ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ηλεκτρονικά Ισχύος

1η Εργασία

Λογισμικό

MATLAB

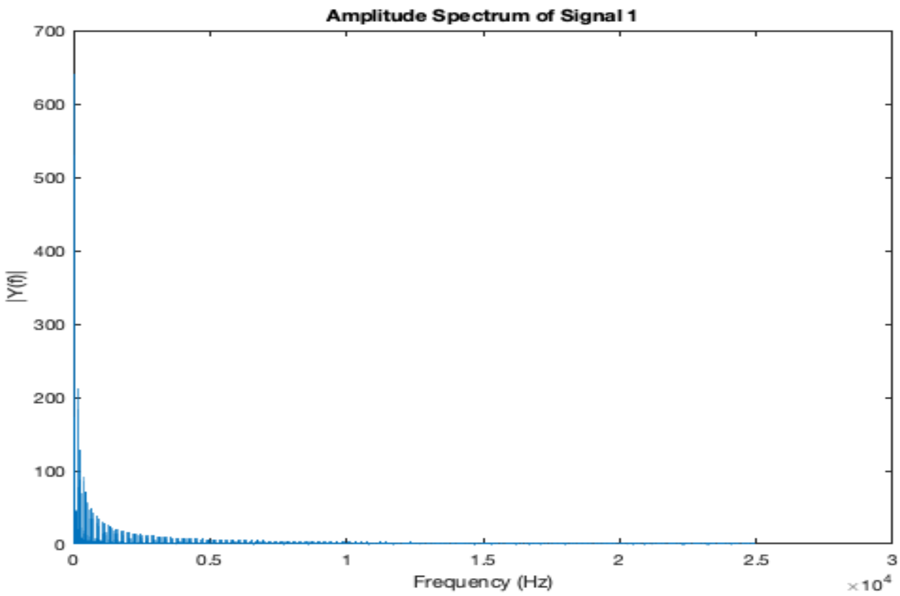
Θωμάς Χατζής *2018030134*

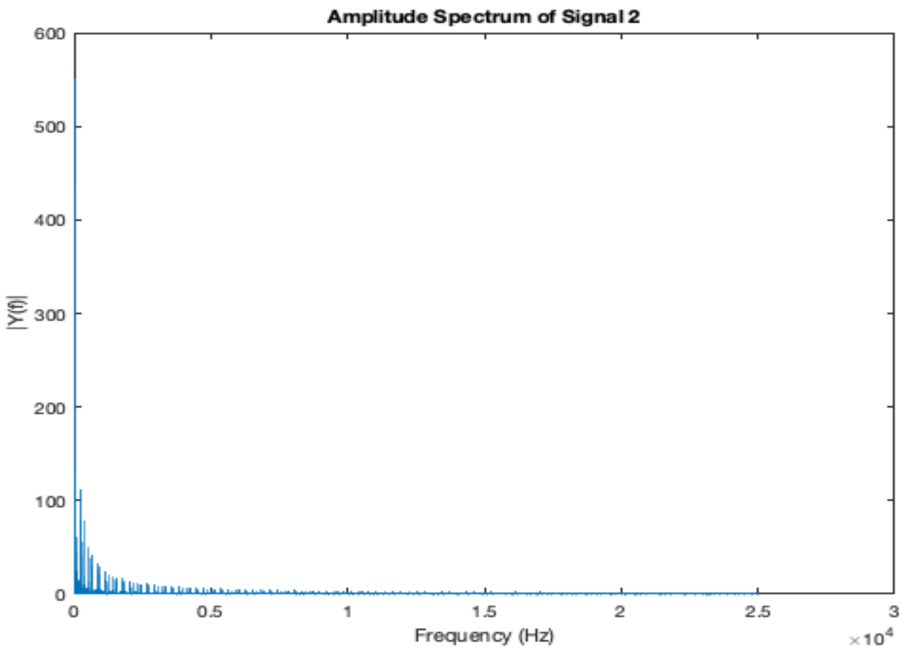
Χρήστος Παυλόπουλος 2018030139

Μάριος Σαλίνας 2018030049

Ερώτημα 1-2

Χρησιμοποιήθηκε η συνάρτηση fft του MATLAB για να γίνει ο μετασχηματισμός Fourier των σημάτων που απεικονίζονται στα σχήματα 7.A και 7.B αντίστοιχα. Ο αριθμός των σημείων του μετασχηματισμού Fourier, NFFT = 1001000 και μήκος L = 1001. Στην συνέχεια, δημιουργήθηκαν οι γραφικές παραστάσεις των φασματικών περιεχομένων των παραπάνω σημάτων, όπως φαίνονται και παρακάτω.





