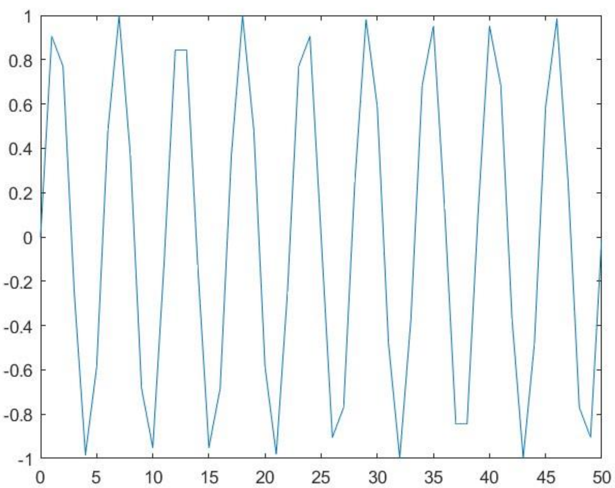
ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

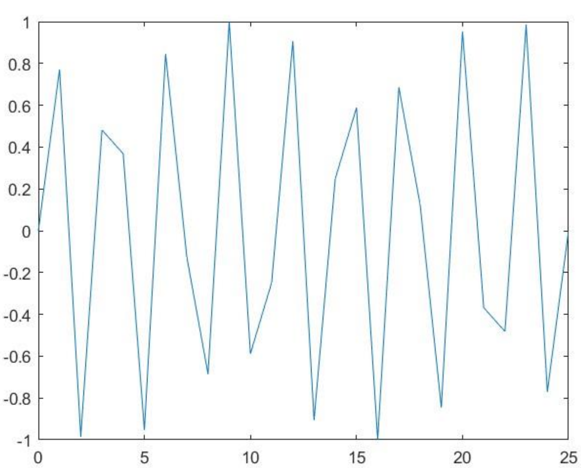
ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

Άσκηση 1

(α) Σχεδιάστε το σήμα διακριτού χρόνου που προκύπτει για Ts = 0.02, 0.04 και 0.1 sec.



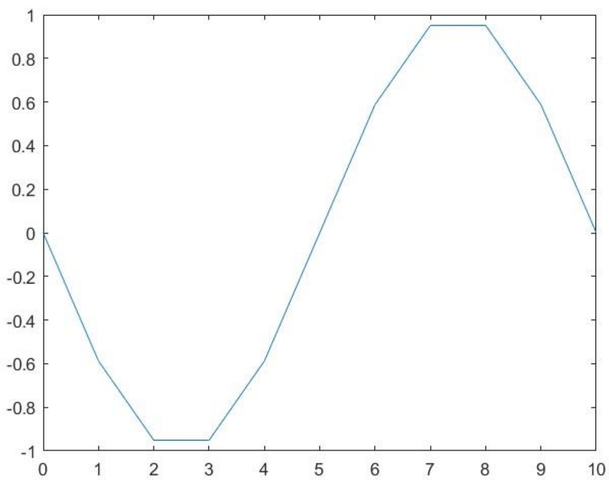
Ts = 0.02s



Ts = 0.04s

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB



Ts = 0.1s

- Τι παρατηρείτε εάν αντί για *Ts*=0.02s ή 0.04s θέσετε *Ts*=0.1s ; Αιτιολογήστε την απάντησή σας

Απάντηση:

Το σήμα μου έχει συχνότητα 9 Hz.

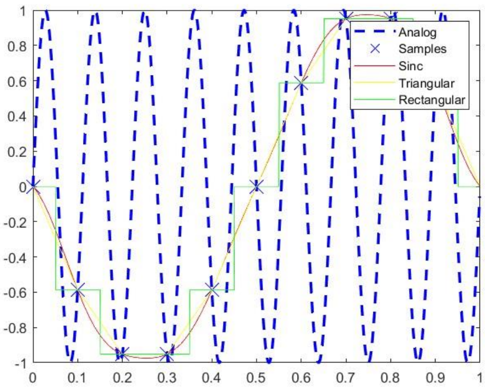
- **Για *Ts*=0.02s έχω fs = 50 Hz ,**
- **Για *Ts*=0.04s έχω fs = 25 Hz , Για *Ts*=0.1s έχω fs = 10 Hz.**

Παρατηρώ πως όσο αυξάνεται η περίοδος δειγματοληψίας Ts, τόσο πιο αραιά είναι τα δείγματά μου και τόσο λιγότερα δείγματα έχω ανά περίοδο του σήματός μου.

