ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER
ΚΑΙ ΜΕΤΑΤΡΟΠΗ ΨΗΦΙΑΚΟΥ ΣΗΜΑΤΟΣ ΣΕ ΑΝΑΛΟΓΙΚΟ

Δημήτρης Κ. Ιακωβίδης

Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική, Πανεπιστήμιο Θεσσαλίας

Στόχος αυτής της ενότητας είναι η εισαγωγή στη φασματική ανάλυση ψηφιακών σημάτων με τη χρήση του μετασχηματισμού Fourier. Εξηγείται η χρήση του αντίστροφου μετασχηματισμού Fourier στο πλαίσιο της ανάλυσης, επεξεργασίας και επανασύνθεσης ενός συνθετικού σήματος.

5.1 Φάσμα συχνοτήτων σήματος

Σύμφωνα με το θεώρημα του Φουριέ ένα **στατικό** (stationary / δηλ. σταθερό ως προς τις στατιστικές παραμέτρους του στο χρόνο) σήμα μπορεί να συντεθεί ως άθροισμα πολλών ημιτονοειδών σημάτων με διαφορετικές συχνότητες.

Το **φάσμα συχνοτήτων** (frequency spectrum) ενός σήματος δείχνει τι συχνότητες υπάρχουν στο σήμα μας, δηλαδή, τις συχνότητες διαφορετικών ημιτονοειδών που μπορούν να προστεθούν για να συνθέσουν το σήμα μας.

5.2 Ευθύς μετασχηματισμός Φουριέ

Το φάσμα συχνοτήτων μπορεί να υπολογιστεί με τη χρήση του μετασχηματισμού Φουριέ (Fourier Transform), ο οποίος μετασχηματίζει ένα συνεχές σήμα x_c από το πεδίο του χρόνου t στο πεδίο των συχνοτήτων ω:

$$X_c(j\omega) = \int_{-\infty}^{\infty} x_c(t) e^{-j\omega t} dt \tag{5}$$

Για τα ψηφιακά σήματα (δηλ. τα σήματα διακριτού χρόνου και διακριτών τιμών), αποδεικνύεται ότι ο μετασχηματισμός αυτός λαμβάνει τη μορφή:

$$X[k] = \sum_{n=0}^{L-1} x[n] e^{-j(2\pi/N)kn} \qquad k = 0, 1, \ldots, N-1 \tag{6}$$

για $t = nT_s$, όπου T_s η περίοδος δειγματοληψίας και n ο αύξων αριθμός των δειγμάτων (δηλ. ο δείκτης του πίνακα στον οποίο αποθηκεύεται το ψηφιακό σήμα) και η γωνιακή συχνότητα προκύπτει από τη σχέση:

$$\omega_k = \frac{2\pi k}{N T_s} \tag{7}$$

Στο Matlab ο **διακριτός μετασχηματισμός Φουριέ (Discrete Fourier Transform, DFT)** υλοποιείται μέσω του αλγορίθμου που ονομάζεται γρήγορος μετασχηματισμός Φουριέ (Fast Fourier Transform, FFT). Η συνάρτηση fft καλείται ως εξής:

y = fft(x)

όπου x το σήμα εισόδου που περιγράφεται στο πεδίο του χρόνου, και y το μετασχηματισμένο σήμα που περιγράφει το φάσμα του σήματος ως διακριτό σήμα ως προς *k* (5). Για να μπορέσουμε να λάβουμε το φάσμα ως προς τη συχνότητα θα πρέπει να λάβουμε υπόψη μας τη σχέση (6).

Στο Σχ. 2 εικονίζεται μια συνάρτηση Matlab που επιστρέφει το φάσμα συναρτήσει της συχνότητας. Επειδή το επιστρεφόμενο σήμα y είναι μιγαδικό, επιστρέφεται ξεχωριστά το μέτρο (magnitude) και η φάση του (phase).

```
% A Matlab DSP Example by D.K. Iakovidis
% Calculates the Fourier spectrum of any signal x

function [frequency, magnitude, phase] = fftFun(x, fs)

% Fast Fourier Transform
y = fft(x);

% y is a complex discrete signal with m=length(y) samples
% the samples correspond to 0, fs/m, 2fs/m...
m = length(y);
frequencyAxis = (0:m)*fs/m;
fftMagnitude = abs(y);
fftPhase = angle(y);

% Construct the frequency axis
frequency =
[-(frequencyAxis(m)-frequencyAxis(m/2:m)) frequencyAxis(1:m/2)];
magnitude = [fftMagnitude(m/2:m) fftMagnitude(1:m/2)];
phase = [fftPhase(m/2:m) fftPhase(1:m/2)];
```

Σχήμα 2. Συνάρτηση Matlab με την οποία λαμβάνεται το φάσμα ενός σήματος x.

