Πολυτεχνείο Κρήτης

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Τηλεπικοινωνιακά Συστήματα Ι

1η Εργαστηριακή Άσκηση

Τρίμας Χρήστος	***2016030054***
Μιχαήλ Αλέξανδρος	***2014030077***

Σκοπός εργαστηριακής άσκησης:

Στη 1η εργαστηριακή άσκηση του μαθήματος Τηλεπικοινωνιακά Συστήματα Ι, μελετήθηκε η επικοινωνία βασικής ζώνης, με διαμόρφωση 2-PAM και αποκομμένους παλμούς Square Root Raised Cosine ή για συντομία srrc.

Υλοποίηση ερωτημάτων:

A1:

Αρχικά, με την βοήθεια της έτοιμης συνάρτησης matlab που δόθηκε από τον διδάσκοντα, υλοποιήθηκαν παλμοί τύπου srrc, fi(t) σε κοινή απεικόνιση με παραμέτρους T = 10^-2 sec, over = 10, A = 4 και α1=0, α2=0.5, α3=1. Επιπλέον ο χρόνος T_s, ορίζεται ως: T_s = T/over.

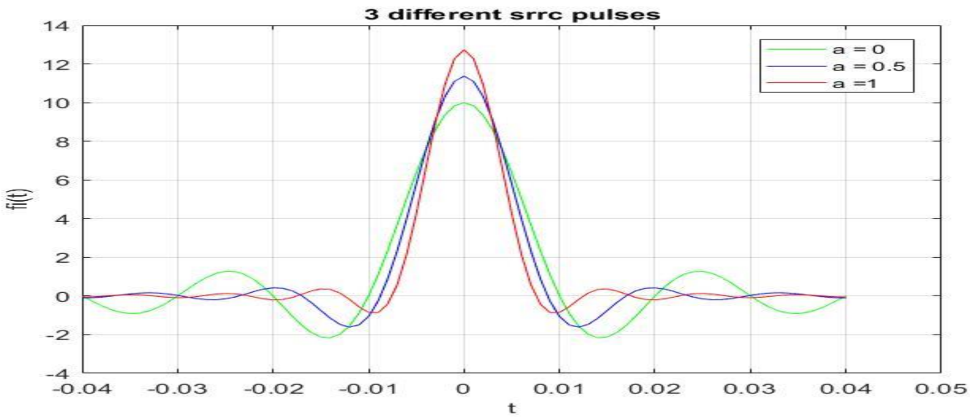
Ο παλμός fi(t) με α=0, υπολογίζεται από την σχέση: $fi(t) = \frac{1}{\sqrt{T}} sinc\left(\frac{\tau}{T}\right)$

Σε περίπτωση που το α παίρνει τιμές στο διάστημα (0,1], τότε ο παλμός υπολογίζεται από την ακόλουθη σχέση:

$$fi(t) = \frac{4\alpha}{\pi\sqrt{T}} \frac{\cos\left((1+\alpha)\pi\frac{\tau}{T}\right) + \frac{\sin\left((1-\alpha)\pi\frac{\tau}{T}\right)}{4\alpha\frac{\tau}{T}}}{1-\left(\frac{4\alpha\tau}{T}\right)^2}, 0 < \alpha \leq 1$$

Αν το α, πάρει οποιαδήποτε άλλη τιμή, τότε η συνάρτηση επιστρέφει ένα διάνυσμα μηδενικών.

Παρακάτω απεικονίζονται σε κοινά plot οι παλμοί.



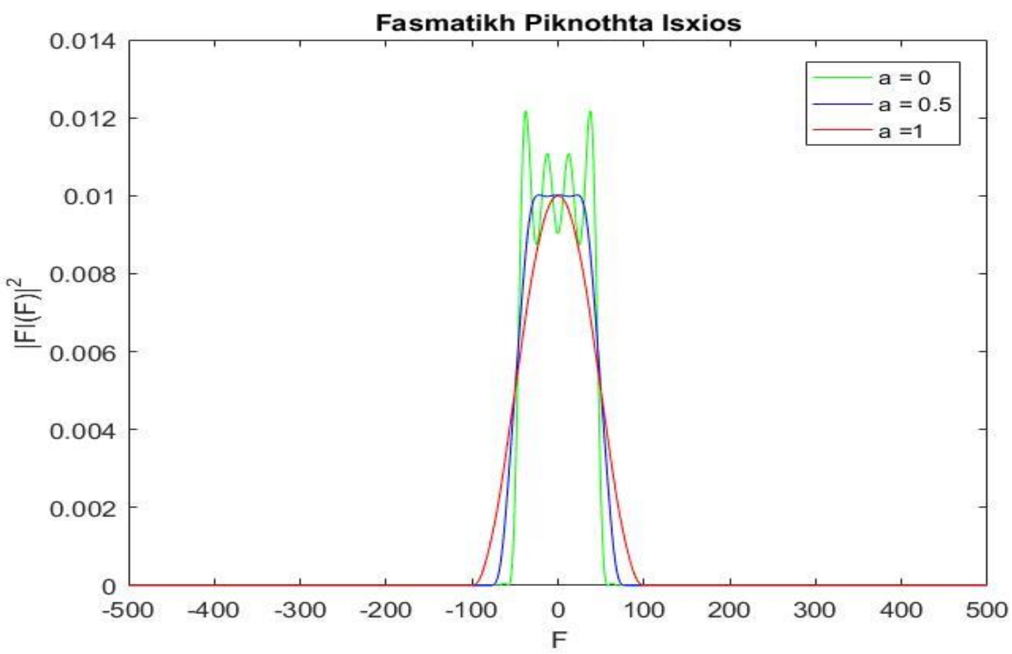
Παρατηρείται ότι, με την αύξηση του συντελεστή α, ο ρυθμός με τον οποίο μειώνεται το πλάτος του παλμού αυξάνεται, με το πέρασμα του χρόνου. Παρατηρείται επίσης, μείωση στην περίοδο της ταλάντωσης με την αύξηση του α, αλλά ταυτόχρονα μια αύξηση στο μέγιστο του παλμού. Ακολουθεί ο αντίστοιχος κώδικας υλοποίησης:

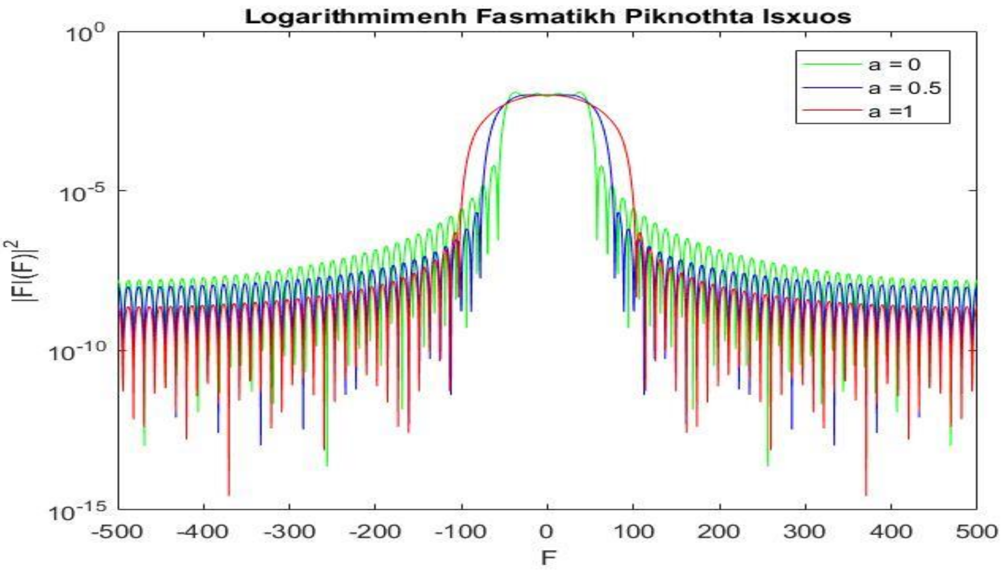
```
T = 10^(-2); %period
over = 10;
Ts = T/over; %periodos deigmatolipsias
A = 4;
a1 = 0;   %green pulse
a2 = 0.5; %blue pulse
a3 = 1;   %red pulse


%creation of srrc pulses
[phi1,t1] = srrc_pulse(T, Ts, A, a1);
[phi2,t2] = srrc_pulse(T, Ts, A, a2);
[phi3,t3] = srrc_pulse(T, Ts, A, a3);

%plots
figure;
plot(t1,phi1,'g')
hold on
plot(t2,phi2,'b')
hold on
plot(t3,phi3,'r')
xlabel('t')
ylabel('fi(t)')
title('3 different srrc pulses')
legend('a = 0','a = 0.5', 'a =1')
```

A2:

Με χρήση των fft και fftshift συναρτήσεων του matlab, υπολογίστηκε ο μετασχηματισμός Φουριέ του σήματος fi(t),FI(F) και σχεδιάστηκε με N = 2048 ισαπέχοντα σημεία στον άξονα της συχνότητας f στο διάστημα [$-F_s/2$, $F_s/2$).

Παρακάτω παρουσιάζεται η φασματική πυκνότητα ισχύος του MF των παλμών τόσο κανονικά, όσο και «συμπιεσμένη» μέσο της συνάρτησης του matlab semilogy.