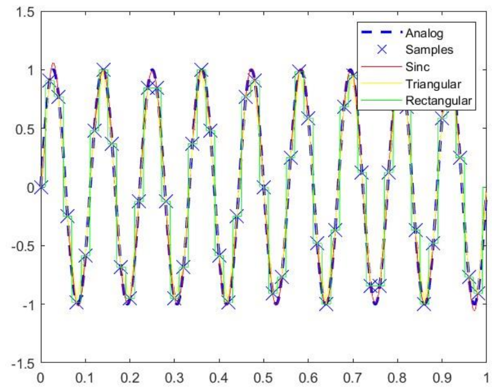
Για Ts = 0.1s έχω 1 δείγμα ανά χρονική περίοδο του δειγματοληπτούμενου σήματος, ενώ για τις άλλες δύο Ts έχω 2 και 5 δείγματα ανά περίοδο αντίστοιχα.

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

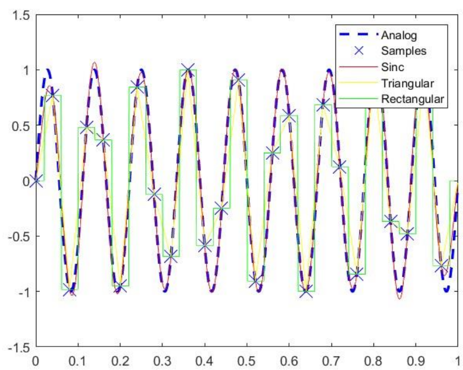
ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB



Ts = 0.1s



Ts = 0.02s



Ts = 0.04s

(β) Πώς επηρεάζει η συχνότητα δειγματοληψίας την ποιότητα ανακατασκευής του σήματος; **Απάντηση:**

- **Σύμφωνα με το θεώρημα του Nyquist πρέπει να έχω συχνότητα δειγματοληψίας fs >= f0 ώστε το σήμα διακριτού χρόνου να πλησιάζει στο αρχικό σήμα συνεχούς χρόνου. Έχω f0 = 9 Hz.**

Για Ts=0.1s είναι fs=10 Hz

Για Ts=0.02 είναι fs=50 Hz

Για Ts=0.04 είναι fs=25 Hz

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

- **Παρατηρώ πως όσο αυξάνω την συχνότητα δειγματοληψίας πάνω από το 2f0 (ή ισοδύναμα μειώνω την περίοδο δειγματοληψίας δλδ την απόσταση μεταξύ των δειγμάτων μου) τόσο περισσότερο αποτελεσματική είναι η δειγματοληψία του σήματός μου και το σήμα διακριτού χρόνου πλησιάζει στο αρχικό αναλογικό.**

Για κάθε συνάρτηση ανακατασκευής χρησιμοποιήστε το μέσο τετραγωνικό σφάλμα, ανάμεσα στο αρχικό και το ανακατασκευασμένο σήμα, και την τυπική απόκλιση , ως μετρικές ποιότητας ανακατασκευής.

Απάντηση:

T_s	MSE_1,STD_1	MSE_2,STD_2	MSE_3, STD_3
0.02s	0.0001, 0.0088	0.0003, 0.0162	0.0105, 0.1028
0.04s	0.0005, 0.0225	0.0041, 0.0638	0.0416, 0.2041
0.1s	0.0288, 0.1698	0.1256, 0.3546	0.2430, 0.4932

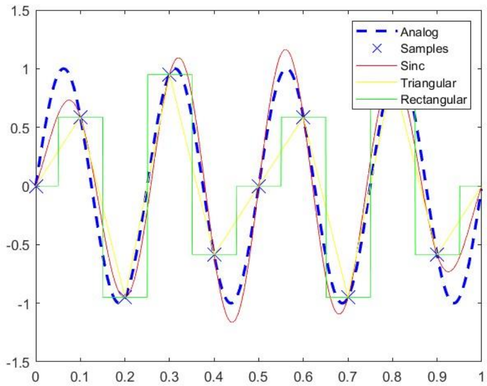
Η συνάρτηση sinc() είναι εκείνη η οποία δίνει το μικρότερο σφάλμα κατά την διαδικασία ανακατασκευής του σήματός μου. Μεγαλύτερο σφάλμα έχω με τη συνάρτηση τριγωνικού παραθύρου και τέλος το μεγαλύτερο σφάλμα έχω στη συνάρτηση τετραγωνικού παραθύρου.

(γ) Σχολιάστε τον ρόλο της αρχικής φάσης του σήματος.

