

TECHNICAL UNIVERSITY OF CRETE

ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ Ι

Άσκηση 1

Author:

Σπυριδάκης Χρήστος

AM: 2014030022

October 31, 2019



**ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ**

1 Εισαγωγή

Για την διευκόλυνση της υλοποίησης είναι καλό να σημειωθεί ότι το κάθε μέρος (A, B και C) υλοποιήθηκε σε διαφορετικό script, αυτό είχε τα θετικά του, στο να είναι περισσότερο διακριτός ο κώδικας για ανάπτυξη και αποσφαλμάτωση, αλλά χρειάστηκε να ξανά δημιουργηθούν σήματα που είχαν δημιουργηθεί σε προηγούμενα ερωτήματα. Δεν επηρεάζονται κάπως τα αποτελέσματα απλά είναι μία διευκρίνιση σχετικά με την δομή που δόθηκε στον κώδικα.

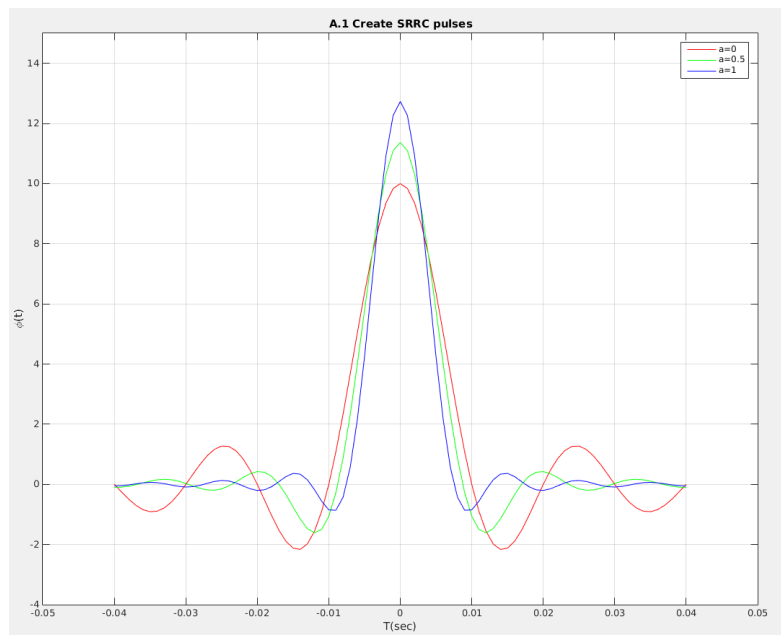
Επίσης τα πέντε σημαντικά αρχεία που υπάρχουν είναι το *srrc_pulses.m* το οποίο είναι αυτό που δόθηκε ως βοηθητικό αρχείο για την δημιουργία των αποκομμένων SRRC παλμών. Τα *part_a.m*, *part_b.m* και *part_c.m* τα οποία αναφέρονται για κάθε θέμα ξεχωριστά καθώς επίσης υπάρχει και το *bits_to_2PAM.m* το οποίο εξηγείται περισσότερο στο θέμα C.

Τέλος, αν λόγω της εκτύπωσης σε χαρτί δεν είναι εμφανές σε ικανοποιητικό βαθμό κάποιο από τα figures μπορούν να βρεθούν όλα τα μέρη του project στο παρακάτω repository όπου υπάρχουν και screenshot αυτών που φαίνονται πιο λεπτομερειακά: <https://github.com/CSpyridakis/CommSys>

2 Ερώτημα A

A.1 Δημιουργία των SRRC $\phi(t)$ παλμών

Αρχικά χρειάστηκε να χρησιμοποιήσουμε το script που μας δόθηκε - *srrc_pulses.m* - προκειμένου να δημιουργήσουμε αποκομμένους παλμούς Square Root Raised Cosine $\phi(t)$. Είσοδος της συνάρτησης είναι η περίοδος συμβόλων T , η περίοδος δειγματοληψίας T_s , ο θετικός αριθμός A και το roll-off factor a . Όπου η περίοδος δειγματοληψίας υπολογίζεται ως το πηλίκο $\frac{T}{over}$, με $over$ ως το συντελεστή υπερδειγματοληψίας. Στην περίπτωση μας για τα δεδομένα της εκφώνησης είχαμε $T = 10^{-2}$ sec, $over = 10$, $A = 4$ και $a = 0, 0.5, 1$. Συνεπώς δημιουργήθηκαν οι παρακάτω παλμοί:

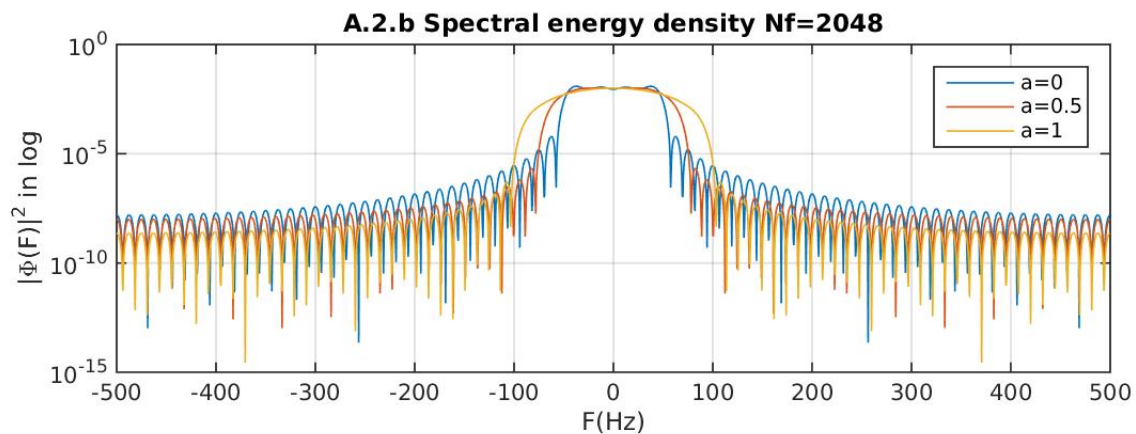
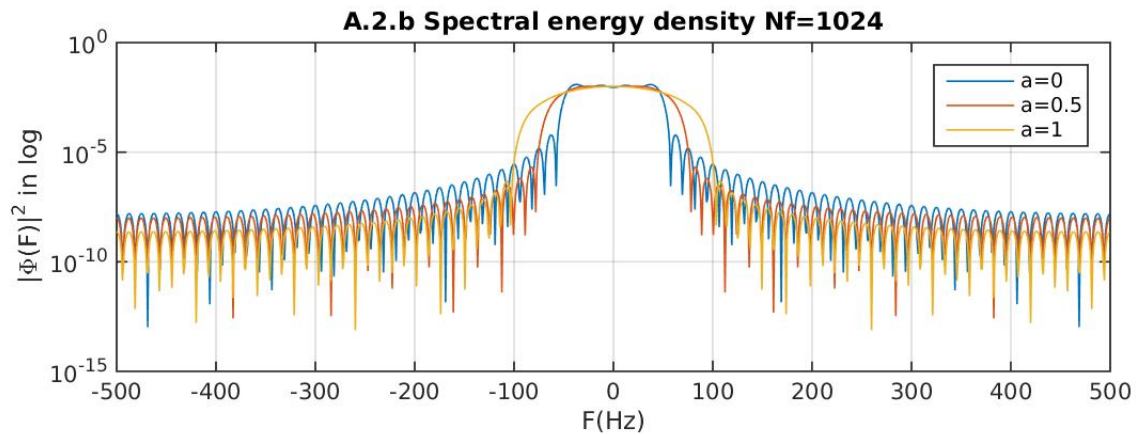
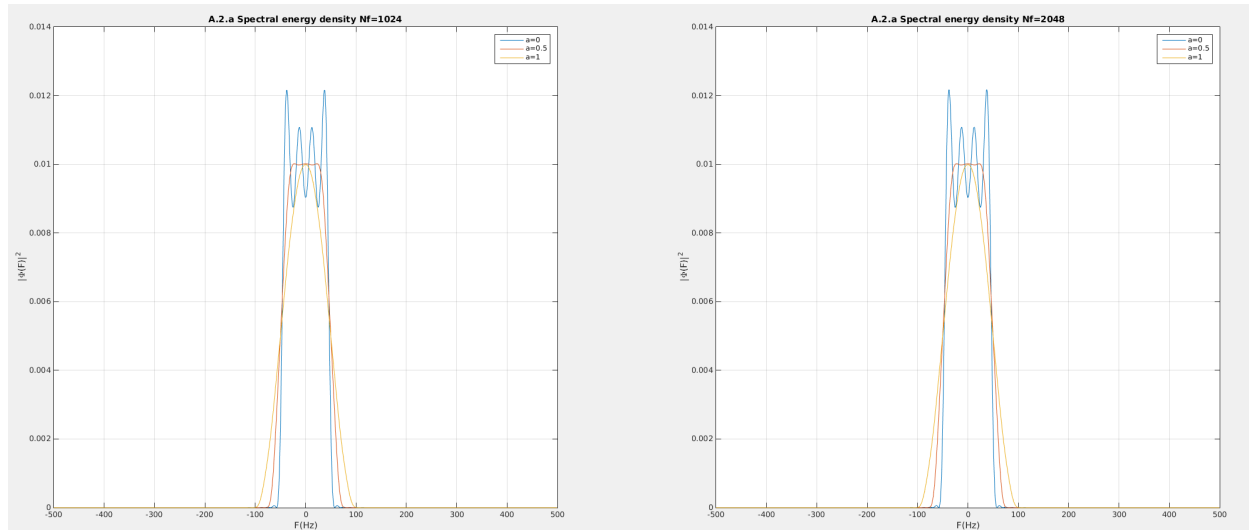


Αυτά που μπορούμε να παρατηρήσουμε και από τα διαγράμματα σχετικά με το ρυθμό “μείωσης” του πλάτους των παλμών είναι τα εξής. Πρώτη παρατήρηση είναι ότι για κάθε τιμή του roll-off γίνεται φθίνουσα ταλάντωση της οποία βέβαια τα χαρακτηριστικά εξαρτώνται από την τιμή του a . Επίσης, όλοι οι παλμοί έχουν την ίδια περίοδο. Ακόμα, βλέπουμε ότι για μεγαλύτερο a έχουμε και μεγαλύτερο αρχικό πλάτος, ενώ ταυτόχρονα όσο μεγαλύτερο είναι το a τόσο μεγαλύτερος είναι και ο ρυθμός απόσβεσης όσο αυξάνεται η απόλυτη τιμή του χρόνου.

A.2 Fourier Transform of SRRC pulses

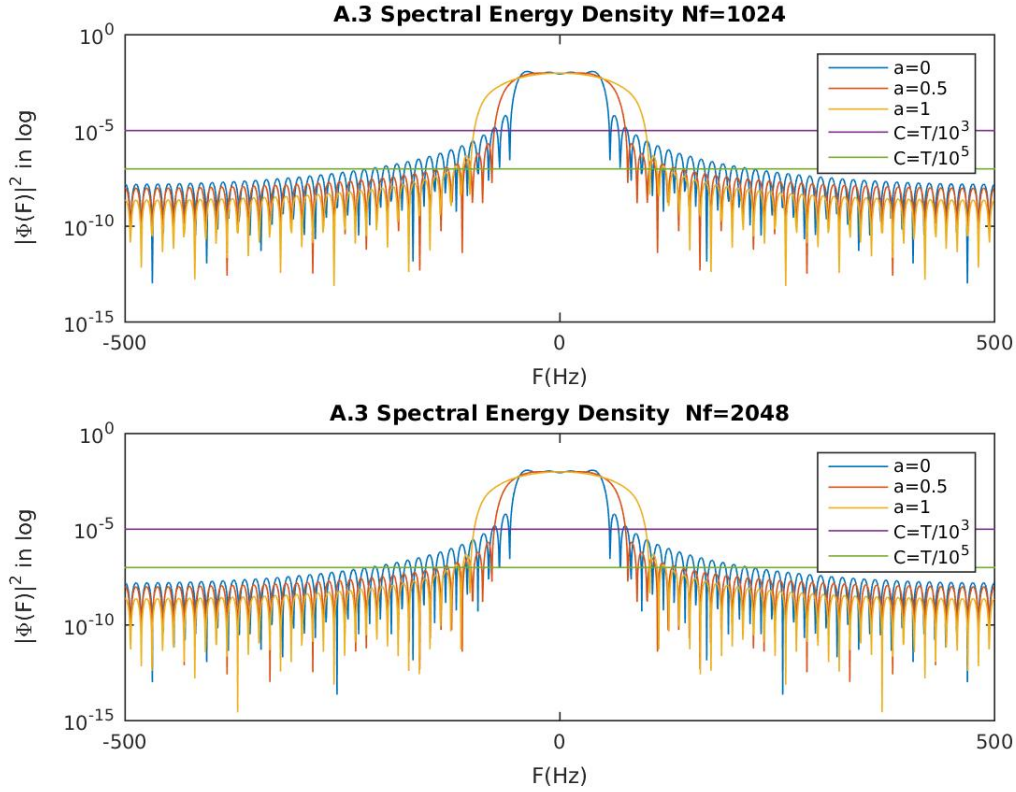
Σε αυτό το ερώτημα ζητήθηκε να χρησιμοποιήσουμε τις συναρτήσεις *fft* και *fftshift* προκειμένου να υπολογίσουμε τον Fourier Transform των παλμών που μόλις δημιουργήσαμε και να σχεδιάσουμε την

