Ο κώδικας για όλα τα ερωτήματα σε συνολο:

```
%TELECOMMUNICATION SYSTEMS I
%exercise 2
%authors-Panagiotis Sklabos/Giannis Peridis

%MATLAB CODE
clear all;
close all;
clc;
format compact;
%======================================================================================
%A.1
%intialization of symbol period T,oversampling factor over,
%parameter A(half duration of the pulse) and roll-off factor a
T=10^-2;
over=10;
A=4;
a=0.5;

%creating srrc pulses
[phi,t]=srrc_pulse(T,over,A,a);

%intialization of sampling period Ts,sampling frequency Fs,
%parameter Nf(number of equidistant points) and F axis
Ts=T/over;
Fs=1/Ts;
Nf=4096;
F=[-Fs/2:Fs/Nf:Fs/2-Fs/Nf];

%creating fast fourier transformation
PHI=fftshift(fft(phi,Nf)*Ts);

%creating the energy density spectrum
edsPHI=abs(PHI).^2;

%creating graph of the energy density spectrums
%in the same plot
disp('press any key to see the energy density spectrum with logarithmic yaxis')
pause
figure();
```

```
semilogy(F,edsPHI);
title('energy density spectrum with logarithmic yaxis');
xlabel('F=frequency in Hz');
ylabel('edsPhi =energy density spectrum of srrc pulses');
legend('a = 0.5');

%==================================================================================
%A.2
%N number of bits
N=100;

%call of the created function that transfers bits to 2PAM
disp('a random sequence of independent and  equally possible bits')
b=(sign(randn(N,1))+1)/2;
o=transpose(b)

%call of the created function that transfers bits to 2PAM
disp('the transformed 2PAM sequence')
Xn=bits_to_2PAM(b)
t1=(0:Ts:N*T-Ts);

%creation of X_delta
X_delta=1/Ts*upsample(Xn,over);

%creation of X(t)=sum(Xn*phi(t-n*T))
%convolution of phi and X_delta
tconv=[t(1)+t1(1):Ts:t(end)+t1(end)];
X_t=conv(phi,X_delta)*Ts;
%creation of the graph of X(t)
disp('press any key to see X(t) waveform ')
pause
figure()
plot(tconv,X_t);
title(' X(t)=sum(Xn*phi(t-n*T))')
ylabel(' X(t)')
xlabel('t=time in seconds')

%==================================================================================
%A3
%creating fast fourier transformation X(F) of  X(t)
X_F=fftshift(fft(X_t,Nf)*Ts);
%the total duration time
Ttotal=length(tconv)*Ts;
```

```
%creation of a periodogram of one of the implementations of X(t)
P_x=abs(X_F).^2/Ttotal;

%creation of periodogram graph in logarithmic y axis
disp('press any key to see the periodogram')
pause
figure();
semilogy(F,P_x);
title('Periodogram with logarithmic y axis');
xlabel('F=frequency in Hz');
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('a = 0.5');

%creation of periodogram graph in plot
disp('press any key to see the periodogram with logarithmic y axis')
pause
figure();
plot(F,P_x);
title('Periodogram of X');
xlabel('F=frequency in Hz');
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('a = 0.5');

%initializing K the number of periodograms Px(F) of the of implemenations of
%X(t) and sum ,helps to find the summary
K =500;
sum=0;
%every loop we create a new implementation of X(t) and a new periodogram
%Px(F) for it, we repeat K times
for k=0:K-1
    b=(sign(randn(N,1))+1)/2;
    Xn=bits_to_2PAM(b);
    X_delta=1/Ts*upsample(Xn,over);