Στη συνάρτηση αυτή η γραμμή 12 εξηγείται από την εξίσωση (7) γιατί ω = 2πf => f = ω / 2π => f = k / NT_s = kf_s / N. Στον κώδικα η μεταβλητή m αντιστοιχεί στο πλήθος N των δειγμάτων της συχνότητας που περιγράφει η εξίσωση (7). Τα μισά από αυτά τα δείγματα αντιστοιχούν στις αρνητικές συχνότητες και τα άλλα μισά στις θετικές.

Στο Σχ.3 εικονίζεται ένα πρόγραμμα Matlab που χρησιμοποιεί τη συνάρτηση fftFun (Σχ.2) για την απεικόνιση του φάσματος ενός ημιτονοειδούς σήματος σε δύο υπο-γραφήματα, το πρώτο για το πλάτος του και το δεύτερο για τη φάση του. Αναμένεται το πλάτος του φάσματος να παρουσιάζει δύο κορυφές στις συχνότητες +f και –f.

```
% A Matlab DSP Example by D.K. Iakovidis
% Calculates the Fourier spectrum using fftFun function

fs = 1000;              % sampling frequency
Ts = 1/fs;              % sampling period
tmin = 0;               % time axis min value
tmax = 10;              % time axis max value
t = tmin:Ts:tmax;       % time axis sampled every Ts

f = 100;                % sinusoid frequency
A = 10;                 % sinusoid amplitude
phi = 0;                % sinusoid phase

% Calculate sinusoid
x = A * cos(2*pi*f*t + phi);

% Spectrum
[frequency, magnitude, phase] = fftFun(x, fs);

% Plot
figure;
subplot(2,1,1);
plot( frequency, magnitude );
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(2,1,2);
plot( frequency, phase );
xlabel('Frequency (Hz)');
ylabel('Phase');
```

Σχήμα 3. Πρόγραμμα που απεικονίζει το πλάτος και τη φάση του φάσματος ενός ημιτονοειδούς σήματος.

5.3 Αντίστροφος μετασχηματισμός Φουριέ

Ο αντίστροφος μετασχηματισμός Φουριέ χρησιμοποιείται για τη σύνθεση ενός σήματος από το φάσμα του. Δηλαδή αποσκοπεί στο μετασχηματισμό του σήματός μας από το πεδίο των συχνοτήτων στο πεδίο του χρόνου.

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}[k]e^{j(2\pi k/N)n}$$

$$0 \le n \le N-1 \tag{8}$$

όπου X(k) το φάσμα και x[n] το ψηφιακό σήμα που λαμβάνουμε ως αποτέλεσμα, και η συχνότητα προκύπτει από τη σχέση:

$$\omega_k = \frac{2\pi k}{NT_s} \tag{9}$$

Στο Matlab ο αντίστοιχος μετασχηματισμός προκύπτει με τη χρήση της συνάρτησης ifft ως εξής:

y = ifft(x)

όπου το y είναι σε μιγαδική μορφή.

