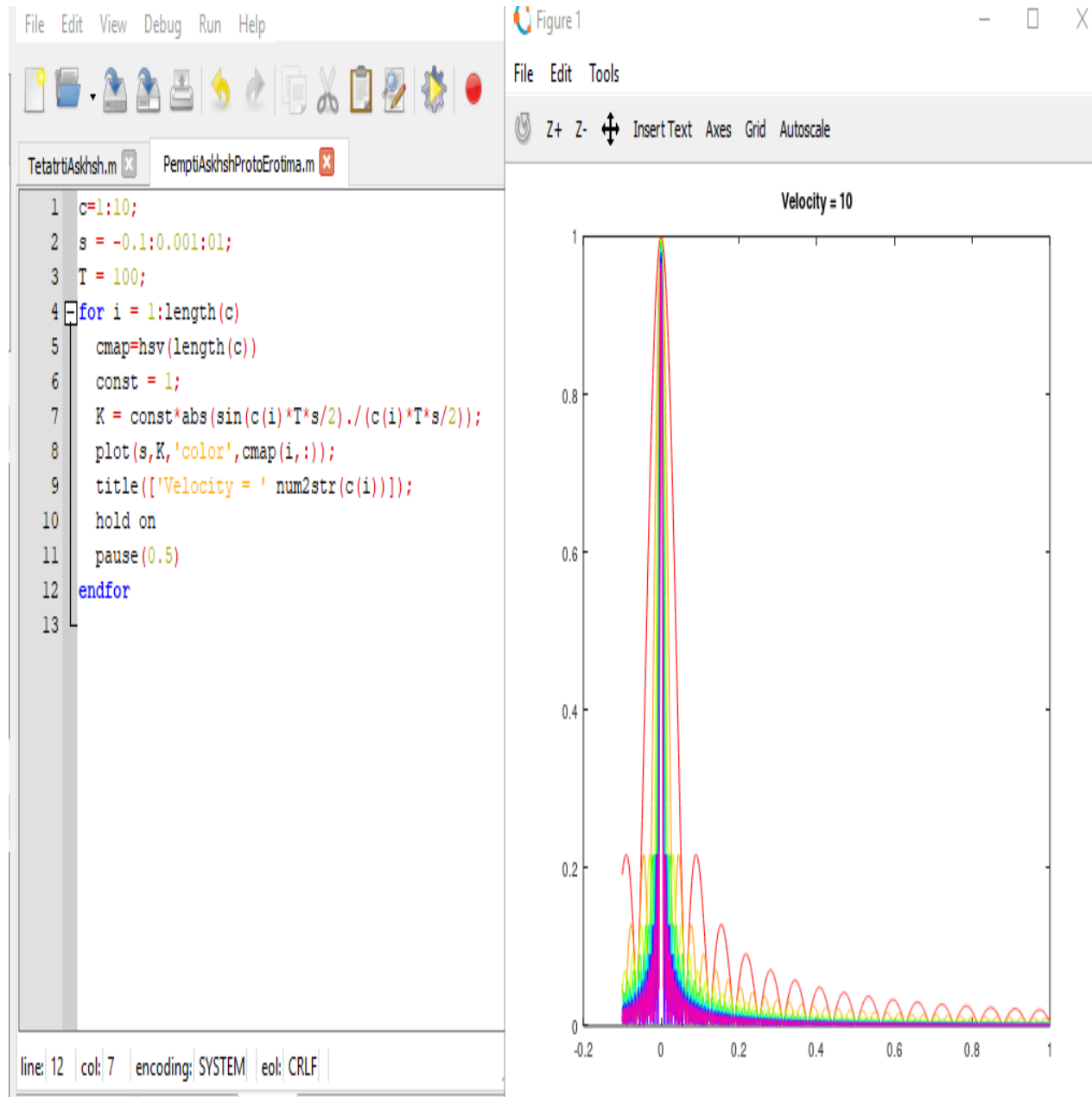
```
1  img = imread('mochart.jpg');
2  image(img);
3  imshow(img);
4  img=double(img)/255;
5  img=rgb2gray(img);
6  subplot(2,1,1)
7  imshow(img)
8  title('Normal iamge');
9  img_dct=dct(img);
10 img_pow=(img_dct).^2;
11 img_pow=img_pow(:);
12 [B,index]=sort(img_pow);
13 B=flipud(index);
14 index=flipud(index);
15 compressed_dct=zeros(size(img));
16 coeff = 500;
17 for k=1:coeff
18     compressed_dct(index(k))=img_dct(index(k));
19 endfor
20 img_dct=idct2(compressed_dct);
21 subplot(2,1,2)
22 imshow(img_dct)
23 title('DCT compress photo')
24 imwrite(img_dct,'compress.bmp')
25
```



## Γ5

**1)** Αποθηκεύουμε την τιμή του μέτρου της συνάρτησης στην μεταβλητή K.

Έτσι, θα εμφανίζονται τα κύματα τιμών για κάθε διαφορετική τιμή του μέτρου c από το 1 έως το 10.