    X_t=conv(phi,X_delta)*Ts;
    X_F=fftshift(fft(X_t,Nf)*Ts);
    P_x=abs(X_F).^2/Ttotal;
    sum=sum+P_x;
end
%creation of the estimated power density spectrum, we divide the  summary by K times
%to find the arithmetic mean
S_x=sum/K;

```

```
%creating the theoritical Sx(F)
variance=1;
S_x_th= (variance*(abs(PHI).^2))/T;

%creating the graph of the estimated Sx(F) and the theoritical Sx(F) in the
%same plot with logarithmic y axis
disp('press any key to see the graph of the estimated Sx(F) and the theoritical Sx(F) in the same plot with logarithmic y axis')
pause
figure();
semilogy(F,S_x,F,S_x_th);
title('estimated Sx(F) and theoritical Sx(F) smilogy');
xlabel('F=frequency in Hz');
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('estimated', 'theoritical');
figure

plot(F,S_x,F,S_x_th);
title('estimated Sx(F) and theoritical Sx(F) plot');
xlabel('F=frequency in Hz');
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('estimated', 'theoritical');
%=====================================================================================
%A4
%creating a random sequence of N bits
b=(sign(randn(N,1))+1)/2;
%call of the created function that transfers bits to 4PAM
Xn_4PAM=bits_to_4PAM(b);
t_4PAM=(0:Ts:length(Xn_4PAM)*T-Ts);

%creation of X_delta
X_delta_4PAM=1/Ts*upsample(Xn_4PAM,over);

%creation of X(t)=sum(Xn*phi(t-n*T))
%convolution of phi and X_delta
tconv_4PAM=[t(1)+t_4PAM(1):Ts:t(end)+t_4PAM(end)];
X_t_4PAM=conv(phi,X_delta_4PAM)*Ts;
%creation of the graph X(t)
disp('press any key to see X(t) waveform ')
pause
figure()
plot(tconv_4PAM,X_t_4PAM);
title(' X(t)=sum(Xn*phi(t-n*T))')
```

```
ylabel(' X(t)')
xlabel('t=time in seconds')

%creation of a periodogram of one of the implementations of X(t)_4PAM
X_F_4PAM=fftshift(fft(X_t_4PAM,Nf)*Ts);
Ttotal=length(tconv_4PAM)*Ts;
P_x_4PAM=abs(X_F_4PAM).^2/Ttotal;

%creation of periodogram graph in plot
disp('press any key to see the periodogram of X(t)_4PAM ')
pause
figure();
semilogy(F,P_x_4PAM);
title('Periodogram with logarithmic y axis');
xlabel('F=frequency in Hz');
ylabel('Px(F)_4PAM=(|X(F)_4PAM|^2)/Ttotal');
legend('a = 0.5');

%creation of periodogram graph in logarithmic y axis
disp('press any key to see the periodogram of X(t)_4PAM with logarithmic y axis')
pause
figure();
plot(F,P_x_4PAM);
title('Periodogram of X_4PAM');
xlabel('F=frequency in Hz');
ylabel('Px(F)_4PAM=(|X(F)_4PAM|^2)/Ttotal');
legend('a = 0.5');

%initializing K the number of periodograms Px(F)_4PAM of the of implemenations of
%X(t)_4PAM and sum ,helps to find the summary
K =500;
sum_4PAM=0;
%every loop we create a new implementation of X(t)_4PAM and a new periodogram
%Px(F)_4PAM for it, we repeat K times
for k=0:K-1
    b=(sign(randn(N,1))+1)/2;
    Xn_4PAM=bits_to_4PAM(b);
    X_delta_4PAM=1/Ts*upsample(Xn_4PAM,over);
    X_t_4PAM=conv(phi,X_delta_4PAM)*Ts;
    X_F_4PAM=fftshift(fft(X_t_4PAM,Nf)*Ts);
    P_x_4PAM=abs(X_F_4PAM).^2/Ttotal;
    sum_4PAM=sum_4PAM+P_x_4PAM;
end
```

```
%creation of the estimated power density spectrum, we divide the  summary by K times
%to find the arithmetic mean
S_x_4PAM=sum_4PAM/K;

%creating the theoritical Sx(F)_4PAM
