Εργασία στο μάθημα Τηλεπικοινωνιακά Συστήματα

Ονοματεπώνυμο : Γεώργιος Χατζηευφραιμίδης

ΑΕΜ : 3503

Η εργασία πραγματοποιήθηκε σε περιβάλλον MATLAB έκδοσης 2016b
Στο παράρτημα της εργασίας υπάρχει ο κώδικας στο Matlab.

Άσκηση 1: Ιδανική τυχαία ακολουθία τετραγωνικών παλμών

1. Κατασκευάστε έναν τετραγωνικό παλμό με πλάτος Α στο διάστημα (-To/2, To/2), όπου το To είναι σε msec. Κάντε τη γραφική παράσταση του παλμού στο χρόνο, υπολογίστε το φάσμα του παλμού μαθηματικά και κάντε τη γραφική παράσταση του μετ/μού Fourier του παλμού.

Απάντηση :

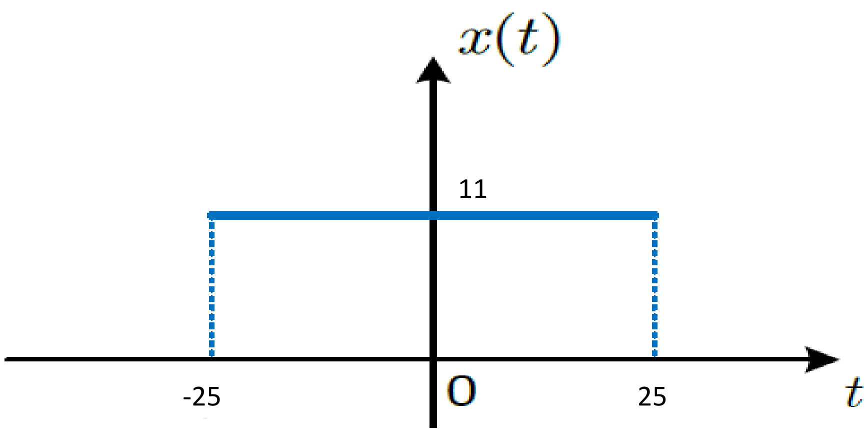
<u>Θεωρητική ανάλυση</u>

Ο τετραγωνικός παλμός ορίζεται με την συνάρτηση :

$$x(t) = \begin{cases} A, & -\frac{T_o}{2} < t < \frac{T_o}{2} \\ 0, & \alpha\lambda\lambda o\acute{\upsilon} \end{cases} = A\mathrm{rect}\left(\frac{t}{T_o}\right)$$

όπου A=7+(k mod 6) = 7+(22 mod 6) = 11 και To = 50+(k div 2) = 50+(1 div 2) = 50 msec,

και θα έχει την μορφή :



Από πίνακα μετασχηματισμού Fourier προκύπτει :

Συνάρτηση	Πεδίο Χρόνου (x(t))	Πεδίο Συχνοτήτων (X(f))
Ορθογωνικός παλμός	$A \cdot \mathrm{rect}\left(\frac{t}{T_o}\right)$	$A \cdot T_o\, sinc(T_o f)$

όπου

$$A \cdot T_o \cdot sinc(T_o f) = \begin{cases} A \cdot T_o \cdot \frac{\sin(T_o \pi f)}{T_o \pi f}, & \gamma\iota\alpha\ f \neq 0 \\ A \cdot T_o, & \gamma\iota\alpha\ f = 0 \end{cases}$$, η συνάρτηση δειγματοληψίας

Αντικαθιστώντας προκύπτει :

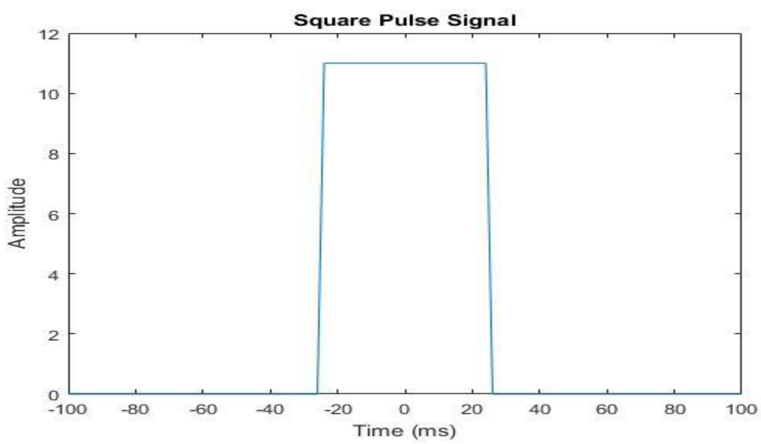
$$X(f) = \begin{cases} 0.55 \dfrac{\sin(0.05\pi f)}{0.05\pi f}, & \gamma\iota\alpha\ f \neq 0 \\ 0.55\,, & \gamma\iota\alpha\ f = 0 \end{cases}$$

Τα σημεία μηδενισμού της συνάρτησης δειγματοληψίας είναι :

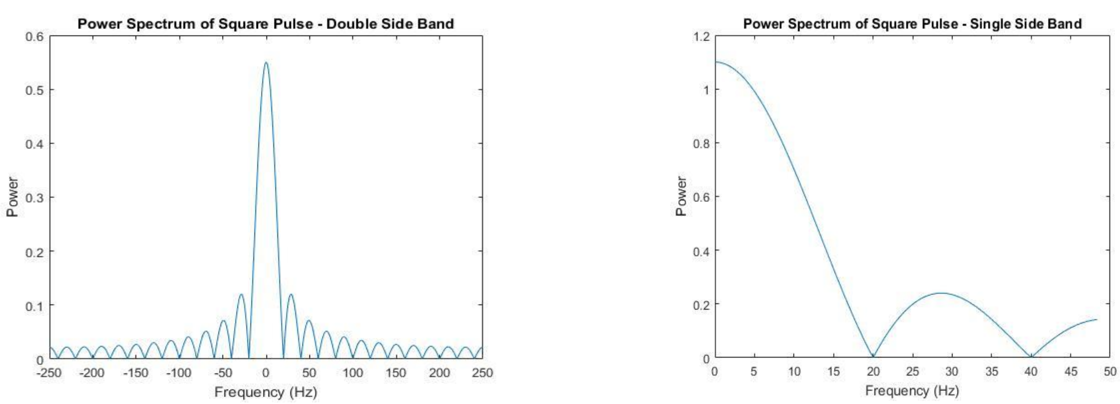
$\frac{\sin(T_o \pi f)}{T_o \pi f} = 0$ ή $T_o \pi f = \kappa\pi$ ή $f = \frac{\kappa}{T_o}$ ή $f = 20\kappa\ (Hz)$, κ ακέραιος

<u>Αποτελέσματα προσομοίωσης σε Matlab</u>

Ο κώδικας που χρησιμοποιήθηκε για την σχεδίαση του σήματος στο πεδίο του χρόνου και της συχνότητας, είναι στο παράρτημα. Τα αποτελέσματα φαίνονται παρακάτω:



Πεδίο χρόνου



Πεδίο *Συχνότητας*

Όπως φαίνεται τα αποτελέσματα ταυτίζονται με τους θεωρητικούς υπολογισμούς.