Παρατήρηση: Η συνάρτηση fftshift κεντράρει τον μετασχηματισμό Φουριέ στο μηδέν, ενώ η συνάρτηση fft υπολογίζει απλά τον μετασχηματισμό.

```
%a2
Fs = 1/Ts;
Nf = 2048;                              %samples
F = [-Fs/2:Fs/Nf:Fs/2-Fs/Nf]; %frequency

y1 = fftshift(fft(phi1,Nf)*Ts); %fourier transformation
y2 = fftshift(fft(phi2,Nf)*Ts);
y3 = fftshift(fft(phi3,Nf)*Ts);
```

```
figure;
plot(F,abs(y1).^2,'g')
hold on
plot(F,abs(y2).^2,'b')
hold on
plot(F,abs(y3).^2,'r')
xlabel('F');
ylabel('|FI(F)|^2')
title('Fasmatikh Piknothta Isxios')
legend('a = 0','a = 0.5', 'a =1')

figure;
semilogy(F,abs(y1).^2,'g');
hold on;
semilogy(F,abs(y2).^2,'b');
hold on;
semilogy(F,abs(y3).^2,'r');
xlabel('F');
ylabel('|FI(F)|^2')
title('Logarithmimenh Fasmatikh Piknothta Isxuos')
legend('a = 0','a = 0.5', 'a =1')
```

Με την χρήση της semilogy συνάρτησης, η οποία σχεδιάζει σε λογαριθμική κλίμακα τον κατακόρυφο άξονα, επιτρέπει την δυνατότητα να μελετηθούν τιμές όπου η συνάρτηση μας παίρνει πολύ μικρές τιμές(για αυτό και ο όρος συμπίεση). Στην συγκεκριμένη περίπτωση, παρατηρείται ότι η φασματική πυκνότητα ενέργειας μειώνεται με την αύξηση του roll-off factor.

A3:

Σε πρώτη φάση, υπολογίστηκε θεωρητικά το bandwidth με την χρήση του τύπου:

BW = 1+a/2T.

Για α=0 και T=0.01: BW = 50

Για α=0,5 και T=0.01: BW = 75

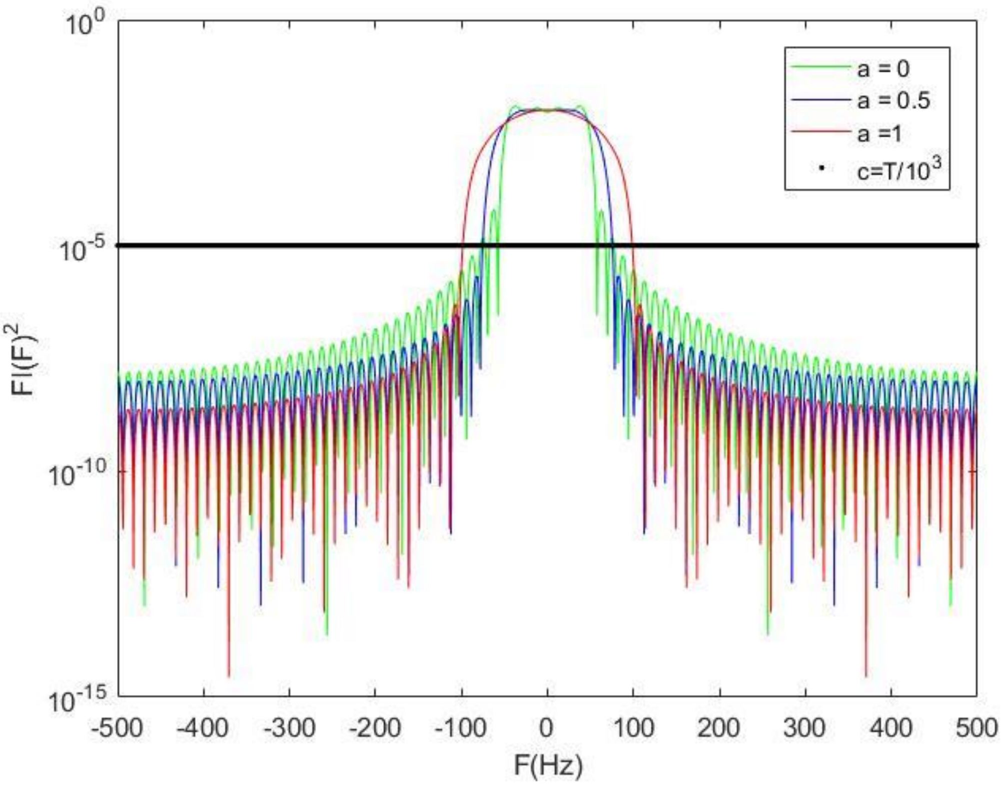
Για α=1 και T=0.01: BW = 100

Με κοινό plot, σχεδιάστηκε η c1 γραμμή με τιμή c1=10^-5 και θεωρήθηκε ότι οι τιμές κάτω από αυτή την ευθεία είναι μηδενικές. Είναι γνωστό ότι η συχνότητα δεν παίρνει αρνητικές τιμές, οπότε fmin=0 και επομένως BW = fmax, το οποίο είναι το τελευταίο σημείο τομής του παλμού με την ευθεία c. Επομένως:

Για α=0 και T=0.01: BW = 56

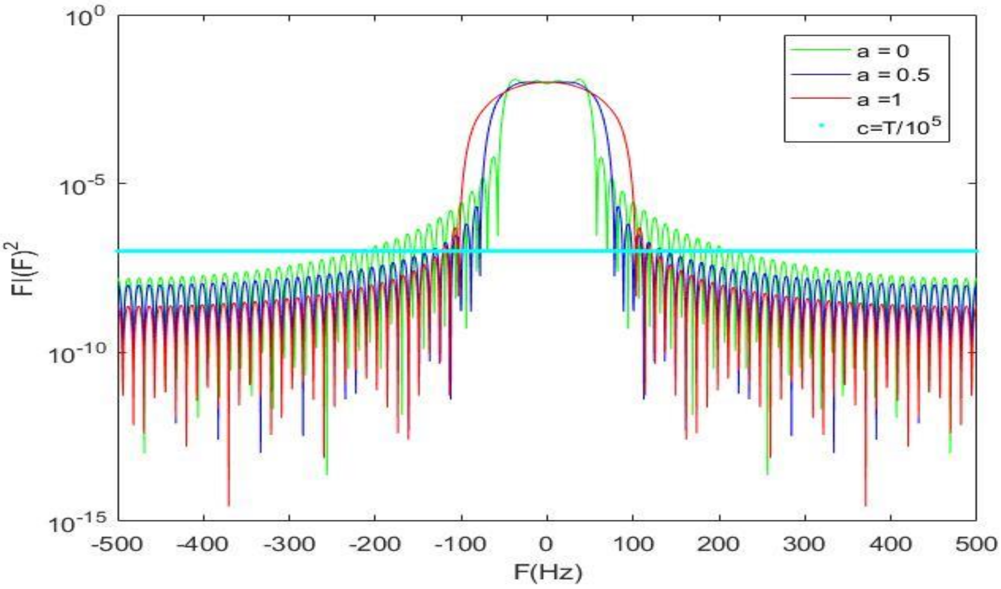
Για α=0,5 και T=0.01: BW = 75

Για α=1 και T=0.01: BW = 99



Για α=0 γίνεται αντιληπτό ότι ο παλμός έχει το μικρότερο bandwidth και είναι αποδοτικότερος από τους υπόλοιπους.

Η ίδια διαδικασία ακολουθείται για T = 10^-7 και έχουμε:



Για α=0 και T=0.01: BW = 200

Για α=0,5 και T=0.01: BW = 105

Για α=1 και T=0.01: BW = 115

Στην προκειμένη περίπτωση παρατηρείται ότι ο παλμός με α=0.5 είναι πιο αποδοτικός από τους υπολοίπους.

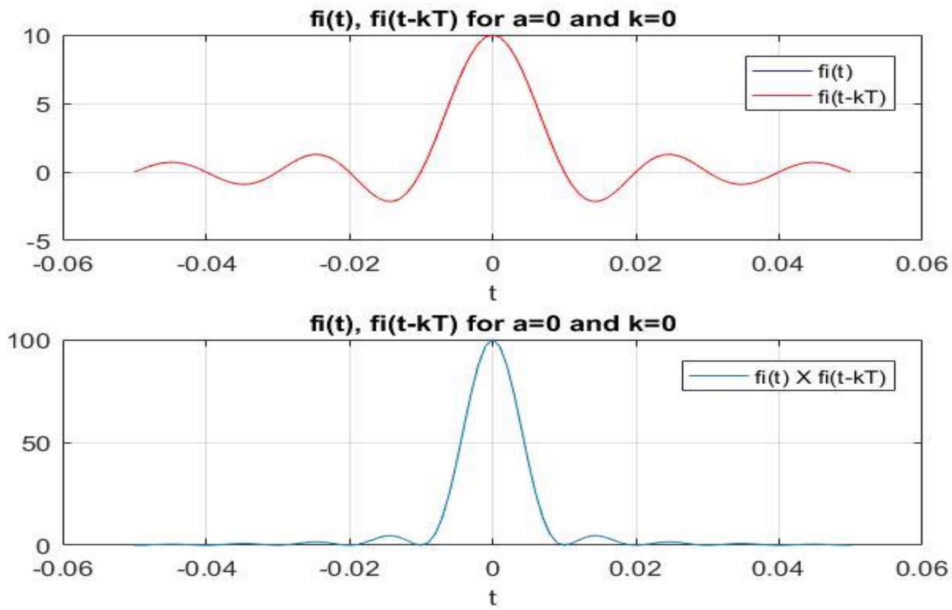
Άρα, γρήγορα συμπεραίνουμε ότι το κριτήριο για να χαρακτηριστεί ένας παλμός βέλτιστος σχετικά με το bandwidth, είναι ο ορισμός του μηδέν.