Ως παράδειγμα δίνεται ο κώδικας του Σχ.4, όπου ένα αρχικό σήμα (άθροισμα τριών ημιτονοειδών), από το **πεδίο του χρόνου**, μετασχηματίζεται με την **fft** στο **πεδίο των συχνοτήτων**, και μετά το μετασχηματισμένο σήμα εισάγεται στην **ifft** για να μετασχηματιστεί αντίστροφα και να μας δώσει το αρχικό σήμα **στο πεδίο του χρόνου**.

```
% A Matlab DSP Example by D.K. Iakovidis
% Demonstrates the use of inverse fft (ifft)

fs = 1000;             % sampling frequency
Ts = 1/fs;             % sampling period
tmin = 0;              % time axis min value
tmax = 0.05;           % time axis max value
t = tmin:Ts:tmax;      % time axis sampled every Ts

f1 = 100;              % sinusoid frequency
f2 = 200;
f3 = 300;
A = 10;                % sinusoid amplitude
phi = 0;               % sinusoid phase
```

```matlab

% Calculate signal
x1 = A * cos(2*pi*f1*t + phi);
x2 = A * cos(2*pi*f2*t + phi);
x3 = A * cos(2*pi*f3*t + phi);
x  = x1+x2+x3;

figure;
plot(t,x,’o’);
title('Original signal');

% Calculate DFT (via Fast Fourier Transform)
y = fft(x);

% Plot DFT
figure;
plot(abs(y));
title('Spectrum');

% Calculate Inverse DFT (to return to the time domain)
x = ifft(y);

% Plot signal
figure;
plot(t,real(x));
title('Reconstructed signal');
```

Σχήμα 4. Παράδειγμα αντίστροφου μετασχηματισμού Φουριέ.

5.4 Ανακατασκευή αναλογικού σήματος από ψηφιακό

Ένα αναλογικό σήμα y(t) μπορεί να ληφθεί από το αντίστοιχο ψηφιακό του y[n] μέσω της ακόλουθης σχέσης:

$$y(t) = \sum_{n=-\infty}^{\infty} y[n]p(t - nT_s) \tag{10}$$

όπου p(t) μια συνάρτηση παλμού όπως είναι ο τετραγωνικός παλμός ή ο ιδανικός παλμός που εκφράζεται μέσω της συνάρτησης sinc(t) = sin(πt)/πt.

Στο Σχ.5 εικονίζεται ένα παράδειγμα υλοποίησης.

```matlab
% A Matlab DSP Example by D.K. Iakovidis
% Digital to Analog signal Conversion (DAC)

fs = 1000;            % sampling frequency
Ts = 1/fs;            % sampling period
tmin = 0;             % time axis min value
tmax = 1;             % time axis max value
t = tmin:Ts:tmax;     % time axis sampled every Ts
```

```

f1 = 10;            % sinusoid frequency
f2 = 20;
f3 = 30;
A = 10;             % sinusoid amplitude
phi = 0;            % sinusoid phase

% Calculate the digital signal
x1 = A * cos(2*pi*f1*t + phi);
x2 = A * cos(2*pi*f2*t + phi);
x3 = A * cos(2*pi*f3*t + phi);
x  = x1 + x2 + x3;

% Make x look clearly discrete
N = length(x);
n = 1:N;

figure;
subplot(311);
plot(t,x);
title('Original signal');

subplot(312);
plot(n,x);
title('Discrete signal');
axis([0 1000 -5*A 5*A]);

% Reconstruct signal from its samples (D/A)
reconstructed = [];
for n=1:N
    %reconstructed(n) = sum(x(n).*sinc(t(n)-n*Ts));  %
using sinc
    reconstructed(n) = sum(x(n).*rectpuls(t(n)-n*Ts));  %
using pulse
end

subplot(313);
plot(t, reconstructed);
title('Reconstructed signal');

% Plot spectra
figure;
[f,a,p] = fftFun(x,fs);
subplot(211);
plot(f,a);
axis([-100 100 0 6000]);
title('Spectrum of the original signal');
[f,a,p] = fftFun(reconstructed,fs);
subplot(212);
plot(f,a);
axis([-100 100 0 6000]);
title('Spectrum of the reconstructed signal');
```

Σχήμα 5. Παράδειγμα ανακατασκευής αναλογικού σήματος από τα δείγματα ενός ψηφιακού σήματος.