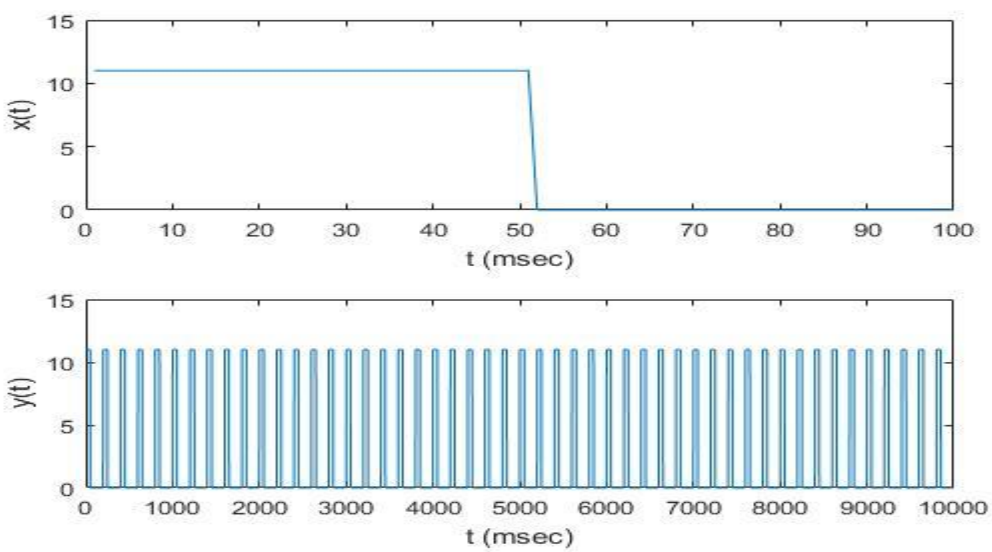
2. Δημιουργείστε μια περιοδική ακολουθία 100 παλμών ρολογιού, όπου κάθε παλμός είναι όμοιος με τον παλμό του ερωτήματος (1) και η περίοδος μεταξύ διαδοχικών παλμών είναι ίση με Tb=200msec. Χρησιμοποιείστε time-step 1 msec και περίοδο μεταξύ των παλμών Tb msec (οπότε από 1-Tb θα είναι ο πρώτος τετραγωνικός παλμός, από Tb+1 μέχρι 2Tb ο 2ος παλμός κ.ο.κ.). Στόχος είναι να κατασκευάσετε μία συνάρτηση y(t) με t=1,2,3…100,…20.000 msec (που θα αναπαριστά το χρόνο οπότε θα έχει τιμές ανά time-step 1msec) και θα περιλαμβάνει έναν τετραγωνικό παλμό σε κάθε διάστημα Tb msec.

Απάντηση :

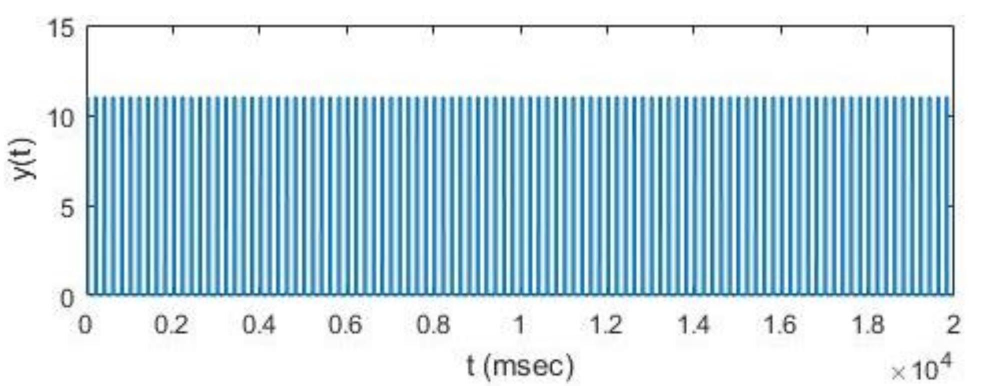
Αρχικά δημιουργείτε ένας μόνο παλμός με χαρακτηριστικά πλάτος A =11 και διάρκεια To= 50 msec.

$$x(t) = \begin{cases} 11, & 1ms < t < 51ms \\ 0, & αλλού \end{cases}$$

Για λόγους ευκολίας, επιλέχθηκε η δημιουργία του παλμού να γίνει με χρήση ενός βρόγχου επανάληψης, προσδιορίζοντας το επιθυμητό χρονικό σημείο έναρξης και λήξης του, αντί της συνάρτησης rectpuls(t,To) του πρώτου ερωτήματος. Στην συνέχεια με χρήση νέου βρόγχου επανάληψης, παράγονται όμοιοι παλμοί με τα ίδια χαρακτηριστικά, δίνοντας μια περιοδική ακολουθία 100 παλμών, περιόδου 0.2 sec, σε συνολική διάρκεια 20 sec. Στο ακόλουθο σχήμα παρουσιάζονται ο ένας παλμός (σε κλίμακα χρόνου από 1 ms μέχρι 100 ms) και η παλμοσειρά (σε κλίμακα χρόνου από 1ms μέχρι 10000 ms).



Η ίδια περιοδική ακολουθία παλμών στο συνολικό χρονικό διάστημα των 20 sec παρουσιάζεται στο ακόλουθο σχήμα :



Ο κώδικας που χρησιμοποιήθηκε για την σχεδίαση των παραπάνω σημάτων στο πεδίο του χρόνου, βρίσκεται στο παράρτημα.

3. Αποτυπώστε γραφικά (πχ μέσω συνάρτησης plot) τη συνολική ακολουθία y(t) ως προς το χρόνο t για t=1,2,3…20.000. Υπολογίστε τη μέση ισχύ του σήματος.

Απάντηση :

Η περιοδική ακολουθία παλμών y(t) έχει σχεδιαστεί στο προηγούμενο ερώτημα.

Ο υπολογισμός της μέσης ισχύς του σήματος βασίζεται στην σχέση :

$$P = \sum_{n=1}^{20000} |y(n)|^2$$

και υπολογίζεται με χρήση της συνάρτησης mean().

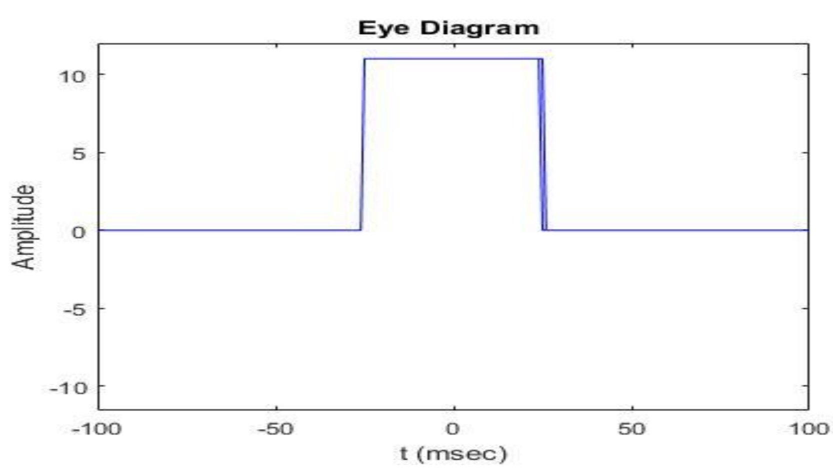
Το αποτέλεσμα που προκύπτει από τον κώδικα που βρίσκεται στο παράρτημα, είναι P = 30.81 :

```
MATLAB Command Window

The average power of the signal is equal to: 3.081265e+01
>>
```

4. Δημιουργείστε γραφικά το διάγραμμα ματιού της ακολουθίας, δλδ σε ένα χρονικό παράθυρο 1-Tb msec να αποτυπώνονται όλοι οι παλμοί του σήματος ρολογιού.