variance=5;
S_x_th_4PAM= (variance*(abs(PHI).^2))/T;

%creating the graph of the estimated Sx(F)_4PAM and the theoritical Sx(F)_4PAM in the
%same plot with logarithmic y axis
disp('press any key to see the graph of the estimated Sx(F)_4PAM and the theoritical Sx(F)_4PAM in the same plot
      with logarithmic y axis')
pause
figure();
semilogy(F,S_x_4PAM);
hold on;
semilogy(F,S_x_th_4PAM);
title('estimated Sx(F)_4PAM and theoritical Sx(F)_4PAM ');
xlabel('F=frequency in Hz');
ylabel('Px(F)_4PAM=(|X(F)_4PAM|^2)/Ttotal');
legend('estimated', 'theoritical');

%comparing the S(X) 2PAM and S(X) 4PAM in plot
disp('press any key to  see the S(X) 2PAM and S(X) 4PAM in the same plot')
pause
figure()
semilogy(F,S_x,F,S_x_4PAM);
title('P(X)_2PAM and P(X)_4PAM')
xlabel('F=frequency in Hz')
ylabel('P(X)_2PAM , P(X)_4PAM')
legend('2PAM','4PAM')
disp('press any key to  see the S(X) 2PAM and S(X) 4PAM in the same plot with logarithmic y axis')
figure()
plot(F,S_x,F,S_x_4PAM);
title('P(X)_2PAM , P(X)_4PAM')
xlabel('F=frequency in Hz')
ylabel('P(X)_2PAM , P(X)_4PAM')
legend('2PAM','4PAM')
%=====================================================================================
%A5
%initializing the new period to double, that means over is doubled too, and
%keeping everything else as it is
T=2*T;
```

```
over=2*over;
%creating srrc pulse
[phi,t]=srrc_pulse(T,over,A,a);
%creating fast fourier transformations
PHI=fftshift(fft(phi,Nf)*Ts);
%creating the energy density spectrum
edsPHI=abs(PHI).^2;

%creating a sequence of N bits
b=(sign(randn(N,1))+1)/2;
%call of the created function that transfers bits to 2PAM
Xn=bits_to_2PAM(b);
t1=(0:Ts:N*T-Ts);
%creating X_delta
X_delta=1/Ts*upsample(Xn,over);

%creating X(t)
%convolution of phi and X_delta
tconv=[t(1)+t1(1):Ts:t(end)+t1(end)];
X_t=conv(phi,X_delta)*Ts;
%creation of the graph of X(t)
disp('press any key to see X(t) waveform ')
pause
figure()
plot(tconv,X_t);
title(' X(t)=sum(Xn*phi(t-n*T)), T_2=2T')
ylabel(' X(t)')
xlabel('t=time in seconds')

%creating fast fourier transformation X(F) of  X(t)
X_F=fftshift(fft(X_t,Nf)*Ts);
%the total duration time
Ttotal=length(tconv)*Ts;
%creation of a periodogram of one of the implementations of X(t)
P_x=abs(X_F).^2/Ttotal;

%creation of periodogram graph in plot
disp('press any key to see the periodogram')
pause
figure();
semilogy(F,P_x);
title('Periodogram with logarithmic y axis, T_2=2T');
xlabel('F=frequency in Hz');
```

```
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('a = 0.5');

%creation of periodogram graph in logarithmic y axis
disp('press any key to see the periodogram with logarithmic y axis')
pause
figure();
plot(F,P_x);
title('Periodogram of X, T_2=2T');
xlabel('F=frequency in Hz');
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('a = 0.5');

%initializing K the number of periodograms Px(F) of the of implemenations of
%X(t) and sum ,helps to find the summary

K =50;
sum=0;
%every loop we create a new implementation of X(t) and a new periodogram
%Px(F) for it, we repeat K times
for k=0:K-1
    b=(sign(randn(N,1))+1)/2;
    Xn=bits_to_2PAM(b);
    X_delta=1/Ts*upsample(Xn,over);