φασματική πυκνότητα ενέργειας αυτών. Αξίζει να γίνουν μερικές σημειώσεις σχετικά με την συγκεκριμένη υλοποίηση. Αρχικά το διάστημα συχνοτήτων του μετασχηματισμού, όπως δίνεται και από την εκφώνηση είναι το $[-\frac{F}{2}s, \frac{F}{2}s)$. Για λόγους κανονικοποίησης έγινε πολλαπλασιασμός του fft result με το Ts ενώ επίσης, παρόλο που δεν ήταν απαραίτητο πραγματοποιήθηκε η ίδια διαδικασία ώστε το Nf (ισαπέχοντα σημεία) να ισούται με 1024 όσο και με 2048. Να αναφέρουμε ότι ο λόγος που χρησιμοποιείται το fftshift είναι για να μεταφέρει τους παλμούς στο κέντρο ώστε να μπορούν να συγκριθούν πιο εύκολα. Αφού είχαμε κάνει τα παραπάνω εμφανίζεται η ζητούμενη πληροφορία $|\Phi(F)|^2$ τόσο σε κανονική κλίμακα με την χρήση της συνάρτησης plot, όσο και σε ημι-λογαριθμική κλίμακα με την χρήση της συνάρτησης semilogy όπου μπορούμε να παρατηρήσουμε περισσότερες λεπτομέρειες, καθώς με αυτήν δίνεται η δυνατότητα να μελετήσουμε τις τιμές των $|\Phi(F)|^2$ σε διαστήματα όπου αυτές είναι πολύ μικρές.

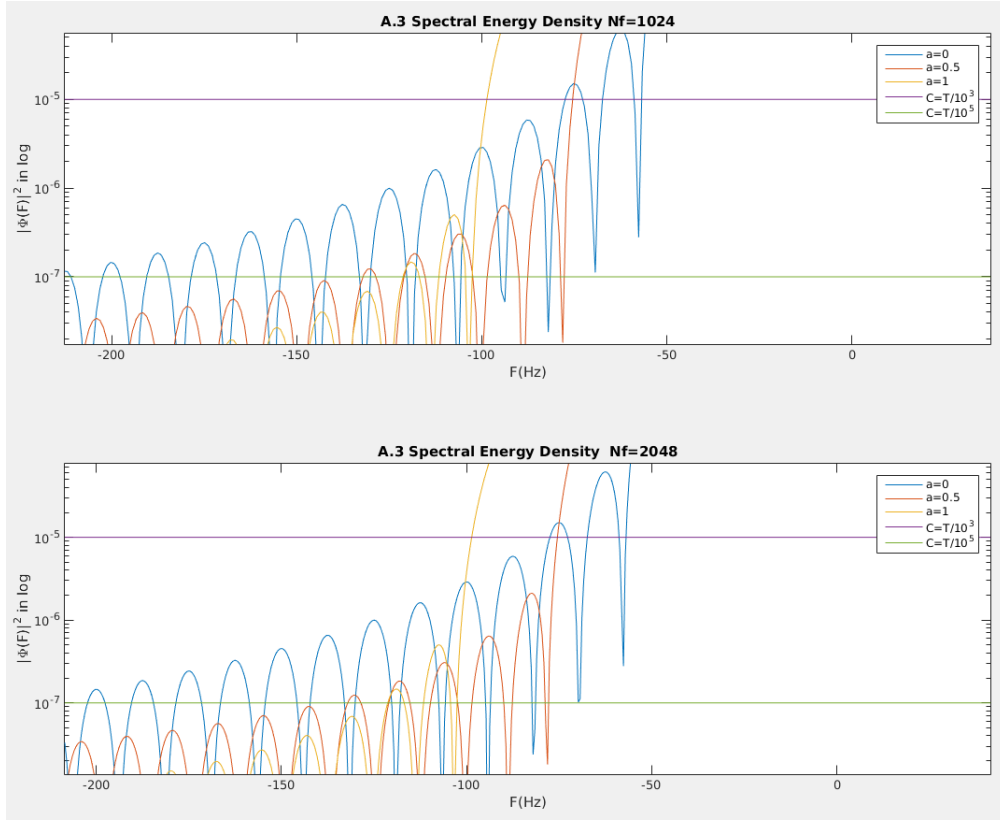


A.3 Bandwidth Calculation

Σκοπός του συγκεκριμένου ερωτήματος ήταν να δούμε τι συμβαίνει σχετικά με το εύρος φάσματος των παλμών, τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο, στο οποίο τότε θεωρητικά έχουν άπειρο, και κατά πόσο αυτό εξαρτάται από το συντελεστή roll-off a . Πρώτο βήμα ήταν να υπολογίσουμε τα θεωρητικά bandwidth για κάθε a μέσω του τύπου $BW = \frac{1+a}{2T}$. Αφού έγινε αυτό, υποθέσαμε μία νοητή ευθεία η οποία θα ήταν το πρώτο φράγμα των πειραμάτων όπου χρησιμοποιήθηκε ως νοητό μηδέν. Σκοπός της ήταν να θεωρήσουμε ότι κάτω από αυτήν, οι παλμοί είναι πρακτικά μηδέν και σύμφωνα με αυτό να δούμε το πρακτικό BW. Μετά επαναλήφθηκε η ίδια διαδικασία για διαφορετική ευθεία προκειμένου να δούμε αν υπάρχουν διαφορετικά αποτελέσματα. Οι ευθείες που θεωρήσαμε ήταν η $C_1 = \frac{T}{10^3}$ και $C_2 = \frac{T}{10^5}$ ενώ εμφανίζονται διαγράμματα και για τα δύο N_f που δόθηκαν ως τυπικές τιμές.