Από τις γραφικές παραστάσεις των φασματικών περιεχομένων των σημάτων 1,2 αντίστοιχα είναι φανερή η χαρακτηριστική μείωση του πλάτους όσο αυξάνεται ο αριθμός της αρμονικής του εκάστοτε σήματος καθώς και το μηδενικό πλάτος σε ορισμένες αρμονικές (τρίτο ερώτημα).

Ερώτημα 3

Στο ερώτημα 3 ζητείται ο υπολογισμός της $3^{ης}$ αρμονικής του σήματος signal_2. Από την γραφική παράσταση του παρακάτω σήματος γίνεται αντιληπτό ότι πρόκειται για ένα σήμα με συμμετρία περιττή ημίσεος κύματος, δηλαδή ισχύει ότι f(-t) = -f(t) και f(t) = -f(t+T/2). Έτσι, για τον υπολογισμό της $3^{ης}$ αρμονικής χρησιμοποιήθηκε ο τύπος $f(t) = a_0 + \sum_{n=1}^{\infty}[a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)]$, αφότου υπολογίστηκαν οι μεταβλητές a_n ,a_0 και b_n και n = 3 ($3^η$ αρμονική).

$$a_0 = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)dt$$

a_n = 0 for all n

$$b_n = \begin{cases} \frac{4}{\pi}\int_0^{\frac{\pi}{2}} f(t)\sin(n\omega t)\, d(\omega t) & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

Παρακάτω υπολογίζεται η παράμετρος b_n με n = 3 και f(t) = 100

$$b_3 = \begin{cases} \frac{4}{\pi}\int_0^{\frac{\pi}{2}} f(t)\sin(n\omega t)\, d(\omega t) & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

$$\Leftrightarrow b_3 = \begin{cases} \frac{4}{\pi}\int_0^{\frac{\pi}{2}} 100\sin(3\omega t)\, d(\omega t) & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

$$\Leftrightarrow b_3 = \begin{cases} \frac{400}{\pi}\left[\int_0^{\frac{\pi}{6}} 0\mathrm{dt} + \int_{\frac{\pi}{6}}^{\frac{5\pi}{6}} 100\sin(3\omega t)d(\omega t) + \int_{\frac{5\pi}{6}}^{\frac{\pi}{2}} 0\mathrm{dt}\right] & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

$$\Leftrightarrow b_3 = \begin{cases} \frac{400}{\pi}\int_{\frac{\pi}{6}}^{\frac{5\pi}{6}} 100\sin(3\omega t)d(\omega t) & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

$$\Leftrightarrow b_3 = \begin{cases} \frac{400}{\pi}\left[-\frac{\cos(3\omega t)}{3}\right]_{\frac{\pi}{6}}^{\frac{5\pi}{6}} & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

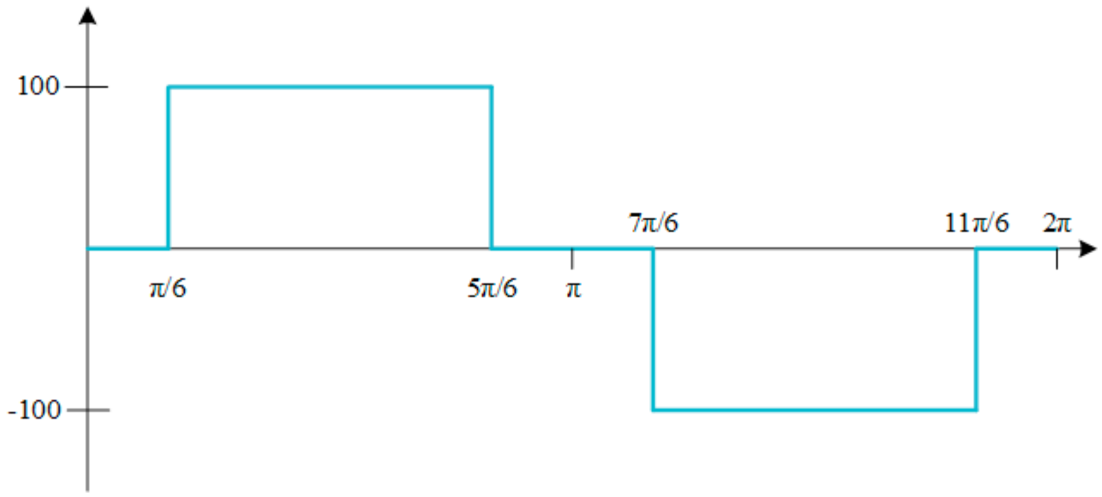
$$\Leftrightarrow \mathrm{b}_3 = \begin{cases} \frac{400}{\pi}\left[-\frac{\cos\left(\frac{15\pi}{6}\right)}{3} + \frac{\cos\left(\frac{3\pi}{6}\right)}{3}\right] & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

$$\Leftrightarrow \mathrm{b}_3 = \begin{cases} 0 & for\ odd\ n \\ 0 & for\ even\ n \end{cases}$$

$$\Leftrightarrow \mathrm{b}_3 = 0$$

Άρα η 3η αρμονική είναι: $f(t) = \alpha_0 + a_3\cos(3\omega_0 t) + b_3\sin(3\omega_0 t) = 0$