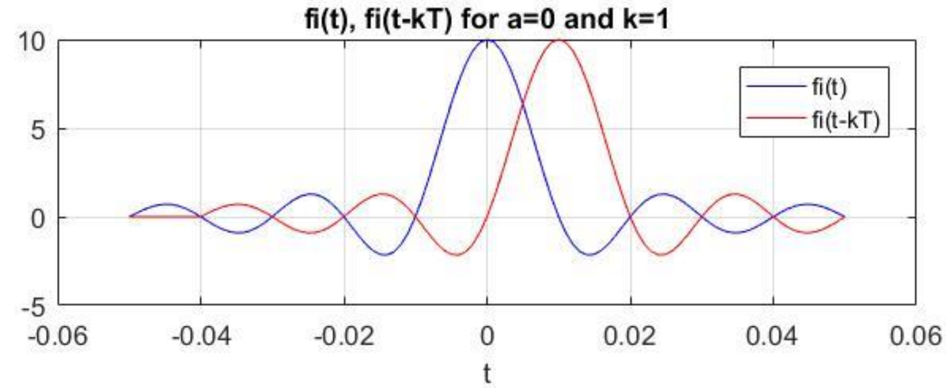
```
%bandwith based in theory
BW1=(1+a1)./(2*T);
BW2=(1+a2)./(2*T);
BW3=(1+a3)./(2*T);

%c1,c2
c1=T./(10^3);
c2=T./(10^5);


%emfanisi fasmatikis piknotias energeias(semilogy) mazi me tin c1
figure;
semilogy(F,abs(y1).^2,'g');
hold on;
semilogy(F,abs(y2).^2,'b');
hold on;
semilogy(F,abs(y3).^2,'r');
hold on;
plot(F, c1, 'k.')
hold on;
xlabel('F(Hz)');
ylabel('FI(F)^2')
legend('a = 0','a = 0.5', 'a =1', 'c=T/10^3')
```

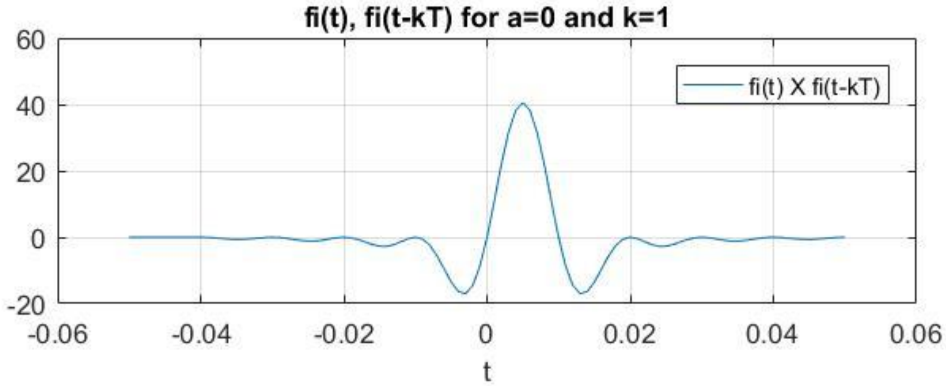
```
%emfanisi fasmatikis piknotias energeias(semilogy) mazi me tin c2
figure(5)
semilogy(F,abs(y1).^2,'g');
hold on;
semilogy(F,abs(y2).^2,'b');
hold on;
semilogy(F,abs(y3).^2,'r');
hold on;
plot(F, c2, 'c.')
hold on;
xlabel('F(Hz)');
ylabel('FI(F)^2')
legend('a = 0','a = 0.5', 'a =1','c=T/10^5')
```

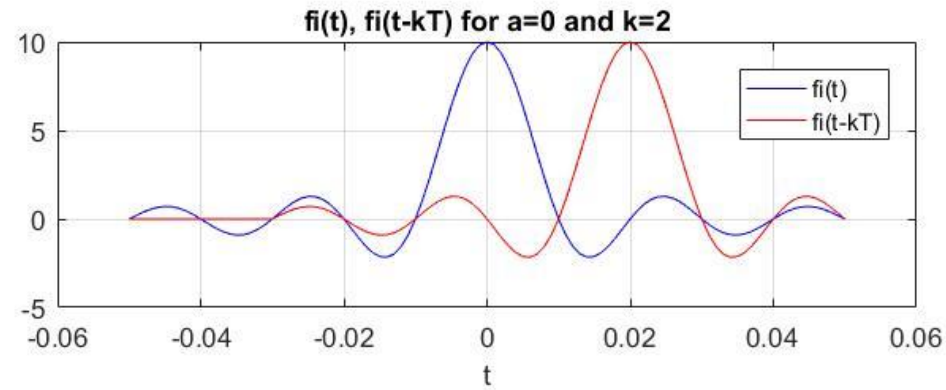
B:

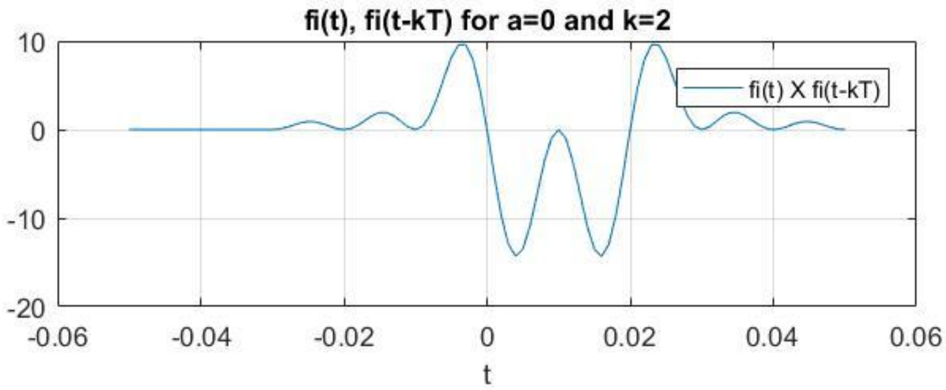
Χρησιμοποιώντας τους παλμούς του μέρους Α, κατασκευάστηκε μια ρουτίνα για κάθε παλμό. Συγκεκριμένα για κάθε επανάληψη δημιουργήθηκε το μετατοπισμένο σήμα fi(t-kT) με το k να παίρνει τιμές {0,1,2,4}. Με την μέθοδο του zero padding, έγινε η κατάλληλη μετατόπιση και εν τέλει σχεδιάστηκε τόσο τα σήματα fi(t),fi(t-kT), όσο και το γινόμενο τους.