Απάντηση :

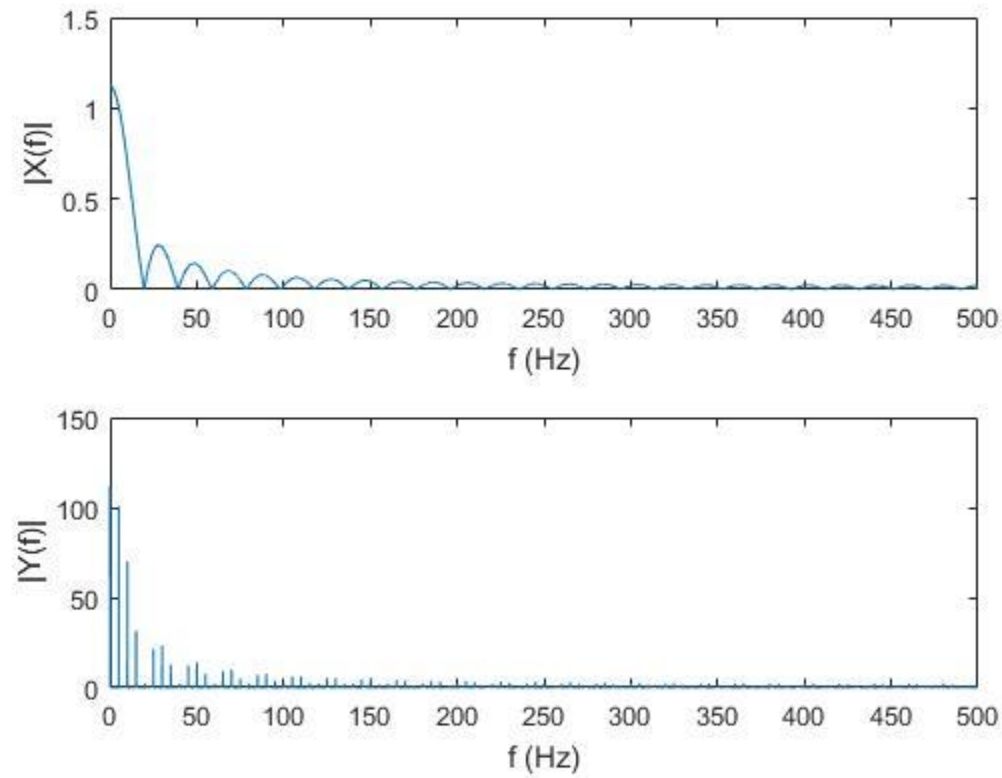
Για τον υπολογισμό του διαγράμματος ματιού χρησιμοποιείται η συνάρτηση eyediagram (Signal,samplePerBit,timeSlot,offset), όπου ως αριθμό δειγμάτων και χρονικό παράθυρο χρησιμοποιήθηκαν τα 200 ms που είναι η περίοδος του σήματος μας και εφαρμόστηκε μια ολίσθηση 25ms ώστε ο παλμός να είναι συμμετρικός ως προς το μηδέν. Στο ακόλουθο σχήμα παρουσιάζεται το αποτέλεσμα που προκύπτει από τον κώδικα που βρίσκεται στο παράρτημα :



5. Υπολογίστε μέσω της συνάρτησης fft το μετασχηματισμό Fourier της ακολουθίας παλμών και αποτυπώστε το επίσης γραφικά.

Απάντηση :

Αφού υπολογίστηκαν και σχεδιάστηκαν τα σήματα ενός μόνο παλμού και ακολουθίας παλμών στο πεδίο του χρόνου, με χρήση της συνάρτησης fft() υπολογίζεται ο μετασχηματισμός Fourier των σημάτων αυτών. Με βάση των αριθμό των δειγμάτων που επιλέχθηκαν υπολογίζεται ο αντίστοιχος πίνακας συχνοτήτων. Κατά την σχεδίαση λαμβάνεται το μονόπλευρο φάσμα συχνοτήτων και γι’ αυτό διπλασιάζεται το πλάτος της κάθε συχνότητας. Στο ακόλουθο σχήμα παρουσιάζονται τα φάσματα ενός μόνο παλμού και της ακολουθίας παλμών, που προκύπτουν από τον κώδικα που βρίσκεται στο παράρτημα :



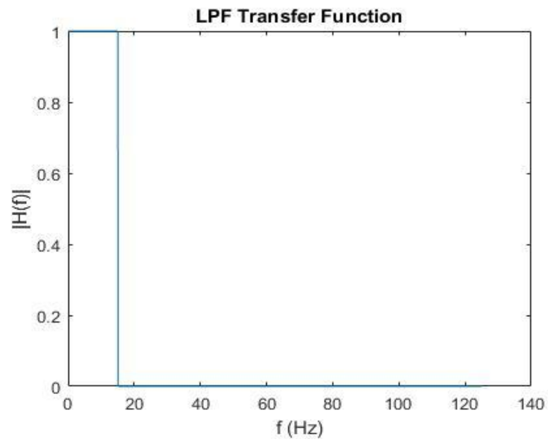
Το πρώτο φάσμα είναι συνεχές και ακολουθεί την μορφή $A \cdot T_o\, sinc(T_o f)$ (συνάρτηση δειγματοληψίας). Το δεύτερο φάσμα είναι μια ακολουθία κρουστικών συναρτήσεων, που έχουν περιβάλλουσα την μορφή $sinc(T_o f)$, όπως προκύπτει από την ιδιότητα :

$$Y(f) = \sum_{n=-\infty}^{\infty} \frac{1}{T_b} X(f) \cdot \delta(f - f_n), \text{ όπου } f_n = {}^{n}/_{T_b}$$

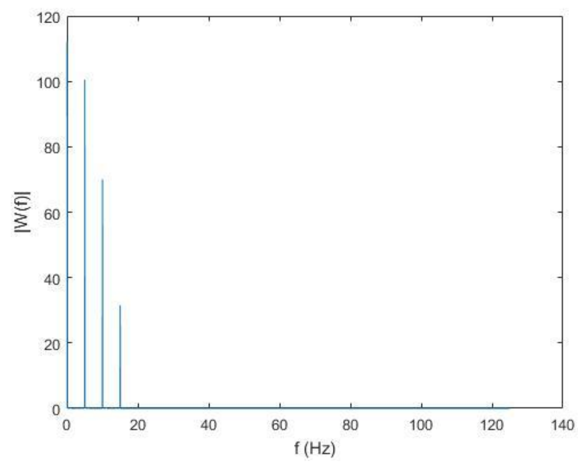
6. Θεωρείστε ότι το σήμα αυτό διέρχεται μέσα από ιδανικό βαθυπερατό φίλτρο με συνάρτηση μεταφοράς ίση με 1 μέχρι τη συχνότητα 15Hz. Υπολογίστε το μετασχηματισμό Fourier της ακολουθίας παλμών στην έξοδο του φίλτρου και μέσω της συνάρτησης αντίστροφου μετασχηματισμού Fourier απεικονίστε γραφικά την ακολουθία παλμών ως προς το χρόνο στην έξοδο του φίλτρου.