Ασκήσεις

1. Να κατασκευάσετε το άθροισμα x των ημιτονοειδών που υλοποιείται στις γραμμές 1-20 του παραδείγματος του Σχ. 4.
    a. Εφαρμόστε τη συνάρτηση fft του Matlab και απεικονίστε το μέτρο και τη φάση του DFT του x. Παρατηρήστε ότι ο οριζόντιος άξονας μετράει δείγματα και όχι συχνότητα.
    b. Εφαρμόστε τη συνάρτηση fftFun του Σχ.2 για να απεικονίσετε το μέτρο και τη φάση του DFT του σήματος x ως προς τη συχνότητα.
    c. Χρησιμοποιήστε τον αντίστροφο μετασχηματισμό Φουριέ (IDFT) όπως στο Σχ.4 για να λάβετε το σήμα x και πάλι στο πεδίο του χρόνου. Παρατηρήστε ότι το σήμα x παραμένει αναλλοίωτο.
    d. Δοκιμάστε να μηδενίσετε κάποιες κορυφές του φάσματος πριν εφαρμόσετε τον αντίστροφο μετασχηματισμό Φουριέ και παρατηρείστε το αποτέλεσμα.

2. Να κατασκευάσετε τη συνάρτηση sinc συναρτήσει του χρόνου και να την απεικονίσετε.

3. Να υλοποιήσετε το παράδειγμα του Σχ.5. Δοκιμάστε διαφορετικές συχνότητες δειγματοληψίας fsR και παρατηρείστε το αποτέλεσμα.

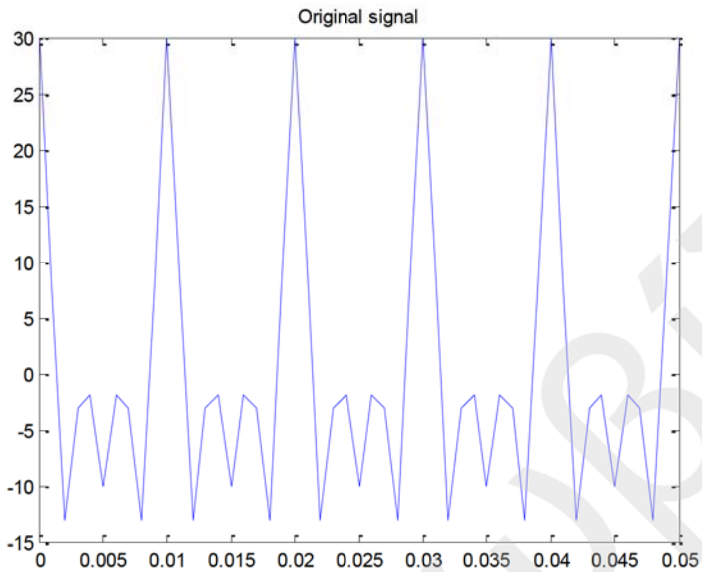
4. Να τροποποιήσετε το παράδειγμα του Σχ.5 ώστε αντί για τη συνάρτηση sinc να χρησιμοποιείται ένας μοναδιαίος τετραγωνικός παλμός.

5. Δοκιμάστε να εφαρμόσετε τη συνάρτηση του Σχ.2 για να δείτε το φάσμα ενός chirp με αρχική συχνότητα f1=220Hz, και τελική συχνότητα f2=2320Hz μετά από T2=3sec. Είναι σωστό αυτό που βλέπετε? Πως το δικαιολογείτε; (σκεφτείτε αν το σήμα είναι στατικό ή όχι).

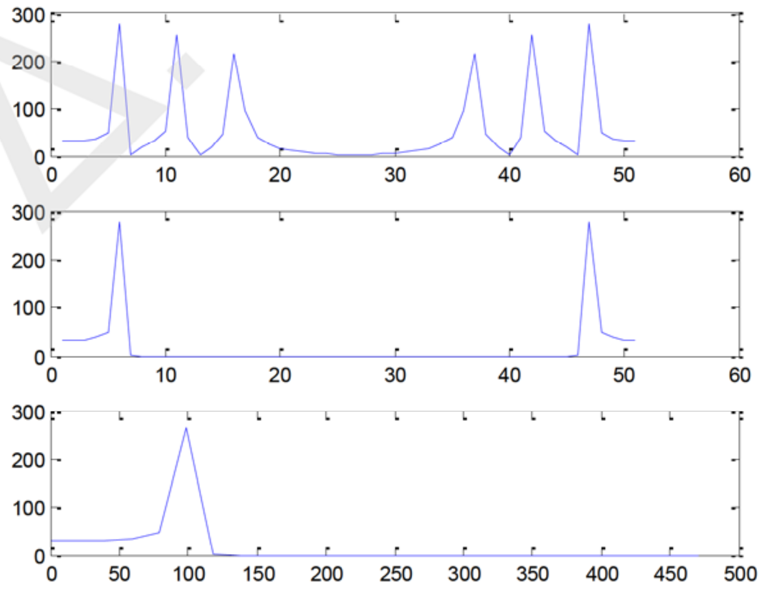
ΕΝΔΕΙΚΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Άσκηση 1