Προκειμένου να μπορέσουμε να διακρίνουμε προσεγγιστικά το εύρος φάσματος του κάθε παλμού μας βοήθησε η χρήση του zoom στα figures.



Σύμφωνα με τα παραπάνω δημιουργήθηκε ο παρακάτω πίνακας ο οποίος συνοψίζει όλες τις πληροφορίες.

$a \backslash BW$	Theoretical ($BW = \frac{1+a}{2T}$)	$c = \frac{T}{10^3}$	$c = \frac{T}{10^5}$
0	50	77.6	214
0.5	75	75.5	132
1	100	98.6	121

Αυτό που μπορούμε να παρατηρήσουμε είναι ότι για $c = \frac{T}{10^3}$ τότε πιο αποδοτικός παλμός (μικρότερος σε εύρος φάσματος) είναι αυτός με roll-off $a=0.5$ ενώ αντίθετα για $c = \frac{T}{10^5}$ είναι αυτός με $a=1$. Πράγμα που σημαίνει ότι στην πράξη δεν είναι μονοσήμαντα ορισμένο ποιος παλμός είναι βέλτιστος ως προς το εύρος φάσματος καθώς εξαρτάται από διάφορους παράγοντες όπως τη θεωρητική ευθεία που θα θεωρήσουμε ως “πρακτικά μηδέν”.

3 Ερώτημα Β

Σε αυτό το θέμα της εργαστηριακής άσκησης χρειάστηκε να μελετήσουμε τους παλμούς που είχαμε ήδη δημιουργήσει SRRC $\varphi(t)$ ως προς την ορθοκανονικότητα τους ως προς τις μετατοπίσεις τους κατά ακέραια πολλαπλάσια k της περιόδους τους T .

B.1.1 Plot $\varphi(t)$ and $\varphi(t-kT)$

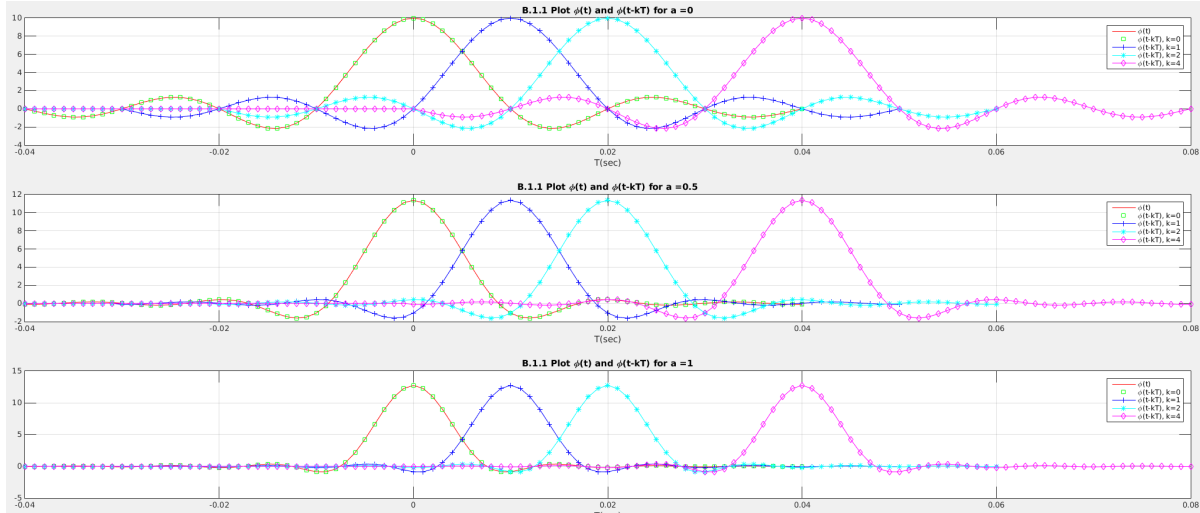
Πρώτο ζητούμενο ήταν να σχεδιάσουμε σε κοινό plot τους παλμούς $\varphi(t)$ και $\varphi(t-kT)$ για $a = 0, 0.5, 1$ και $k = 0, 1, 2, 4$. Για να γίνει αυτό πρώτα έπρεπε να ορίσουμε κατάλληλα τον χρόνο ο οποίος εξαρτάται από τον αριθμό k ενώ έπειτα για τα σήματα χρειάστηκε να κάνουμε zero padding όσα μηδενικά κάθε φορά έπρεπε σύμφωνα με την μετατόπιση που γινόταν. Για το $\varphi(t)$ το zero padding θα γίνει προς τα δεξιά καθώς προς τα εκεί μετακινείται το άλλο σήμα ενώ για το $\varphi(t-kT)$ προς τα αριστερά, αντίθετο δηλαδή με την φορά της κίνησης του. Τα σχήματα τα οποία δημιουργήθηκαν σύμφωνα με τα παραπάνω εμφανίζονται στα ακόλουθα διαγράμματα.