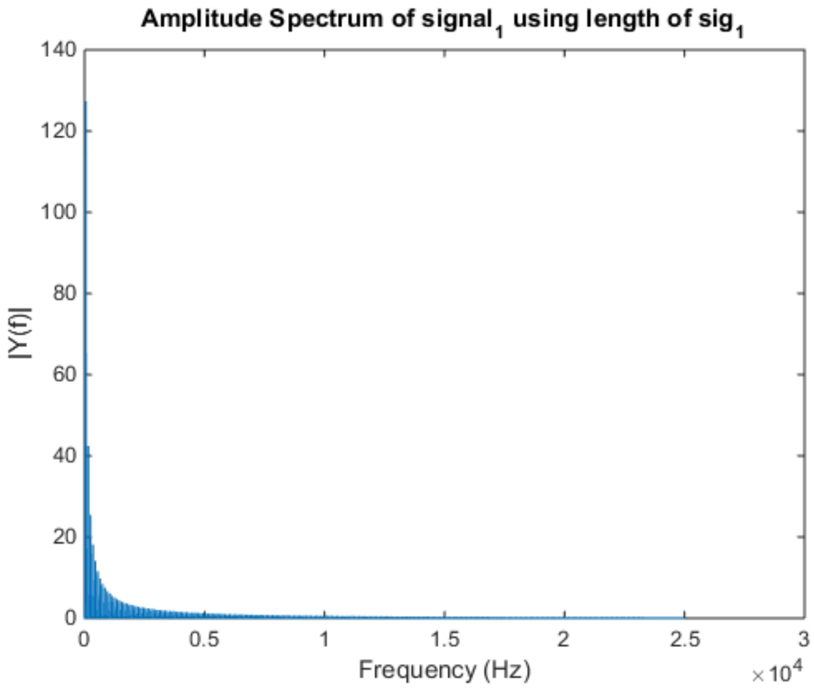
Παρατηρείται ότι ο θεωρητικός υπολογισμός της 3ης αρμονικής επιβεβαιώνει τα πειραματικά αποτελέσματα. Στο αντίστοιχο διάγραμμα του MATLAB διαπιστώνεται στην συχνότητα της 3ης αρμονικής το πλάτος να είναι σχεδόν μηδέν όσο δηλαδή υπολογίστηκε παραπάνω και με την μέθοδο της ανάλυσης σε σειρά Fourier.

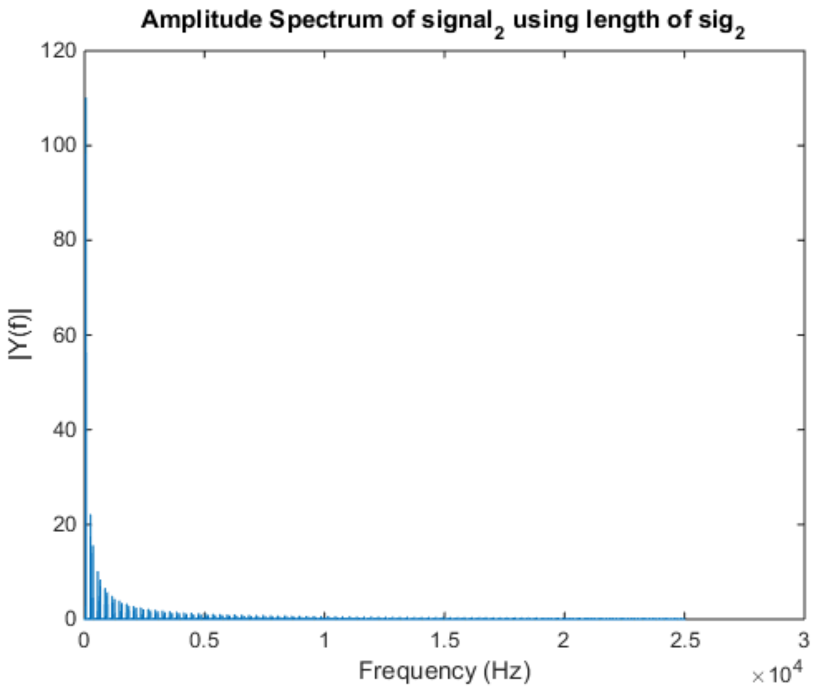


Απεικόνιση μιας περιόδου του signal 2

Ερώτημα 4-5

Επαναλήφθηκαν τα βήματα του 1[ου] ερωτήματος χρησιμοποιώντας αριθμό σημείων μετασχηματισμού Fourier, NFFT = L, όπου L το μήκος του εκάστοτε σήματος. Παρακάτω παρουσιάζονται οι γραφικές παραστάσεις των σημάτων 1,2 αντίστοιχα, αλλά με διαφορετικό μήκος L αυτή την φορά.