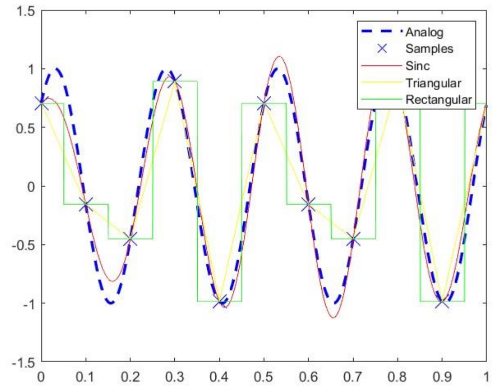
Απάντηση:

T_s	MSE_1,STD_1	MSE_2,STD_2	MSE_3, STD_3
0.1s	0.0149, 0.1218	0.1256, 0.3546	0.2430, 0.4932

Παρατηρώ πως εισάγοντας αρχική φάση τα δείγματα που προέκυψαν ύστερα από δειγματοληψία αλλάζουν θέση στο γράφημα (μετατόπιση κατά την αρχική φάση).



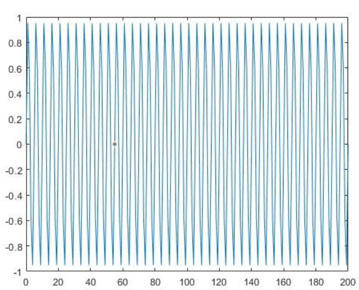
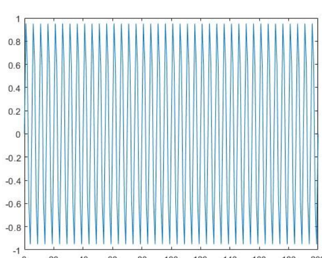
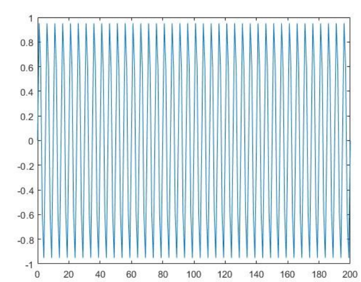
Μηδενική αρχική φάση



Αρχική φάση pi/4

(δ) Συμπληρώστε τον παρακάτω πίνακα με τα δικά σας γραφήματα.

Απάντηση:

$f_0 = 40Hz$	$f_0 = 240Hz$	$f_0 = 4040Hz$
		

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

Ερώτηση 5 (δ συνέχεια) Τι παρατηρείτε στις παραπάνω γραφικές παραστάσεις σας; Ποια η συχνότητα των ανακατασκευασμένων σημάτων; Εξηγήστε.

Απάντηση:

1.)Ελέγχουμε σε ποιες περιπτώσεις έχω αναδίπλωση:

f_1 = 40 Hz → ΔΕΝ ΕΧΩ ΑΝΑΔΙΠΛΩΣΗ.

f_2 = 240 Hz → ΑΝΑΔΙΠΛΩΝΕΤΑΙ. Η τελική συχνότητα που θα δειγματοληπτηθεί είναι η $f_2' = \min_a|a*f_s - f_2|, a \in N^*$ = 40 Hz

f_3 = 4040 Hz → ΑΝΑΔΙΠΛΩΝΕΤΑΙ. Η τελική συχνότητα που θα δειγματοληπτηθεί είναι η $f_3' = \min_a|a*f_s - f_3|, a \in N^*$ = 40 Hz (a=20)

2.)Υπολογίζουμε τις ψηφιακές κυκλικές συχνότητες που προκύπτουν από τη δειγματοληψία σύμφωνα με το γενικό τύπο:

$\omega_n = \Omega * T_s = \Omega / f_s = 2\pi f / f_s$ όπου f: η τελική μέγιστη συχνότητα που θα δειγματοληπτηθεί πριν την αναδίπλωση.

Δηλαδή έχω: $\omega_1 = 2\pi * 40 / 200 = 0.4\pi$

$\omega_2 = 2\pi * 40 / 200 = 0.4\pi$

$\omega_3 = 2\pi * 40 / 200 = 0.4\pi$

3.)Υποθέτω ότι το σύστημα ανακατασκευής είναι ιδανικό και άρα η συχνότητα ανακατασκευής f_{AN} ταυτίζεται με τη συχνότητα δειγματοληψίας f_s. Οι συχνότητες των ανακατασκευασμένων αναλογικών σημάτων δίνονται από τον τύπο:

$\omega_n = 2\pi f_n / f_{AN}$ → *$f_n = \omega_n * f_{AN} / 2\pi$*

Δηλαδή έχω: $f_1 = (\omega_1 * f_s) / 2\pi = (0.4\pi * 200\ Hz) / 2\pi \rightarrow f_1 = 40\ Hz$

$f_2 = (\omega_2 * f_s) / 2\pi = (0.4\pi * 200\ Hz) / 2\pi \rightarrow f_2 = 40\ Hz$

$f_3 = (\omega_3 * f_s) / 2\pi = (0.4\pi * 200\ Hz) / 2\pi \rightarrow f_3 = 40\ Hz$

ΕΠΟΜΕΝΩΣ ΤΑ ΑΝΑΚΑΤΑΣΚΕΥΑΣΜΕΝΑ ΣΗΜΑΤΑ ΕΧΟΥΝ ΤΗΝ ΙΔΙΑ ΣΥΧΝΟΤΗΤΑ.