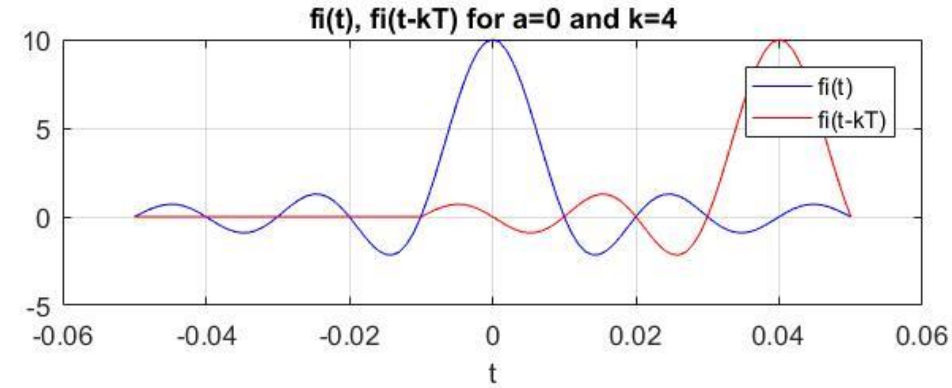


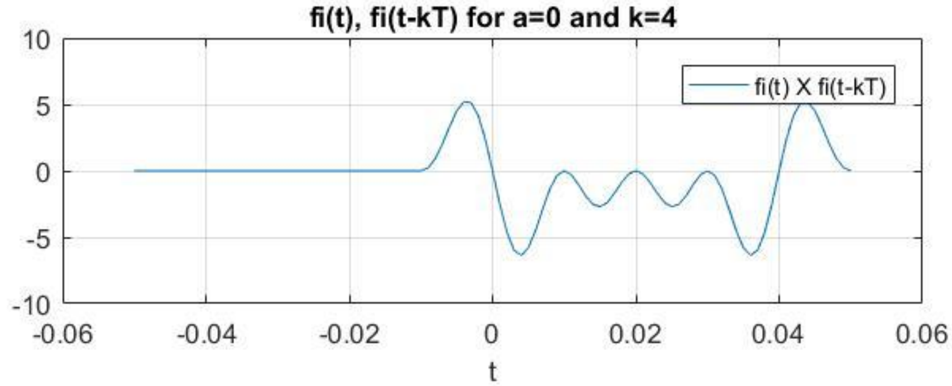

fi(t), fi(t-kT) for a=0 and k=1
fi(t)
fi(t-kT)
t

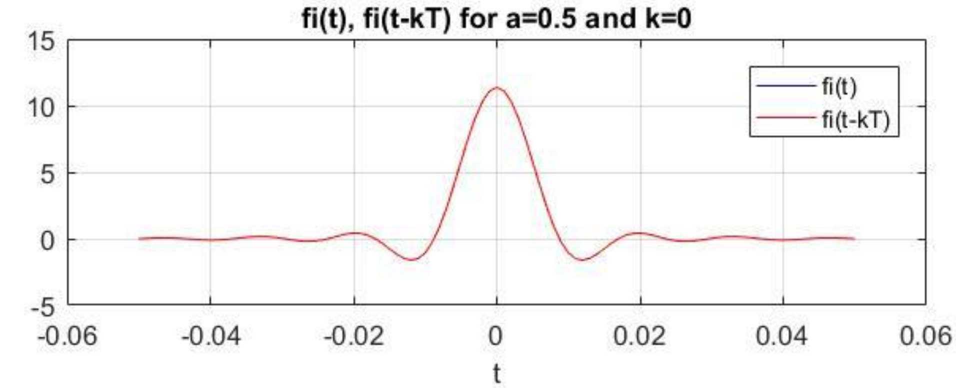

fi(t), fi(t-kT) for a=0 and k=1
fi(t) X fi(t-kT)
t

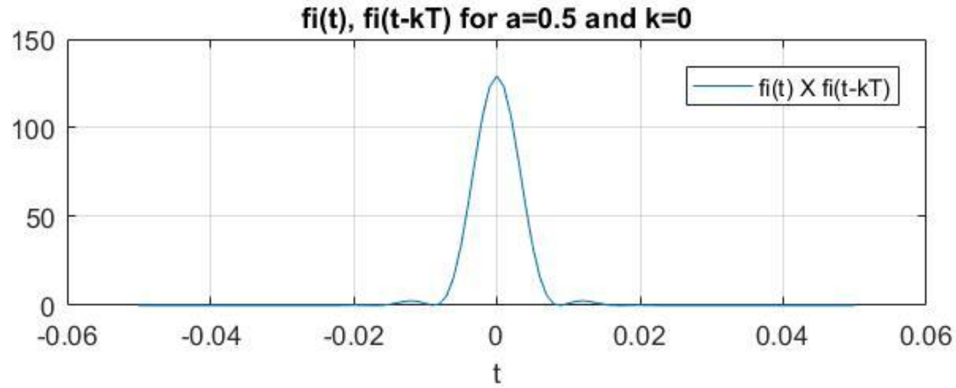

fi(t), fi(t-kT) for a=0 and k=2
fi(t)
fi(t-kT)
t

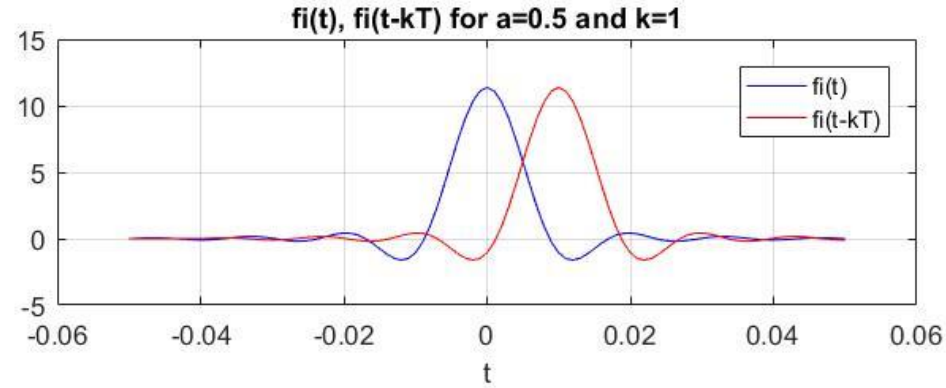

fi(t), fi(t-kT) for a=0 and k=2
fi(t) X fi(t-kT)
t

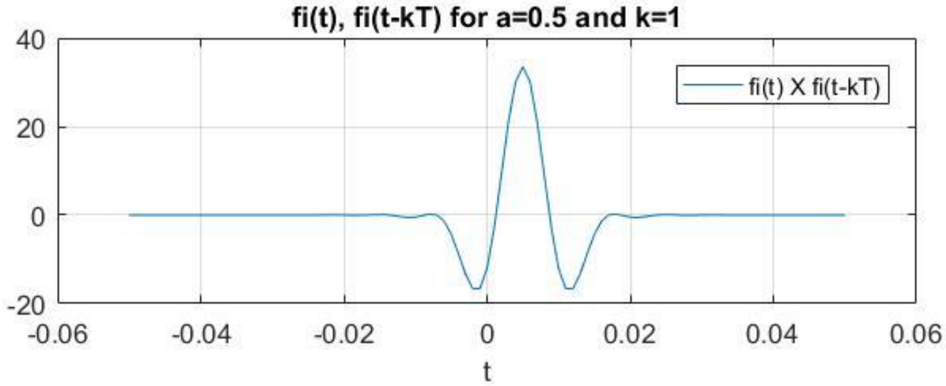

fi(t), fi(t-kT) for a=0 and k=4
fi(t)
fi(t-kT)
t

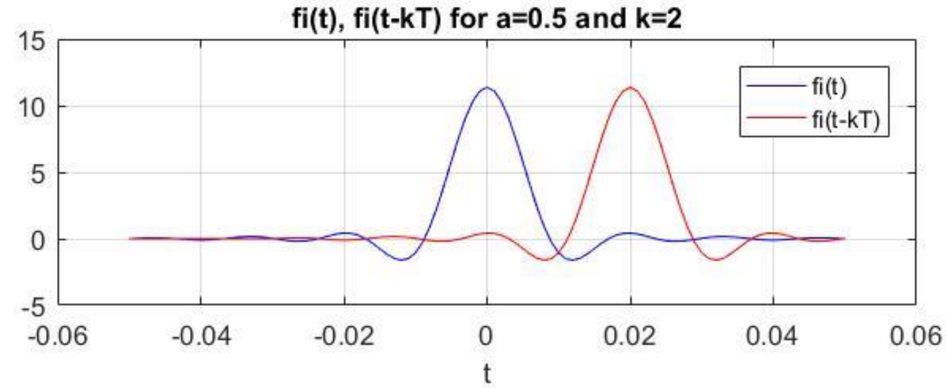

fi(t), fi(t-kT) for a=0 and k=4
fi(t) X fi(t-kT)
t

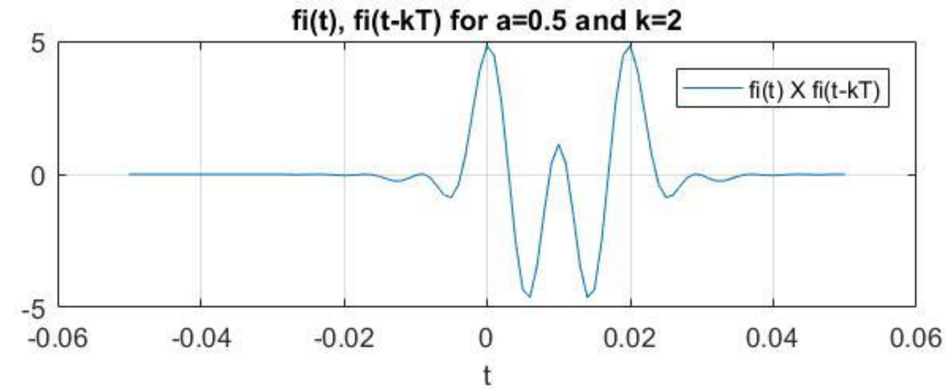

fi(t), fi(t-kT) for a=0.5 and k=0
fi(t)
fi(t-kT)
t

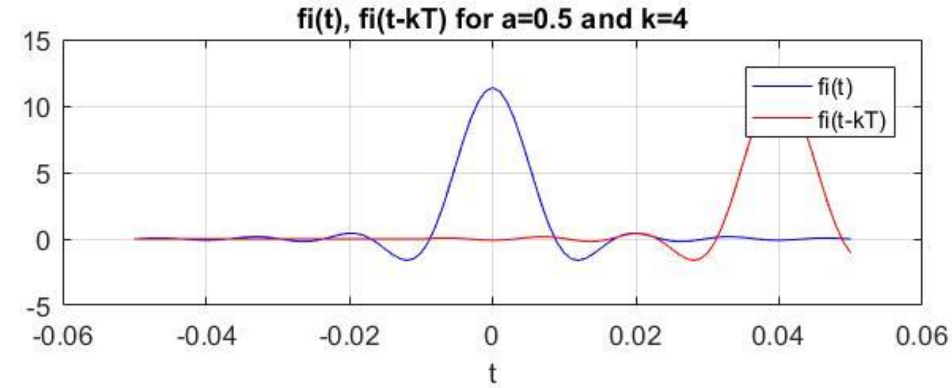

fi(t), fi(t-kT) for a=0.5 and k=0
fi(t) X fi(t-kT)
t

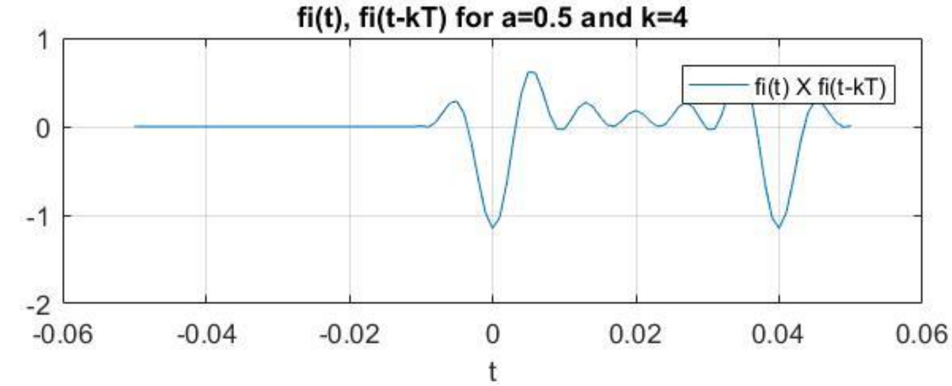

fi(t), fi(t-kT) for a=0.5 and k=1
fi(t)
fi(t-kT)
15
10
5
0
-5
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
t