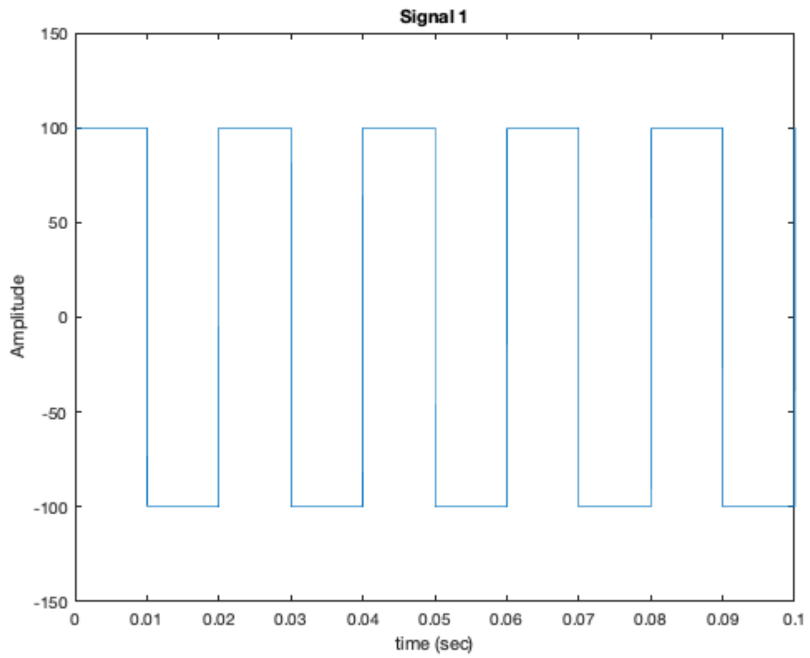




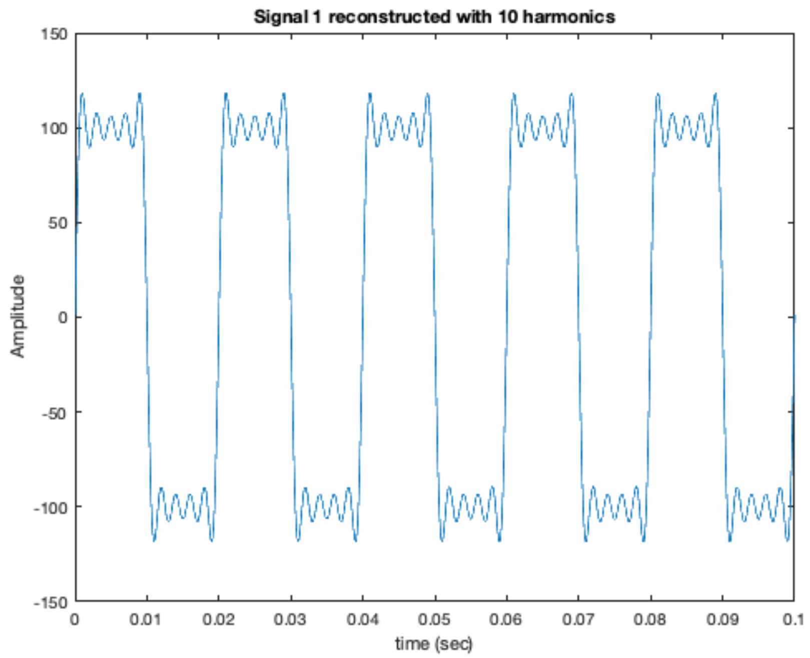
Στη συνέχεια ανακατασκευάστηκαν τα αρχικά σήματα signal_1 και signal_2 στο πεδίο του χρόνου λαμβάνοντας υπόψη κάθε φόρα για το κάθε σήμα