ΚΑΙ ΟΝΤΩΣ ΑΥΤΟ ΕΠΑΛΗΘΕΥΕΤΑΙ ΑΠΟ ΤΑ ΠΑΡΑΠΑΝΩ ΓΡΑΦΗΜΑΤΑ.

<u>Ασκηση 2</u>

(α) Αιτιολογήστε αν το σύστημα είναι αιτιατό ή όχι **Απάντηση:**

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

Ένα σύστημα είναι αιτιατό (causal) όταν η παρούσα τιμή της εξόδου του δεν εξαρτάται από μελλοντικές τιμές της εισόδου. Εδώ αυτό προφανώς ισχύει και άρα το σύστημα είναι αιτιατό.

(β.1) Υπολογίστε κρουστική απόκριση του συστήματος (μόνο θεωρητικά).

Απάντηση:

Η κρουστική απόκριση του συστήματος είναι η έξοδος/απόκριση του συστήματος για είσοδο την κρουστική συνάρτηση δ[n].

Άρα h[n] = ½ , για n=0

1 , για n=1

-1/2 , για n=2

0 , για όλα τα άλλα n

(β.2) Σχεδιάστε το μέτρο και τη φάση της απόκρισης συχνότητας θεωρητικά και χρησιμοποιώντας τη συνάρτηση *freqz()* της Matlab).

Απάντηση:

$y[n] = 1/2x[n] + x[n - 1] – 1/2x[n – 2] \rightarrow$

$Y(j\omega) = 1/2X(j\omega) + X(j\omega) * e^{-j\omega} – 1/2X(j\omega) * e^{-2j\omega} \rightarrow$

$Y(j\omega) / X(j\omega) = 1/2 + e^{-j\omega} -(½) * e^{-2j\omega} = e^{-j\omega} * [(e^{j\omega}/2) + 1 – (e^{-j\omega}/2)] \rightarrow$

$H(j\omega) = e^{-j\omega} (1 + j*\sin\omega)$

Είπαμε πριν ότι ισχύει: $H(j\omega) = e^{-j\omega} (1 + j*\sin\omega)$

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

$$= e^{-j\omega}+je^{-j\omega}\sin(\omega)$$

Έχω:

$e^{-j\omega} = \cos(-\omega)+j\sin(-\omega)=\cos(\omega)-j\sin(\omega)$

$e^{-j\omega}\sin(\omega) = \sin(\omega)\cos(\omega)-j\sin2(\omega)$

Άρα:

$H(j\omega) = (\cos(\omega)+j\sin(\omega))+j(\sin(\omega)\cos(\omega)-j\sin2(\omega))$

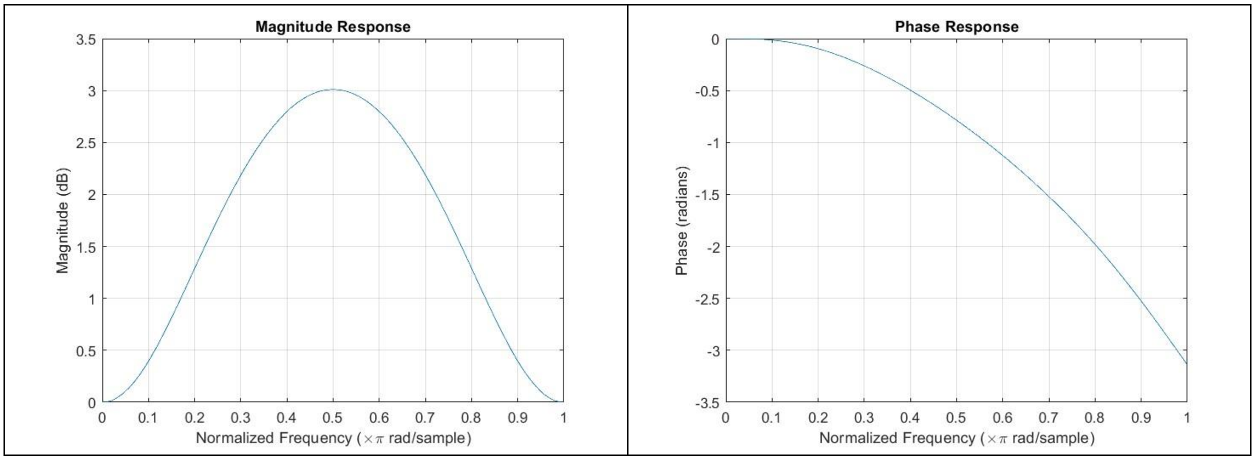
$= \cos(\omega)+j\sin(\omega)+j\sin(\omega)\cos(\omega)+\sin2(\omega)$