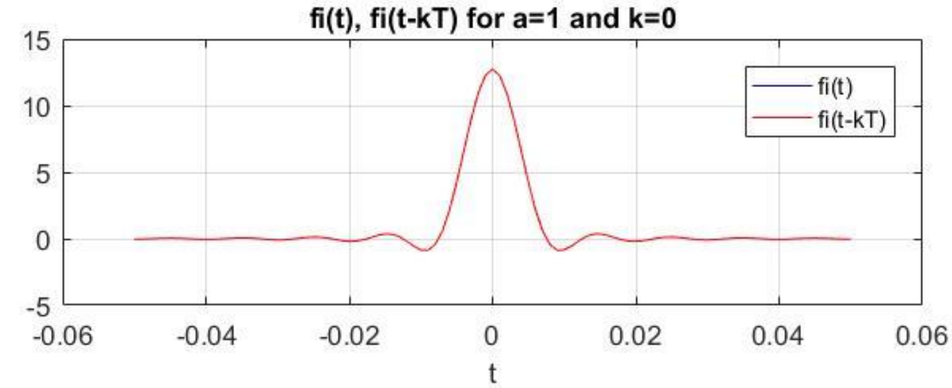

fi(t), fi(t-kT) for a=0.5 and k=1
fi(t) X fi(t-kT)
40
20
0
-20
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
t

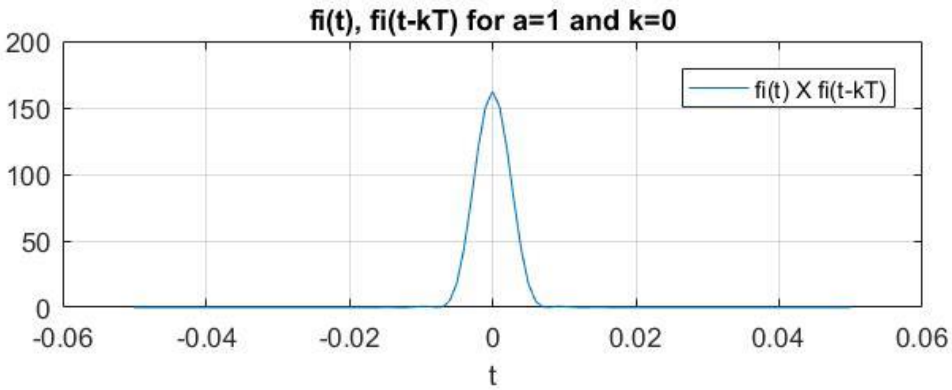

fi(t), fi(t-kT) for a=0.5 and k=2
fi(t)
fi(t-kT)
15
10
5
0
-5
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
t

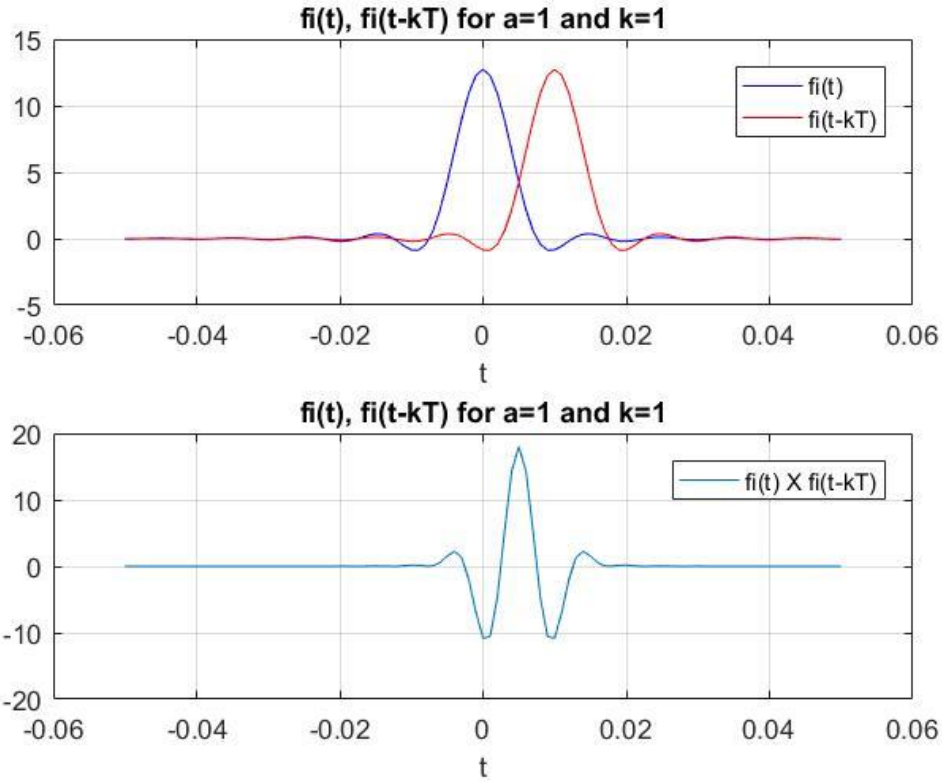

fi(t), fi(t-kT) for a=0.5 and k=2
fi(t) X fi(t-kT)
5
0
-5
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
t

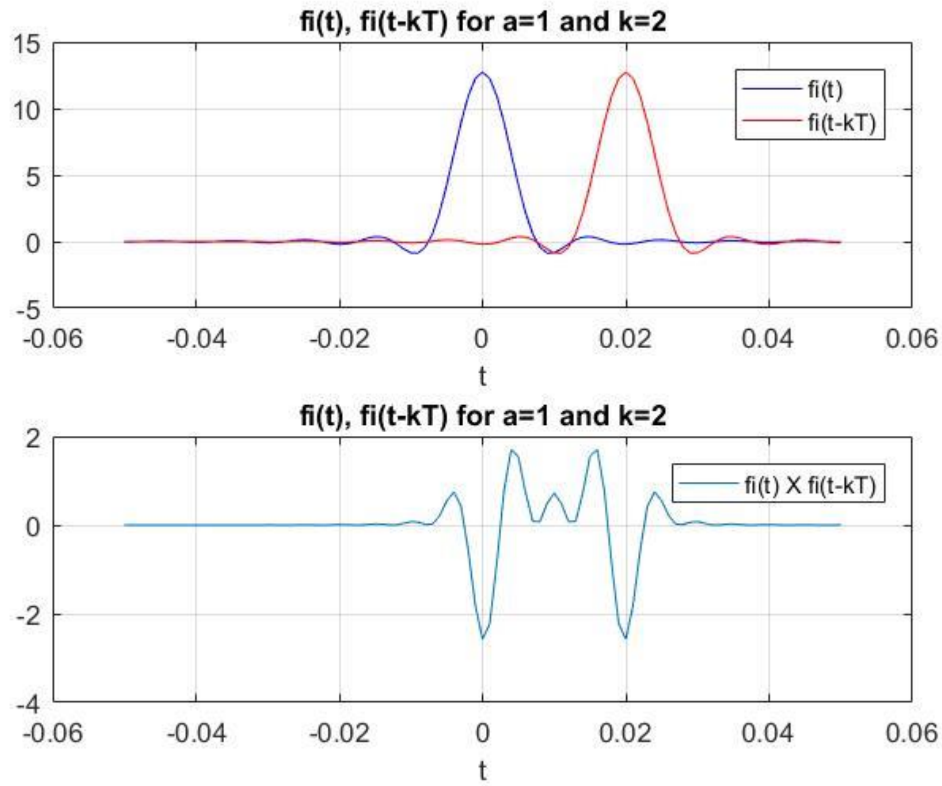

fi(t), fi(t-kT) for a=0.5 and k=4
fi(t)
fi(t-kT)
t

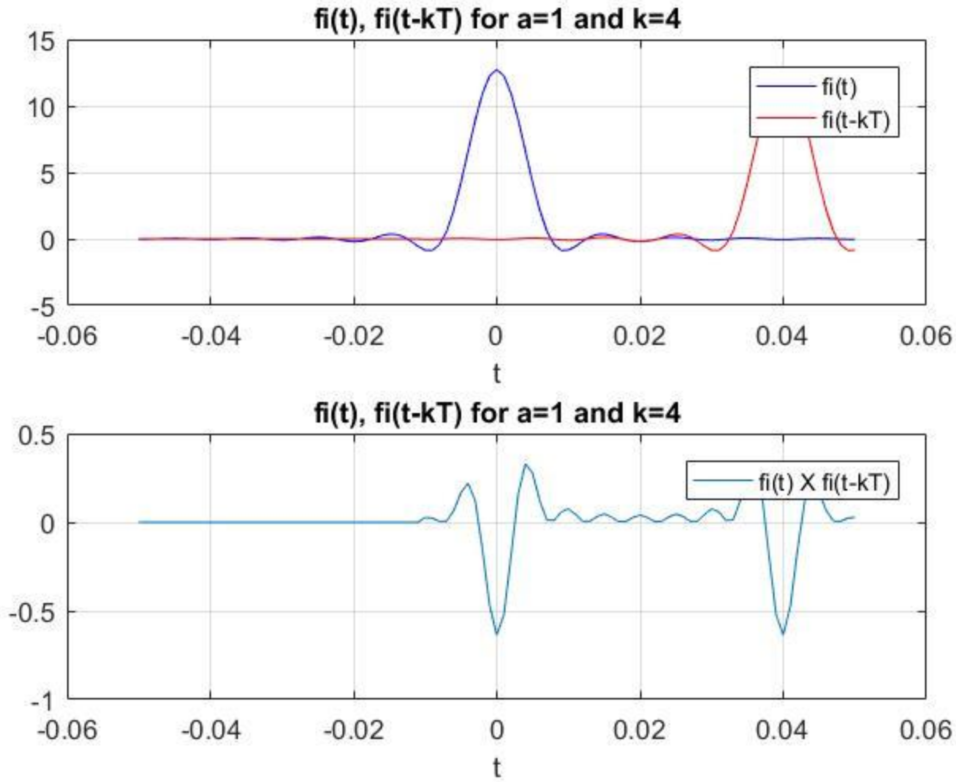

fi(t), fi(t-kT) for a=0.5 and k=4
fi(t) X fi(t-kT)
t


fi(t), fi(t-kT) for a=1 and k=0
fi(t)
fi(t-kT)
t


fi(t), fi(t-kT) for a=1 and k=0
fi(t) X fi(t-kT)
t


fi(t), fi(t-kT) for a=1 and k=1
fi(t)
fi(t-kT)
t
fi(t), fi(t-kT) for a=1 and k=1
fi(t) X fi(t-kT)
t