A) τις 10 πρώτες αρμονικές
B) τις 20 πρώτες αρμονικές
Γ) όλες τις αρμονικές

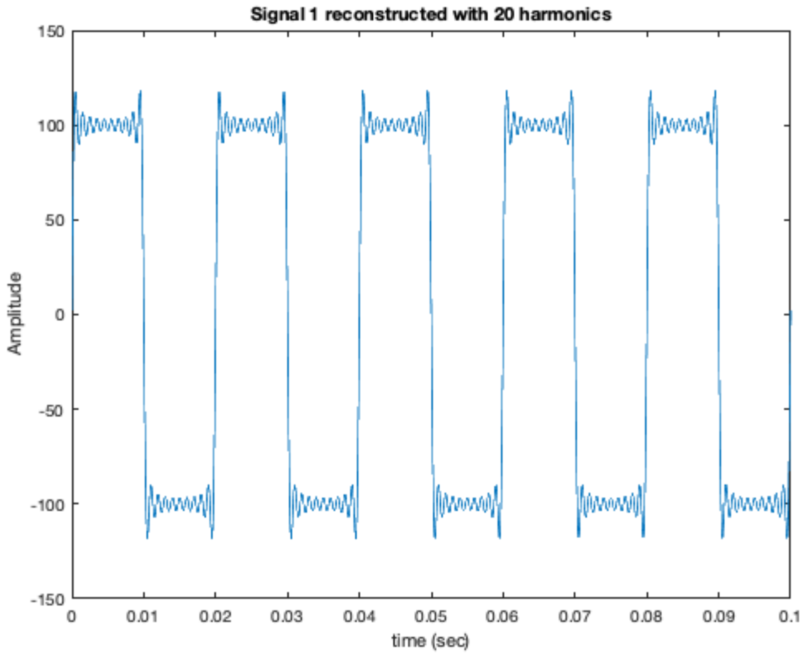
Παρακάτω απεικονίζεται το Signal 1:



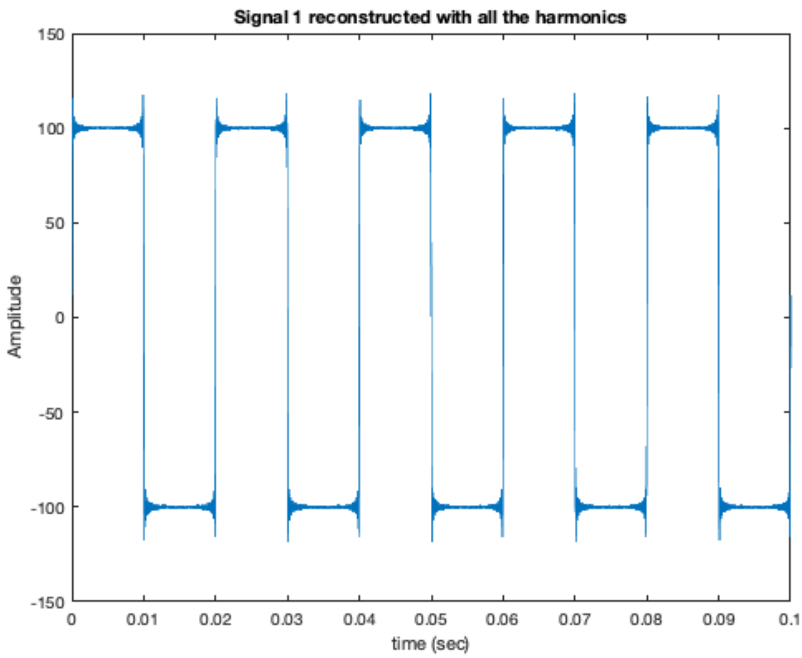
Ανακατασκευή του Signal 1 λαμβάνοντας υπόψη τις 10 πρώτες αρμονικές



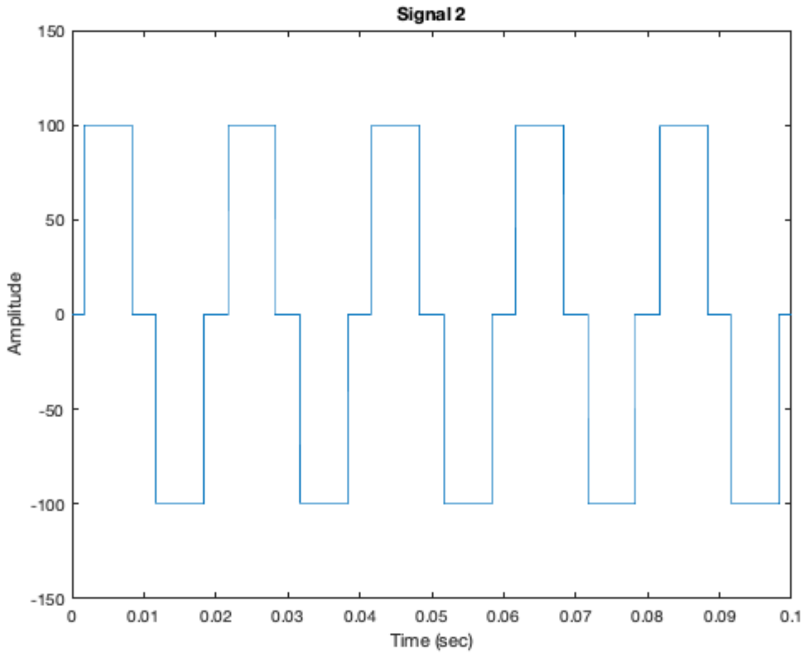
Ανακατασκευή του Signal 1 λαμβάνοντας υπόψη τις 20 πρώτες αρμονικές