$= (\cos(\omega)-\sin2(\omega))+j(\sin(\omega)+\sin(\omega)\cos(\omega))$

$| H(j\omega)|= \sqrt{} (\cos(\omega)-\sin2(\omega))2+(\sin(\omega)+\sin(\omega)\cos(\omega))^2$

$\phi=\arctan(\cos(\omega)-\sin2(\omega)\sin(\omega)+\sin(\omega)\cos(\omega))$

Μέτρο απόκρισης συχνότητας	**Φάση απόκρισης συχνότητας**



(γ) Ποιες συχνότητες του σήματος εισόδου διατηρεί το παραπάνω σύστημα; **Απάντηση:**

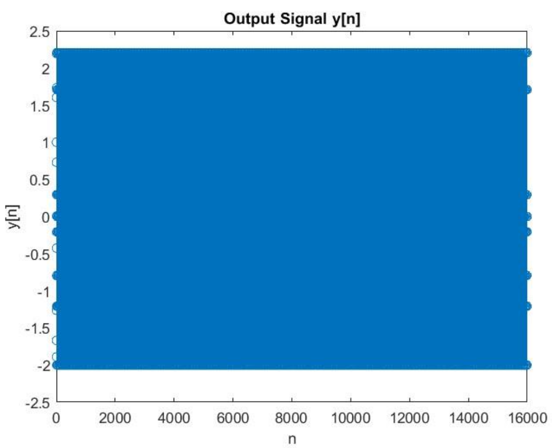
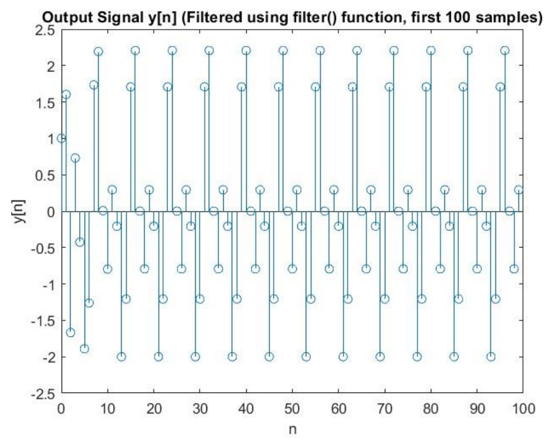
Το παραπάνω σύστημα διατηρεί εκείνες τις συχνότητες του σήματος εισόδου στις οποίες το μέτρο της απόκρισης συχνότητας |H(jω)| είναι ακριβώς ίσο με 1 (στον παραπάνω γράφο |H(jΩ)| = 0 dB). Πιο συγκεκριμένα, βλέποντας τον πάνω αριστερά γράφο συμπεραίνουμε ότι οι συχνότητες που είναι ακέραια πολλαπλάσια του π διατηρούνται από το σύστημα (χωρίς να γίνονται amplified).

(δ) Χρησιμοποιώντας τη συναρτηση *filter()*, υπολογίστε και σχεδιάστε την έξοδο του συστήματος για την είσοδο $x[n]$ (μόνο για τα πρώτα 100 δείγματα). Ποιες οι διαφορές;

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

Απάντηση:

Έξοδος για *conv()*	**Έξοδος για *filter()***
	

Παρατηρώ πως τα παραπάνω γραφήματα είναι ίδια (με εξαίρεση προφανώς τον αριθμό των δειγμάτων που έχω επιλέξει να αναπαραστήσω). **Οι συναρτήσεις δηλαδή conv() και filter()** ***κάνουν την ίδια δουλειά***, υπολογίζουν την έξοδο y ενός συστήματος για κάποια είσοδο x(εδώ y[n] και x[n]).

Η βασική διαφορά τους έγκειται στο γεγονός ότι η conv() υπολογίζει την έξοδο απευθείας μέσω της πράξης του συνελικτικού αθροίσματος. Αντίθετα, η filter() αντιμετωπίζει το σύστημα ***ΜΕ ΠΙΟ ΓΕΝΙΚΕΥΜΕΝΟ ΤΡΟΠΟ ως ένα FIR φίλτρο***. Εδώ το σύστημα που περιγράφεται από την $y[n] = 1/2x[n] + x[n - 1] - 1/2x[n - 2]$ πρόκειται για FIR διότι:

- ✓ Είναι ένα σύστημα **χωρίς feedback**(a_k=0 foreach k). Η έξοδος ενός FIR filter εξαρτάται μόνο από τις τιμές στα τρέχοντα και παρελθοντικά δείγματα εισόδου, και όχι από τις προηγούμενες τιμές εξόδου του.
- ✓ Έχει **πεπερασμένη απόκριση συχνότητας** .
- ✓ Τα δείγματα εισόδου είναι πεπερασμένα ($b_k \neq \infty$). ✓ Είναι LTI (ΓΧΑ).