fi(t), fi(t-kT) for a=1 and k=2
fi(t)
fi(t-kT)
t
fi(t), fi(t-kT) for a=1 and k=2
fi(t) X fi(t-kT)
t



Υπολογίστηκαν επίσης, οι τιμές των ολοκληρωμάτων από τα γινόμενα fi(t)fi(t-kT).

Από την θεωρία ισχύει ότι:

$$\int_{-\infty}^{\infty} \varphi(\tau)\varphi(\tau-\kappa T)d\tau = \begin{Bmatrix} 1, \alpha\nu\ k = 0 \\ 0, \alpha\nu\ k \neq 0 \end{Bmatrix}$$

Επομένως:

	K=0	**K=1**	**K=2**	**K=4**
α=0	0.9798	0.0226	-0.0258	-0.0402
α=0.5	0.9999	-7.2284e-06	1.5853e-04	-9.1665e-04
α=1	1.0000	-2.2124e-05	-3.3230e-05	-1.3955e-04

Οι παλμοί, παρατηρείται ότι είναι «οριακά» ορθοκανονικοί ως προς τις μετατοπίσεις του κατά κT, καθώς για κ=0 προσεγγίζουν την μονάδα. Για οποιαδήποτε άλλη τιμή του κ, τείνουν να φτάσουν το μηδέν. Τέλος, για α=1 η προσέγγιση για κ=0 είναι ακριβέστατη, άρα όσο αυξάνεται η τιμή του α, τόσο καλύτερη προσέγγιση έχουμε.

```
%a=0
for k = setdiff(0:4,3)
    %zero padding to create the signal moved by kT
    phi_1_kt = [zeros(1,(1/Ts)*k*T) phi_1(1:end-(1/Ts)*k*T)];
    z1 = phi_1.*phi_1_kt; %ginomeno

    figure;
    subplot(2,1,1);
    plot(t1,phi_1,'b');
    hold on;
    plot(t1,phi_1_kt,'r');
    grid on;
    xlabel('t');

    if(k==0)
        title('fi(t), fi(t-kT) for a=0 and k=0')
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=0 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=0 and k=2')
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=0 and k=4')
    end;
    legend('fi(t)', 'fi(t-kT)')

    subplot(2,1,2);
    plot(t1,z1); %ginomeno
    xlabel('t');

    if(k==0)
        title('fi(t), fi(t-kT) for a=0 and k=0')
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=0 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=0 and k=2')
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=0 and k=4')
    end;
    legend('fi(t) X fi(t-kT)');

    integral_1 = sum(z1)*Ts %ypologismos oloklirwmatos
    grid on

end;
```

Για τις υπόλοιπες τιμές του α είναι ίδιος κώδικας απλά αλλάζουν οι τιμές σε 0.5 και 1. Επίσης, το A = 5 σε αντίθεση με το ερώτημα α της άσκησης όπου ζητήθηκε A=4.

C1:

Στο 3ο και τελευταίο κομμάτι της εργασίας, ζητήθηκε να προσομοιωθεί ένα 2-PAM σύστημα βασικής ζώνης, το οποίο μεταφέρει N bits με διαμόρφωση προφανώς 2-PAM. Αρχικά, δημιουργήθηκαν 100 bits με την χρήση της εντολής b=(sign(randn(N,1))+1)/2. Η συνάρτηση του matlab randn επιστρέφει ένα πίνακα διαστάσεων N*1(στην περίπτωση μας N=100), γεμάτο τυχαίους αριθμούς, οι οποίοι ανήκουν στο σύνολο των πραγματικών αριθμών. Η συνάρτηση sign με την σειρά της, δέχεται σαν όρισμα αυτούς τους αριθμούς και ανάλογα το πρόσημο τους, τους μετατρέπει σε -1 ή +1 ανάλογα αν είναι αρνητικός ή θετικός αριθμός αντίστοιχα. Επειδή σκοπός μας ήταν να μπορέσουμε να καλύψουμε κάθε περίπτωση, προσθέσαμε επίσης την μονάδα για να καλύψουμε την περίπτωση του μηδέν(το -1 αντιστοιχεί σε 0).

```
%c1
a = 0.5;
A = 5;
T = 0.1;
over = 10;
Ts = T/over;
N = 100;

%creation of the N=50 random bits
b = (sign(randn(N, 1)) + 1)/2;
```

C2:

Για το τελευταίο μέρος, υλοποιήθηκε η συνάρτηση bits_to_2PAM, η οποία δεχόταν σαν είσοδο την ακόλουθια bits του ερωτήματος c1 και έκανε mapping με τον ακόλουθο κανόνα:

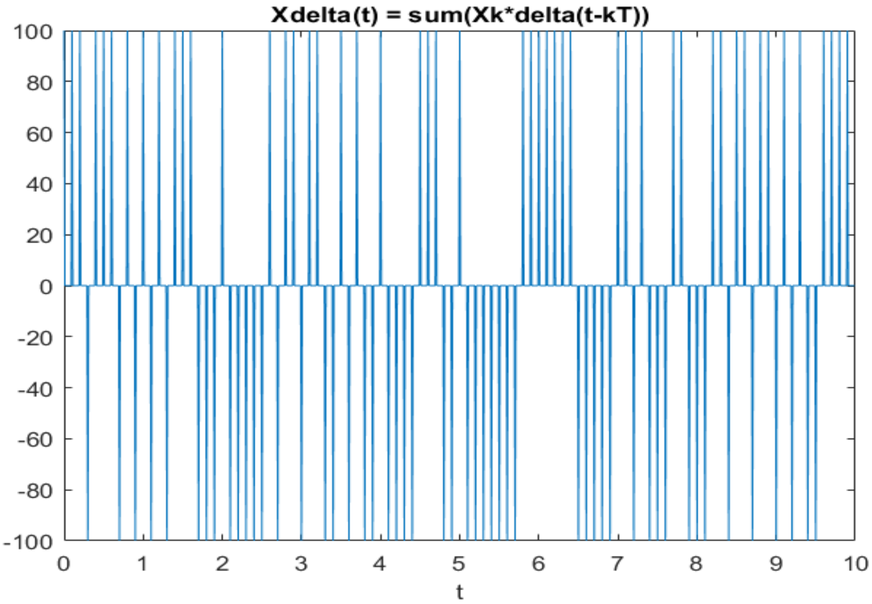
0 → +1

1 → -1

Με την χρήση μιας λούπας λοιπόν και μιας εντολής ελέγχου έγινε η κατάλληλη μετατροπή, έτσι ώστε να είμαστε πιστή στον κανόνα της εκφώνησης.

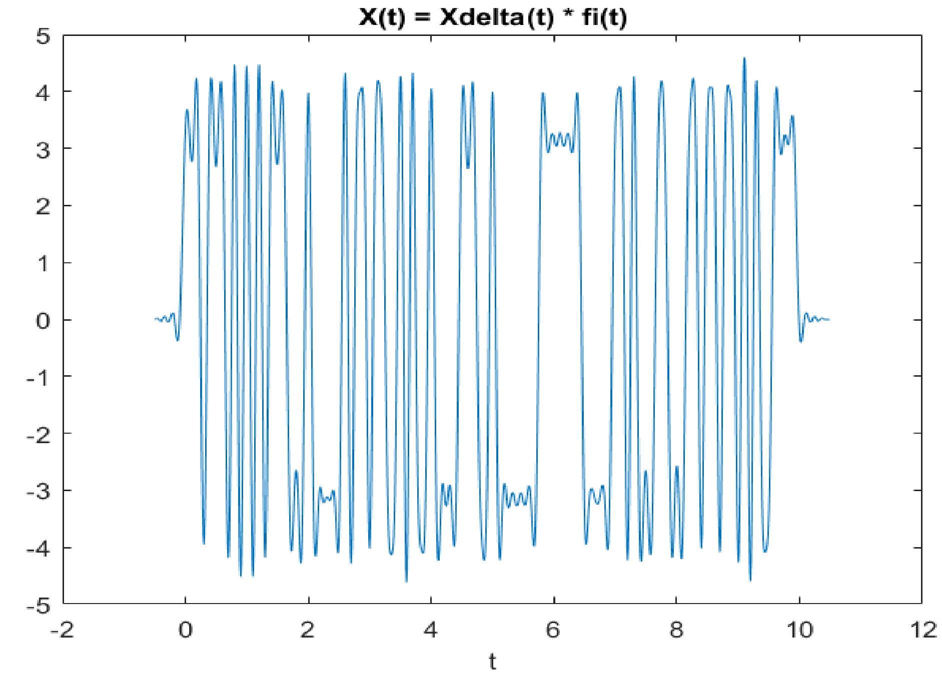
```
%function bits_to_2PAM
% Project Name: Thl_1
% Engineer: Christos Trimas, Alexandros Michael


function X = bits_to_2PAM(b)

%creating vector of zeros
    X=zeros(size(b));

    %matching 0 --> +1, 1 --> -1
    for k = 1:size(b)
        if(b(k)==0)
        X(k) = 1;

        elseif(b(k)==1)
        X(k) = -1;
        end;
    end;
end
```