Απάντηση :

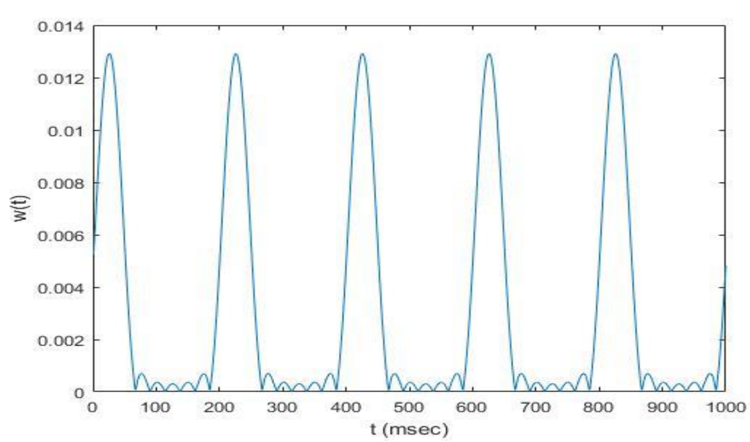
Σχεδιάζεται ιδανικό βαθυπερατό φίλτρο με συχνότητα αποκοπής $^{0.75}/_{T_o} = 15\,Hz$, του οποίου η συνάρτηση μεταφοράς φαίνεται στο ακόλουθο σχήμα:



Στην συνέχεια το σήμα Y(f) διέρχεται από το φίλτρο, οπότε η έξοδος του φίλτρου W(f), υπολογίζεται από την σχέση W(f) = Y(f) · H(f), και απεικονίζεται στο ακόλουθο σχήμα :



Όπως φαίνεται στην έξοδο του φίλτρου παραμένουν η dc συνιστώσα του σήματος (n=0), η 1^{η} αρμονική (n=1, 5Hz), η 2^{η} αρμονική (n=2, 10Hz) και η 3^{η} αρμονική (n=3, 15Hz). Οι συγκεκριμένες αρμονικές δηλώνουν την παρουσία τριών συνημιτονικών σημάτων στο πεδίο του χρόνου με συχνότητες 5Hz, 10Hz και 15 Hz με διαφορετικά πλάτη που αθροίζονται. Χρησιμοποιώντας την συνάρτηση ifft(W(f) υπολογίζεται ο αντίστροφος μετασχηματισμός Fourier που παρουσιάζεται στο ακόλουθο σχήμα :



Άσκηση 2: Ακολουθία τετραγωνικών παλμών με θόρυβο

Δημιουργείστε, για κάθε μία από τις παρακάτω περιπτώσεις χαρακτηριστικών θορύβου, μια ακολουθία λευκού Γκαουσιανού θορύβου (white Gaussian noise) n(t), συνολικού μήκους ίσου με το μήκος της παλμοσειράς y(t), και προσθέστε κάθε τιμή θορύβου στην αντίστοιχη τιμή του y(t), ώστε να αποκτήσει κατά αυτόν τον τρόπο θόρυβο η παλμοσειρά και να δημιουργηθεί η τελική θορυβώδης παλμοσειρά y1(t)=y(t)+n(t):

a. Θόρυβο με μέση τιμή 0 και τυπική απόκλιση 0,2

b. Θόρυβο με μέση τιμή 0 και τυπική απόκλιση 0,5

c. Θόρυβο με μέση τιμή 2 και τυπική απόκλιση 0,5

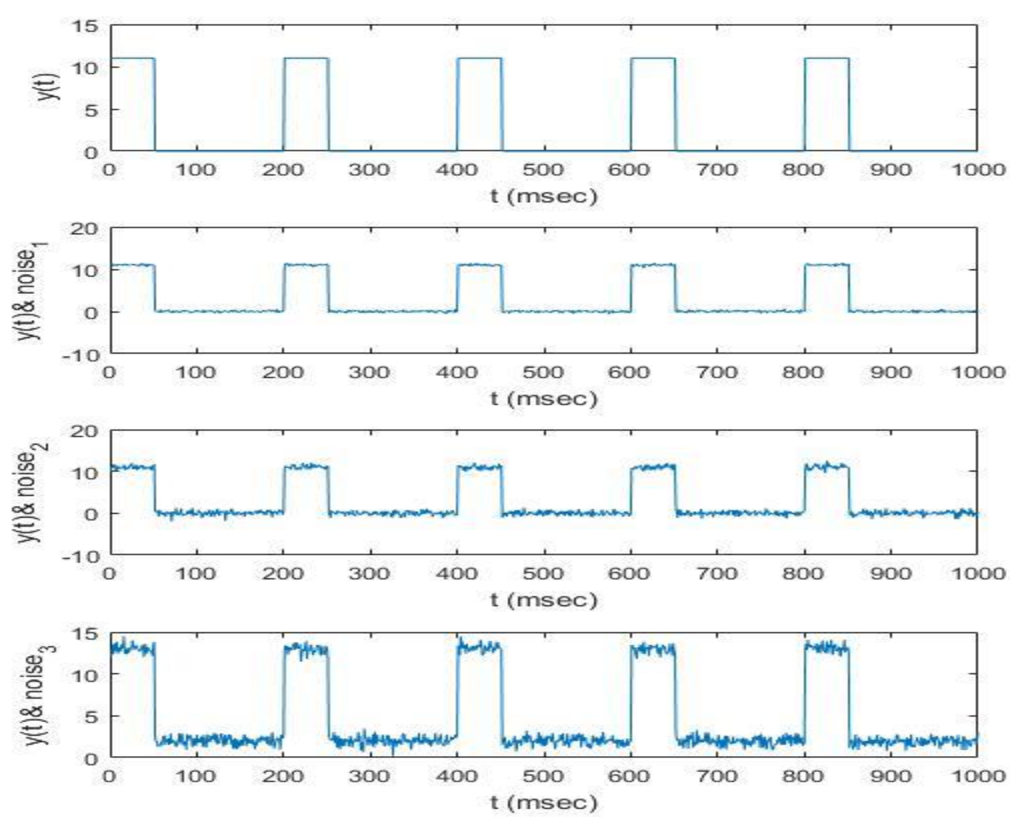
Για κάθε μία από τις παραπάνω περιπτώσεις:

1. Αποτυπώστε γραφικά (πχ μέσω συνάρτησης plot) τη συνολική ακολουθία y1(t) ως προς το χρόνο t για t=1,2,3…20.000 msec. Υπολογίστε τη μέση ισχύ του σήματός σας και το σηματοθορυβικό λόγο σε dB.

2. Δημιουργείστε γραφικά το διάγραμμα ματιού της ακολουθίας.