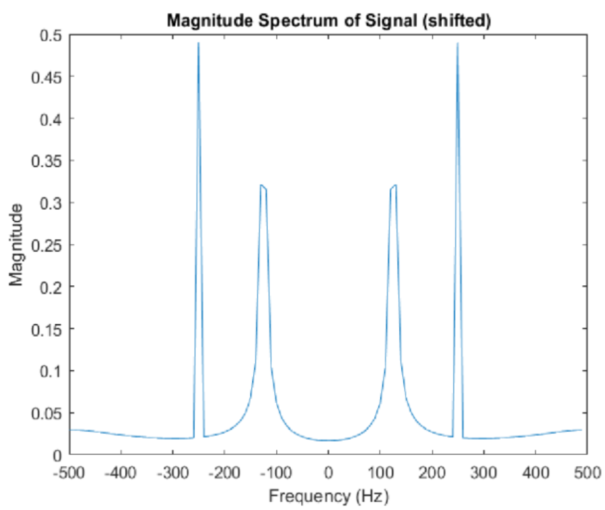
Με άλλα λόγια, επειδή το παραπάνω σύστημα είναι FIR χωρίς feedback, η filter() παίρνει σαν όρισμα a=1 και κάνει ότι και η conv() για δοσμένα h[n] και x[n].

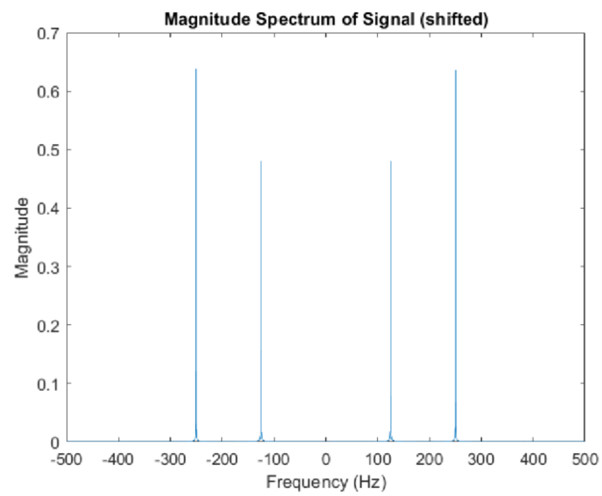
ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

(ε) Σχεδιάστε το `abs(fftshift(fft(x)))` και `abs(fftshift(fft(y)))`.

Απάντηση:

<table>
<tr><th>$|DFT\{x\}|$</th><th>$|DFT\{y\}|$</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Πώς επιδρά στις συχνότητες του σήματος εισόδου το σύστημα;

- Από τον γράφο του μέτρου της απόκρισης συχνότητας συμπεραίνω ότι σε κάθε ακέραιο πολλαπλάσιο του π το μέτρο της H(jω) είναι 1 (0 dB αλλιώς) , επομένως σε αυτά τα ακέραια πολλαπλάσια το πλάτος του σήματος εισόδου διατηρείται. Σε όλες τις υπόλοιπες συχνότητες ενισχύεται (amplified).
- Από τον γράφο της φάσης της απόκρισης συχνότητας συμπεραίνω ότι σε κάθε ζυγό ακέραιο πολλαπλάσιο του π η φάση της H(jω) είναι 0 rad, επομένως σε αυτά τα πολλαπλάσια η φάση του σήματος εισόδου διατηρείται. Σε όλες τις υπόλοιπες συχνότητες η φάση μεταβάλλεται από το σύστημα (shifted).

Με απλά λόγια, το παραπάνω σύστημα πρόκειται για ένα bandpass φίλτρο, το οποίο συμπεριφέρεται ως amplifier για συγκεκριμένο εύρος συχνοτήτων.

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

Ο κατάλληλος τύπος μετασχηματισμού Fourier για τα δεδομένα της άσκησης, είναι ο DFT (σκεφτείτε και εξηγήστε γιατί).

Αυτό οφείλεται στο γεγονός ότι έχω πεπερασμένο αριθμό δειγμάτων n και επιπλέον έχω στη διάθεσή μου τον FFT ο οποίος είναι ένας γρήγορος αλγόριθμος υπολογισμού του DFT.