Listing 1: Create signals

```

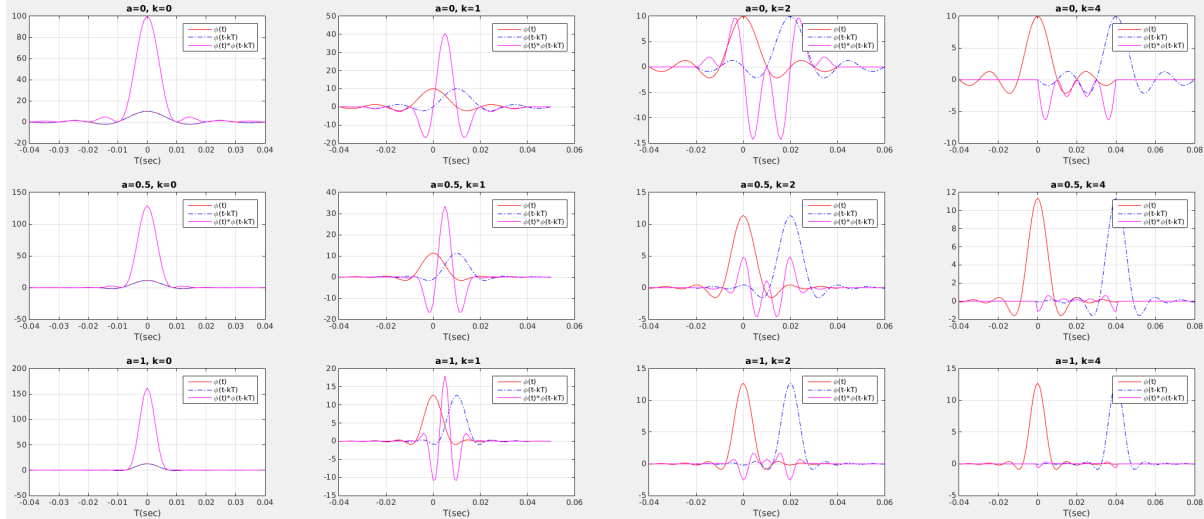
1 %Create signals
2 t_s=[-A*Ts:(A+k)*T];      % time vector with needed extra time added for shifting
3 phi_t_za=[phi_t(i,:) zeros(1,k*over)];      % phi(t) (with zeros added)
4 phi_kt_za=[zeros(1,k*over) phi_t(i,:)];      % phi(t-kT) (with zeros added)

```



B.1.2 Plot $\varphi(t)\varphi(t - kT)$

Αφού είχαμε κάνει το παραπάνω υπολογίσαμε και σχεδιάσαμε τα γινόμενα $\varphi(t)\varphi(t - kT)$ για $a = 0, 0.5, 1$ και $k = 0, 1, 2, 4$ και παρακάτω εμφανίζονται τα αποτελέσματα που πήραμε.



B.1.3 Calculate $\int \varphi(t)\varphi(t - kT)dt$

Αφού κάναμε αυτά προσεγγίσαμε αριθμητικά το ολοκλήρωμα του γινομένου $\varphi(t)\varphi(t-kT)$ για $a = 0, 0.5, 1$ και $k = 0, 2, 4$. Ο τρόπος με τον οποίο το κάναμε ήταν ουσιαστικά με το να αθροίσουμε τις τιμές για τα γινόμενα που ήδη έχουμε υπολογίσει. Παρακάτω παρατίθεται ο πίνακας με τις τιμές των integrals που ζητήθηκαν.

$a \backslash k$	0	2	4
0	0.974749	-0.034885	-0.095912
0.5	0.999876	0.000333	0.000735
1	0.999969	-0.000082	-0.000159

Όπως μπορούμε να παρατηρήσουμε και από τον πίνακα μπορούμε να δούμε ότι οι αποκομμένοι SRRC παλμοί είναι προσεγγιστικά ορθοκανονικοί, ως προς τις μετατοπίσεις τους κατά kT καθώς για $k = 0$ και οι τρεις περιπτώσεις ισούνται σχεδόν με την μονάδα ενώ για $k \neq 0$ πλησιάζουν το 0 με την προσέγγιση να βελτιώνεται όσο το a πλησιάζει τη μονάδα.

4 Ερώτημα C

Σε αυτό το ζητούμενο είχαμε να προσομοιώσουμε ένα 2-PAM σύστημα βασικής ζώνης το οποίο μεταφέρει N bits. Αρχικά δημιουργήσαμε ένα SRRC παλμό με χαρακτηριστικά $T = 0.1$ sec, $over = 10$, $a = 0.5$, και $A = 5$.

C.1 Δημιουργία N bits

Πρώτο πράγμα που μας ζητήθηκε ήταν να δημιουργήσουμε τα N bits, σε αυτή την υλοποίηση επιλέχθηκε $N = 100$. Ο τρόπος με τον οποίο δημιουργήθηκαν ήταν με την χρήση της σύνθετης εντολής:

$$b = (\text{sign}(\text{randn}(N, 1)) + 1)/2;$$

Όπου πρώτα υπάρχει η χρήση της $\text{randn}(N, 1)$ ώστε να δημιουργήσουμε ένα πίνακα $N \times 1$ από αριθμούς κανονικής κατανομής έπειτα με την χρήση της sign κρατάμε το πρόσημο του κάθε αριθμού πολλαπλασιασμένο με την μονάδα άρα μέχρι αυτό το σημείο έχουμε ένα πίνακα με -1 και 1, στην συνέχεια προσθέτουμε σε αυτούς την μονάδα για να δημιουργήσουμε ένα πίνακα με στοιχεία 0 και 2 και προκειμένου να δημιουργήσουμε τελικά τον πίνακα από bits διαιρούμε αυτά τα στοιχεία με το 2 ώστε να πάρουμε 0 και 1.

C.2.a 2-PAM διαμόρφωση βασικής ζώνης

Πλέον σκοπός ήταν να κωδικοποιήσουμε τα bits που μόλις δημιουργήσαμε σε 2-PAM βασικής ζώνης. Για να το κάνουμε αυτό δημιουργήθηκε το script `bits_to_2PAM(b)` ο οποίο παίρνει ως όρισμα ένα πίνακα b - ακολουθία bits - $N \times 1$ στοιχεία και μέσω απλών συνθηκών μετασχηματίζει σύμφωνα με την εκφώνηση το 0 σε +1 και 1 σε -1. Δηλαδή επιστρέφει την ακολουθία από 2-PAM σύμβολα X .