    X_t=conv(phi,X_delta)*Ts;
    X_F=fftshift(fft(X_t,Nf)*Ts);
    P_x=abs(X_F).^2/Ttotal;
    sum=sum+P_x;
end
%creation of the estimated power density spectrum, we divide the  summary by K times
%to find the arithmetic mean
S_x=sum/K;

%creating the theoritical Sx(F)
variance=1;
S_x_th= (variance*(abs(PHI).^2))/T;

%creating the graph of the estimated Sx(F) and the theoritical Sx(F) in the
%same plot with logarithmic y axis
disp('press any key to see the graph of the estimated Sx(F) and the theoritical Sx(F) in the same plot with logarithmic y axis')
pause
figure();
```

```
semilogy(F,S_x,F,S_x_th);
title('estimated Sx(F) and theoritical Sx(F) smilogy');
xlabel('F=frequency in Hz');
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('estimated', 'theoritical');
figure

plot(F,S_x,F,S_x_th);
title('estimated Sx(F) and theoritical Sx(F) plot');
xlabel('F=frequency in Hz');
ylabel('Px(F)=(|X(F)|^2)/Ttotal');
legend('estimated', 'theoritical');
%making the period and the over back to the staring ones
T=T/2;
over=over/2;
%===================================================================================
%B4
%initializing the modulation frequency f0 between 1/2T and (FS/2-1/2T)=>50<f0<450 and
% K number of waveforms
f0=300;
K = 4;

%creating srrc pulse
[phi,t]=srrc_pulse(T,over,A,a);
t1=(0:Ts:N*T-Ts);
variance=1;
%for every loop we make a different modulated waveform that comes from 2PAM
%its different , due to the change of theta and the change of X(t) sequence
for k=0:K-1
    b=(sign(randn(N,1))+1)/2;
    Xn=bits_to_2PAM(b);
    X_delta=1/Ts*upsample(Xn,over);
    X_t=conv(phi,X_delta)*Ts;
    tconv=[t(1)+t1(1):Ts:t(end)+t1(end)];
    %theta is a random variable evenly distributed
    Theta=2*pi*rand;
    %modulation
    Y_t=X_t.*cos(2*pi*f0*tconv+Theta);
    %making fast fourier transformation
    Y_F=fftshift(fft(Y_t,Nf)*Ts);
    absY=abs(Y_F);
    %making the graphs ,we put absY , because Y has a complex part too, and
    %we just want to see the meter
```

```
    figure()
    plot(F,absY)
    xlabel('F=frequency in Hz')
    title('the modulated :Y(F)=X(t)*cos(2*pi*f0+\theta)')
    ylabel('Y(F)')
end

%initializing K number of periodograms , total duration time and the sums
%we use to create the Py(F) and Px(F)
K=100;
Ttotal=length(tconv)*Ts;
sumX=0;
sumY=0;
%every loop we make a different implementation of a  periodogramm for the same Y(t)
%its different , due to the change of theta
for k=0: K-1
   Theta=2*pi*rand;
   Y_t=X_t.*cos(2*pi*f0*tconv+Theta);
   Y_F=fftshift(fft(Y_t,Nf)*Ts);
   absY=abs(Y_F);
   P_y=absY.^2/Ttotal;
   sumY=sumY+P_y;
end
%divide the summary by K so we can find the power density spectrum
S_y=sumY/K;

%creation of the graph of Sy(F) for plot an dplot with logarithmic y axis
disp('press an key to see the power density spectrum :Sy(F)in logarithmic y axis ')
pause
semilogy(F,S_y)
title('power density spectrum :Sy(F)in logarithmic y axis')
xlabel('Sy(F)')
ylabel('F=frequency in Hz')
disp('press any key to see the power density spectrum :Sy(F)')
pause
figure();
plot(F,S_y)
title('power density spectrum :Sy(F)')
xlabel('Sy(F)')
ylabel('F=frequency in Hz')
```