Απάντηση :

1. Ο θόρυβος παράγεται με χρήση συνάρτησης randn(), η οποία δημιουργεί 20.000 ψευδοτυχαίους αριθμούς που ακολουθούν κανονική κατανομή. Στην συνέχεια οι αριθμοί πολλαπλασιάζονται με την τυπική απόκλιση και προσθέτουμε την μέση τιμή του θορύβου, έτσι ώστε να προκύψουν οι τρεις διαφορετικές περιπτώσεις της άσκησης, όπως φαίνονται στο ακόλουθο σχήμα :

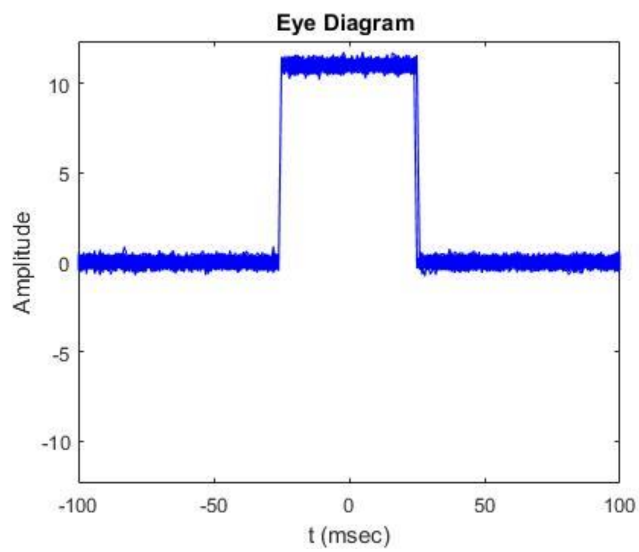


Επίσης υπολογίζεται η μέση ισχύς του σήματος με τον θόρυβο όπως έγινε στην άσκηση 1, δίνοντας :

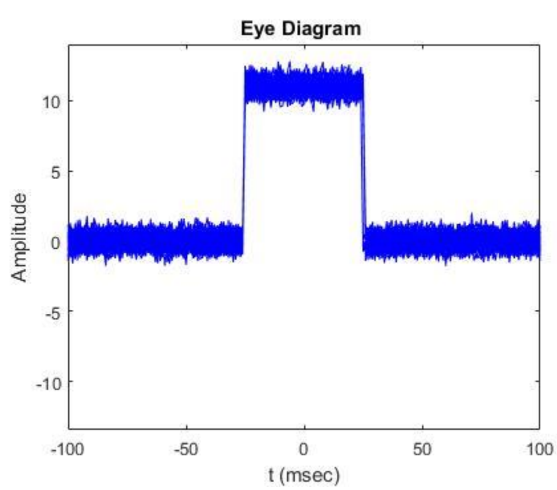
```
MATLAB Command Window                                          Page 1

The average power of the signal is equal to: 3.081265e+01
The average power of the signal with noise (mean value:0, deviation:0.2) is equal to:
3.085848e+01
The average power of the signal with noise (mean value:0, deviation:0.5) is equal to:
3.099667e+01
The average power of the signal with noise (mean value:2, deviation:0.5) is equal to:
4.625367e+01
>>
```

Όπως φαίνεται από τις τιμές της ισχύος, η επίδραση του θορύβου στο σήμα είναι πολύ μεγαλύτερη στην περίπτωση που η μέση τιμή του είναι μη μηδενική.

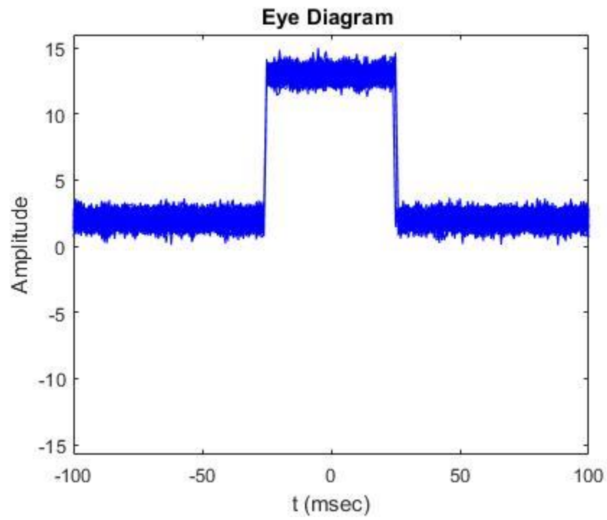
2. Ακολουθούν τα αντίστοιχα διαγράμματα ματιού του σήματος για κάθε μια από τις περιπτώσεις:



μ = 0, σ = 0.2



μ = 0, σ = 0.5



μ = 2, σ = 0.5

Από τα διαγράμματα ματιού φαίνεται ότι αύξηση της τυπικής απόκλισης του θορύβου μεταβάλλει αισθητά τις τιμές γύρω από το 0 και το 1.

Άσκηση 3: Διαμόρφωση ΑΜ

Θεωρείστε φέρον συνημίτονο Accos(2πfct), όπου Ac=A και fc=100Hz και σχηματίστε τις εξής διαμορφώσεις, όπου σε κάθε διαμόρφωση να πλοτάρετε το διαμορφωμένο σήμα στο χρόνο και στο φάσμα:

1. Διαμορφώστε το σήμα y(t) της Άσκησης 1, ερώτημα 3, με διαμόρφωση AM με συνολικό φέρον.
2. Διαμορφώστε το σήμα y(t) της Άσκησης 1, ερώτημα 3, με διαμόρφωση AM με καταπιεσμένο φέρον.
3. Διαμορφώστε το σήμα y1(t) της Άσκησης 2, περίπτωση b, με διαμόρφωση AM με συνολικό φέρον.
4. Διαμορφώστε το σήμα y1(t) της Άσκησης 2, περίπτωση b, με διαμόρφωση AM με καταπιεσμένο φέρον.

Απάντηση :

Το διαμορφωμένο σήμα w1(t), προκύπτει από διαμόρφωση AM με συνολικό φέρον και υπολογίζεται από την σχέση :

$$w1(t) = (1 + y(t)) \cdot x_c(t)$$

ενώ το διαμορφωμένο σήμα w2(t), προκύπτει από διαμόρφωση AM με καταπιεσμένο το φέρον και υπολογίζεται από την σχέση :

$$w2(t) = y(t) \cdot x_c(t)$$

όπου y(t) το σήμα πληροφορίας (παλμοσειρά του ερωτήματος 3 χωρίς θόρυβο) και $x_c(t)$ το συνημιτονικό φέρον.