Listing 2: `bits_to_2PAM.m`

```

1 function [ Xo ] = bits_to_2PAM( Xi )
2     Xo=zeros(1,length(Xi));
3     for i=1:length(Xi)
4         if(Xi(i)>0)
5             Xo(i)=-1;
6         else
7             Xo(i)=1;
8         end
9     end
10 end

```

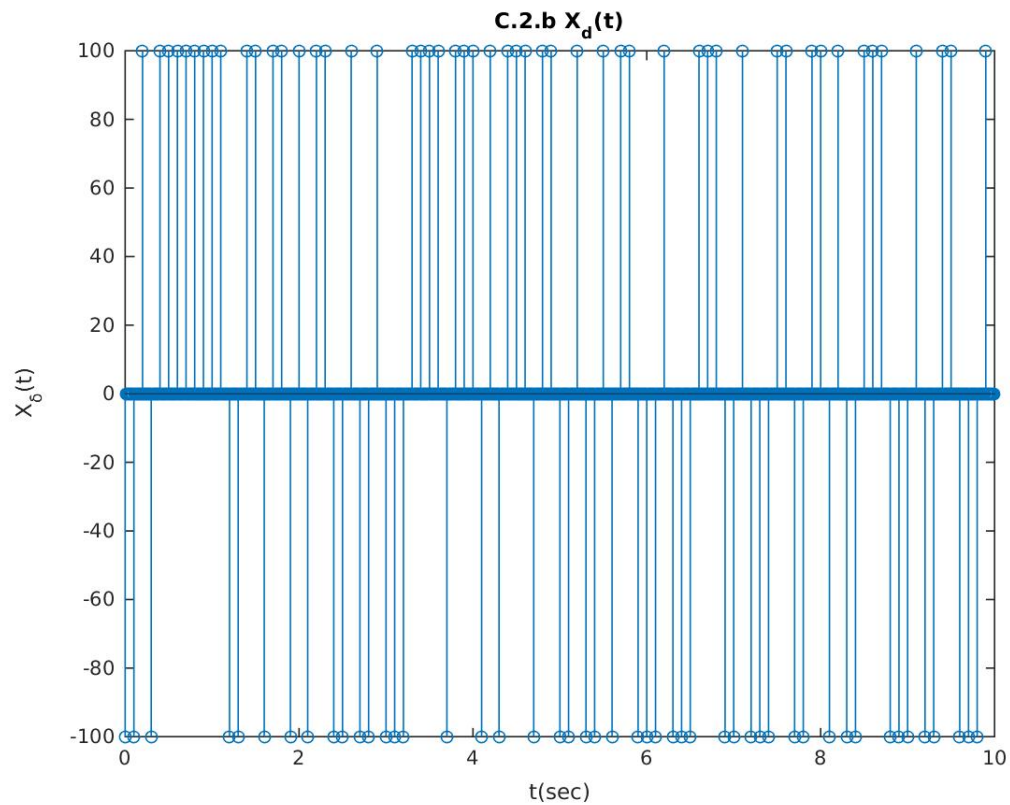
C.2.b Δημιουργία $X_\delta(t)$

Αφού είχαμε δημιουργήσει το 2-PAM βασικής ζώνης χρησιμοποιούμε την `upsample` η οποία αυξάνει το ρυθμό δειγματοληψίας του σήματος X προσθέτοντας ανάμεσα στα δείγματα $over-1$ μηδενικά. Αυτό που θέλαμε να πετύχουμε είναι να προσομοιώσουμε το σήμα $X_\delta(t)$ μέσω της εντολής

$$X_delta = 1/Ts * \text{upsample}(X, over);$$

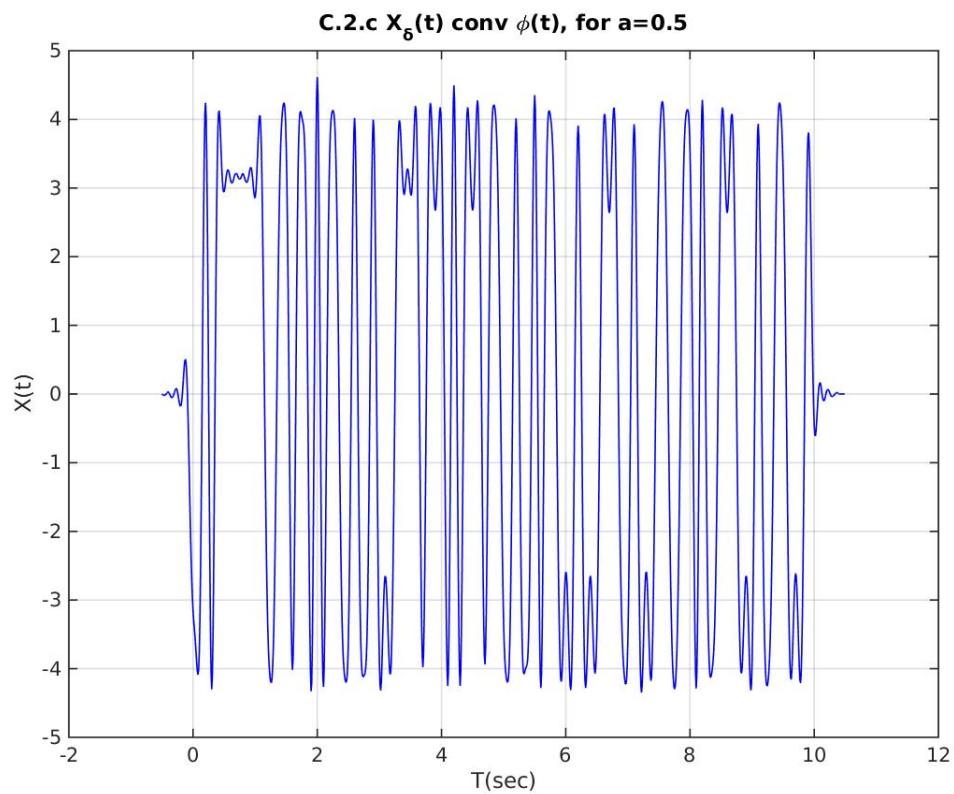
Ο υπολογισμός του σήματος X_δ ήταν εύκολος, αυτό που έπρεπε όμως να κάνουμε έξτρα ήταν να ορίσουμε κατάλληλα τον άξονα του χρόνου και να το σχεδιάσουμε. Σύμφωνα με τα παραπάνω ο χρόνος θα έπρεπε να είναι:

$$t_delta = [0 : Ts : (N * over - 1) * Ts];$$



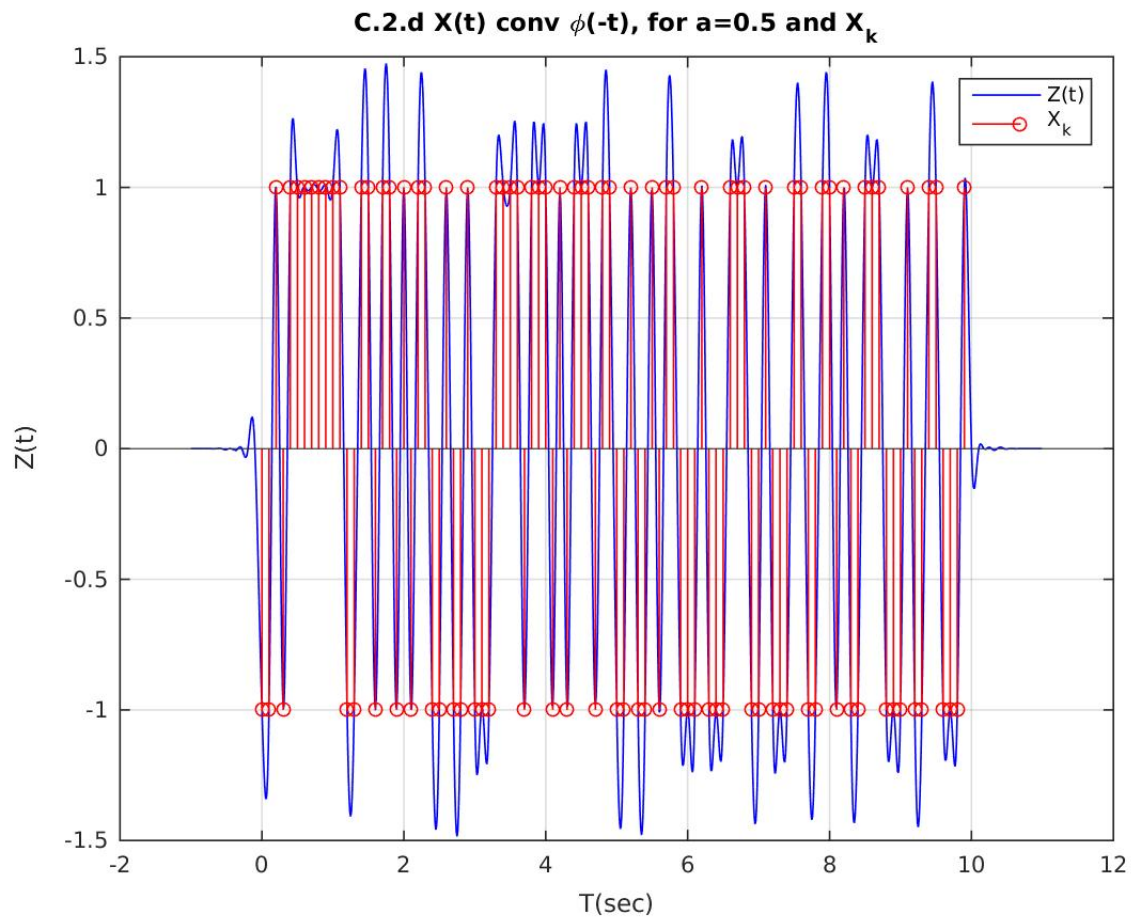
C.2.c Δημιουργία $X(t)$

Προκειμένου να υπολογίσουμε την συνέλιξη του $X(t) = X_d(t) \otimes \varphi(t)$. Χρησιμοποιήσαμε την έτοιμη συνάρτηση του MATLAB ενώ όμως αναφέρεται και στην εκφώνηση για τον ορισμό του άξονα του χρόνου δεν είχαμε παρά να αθροίσουμε τα όρια των δύο σημάτων. Παρακάτω παρουσιάζεται η ζητούμενη συνέλιξη.



C.2.d Δημιουργία $Z(t)$

Τέλος το μόνο που είχαμε να κάνουμε είναι να υποθέσουμε ιδανικό κανάλι και να υπολογίσουμε το αποτέλεσμα του δέκτη. Το οποίο ουσιαστικά είναι η συνέλιξη $Z(t) = X(t) \otimes \varphi(-t)$. Ακολουθήθηκε παρόμοια διαδικασία με το ερώτημα C.2.c με την διαφορά ότι έπρεπε να ανακλάσουμε το σήμα $\varphi(t)$. Στο ίδιο σχεδιάγραμμα εμφανίζουμε και τις τιμές του X_k , για $k = 0, \dots, N-1$ που είναι ουσιαστικά τα σύμβολα που δημιουργήσαμε στον πομπό. Αυτό που παρατηρούμε είναι ότι οι τιμές της δειγματοληψίας συμπίπτουν με ικανοποιητική ακρίβεια με τον παλμό μας. Άρα μπορούμε να ανακτήσουμε πλήρως την αρχική πληροφορία αφού το $\varphi(t)$ είναι ορθοκανονική όπως είδαμε παραπάνω.



Σημείωση: Λόγω της ιδιαιτερότητας του LaTeX για κάποια σύμβολα και του τρόπου που προστέθηκαν οι κώδικες μέσα σε αυτό, κάποια ονόματα των titles/xlabel/ylabel/comments ίσως είναι λίγο διαφορετικά από αυτά που εμφανίζονται στα figures ή που υπάρχουν στα ακριβές αρχεία που βρίσκονται στο GitHub. Οι διαφορές αφορούν KATEΞΟΧΗΝ θέματα ονόματα, παραδείγματος χάριν με το ελληνικό φ και δ τα οποία προκαλούσαν error στο LaTeX στα πλαίσια της προσθήκης του MATLAB κώδικα. ΔΕΝ συσχετίζεται με την λειτουργικότητα.