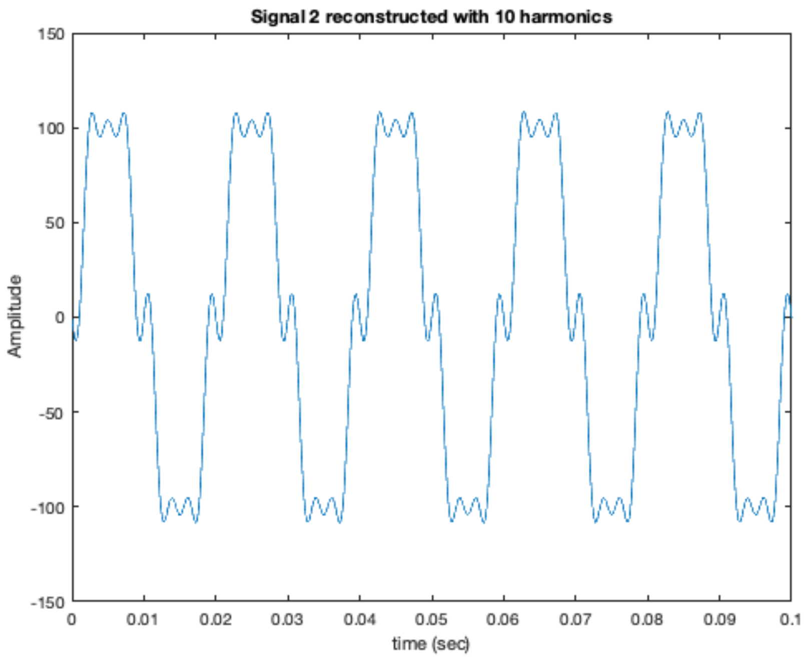
Ανακατασκευή του Signal 1 λαμβάνοντας υπόψη όλες τις αρμονικές



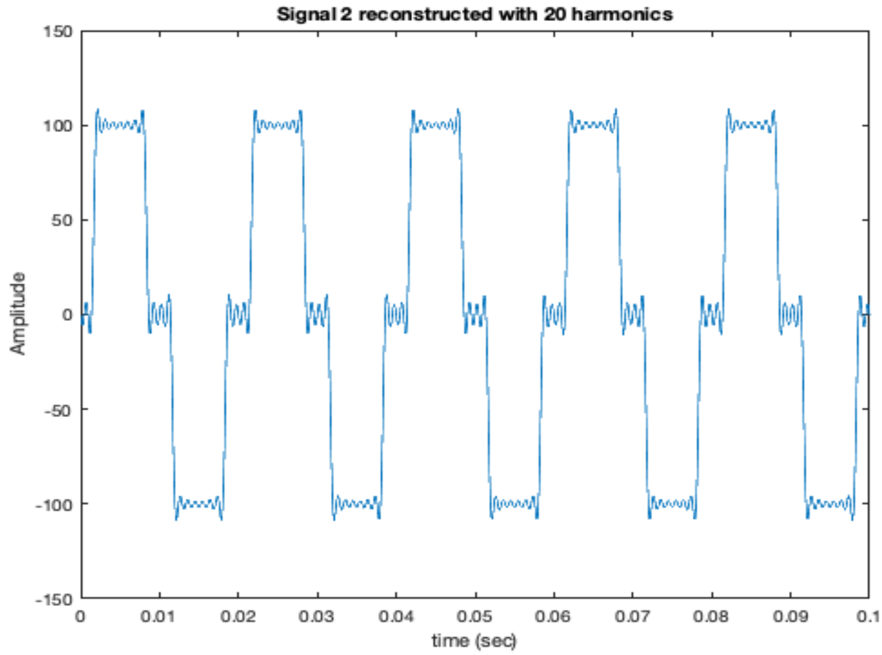
Παρακάτω απεικονίζεται το Signal 2:



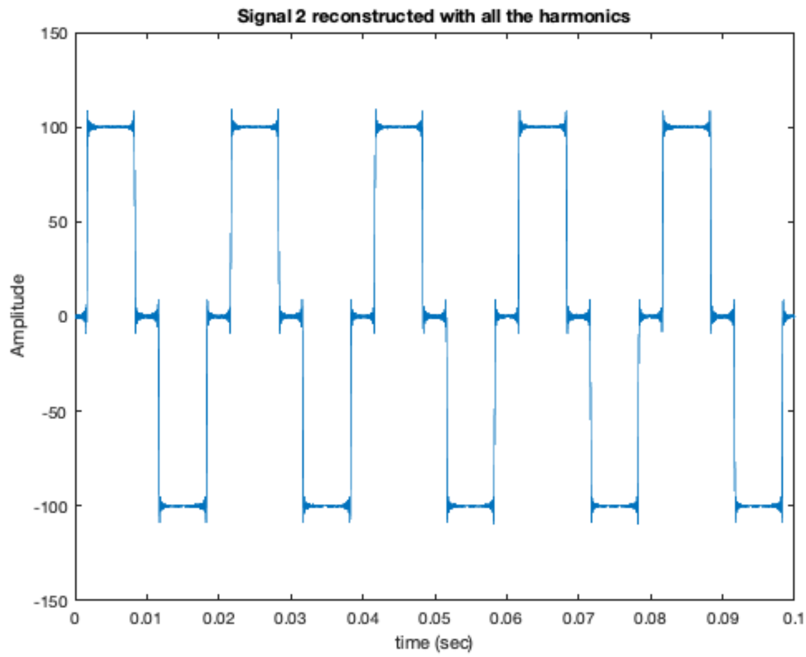
Ανακατασκευή του Signal 2 λαμβάνοντας υπόψη τις 10 πρώτες αρμονικές



Ανακατασκευή του Signal 2 λαμβάνοντας υπόψη τις 20 πρώτες αρμονικές



Ανακατασκευή του Signal 2 λαμβάνοντας υπόψη όλες τις αρμονικές



Γίνεται εύκολα αντιληπτή η σταδιακή βελτίωση και η πιο ακριβής ανακατασκευή του κάθε σήματος χρησιμοποιώντας μεγαλύτερο αριθμό αρμονικών κάθε φορά, αφού μειώνονται και οι κυματώσεις των σημάτων με την αύξηση του αριθμού των αρμονικών. Τέλος, χρησιμοποιώντας όλες τις αρμονικές το αποτέλεσμα είναι αρκετά πανομοιότυπο με αυτό του αρχικού σήματος.