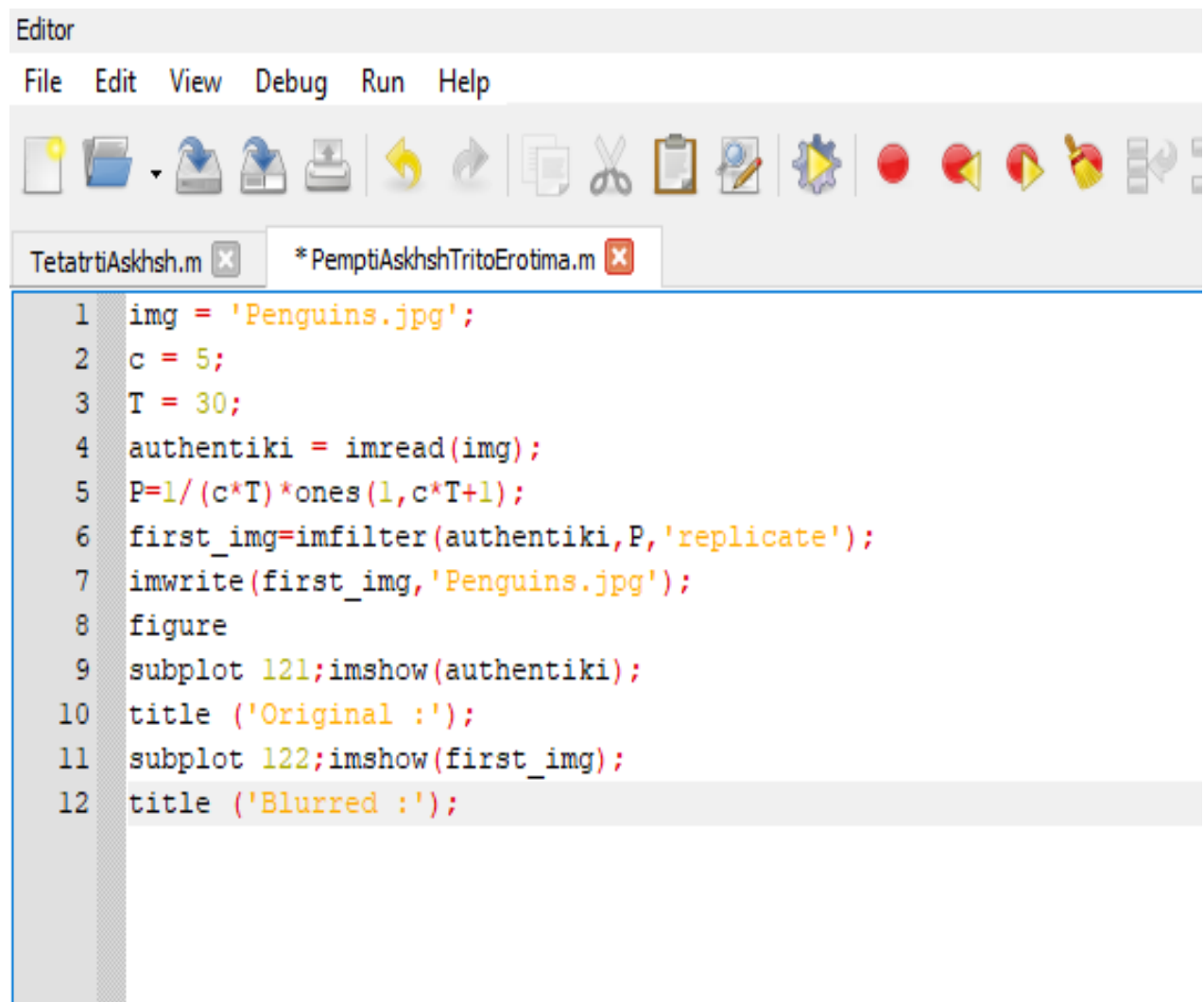


**2)**

**Η αύξηση της γραμμικής ταχύτητας  $c$  μικραίνει τις τιμές του μέτρου της συνάρτησης μεταφοράς  $K$ . Αυτό δικαιολογείται από το γεγονός πως κατά την διάρκεια εξέτασης της θέσης του πρώτου μηδενισμού στην πρώτη λύση της εξίσωσης ως προς το μέτρο  $s$ , ο μηδενισμός παρατηρείται συγκριτικά λιγότερες φορές από όταν το  $s$  δέχεται μεγαλύτερες τιμές. Επομένως, καταλήγουμε στο συμπέρασμα ότι το φαινόμενο του θολώματος γίνεται πιο έντονο και η εικόνα πιο δυσδιάκριτη.**



3)





The image shows a MATLAB Editor window with a menu bar (File, Edit, View, Debug, Run, Help) and a toolbar with various icons. Two tabs are open: 'TetatrtiAskhsh.m' and '\*PemptiAskhshTritoErotima.m'. The active tab contains the following MATLAB code:

```
1  img = 'Penguins.jpg';
2  c = 5;
3  T = 30;
4  authentiki = imread(img);
5  P=1/(c*T)*ones(1,c*T+1);
6  first_img=imfilter(authentiki,P,'replicate');
7  imwrite(first_img,'Penguins.jpg');
8  figure
9  subplot 121;imshow(authentiki);
10 title ('Original :');
11 subplot 122;imshow(first_img);
12 title ('Blurred :');
```

Figure 1



File Edit Tools

 Z+ Z-  Insert Text Axes Grid Autoscale

**Original :**



**Blurred :**