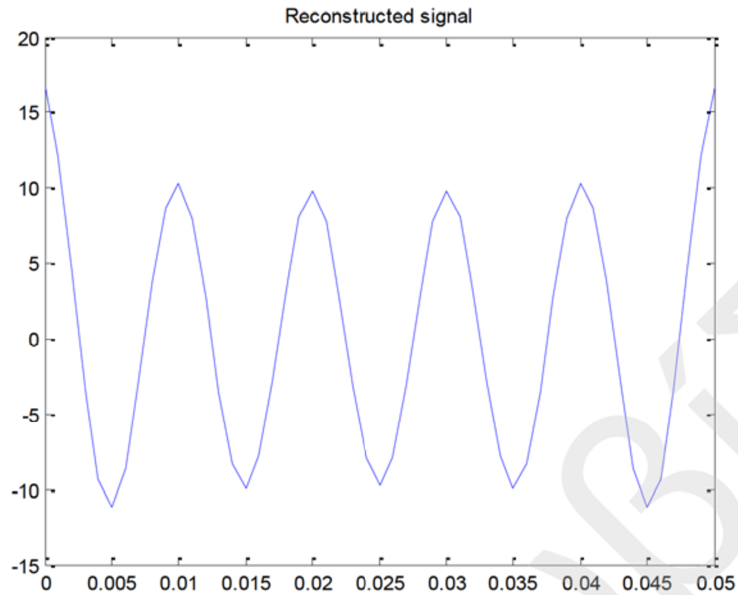
Το άθροισμα x των τριών ημιτονοειδών που παρουσιάζεται ως εξής:



Στο πρώτο από τα παρακάτω διαγράμματα παρουσιάζεται ο FFT του x. Το δεύτερο προκύπτει μηδενίζοντας χειροκίνητα τις τιμές του φάσματος, και το τρίτο προκύπτει εφαρμόζοντας το «τρικ» που χρησιμοποιεί η fftFun για να αλλάξει τον οριζόντιο άξονα ώστε να απεικονίζει συχνότητες και όχι δείγματα (παρουσιάζονται μόνο οι θετικές συχνότητες). Παρατηρούμε ότι έχουμε μόνο μια συχνότητα στα 100Hz.

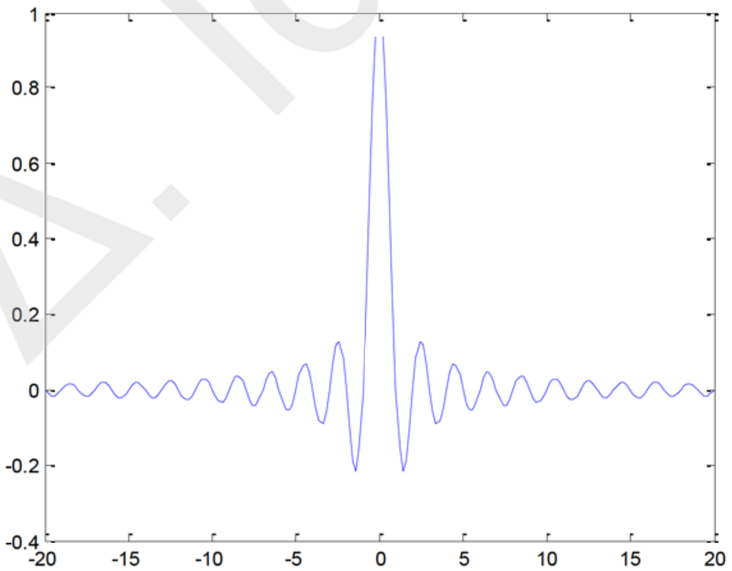


Εφαρμόζοντας τον αντίστροφο μετασχηματισμό Φουριέ στο τελευταίο φάσμα, λαμβάνουμε ένα ημιτονοειδές συχνότητας 100Hz.