Ερώτημα 6

ΚΩΔΙΚΑΣ MATLAB

```
close all
clear all
load('signal_1.mat');
load('signal_2.mat');
load('time.mat');

%% Exercise_1-2

% Sampling period
Ts = 2*(10^-5);
% Sampling frequency
Fs = 1/Ts;
% Length of signal
L = 1001;

%Fourrier Transformation of signal_1
NFFT = 1001000;
Y1 = fft(signal_1,NFFT)/L;
f2 = Fs/2*linspace(0,1,NFFT/2+1);

%Fourrier Transformation of signal_2
Y2 = fft(signal_2,NFFT)/L;
f2 = Fs/2*linspace(0,1,NFFT/2+1);

%Plot Fourrier Transform of signal_1
figure(1);
plot(f2,2*abs(Y1(1:NFFT/2+1)));
title('Amplitude Spectrum of Signal 1');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

%Plot Fourrier Transform of signal_2
figure(2);
plot(f2,2*abs(Y2(1:NFFT/2+1)));
title('Amplitude Spectrum of Signal 2');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

%% Exercise 4-5

% Sampling period
Ts = 2*(10^-5);
% Sampling frequency
Fs = 1/Ts;
% Length of signal_1
L1 = length(signal_1);
% Length of signal_2
L2 = length(signal_2);
```

```
%Fourier Transformation of signal_1 using length of signal_1
NFFT1 = L1;
Y1_2 = fft(signal_1,NFFT1)/L1;
f1 = Fs/2*linspace(0,1,NFFT1/2+1);

%Fourier Transformation of signal_2 using length of signal_2
NFFT2 = L2;
Y2_2 = fft(signal_2,NFFT2)/L2;
f2 = Fs/2*linspace(0,1,NFFT2/2+1);

%Plot Fourrier Transform of signal_1 using signal_1's length
%It's not required to plot the results but we do it in order
to
%see the difference between the 2 length vectors.
figure(3);
plot(f1,2*abs(Y1_2(1:NFFT1/2+1)))
title('Amplitude Spectrum of signal 1 using length of Signal
1');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

%Plot Fourrier Transform of signal_2 using signal_2's length
%It's not required to plot the results but we do it in order
to
%see the difference between the 2 length vectors.
figure(4);
plot(f2,2*abs(Y2_2(1:NFFT2/2+1)));
title('Amplitude Spectrum of signal 2 using length of Signal
2');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

%% SIGNAL 1

%Ploting the given signal_1
figure(5);
plot(time, signal_1)
title('Signal 1');
xlabel('time (sec)');
ylabel('Amplitude');
axis ([0 0.1 -150 150]);

% We reconstruct the signals 1,2 using fourier series
% signal(t) = a0 + Sum(an*cos(omega*n*t)+bn*sin(omega*n*t))

%First real value of Y1
a0 = real(Y1_2(1));

%Initializing signal1 to be used on for loop
signal1_10har = a0*ones(1,L1);

%Rounding the frequency vector in order to be at 150hz,250Hz,
etc
round_f1=round(f1,-1);
```

```
%Using for loop to take the first 10 harmonics
for k=1:10
    %finding the fourier coefficient
    fc=Y1_2(k*50 == round_f1);
    an=2*real(fc);
    bn=(-2)*imag(fc);
    signal1_10har = signal1_10har +
an*cos(k*2*pi*50*time)+bn*sin(k*2*pi*50*time);
end

figure(6);
plot(time, signal1_10har);
title('Signal 1 reconstructed with 10 harmonics')
xlabel('time (sec)');
ylabel('Amplitude');

%Initializing signal1 to be used on for loop
signal1_20har = a0*ones(1,L1);

%20 Harmonincs signal recreation
for k=1:20
    fc=Y1_2(k*50 == round_f1);
    an=2*real(fc);
    bn=(-2)*imag(fc);
    signal1_20har = signal1_20har +
an*cos(k*2*pi*50*time)+bn*sin(k*2*pi*50*time);
end

figure(7);
plot(time, signal1_20har);
title('Signal 1 reconstructed with 20 harmonics')
xlabel('time (sec)');
ylabel('Amplitude');


%Initializing signal1 to be used on for
signal1_allhar = a0*ones(1,L1);

%All harmonics signal recreation
for k=1:L1/50
    fc=Y1_2(k*50 == round_f1);
    an=2*real(fc);
    bn=(-2)*imag(fc);
    signal1_allhar = signal1_allhar +
an*cos(k*2*pi*50*time)+bn*sin(k*2*pi*50*time);
end

figure(8);
plot(time, signal1_allhar);
title('Signal 1 reconstructed with all the harmonics')
xlabel('time (sec)');
ylabel('Amplitude');
```

```
%% SIGNAL 2

%Plotting the given signal 2
figure(9);
plot(time, signal_2)
title('Signal 2');
xlabel('Time (sec)');
ylabel('Amplitude');
axis ([0 0.1 -150 150]);

%A0 is the first real value of Y2 vector
a0_2 = real(Y2_2(1));

%Initializing signal2 to be used on for loop
signal2_10har = a0_2*ones(1,L2);

%Rounding the f2 vector values so it's 150hz,250hz,etc
round_f2=round(f2,-1);

%For method to obtain signal 2 from 10 harmonics
for p=1:10
    fc2=Y2_2(p*50 == round_f2);
    an2=2*real(fc2);
    bn2=(-2)*imag(fc2);
    signal2_10har = signal2_10har +
an2*cos(p*2*pi*50*time)+bn2*sin(p*2*pi*50*time);
end

figure(10);
plot(time, signal2_10har);
title('Signal 2 reconstructed with 10 harmonics');
xlabel('time (sec)');
ylabel('Amplitude');


signal2_20har = a0_2*ones(1,L2);

%20 harmonics signal2 recreation
for p=1:20
    fc2=Y2_2(p*50 == round_f2);
    an2=2*real(fc2);
    bn2=(-2)*imag(fc2);
    signal2_20har = signal2_20har +
an2*cos(p*2*pi*50*time)+bn2*sin(p*2*pi*50*time);
end

figure(11);
plot(time, signal2_20har);
title('Signal 2 reconstructed with 20 harmonics');
xlabel('time (sec)');
ylabel('Amplitude');
```

```
%All harmonics signal2 recreation
signal2_allhar = a0*ones(1,L2);

for p=1:L2/50
    fc2=Y2_2(p*50 == round_f2);
    an2=2*real(fc2);
    bn2=(-2)*imag(fc2);
    signal2_allhar = signal2_allhar +
an2*cos(p*2*pi*50*time)+bn2*sin(p*2*pi*50*time);
end

figure(12);
plot(time, signal2_allhar);
title('Signal 2 reconstructed with all the harmonics');
xlabel('time (sec)');
ylabel('Amplitude');
```