(στ)

Μήκος σήματος	Μέσος χρόνος σε μsec	Μήκος σήματος	Μέσος χρόνος σε μsec
2^6	8.0722	**2^6-1**	6.0629
2^7	2.4753	**2^7-1**	10.439
2^8	3.2118	**2^8-1**	8.4313
2^9	3.3358	**2^9-1**	15.945
2_{10}	6.0346	**2^{10}-1**	21.419
2_{11}	10.089	**2^{11}-1**	98.994
2_{12}	25.199	**2^{12}-1**	51.016
2_{13}	54.373	**2^{13}-1**	398.26
2_{14}	107.63	**2^{14}-1**	1086.1
2_{15}	230	**2^{15}-1**	2619.7

ΠΑΡΑΡΤΗΜΑ

Κώδικας Άσκησης 1

```
% Ts: sampling rate
% f0: frequency of signal in Hz
% initial_phase: initial phase of signal
%=======================================================================%
% clear
% clc
% close all
%=======================================================================%
Ts  =  0.005;  f0  =
4040; initial_phase
=  0;
  n  = 0:1/Ts; %discrete samples %x =
sin(2*pi*f0*n*Ts+initial_phase); x =
sin(2*pi*f0*n*Ts+initial_phase);
plot(n,x) dt = 0.001;
t = 0:dt:1; %continuous time x_cont=sin(2*pi*f0*t'+initial_phase);

% Initialize Arrays
sinc_array = zeros(length(t),length(n));
triangular_array = sinc_array; rec_array
= sinc_array;

% indx:(t/Ts-n)
indx = t'*ones(1,length(n))/Ts-ones(length(t),1)*n;
 %
Sinc
sinc_array = sinc(indx);

% Triangular
triangular_array(abs(indx)>1)=0; %x in [-1, 1], so delete the rest
triangular_array(abs(indx)<1) = 1 - abs(indx(abs(indx)<1));

% Rectangular
rec_array(abs(indx)<1/2) = 1; rec_array(indx
==1/2) = 1; rec_array(abs(indx)>1/2) = 0;


% Reconstructed Signals
```

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

```
x_analog1 = sum((ones(length(t),1)*x).*sinc_array,2); % Sinc Reconstruction
x_analog2 = sum((ones(length(t),1)*x).*triangular_array,2); %Triangular
Reconstruction
x_analog3 = sum((ones(length(t),1)*x).*rec_array,2); % Rectangular Reconstruction


% Residual Signals r1=x_cont-x_analog1;
r2=x_cont-x_analog2; r3=x_cont-
x_analog3;


% Plot Reconstructed Signals figure;
plot(t(1:1000),x_cont(1:1000),'b--','LineWidth',2) % Plot original analog signal
hold on
plot(n(1:dt/Ts*1000)*Ts,x(1:dt/Ts*1000),'bx','MarkerSize',14) % Plot Sample Points
plot(t(1:1000),x_analog1(1:1000),'r') % Plot sinc reconsruction
plot(t(1:1000),x_analog2(1:1000),'y') % Plot triangular reconstruction
plot(t(1:1000),x_analog3(1:1000),'g') % Plot rectangular reconsturction hold off
legend('Analog','Samples','Sinc','Triangular','Rectangular')


% Plot Error of Reconstruction
figure hold on
plot(t(1:100),sin(10*pi*t(1:100)')-x_analog1(1:100)) % Plot sinc Error
plot(t(1:100),sin(10*pi*t(1:100)')-x_analog2(1:100)) % Plot triangular Error
plot(t(1:100),sin(10*pi*t(1:100)')-x_analog3(1:100)) % Plot rectangular Error hold
off
legend('Sinc','Triangular','Rectangular')

% Plot of Distributions of residuals
 figure
hist(r1,200) % Histogram of r1
legend('Sinc Residual') figure
hist(r2,200) % Histogram of r2
legend('Triangular Residual') figure
hist(r3,200) % Histogram of r3 legend('Rectangular
Residual')


MSE = [mean(r1.^2) mean(r2.^2) mean(r3.^2) ]
STD = [std(r1) std(r2) std(r3) ]
```

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

Κώδικας Άσκησης 2

```
% Generate a test signal Fs = 1000; %
Sampling frequency (Hz) t = 0:1/Fs:1-1/Fs;
% Time vector (seconds) f1 = 10; %
Frequency of signal 1 (Hz) f2 = 100; %
Frequency of signal 2 (Hz)
x = sin(2*pi*f1*t) + sin(2*pi*f2*t); % Signal with two frequencies

% Compute the FT of the signal
N = length(x); % Number of samples
X = fft(x); % Compute FFT
X_shifted = fftshift(X); % Shift zero frequency to center f
= Fs*(-N/2:N/2-1)/N; % Frequency vector (Hz)

% Plot the unshifted magnitude spectrum of the signal
figure; plot(abs(X)/N); xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Magnitude Spectrum of Test Signal (unshifted)');


% Plot the magnitude spectrum of the signal figure;
plot(f, abs(X_shifted)/N); xlabel('Frequency
(Hz)'); ylabel('Magnitude');
title('Magnitude Spectrum of Test Signal (shifted)');


% Plot the phase spectrum of the signal
figure; plot(f, angle(X));
xlabel('Frequency (Hz)');
ylabel('Phase');
title('Phase Spectrum of Test Signal');
```