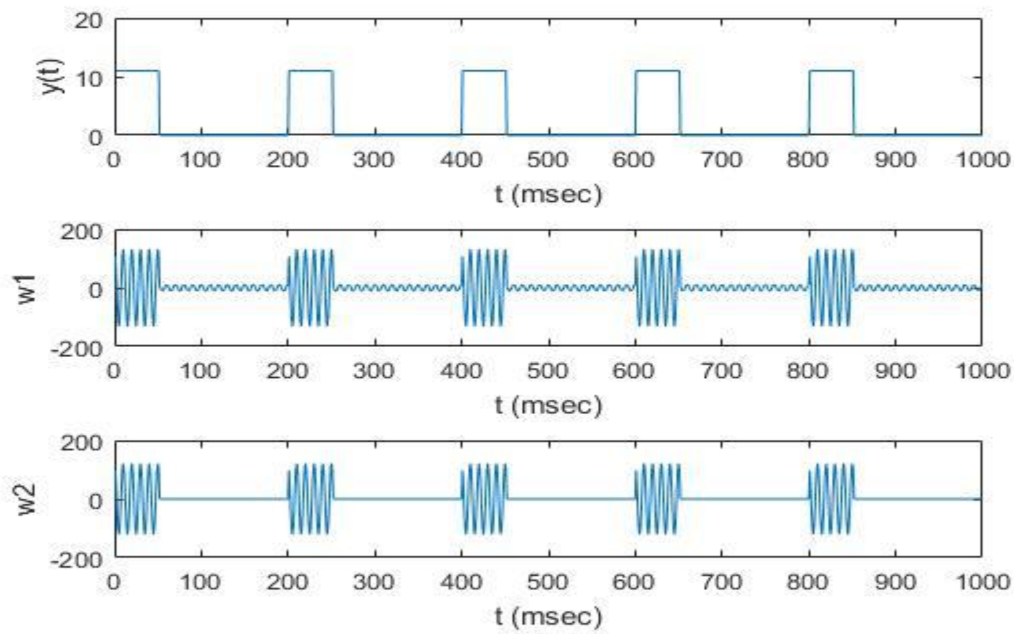
Από τον μετασχηματισμό Fourier προκύπτουν τα αντίστοιχα φάσματα των διαμορφωμένων σημάτων W1(f) και W2(f), που δίνονται από τις σχέσεις :

$$W1(f) = \frac{1}{2}[\delta(f - f_c) + \delta(f + f_c)] + \frac{1}{2}\,[Y(f - f_c) + Y(f + f_c)]$$

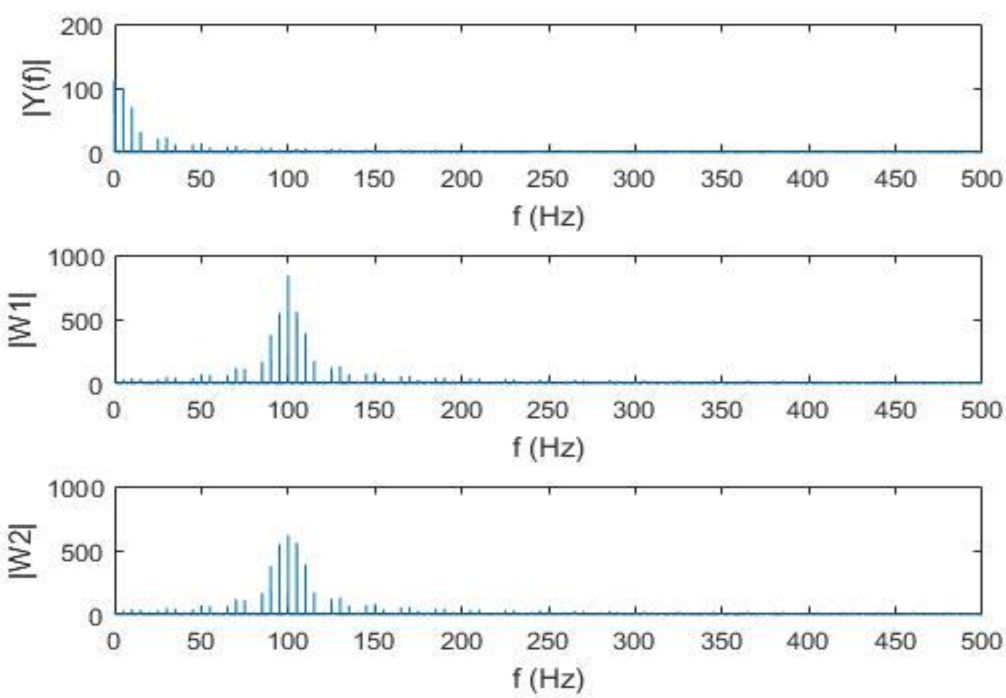
και

$$W2(f) = \frac{1}{2}\,[Y(f - f_c) + Y(f + f_c)]$$

Στο ακόλουθο σχήμα σχεδιάζονται τα σήματα y(t), w1(t) και w2(t) στο πεδίο του χρόνου:

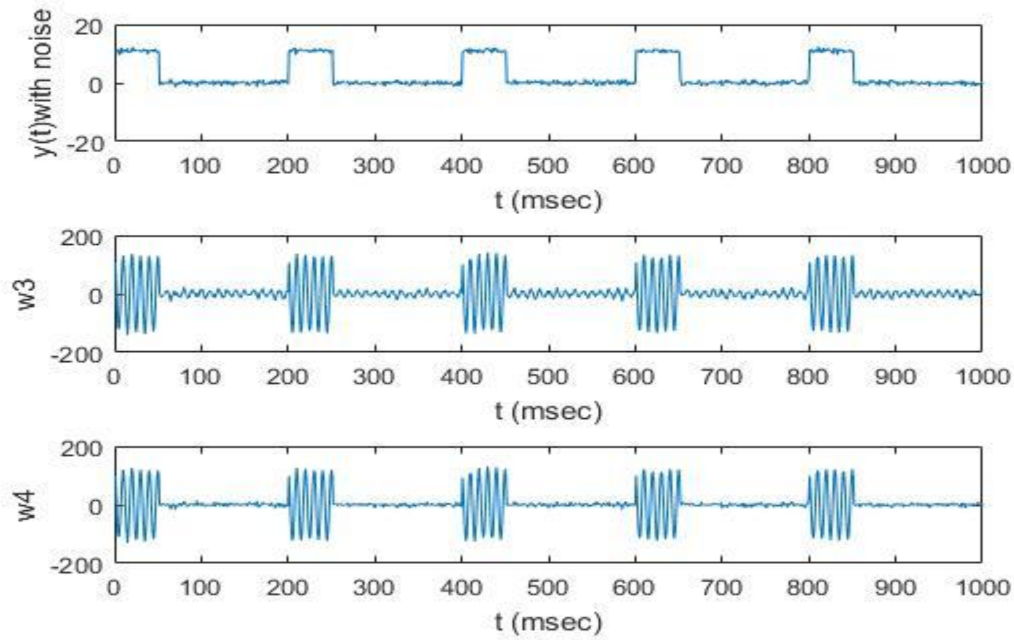


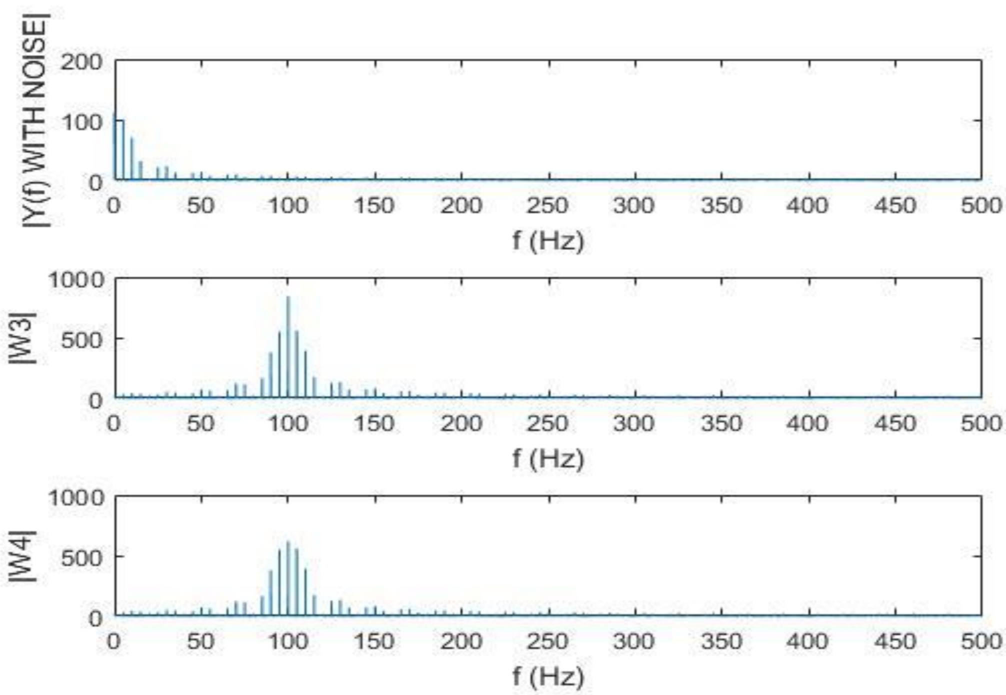
Στο ακόλουθο σχήμα σχεδιάζονται τα σήματα Y(f), W1(f) και W2(f) στο πεδίο της συχνότητας:



Όπως φαίνεται και από τα διαγράμματα και οι δύο διαμορφώσεις, δίνουν μετατόπιση του αρχικού φάσματος του σήματος πληροφορίας στην συχνότητα του φέροντος. Στο σήμα W1(f) επιπλέον υπάρχει το φέρον ως κρουστική συνάρτηση στα 100 Hz, ενώ στο σήμα W2(f) έχει καταπιεστεί.

Με όποιο τρόπο υπολογίζονται τα σήματα w3(t) και w4(t) και W3(f) και W4(f) για διαμόρφωση πλάτους με συνολικό και με καταπιεσμένο το φέρον, χρησιμοποιώντας ως σήμα πληροφορίας την παλμοσειρά του προηγούμενου ερωτήματος εισάγοντας όμως και θόρυβο κανονικής κατανομής στο σήμα μας. Έτσι προκύπτουν τα αντίστοιχα διαγράμματα :





Από τα παραπάνω διαγράμματα φαίνεται η επίδραση του θορύβου στο σήμα όπου μεταβάλει τις τιμές του πλάτους του αρχικού σήματος, τόσο στο πεδίο του χρόνου όσο και στο πεδίο των συχνοτήτων.

Παράρτημα. Κώδικας Matlab

Exam1_1

```
clear all

clc

Fs = 500;                    % Sampling frequency

t = -0.1:1/Fs:0.1;           % Time vector (sec)

To = 0.05;                   % Width of square pulse (sec)

A = 11;                      % Amplitude of square pulse

L= 1000;                     % Length to calculate the points of FFT

x = A*rectpuls(t,To);        % Generate Square Pulse

nfft = 2^nextpow2(L);        % Length of FFT

X = fft(x,nfft);             % Take symmetric FFT

X_symmentric = abs(X/Fs);           % Take the normalized magnitude of fft of X

X_symmentric = fftshift(X_symmentric);      % Rearranges a Fourier transform of X

f_symmetric = (-nfft/2:nfft/2-1)*Fs/nfft;   % Frequency vector

X_single_band = 2*abs(X/Fs) ;

X_single_band = X_single_band(1:nfft/2);    % Take side band spectrum and double the amplitude

ff = (0:nfft/2-1)*Fs/nfft;          % Frequency vector for side band spectrum

% Generate the plots time and frequency domains

figure(1);

plot(1000*t,x);

title('Square Pulse Signal');

xlabel('Time (ms)');

ylabel('Amplitude');

figure(2);

plot(f_symmetric,X_symmentric);

title('Power Spectrum of Square Pulse - Double Side Band');

xlabel('Frequency (Hz)');

ylabel('Power');

figure (3)

plot (ff (1:100), X_single_band (1:100))

title('Power Spectrum of Square Pulse - Single Side Band');
```

```
xlabel('Frequency (Hz)');
ylabel('Power');
```

Exam1_2_6

```
clear all
clc
A = 11;           % Amplitude of square pulse
Fs = 1000;        % Sampling frequency (Hz)
Ts = 1/Fs;        % Sampling period (sec)
Nsample = 20000;   % Number of samples
Df=Fs/Nsample ;    % Frequency resolution
bins=[1:Nsample];  % Number of bins - the same number of samples
Tb = 0.2;         % Period (sec)
To = 0.05;        % Width of square pulse (sec)
t = bins*Ts;      % time scale
f = bins*Df;      % frequency scale
%produce the basic rectangular pulsewaveform
S1=zeros(1,length(bins));
for i=[(1*To/50)/Ts:(51*To/50)/Ts]
   i=round(i);
   S1(i)=A;
end
%produce the clock signal of rectangular pulsewaveforms
k=0;
S2=zeros(1,length(bins));
while (51*To/50+k*Tb)/Ts<Nsample
      for i=[(1*To/50+k*Tb)/Ts:(51*To/50+k*Tb)/Ts]
      i=round(i);
      S2(i)=A;
   end
   k=k+1;
end
% Generate the plots time domains
```

```
figure (1)
title('Square Pulse Signal')
subplot (2,1,1)
plot (1000*t(1:100), S1(1:100))
xlabel ('t (msec)')
ylabel('x(t)')
subplot (2,1,2)
title('Clock signal');
plot (1000*t, S2)
xlabel ('t (msec)')
ylabel('y(t)')
% Calculate the average power of the signal
fprintf('The average power of the signal is equal to: %d\n', mean(S2.^2));
% Take FFT
YS1 = fft(S1)/(length(t)*Df); % Normalized fft
YS2 = fft(S2)/(length(t)*Df); % Normalized fft
% Generate the plots frequency domains - one sided spectrum
figure (2)
subplot (2,1,1)
plot (f(1:Nsample/2),abs(2*YS1(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|X(f)|')
subplot (2,1,2)
plot (f(1:Nsample/2),abs(2*YS2(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|Y(f)|')
% Generate the eyediagram
samplePerBit = 200;
timeSlot = 200;
offset = 25;
eyediagram (S2,samplePerBit,timeSlot,offset)
xlabel ('t (msec)')
ylabel('Amplitude')
```

```
%Design a LPF
LPF = zeros(1,length(bins));    % initialize LPF to have all zeros
BW=0.75/To;                     % defines bandwidth of LP filter
for i=1:BW/Df+1
   LPF(i)=1;
end
   for i=length(bins):-1:length(bins)-BW/Df
   LPF(i)=1;
end
figure (5)                      % plot the transfer function of LPF
plot (f(1:Nsample/8),LPF(1:Nsample/8))
title('LPF Transfer Function')
xlabel('f (Hz)')
ylabel('|H(f)|')
figure (6)                      % plot the output of the filter - frequency domain
W = LPF.*YS2;
plot (f(1:Nsample/8),abs(2*W(1:Nsample/8)))
xlabel('f (Hz)')
ylabel('|W(f)|')
figure (7)
w = ifft(W);                    % plot the output of the filter - time domain
plot (1000*t(1:1000), (abs(w(1:1000))))
xlabel ('t (msec)')
ylabel('w(t)')
```