Στη συνέχεια, έγινε προσομοίωση του σήματος: $X_\delta(t) = \sum_{k=0}^{N-1} X_k \delta(t-kT)$,με χρήση της εντολής X_delta= 1/T_s *upsample(X,over). Ο ρόλος της upsample είναι να δέχεται σαν όρισμα τα σύμβολα της 2-PAM (τα X δηλαδή) και το over και να επιστρέφει ένα stream από over-1 μηδενικά μεταξύ των συμβόλων X. Ο άξονας του χρόνου για το παραπάνω σήμα ορίστηκε από το μηδέν έως το (N+N*(over-1))-1)*Ts, με N*(over-1) να είναι τα μηδενικά ανάμεσα στα σύμβολα.



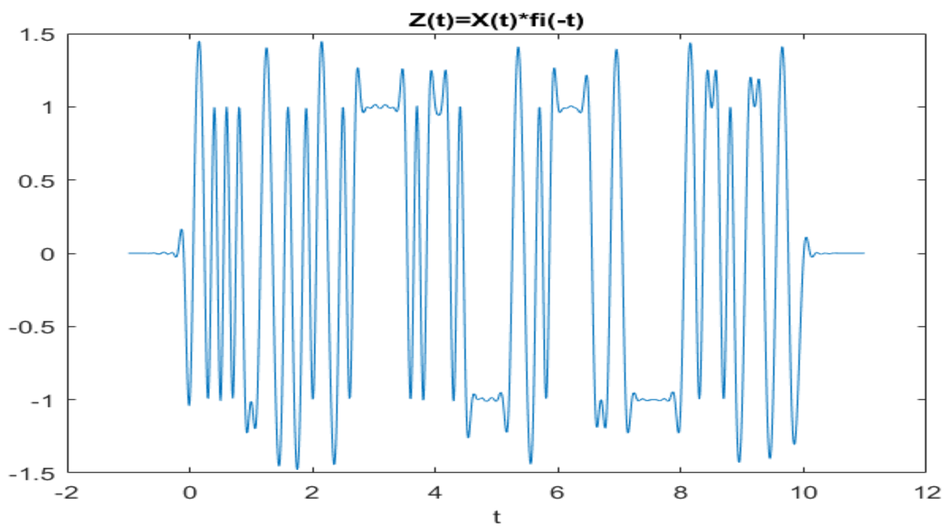

```
%b
%simulation of the signal X{delta}(t)
X_delta = 1/Ts*upsample(X, over);

%time
t = 0:Ts:((N+N*(over-1))-1)*Ts; %exoume prosthiki over-1 midenikwn meta3i simvolwn

figure;
plot(t,X_delta);
xlabel('t');
title('X{delta}(t) = sum(Xk*delta(t-kT))');
```

Έπειτα, δημιουργήθηκε ένας παλμός srrc και με την χρήση της συνάρτησης του matlab conv, προσομοιώθηκε η συνέλιξη του παλμού, με το προηγούμενο σήμα που κατασκευάστηκε. Από τα σήματα και συστήματα, και από την ψηφιακή επεξεργασία σήματος, ξέρουμε ότι για να σχεδιαστεί επιτυχώς η συνέλιξη χρειάζεται να ορίσουμε τον χρόνο από το άθροισμα των ελάχιστων τιμών του χρόνου στα οποία ορίζονται τα σήματα που θέλουμε να συνελίξουμε, μέχρι το άθροισμα των μέγιστων τιμών τους.



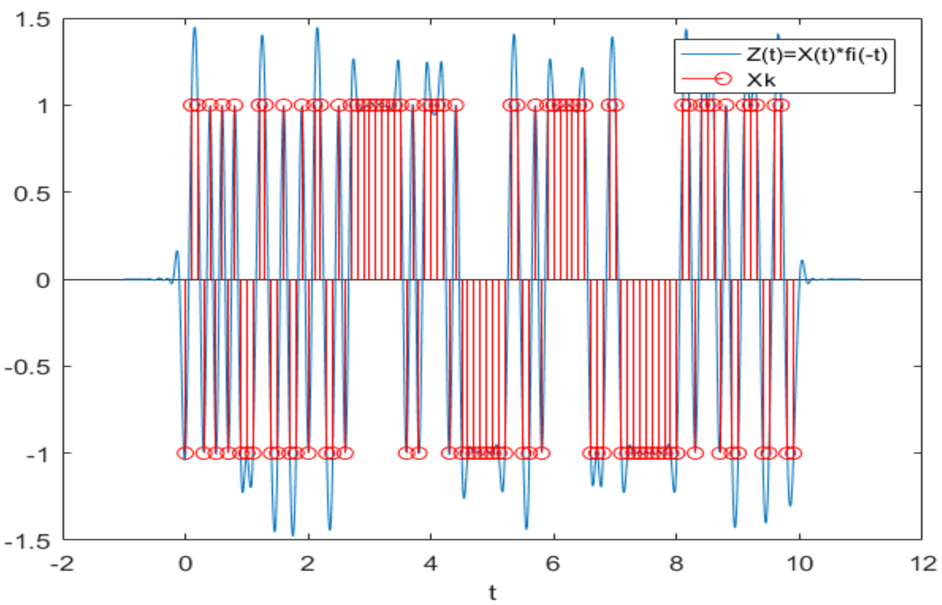

```
%creation of fi(t) pulse
[phi_1,t1] = srrc_pulse(T, Ts, A, a);

%time of convolution
tx = t(1)+t1(1) :Ts: t(end)+t1(end);

%conv function for the convolution between fi(t) and X_delta
X_conv = conv(X_delta,phi_1)*Ts;

figure;
plot(tx,X_conv);
xlabel('t');
title('X(t) = X{delta}(t) * fi(t)')
```

Έγινε επίσης η συνέλιξη Z(t) = X(t)*fi(-t). Για την δημιουργία του fi(-t) χρησιμοποιήθηκε ο παλμός αλλά ανακλασμένος και η ίδια τεχνική έγινε και για τον χρόνο, έτσι ώστε να σχεδιαστεί σωστά η συνέλιξη.



```
%creation of fi(-t) pulse
phi_2 = phi_1(end:-1:1);
t2 = -t1(end:-1:1);

%convolution of fi(-t) and X_conv
z = conv(X_conv,phi_2)*Ts;

%time of the convolution
tz= tx(1)+t2(1) :Ts: tx(end)+t2(end);

figure;
plot(tz,z);
xlabel('t');
title('Z(t)=X(t)*fi(-t)')
```

Τέλος, συγκρίναμε γραφικά την συνάρτηση Z(kT) με τις τιμές Xκ, και επιλέχτηκε hold on στην απεικόνιση του Z(t) εκτελώντας την εντολή stem([0:N-1]*T,X).



Με την συνέλιξη των σημάτων, αυτό που επιτεύχθηκε ήταν η διαμόρφωση PAM ή αλλιώς η μετατροπή του stream σε αναλογικό σήμα. Έπειτα φιλτράραμε με χρήση προσαρμοσμένου φίλτρου(το οποίο με βάση την θεωρία των Τηλ2 θεωρείται ιδανικό) και προέκυψε το σήμα Z(t). Με δειγματοληψία ανακτούμε πίσω το αρχικό stream πληροφορίας και αυτό γίνεται αντιληπτό από την παραπάνω απεικόνιση του σήματος Z(t) και του vector X, όπου οι τιμές του Xκ ταυτίζονται με αυτές των Z(κT).