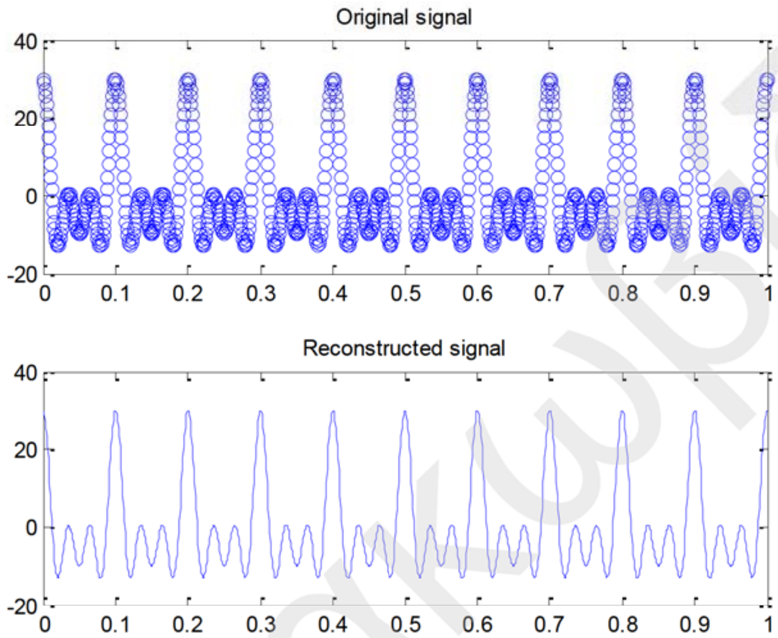
Άσκηση 2

Η sinc που παράγεται μοιάζει ως εξής:

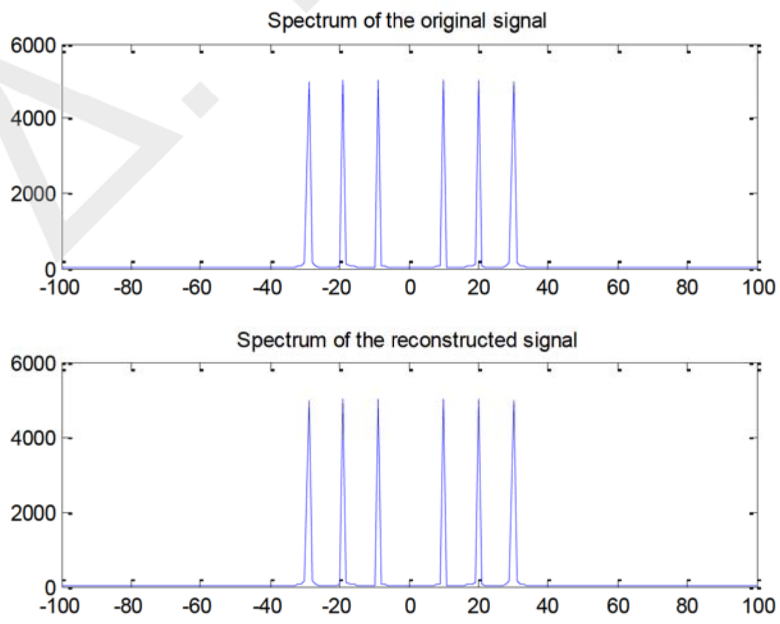


Ασκήσεις 3-4

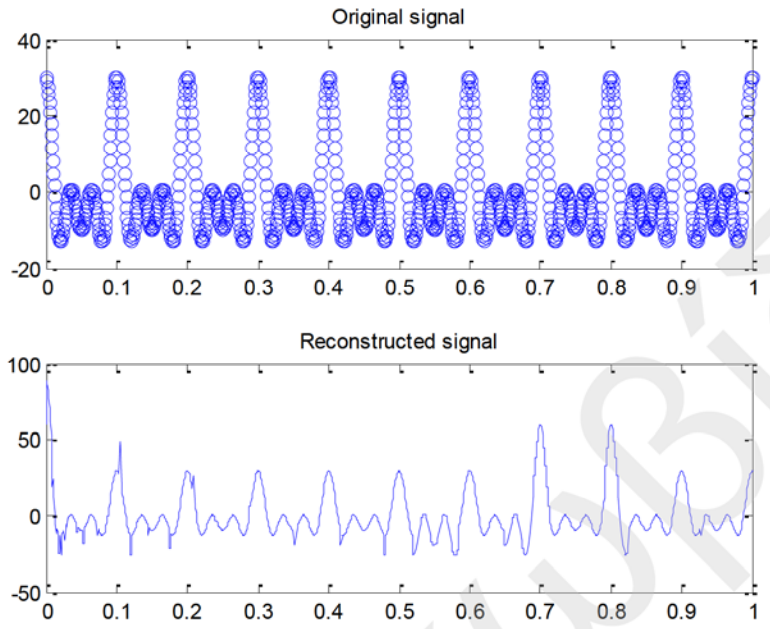
Χρησιμοποιώντας της sinc για την ανακατασκευή του αναλογικού σήματος από το αντίστοιχο ψηφιακό, το αποτέλεσμα του Σχ.5 για fsR=1000 είναι ένα αναλογικό σήμα που έχει ίδια μορφή με το ψηφιακό σήμα εισόδου αλλά και ίδιο φάσμα. Για λόγους απεικόνισης το σήμα εισόδου παρουσιάζεται χωρίς γραμμική παρεμβολή. Αν και η άσκηση αποσκοπεί στην επίδειξη της μετατροπής ψηφιακού σήματος σε αναλογικό, το αναλογικό σήμα που λαμβάνεται στο Matlab δε μπορεί να είναι ποτέ αναλογικό γιατί οτιδήποτε αναπαριστάται σε υπολογιστή είναι αναπόφευκτα ψηφιακό.

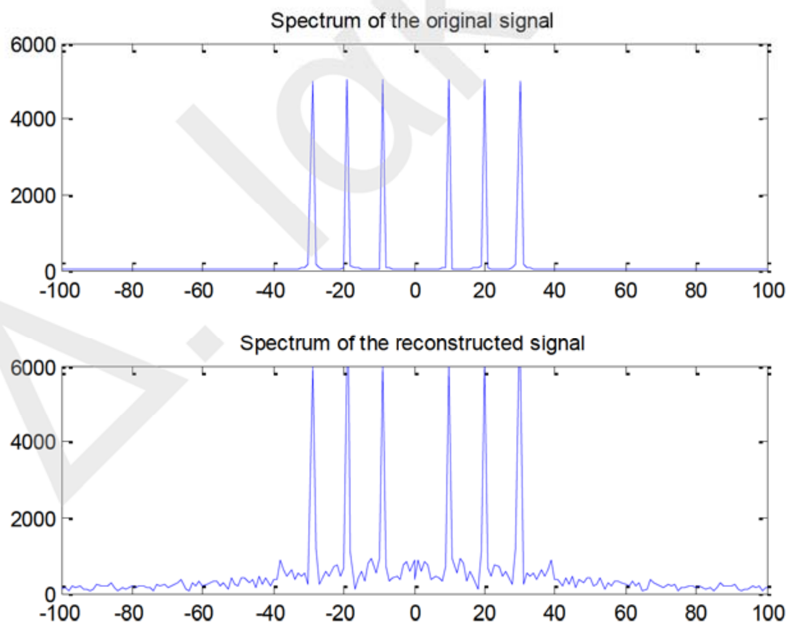


και τα αντίστοιχα φάσματα



Χρησιμοποιώντας το μοναδιαίο παλμό που είχαμε κατασκευάσει σε προηγούμενη άσκηση (pulseFun), παρατηρούμε ότι το αποτέλεσμα της ανακατασκευής δεν είναι τόσο καλό:





Άσκηση 5

Όποιο και αν είναι το αποτέλεσμα, δε θα είναι σωστό διότι το chirp είναι μεταβλητής συχνότητας και ο μετασχηματισμός Φουριέ είναι κατάλληλος μόνο για σήματα των οποίων η συχνότητα δε μεταβάλλεται χρονικά.