Exam2

```
clear all
clc
clf
A = 11;            % Amplitude of square pulse
Fs = 1000;         % Sampling frequency (Hz)
Ts = 1/Fs;         % Sampling period (sec)
Nsample = 20000;   % Number of samples
```

```
bins=[1:Nsample];   % Number of bins - the same number of samples
Tb = 0.2;           % Period (sec)
To = 0.05;          % Width of square pulse (sec)
t = bins*Ts;        % time scale
%produce the clock signal of rectangular pulsewaveforms
k=0;
S2=zeros(1,length(bins));
while (51*To/50+k*Tb)/Ts<Nsample
       for i=[(1*To/50+k*Tb)/Ts:(51*To/50+k*Tb)/Ts]
       i=round(i);
       S2(i)=A;
    end
    k=k+1;
end
% generate noise
mu1=0;
sigma1=0.2;
no1 = randn(1, length(bins))*sigma1 + mu1;
mu2=0;
sigma2=0.5;
no2 = randn(1, length(bins))*sigma2 + mu2;
mu3=2;
sigma3=0.5;
no3 = randn(1, length(bins))*sigma3 + mu3;
% add noise to the signal
signal_noise1 = S2 + no1;
signal_noise2 = S2 + no2;
signal_noise3 = S2 + no3;
% Calculate the average power of the signal
fprintf('The average power of the signal is equal to: %d\n', mean(S2.^2));
fprintf('The average power of the signal with noise (mean value:%d, deviation:%2.1f) is equal to: %d\n',mu1,sigma1,mean(signal_noise1.^2));
```

```
fprintf('The average power of the signal with noise (mean value:%d, deviation:%2.1f) is equal to: %d\n',mu2,sigma2,mean(signal_noise2.^2));

fprintf('The average power of the signal with noise (mean value:%d, deviation:%2.1f) is equal to: %d\n',mu3,sigma3,mean(signal_noise3.^2));

% Generate the plots time domains

figure (1)

title('Pulse train Signal')

subplot (4,1,1)

plot (1000*t(1:1000), S2(1:1000))

xlabel ('t (msec)')

ylabel('y(t)')

subplot (4,1,2)

title('Clock signal');

plot (1000*t(1:1000), signal_noise1(1:1000))

xlabel ('t (msec)')

ylabel('y(t)& noise_1')

subplot (4,1,3)

title('Clock signal');

plot (1000*t(1:1000), signal_noise2(1:1000))

xlabel ('t (msec)')

ylabel('y(t)& noise_2')

subplot (4,1,4)

title('Clock signal');

plot (1000*t(1:1000), signal_noise3(1:1000))

xlabel ('t (msec)')

ylabel('y(t)& noise_3')

% Generate the eyediagram

samplePerBit = 200;

timeSlot = 200;

offset = 25;

eyediagram (S2,samplePerBit,timeSlot,offset)

xlabel ('t (msec)')

ylabel('Amplitude')
```

```
eyediagram (signal_noise1,samplePerBit,timeSlot,offset)
xlabel ('t (msec)')
ylabel('Amplitude')
eyediagram (signal_noise2,samplePerBit,timeSlot,offset)
xlabel ('t (msec)')
ylabel('Amplitude')
eyediagram (signal_noise3,samplePerBit,timeSlot,offset)
xlabel ('t (msec)')
ylabel('Amplitude')
```

<u>Exam3</u>

```
clear all
clc
A = 11;           % Amplitude of square pulse
Fs = 1000;        % Sampling frequency (Hz)
Ts = 1/Fs;        % Sampling period (sec)
Nsample = 20000;  % Number of samples
Df=Fs/Nsample ;   % Frequency resolution
bins=[1:Nsample]; % Number of bins - the same number of samples
Tb = 0.2;         % Period (sec)
To = 0.05;        % Width of square pulse (sec)
t = bins*Ts;      % time scale
f = bins*Df;      % frequency scale
Ac = A;           % Amplitude of carrier
fc = 100;         % Carrier frequency (Hz)
% produce the clock signal of rectangular pulsewaveforms
k=0;
y=zeros(1,length(bins));
while (51*To/50+k*Tb)/Ts<Nsample
    for i=[(1*To/50+k*Tb)/Ts:(51*To/50+k*Tb)/Ts]
    i=round(i);
    y(i)=A;
  end
```

```
    k=k+1;
end
% produce the carrier signal
xc = Ac * cos(2*pi*fc*t);
% generate noise
mu = 0;
sigma = 0.5;
no = randn(1, length(bins))*sigma + mu;
% add noise to the signal
signal_noise = y + no;
% AM modulation
w1 = (1+y).*xc;                  %Equation of Amplitude modulated signal DSB TC
w2 = y.*xc;                      %Equation of Amplitude modulated signal DSB SC
w3 = (1+signal_noise).*xc;       %Equation of Amplitude modulated signal DSB TC with noise
w4 = signal_noise.*xc;           %Equation of Amplitude modulated signal DSB SC with noise
% Generate the plots time domains
figure (1)
title('Clock signal')
subplot (3,1,1)
plot (1000*t(1:1000), y(1:1000))
xlabel ('t (msec)')
ylabel('y(t)')
subplot (3,1,2)
title('Amplitude Modulated signal DSB TC');
plot (1000*t(1:1000), w1(1:1000))
xlabel ('t (msec)')
ylabel('w1')
subplot (3,1,3)
title('Amplitude Modulated signal DSB SC');
plot (1000*t(1:1000), w2(1:1000))
xlabel ('t (msec)')
ylabel('w2')
figure (2)
```

```
title('Clock signal')
subplot (3,1,1)
plot (1000*t(1:1000), signal_noise(1:1000))
xlabel ('t (msec)')
ylabel('y(t)with noise')
subplot (3,1,2)
title('Amplitude Modulated signal DSB TC with noise ');
plot (1000*t(1:1000), w3(1:1000))
xlabel ('t (msec)')
ylabel('w3')
subplot (3,1,3)
title('Amplitude Modulated signal DSB SC with noise ');
plot (1000*t(1:1000), w4(1:1000))
xlabel ('t (msec)')
ylabel('w4')
% Take normalized FFT
Y = fft(y)/(length(t)*Df);
SIGNAL_NOISE = fft(signal_noise)/(length(t)*Df);
W1 = fft(w1)/(length(t)*Df);
W2 = fft(w2)/(length(t)*Df);
W3 = fft(w3)/(length(t)*Df);
W4 = fft(w4)/(length(t)*Df);
% Generate the plots frequency domains - one sided spectrum
figure (3)
subplot (3,1,1)
plot (f(1:Nsample/2),abs(2*Y(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|Y(f)|')
subplot (3,1,2)
plot (f(1:Nsample/2),abs(2*W1(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|W1|')
subplot (3,1,3)
```

```
plot (f(1:Nsample/2),abs(2*W2(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|W2|')
figure (4)
subplot (3,1,1)
plot (f(1:Nsample/2),abs(2*SIGNAL_NOISE(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|Y(f) WITH NOISE|')
subplot (3,1,2)
plot (f(1:Nsample/2),abs(2*W3(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|W3|')
subplot (3,1,3)
plot (f(1:Nsample/2),abs(2*W4(1:Nsample/2)))
xlabel ('f (Hz)')
ylabel('|W4|')
```