Με λίγα λόγια κάναμε τόσο διαμόρφωση, όσο και αποδιαμόρφωση ενός σήματος με μερικές πολύ απλές τεχνικές, σε ένα ιδανικό κανάλι.

```
figure;
plot(tz,z);
hold on;
%plot in the same figure Z and X
stem([0:N-1]*T,X,'r');
xlabel('t');
legend('Z(t)=X(t)*fi(-t)','X{k}')
```

Κώδικας:

```
clc;
close all;
clear all;
%a1

T = 10^(-2); %period
over = 10;
Ts = T/over; %periodos deigmatolipsias
A = 4;
a1 = 0;   %green pulse
a2 = 0.5; %blue pulse
a3 = 1;   %red pulse


%creation of srrc pulses
[phi1,t1] = srrc_pulse(T, Ts, A, a1);
[phi2,t2] = srrc_pulse(T, Ts, A, a2);
[phi3,t3] = srrc_pulse(T, Ts, A, a3);

%plots
figure;
plot(t1,phi1,'g')
hold on
plot(t2,phi2,'b')
hold on
plot(t3,phi3,'r')
xlabel('t')
ylabel('fi(t)')
title('3 different srrc pulses')
legend('a = 0','a = 0.5', 'a =1')
grid on

%a2
Fs = 1/Ts;
Nf = 2048;                          %samples
F = [-Fs/2:Fs/Nf:Fs/2-Fs/Nf]; %frequency

y1 = fftshift(fft(phi1,Nf)*Ts); %fourier transformation
y2 = fftshift(fft(phi2,Nf)*Ts);
y3 = fftshift(fft(phi3,Nf)*Ts);

figure;
plot(F,abs(y1).^2,'g')
hold on
plot(F,abs(y2).^2,'b')
hold on
plot(F,abs(y3).^2,'r')
xlabel('F');
ylabel('|FI(F)|^2')
title('Fasmatikh Piknothta Isxios')
legend('a = 0','a = 0.5', 'a =1')

figure;
semilogy(F,abs(y1).^2,'g');
hold on;
semilogy(F,abs(y2).^2,'b');
hold on;
semilogy(F,abs(y3).^2,'r');
xlabel('F');
ylabel('|FI(F)|^2')
title('Logarithmimenh Fasmatikh Piknothta Isxuos')
legend('a = 0','a = 0.5', 'a =1')

%a3

%bandwith based in theory
BW1 = (1+a1)./(2*T)
BW2 = (1+a2)./(2*T)
BW3 = (1+a3)./(2*T)

%c1,c2
c1 = T./(10^3);
c2 = T./(10^5);

%emfanisi fasmatikis piknotias energeias(semilogy) mazi me tin c1
```

```
figure;
semilogy(F,abs(y1).^2,'g');
hold on;
semilogy(F,abs(y2).^2,'b');
hold on;
semilogy(F,abs(y3).^2,'r');
hold on;
plot(F, c1, 'k.')
hold on;
xlabel('F(Hz)');
ylabel('FI(F)^2')
legend('a = 0','a = 0.5', 'a =1', 'c=T/10^3')

%emfanisi fasmatikis piknotias energeias(semilogy) mazi me tin c2
figure;
semilogy(F,abs(y1).^2,'g');
hold on;
semilogy(F,abs(y2).^2,'b');
hold on;
semilogy(F,abs(y3).^2,'r');
hold on;
plot(F, c2, 'c.')
hold on;
xlabel('F(Hz)');
ylabel('FI(F)^2')
legend('a = 0','a = 0.5', 'a =1','c=T/10^5')

%b
clc;
clear all;
close all;

T = 10^(-2);
over = 10;
Ts = T/over;
A = 5;
a1= 0;
a2 = 0.5;
a3 = 1;


%dimiourgia palmwn srrc
[phi_1,t1] = srrc_pulse(T, Ts, A, a1);
[phi_2,t2] = srrc_pulse(T, Ts, A, a2);
[phi_3,t3] = srrc_pulse(T, Ts, A, a3);

%a=0
for k = setdiff(0:4,3)
    %zero padding to create the signal moved by kT
    phi_1_kt = [zeros(1,(1/Ts)*k*T) phi_1(1:end-(1/Ts)*k*T)];
    z1 = phi_1.*phi_1_kt; %ginomeno

    figure;
    subplot(2,1,1);
    plot(t1,phi_1,'b');
    hold on;
    plot(t1,phi_1_kt,'r');
    grid on;
    xlabel('t');

    if(k==0)
        title('fi(t), fi(t-kT) for a=0 and k=0')
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=0 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=0 and k=2')
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=0 and k=4')
    end;
    legend('fi(t)', 'fi(t-kT)')

    subplot(2,1,2);
    plot(t1,z1); %ginomeno
    xlabel('t');

    if(k==0)
        title('fi(t), fi(t-kT) for a=0 and k=0')
```

```
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=0 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=0 and k=2')
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=0 and k=4')
    end;
    legend('fi(t) X fi(t-kT)');

    integral_1 = sum(z1)*Ts %ypologismos oloklirwmatos
    grid on

end;

%a=0,5
for k = setdiff(0:4,3)
    phi_2_kt = [zeros(1,(1/Ts)*k*T) phi_2(1:end-(1/Ts)*k*T)];
    z2 = phi_2.*phi_2_kt;

    figure;
    subplot(2,1,1)
    plot(t2,phi_2,'b')
    hold on;
    plot(t2,phi_2_kt,'r')
    grid on
    xlabel('t')

    if(k==0)
        title('fi(t), fi(t-kT) for a=0.5 and k=0')
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=0.5 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=0.5 and k=2')
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=0.5 and k=4')
    end;
    legend('fi(t)', 'fi(t-kT)')

    subplot(2,1,2)
    plot(t2,z2)
    xlabel('t')

    if(k==0)
        title('fi(t), fi(t-kT) for a=0.5 and k=0')
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=0.5 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=0.5 and k=2')
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=0.5 and k=4')
    end;
    legend('fi(t) X fi(t-kT)')

    integral_2 = sum(z2)*Ts
    grid on

end;

%a=1
for k = setdiff(0:4,3)
    phi_3_kt = [zeros(1,(1/Ts)*k*T) phi_3(1:end-(1/Ts)*k*T)];
    z3 = phi_3.*phi_3_kt;

    figure;
    subplot(2,1,1);
    plot(t3,phi_3,'b');
    hold on;
    plot(t3,phi_3_kt,'r');
    grid on;
    xlabel('t');

    if(k==0)
        title('fi(t), fi(t-kT) for a=1 and k=0')
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=1 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=1 and k=2')
```

```
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=1 and k=4')
    end;
    legend('fi(t)', 'fi(t-kT)')

    subplot(2,1,2);
    plot(t3,z3);
    xlabel('t');

    if(k==0)
        title('fi(t), fi(t-kT) for a=1 and k=0')
    elseif(k==1)
        title('fi(t), fi(t-kT) for a=1 and k=1')
    elseif(k==2)
        title('fi(t), fi(t-kT) for a=1 and k=2')
    elseif(k==4)
        title('fi(t), fi(t-kT) for a=1 and k=4')
    end;
    legend('fi(t) X fi(t-kT)')

    integral_3 = sum(z3)*Ts
    grid on;

end;


%c
clc;
close all;
clear all;

%c1
a = 0.5;
A = 5;
T = 0.1;
over = 10;
Ts = T/over;
N = 100;

%creation of the N=50 random bits
b = (sign(randn(N, 1)) + 1)/2;

%c2
%a
%transforming the N bits in to 2-pam
X = bits_to_2PAM(b);

%b
%simulation of the signal X{delta}(t)
X_delta = 1/Ts*upsample(X, over);

%time
t = 0:Ts:((N+N*(over-1))-1)*Ts; %exoume prosthiki over-1 midenikwn meta3i simvolwn

figure;
plot(t,X_delta);
xlabel('t');
title('X{delta}(t) = sum(Xk*delta(t-kT))');

%creation of fi(t) pulse
[phi_1,t1] = srrc_pulse(T, Ts, A, a);

%time of convolution
tx = t(1)+t1(1) :Ts: t(end)+t1(end);

%conv function for the convolution between fi(t) and X_delta
X_conv = conv(X_delta,phi_1)*Ts;

figure;
plot(tx,X_conv);
xlabel('t');
title('X(t) = X{delta}(t) * fi(t)')

%creation of fi(-t) pulse
phi_2 = phi_1(end:-1:1);
t2 = -t1(end:-1:1);
```

```
%convolution of fi(-t) and X_conv
z = conv(X_conv,phi_2)*Ts;

%time of the convolution
tz= tx(1)+t2(1) :Ts: tx(end)+t2(end);

figure;
plot(tz,z);
xlabel('t');
title('Z(t)=X(t)*fi(-t)')

figure;
plot(tz,z);
hold on;
%plot in the same figure Z and X
stem([0:N-1]*T,X,'r');
xlabel('t');
legend('Z(t)=X(t)*fi(-t)','X{k}')

function [phi, t] = srrc_pulse(T, Ts, A, a)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% phi = srrc_pulse(T, Ts, A, a)                                                %
% OUTPUT                                                                       %
%      phi: truncated SRRC pulse, with parameter T,                            %
%                 roll-off factor a, and duration 2*A*T                        %
%      t:   time axis of the truncated pulse                                   %
% INPUT                                                                        %
%      T:  Nyquist parameter or symbol period  (positive real number)          %
%      Ts: sampling period  (Ts=T/over)                                        %
%                where over is a positive INTEGER called oversampling factor   %
%      A:  half duration of the pulse in symbol periods (positive INTEGER)     %
%      a:  roll-off factor (real number between 0 and 1)                       %
%                                                                              %
%    A. P. Liavas, Nov. 2011                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t = [-A*T:Ts:A*T] + 10^(-8); % in order to avoid division by zero problems at t=0.

if (a>0 && a<=1)
   num = cos((1+a)*pi*t/T) + sin((1-a)*pi*t/T) ./ (4*a*t/T);
   denom = 1-(4*a*t./T).^2;
   phi = 4*a/(pi*sqrt(T)) * num ./ denom;
elseif (a==0)
   phi = 1/(sqrt(T)) * sin(pi*t/T)./(pi*t/T);
else
    phi = zeros(length(t),1);
    disp('Illegal value of roll-off factor')
    return
end

%function bits_to_2PAM
% Project Name: Thl_1
% Engineer: Christos Trimas, Alexandros Michael


function X = bits_to_2PAM(b)

%creating vector of zeros
    X=zeros(size(b));

    %matching 0 --> +1, 1 --> -1
    for k = 1:size(b)
        if(b(k)==0)
        X(k) = 1;

        elseif(b(k)==1)
        X(k) = -1;
        end;
    end;
end
```