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

```
% ΖΗΤΟΥΜΕΝΟ β)

  b = [0.5 1 -0.5];  % Numerator coefficients (from the difference equation) a =
1;             % Denominator coefficients (for FIR filter, typically set to 1)


[H, w] = freqz(b, a, 1024);  % Compute frequency response

% Plot the magnitude of the frequency response figure;
plot(w/pi, 20*log10(abs(H))); title('Magnitude
Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)'); grid on;

% Plot the phase of the frequency response
figure; plot(w/pi, angle(H)); title('Phase
Response');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Phase (radians)'); grid on;


% ΖΗΤΟΥΜΕΝΟ δ)


% Define the input signal x[n] n
= 0:16000;
x = cos(pi * n / 4) - sin(pi * n / 2) + (-1/2).^n;

% Define the impulse response h[n]
h = [1/2, 1, -1/2];

% Perform convolution using conv() function y
= conv(x, h);

% Plot the output signal y[n] figure;
stem(n, y(1:length(n)));
xlabel('n'); ylabel('y[n]');
title('Output Signal y[n]');

   m = 0:99; % Consider only the first 100 samples
x = cos(pi * m / 4) - sin(pi * m / 2) + (-1/2).^m;

% Define the impulse response coefficients b
= [1/2, 1, -1/2];
```

```
a = 1;                      % Since it's a FIR filter, denominator coefficients are 1
% Perform filtering using filter() function z
= filter(b, a, x);

% Plot the output signal y[n] figure;
stem(m, z(1:length(m)));
xlabel('n'); ylabel('y[n]');
title('Output Signal y[n] (Filtered using filter() function, first 100 samples)');
```

```
% ZHTOYMENO ε.i)
% |DFT{x}|


% Compute the FT of signal x[n]
N = length(x); % Number of samples
X = fft(x); % Compute FFT
X_shifted = fftshift(X); % Shift zero frequency to center f
= Fs*(-N/2:N/2-1)/N; % Frequency vector (Hz)


% Plot the SHIFTED magnitude spectrum of signal x[n]
figure;
plot(f, abs(X_shifted)/N); xlabel('Frequency
(Hz)'); ylabel('Magnitude');
title('Magnitude Spectrum of Signal (shifted)');
```

```
% ZHTOYMENO ε.ii)
% |DFT{y}|

% Compute the FT of signal y[n] = x[n] * h[n]
N = length(y); % Number of samples
Y = fft(y); % Compute FFT
```

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ MATLAB

```
Y_shifted = fftshift(Y); % Shift zero frequency to center f
= Fs*(-N/2:N/2-1)/N; % Frequency vector (Hz)


% Plot the SHIFTED magnitude spectrum of signal y[n] = x[n] * h[n] figure;
plot(f, abs(Y_shifted)/N); xlabel('Frequency
(Hz)'); ylabel('Magnitude');
title('Magnitude Spectrum of Signal (shifted)');




% ZHTOYMENO στ)

num_experiments = 10000;    % πλήθος πειραμάτων
min_exponent = 6;            max_exponent = 15;

exponents = min_exponent:max_exponent;
lengths_power_2 = 2 .^ exponents;     % Δημιουργία μεγεθών ακολουθίας(δυνάμεις του 2)
lengths = 2 .^ exponents - 1;         % Δημιουργία μεγεθών ακολουθίας(2^x-1)



% Αποθήκευση χρόνων FFT για κάθε μέγεθος ακολουθίας
results_FFT_1 = zeros(length(lengths_power_2), 1);   results_FFT_2
= zeros(length(lengths), 1);

for i = 1:length(exponents)

    % Δημιουργία τυχαίας ακολουθίας
sequence_length_1 = lengths_power_2(i);
sequence_1 = rand(sequence_length_1, 1);

    sequence_length_2 = lengths(i);
sequence_2 = rand(sequence_length_2, 1);



    % Μέτρηση χρόνου για FFT
tic;
    for j = 1:num_experiments
fft_result_1 = fft(sequence_1);      end
    time_FFT_1 = toc;
```

```
tic;
    for j = 1:num_experiments
fft_result_2 = fft(sequence_2);     end
    time_FFT_2 = toc;
          results_FFT_1(i) = (time_FFT_1 / num_experiments) *
10^6;     results_FFT_2(i) = (time_FFT_2 / num_experiments) *
10^6; end

% Εκτύπωση αποτελεσμάτων
disp('Μήκος Ακολουθίας, Χρόνος Εκτέλεσης(μsec)');
format short g;                                % just a command i googled to suppress
% the scaling factor applied by Matlab                                                %
due to numbers being too small disp([lengths_power_2' results_FFT_1]); disp([lengths'
results_FFT_2]);
```