Listing 3: bits_to_2PAM.m

```

1 function [ Xo ] = bits_to_2PAM( Xi )
2     Xo=zeros(1,length(Xi));
3     for i=1:length(Xi)
4         if(Xi(i)>0)
5             Xo(i)=-1;
6         else
7             Xo(i)=1;
8         end
9     end
10 end

```

Listing 4: part_a.m

```

1 % -----
2 %   Exercise 1, part A
3 %
4 %   Authors : Spyridakis Christos
5 %   Created Date : 26/10/2019
6 %   Last Updated : 28/10/2019
7 %
8 %   Description:
9 %               Code created for Exercises of Communication Systems Course
10 %              in Technical University of Crete
11 % -----
12
13 clear all ; close all ; clc ;
14
15 % Just for saving in a separate folder figures as images
16 DEBUG = true ; part = 'A.' ; dirpath = '../doc/photos' ; ext = '.jpg' ; if ~DEBUG && ~
    exist(dirpath,'dir') ; mkdir(dirpath); end
17 % Auxiliary variables for plots, semilogy, etc...
18 colors = ['r' 'g' 'b' 'c' 'm' 'y' 'k'] ;
19 valueStyles = ['o' 's' '+' '*' 'd' '.' 'x' ] ;
20
21
22 %%%%%%%%%%%
23 % A.1
24 %
25 % Init mandatory variables
26 stepName = '1 Create SRRC pulses'; extraInfo = '';
27 T=10^-2 ; over=10 ; Ts=T/over ; A=4 ; a=[0 0.5 1] ; phi_t = [] ; t=[] ;
28
29 % Create srcc pulses and plot them
30 f=figure();

```

```

31 for i=1:length(a)
32     [phi_tmp t_tmp] = srrc_pulse(T, Ts, A, a(i));
33     phi_t = [phi_t; phi_tmp];
34     t = t_tmp;
35
36     % Plot each srrc with a color and save plot to add extra info later
37     plot(t, phi_t(i,:), colors(i), 'DisplayName',strcat('a=',num2str(a(i)))) ; hold on
38     ;
39 end
40 hold off ; axis([-0.05 0.05 -4 15]) ;
41
42 % Add more info to plots
43 legend('Location','NorthEast'); grid on;
44 title(strcat(part,stepName)); ylabel('\phi(t)'); xlabel('T(sec)');
45 if ~DEBUG ; saveas(f,strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end
46 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
47 % A.2
48 %
49 % Init mantatory variable
50 stepName = '2 Fourier Transform ';
51 Phi_F1 = [] ; Phi_F2 = [] ;
52
53 % Calculate frequency vectors
54 Fs = 1/Ts ; Nf = [1024 2048] ;
55 F_1 = [-Fs/2 : Fs/Nf(1) : Fs/2-Fs/Nf(1)]; % Frequency vector for Nf=1024
56 F_2 = [-Fs/2 : Fs/Nf(2) : Fs/2-Fs/Nf(2)]; % Frequency vector for Nf=2048
57
58 %Fourier Transform and save to vector
59 for i=1:length(a)
60     X1 = fftshift(fft(phi_t(i,:),Nf(1))*Ts) ; Phi_F1 = [Phi_F1 ; X1] ;
61     X2 = fftshift(fft(phi_t(i,:),Nf(2))*Ts) ; Phi_F2 = [Phi_F2 ; X2] ;
62 end
63
64 % Plot them
65 f=figure(); extraInfo='--Plots';
66 % Nf = 1024
67 subplot(1,2,1);
68 p1 = plot(F_1, abs(Phi_F1(1,:)).^2); hold on;
69 p2 = plot(F_1, abs(Phi_F1(2,:)).^2); hold on;
70 p3 = plot(F_1, abs(Phi_F1(3,:)).^2); hold off;
71 legend([p1,p2,p3], 'a=0', 'a=0.5', 'a=1'); legend('Location','NorthEast'); grid on;
72 title('A.2.a Spectral energy density Nf=1024'); ylabel('|Phi(F)|^2'); xlabel('F(Hz)');
73 % Nf = 2048
74 subplot(1,2,2);
75 p1 = plot(F_2, abs(Phi_F2(1,:)).^2); hold on;
76 p2 = plot(F_2, abs(Phi_F2(2,:)).^2); hold on;
77 p3 = plot(F_2, abs(Phi_F2(3,:)).^2); hold off;
78 legend([p1,p2,p3], 'a=0', 'a=0.5', 'a=1'); legend('Location','NorthEast'); grid on;
79 title('A.2.a Spectral energy density Nf=2048'); ylabel('|Phi(F)|^2'); xlabel('F(Hz)');
80

```

```

81 if ~DEBUG ; saveas(f, strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end
82
83
84 % Semilogy
85 f=figure(); extraInfo='--Semilogy';
86 % Nf = 1024
87 subplot(2,1,1);
88 p1 = semilogy(F_1, abs(Phi_F1(1,:)).^2); hold on;
89 p2 = semilogy(F_1, abs(Phi_F1(2,:)).^2); hold on;
90 p3 = semilogy(F_1, abs(Phi_F1(3,:)).^2); hold off;
91 legend([p1, p2, p3], 'a=0', 'a=0.5', 'a=1'); legend('Location', 'NorthEast'); grid on;
92 title('A.2.b Spectral energy density Nf=1024'); ylabel('|Phi(F)|^2 in log'); xlabel(
    ('F(Hz)'));
93 % Nf = 2048
94 subplot(2,1,2);
95 p1 = semilogy(F_2, abs(Phi_F2(1,:)).^2); hold on;
96 p2 = semilogy(F_2, abs(Phi_F2(2,:)).^2); hold on;
97 p3 = semilogy(F_2, abs(Phi_F2(3,:)).^2); hold off;
98 legend([p1, p2, p3], 'a=0', 'a=0.5', 'a=1'); legend('Location', 'NorthEast'); grid on;
99 title('A.2.b Spectral energy density Nf=2048'); ylabel('|Phi(F)|^2 in log'); xlabel(
    ('F(Hz)'));
100
101 if ~DEBUG ; saveas(f, strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end
102
103 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104 % A.3
105 %
106 % Init mantatory variable
107 stepName = '3 Bandwidth'; extraInfo='';
108
109 % Theoritical Bandwidth
110 disp('Theoritical Bandwidth');
111 BW=(1+a)./(2*T)
112
113 % Practical Bandwith
114 c = [T/(10^3) T/(10^5)];
115 f=figure();
116 % Nf = 1024
117 subplot(2,1,1);
118 p1 = semilogy(F_1, abs(Phi_F1(1,:)).^2); hold on;
119 p2 = semilogy(F_1, abs(Phi_F1(2,:)).^2); hold on;
120 p3 = semilogy(F_1, abs(Phi_F1(3,:)).^2); hold on;
121 p4 = plot(xlim, [c(1) c(1)]); hold on;
122 p5 = plot(xlim, [c(2) c(2)]); hold off
123 legend([p1, p2, p3, p4, p5], 'a=0', 'a=0.5', 'a=1', 'C=T/10^3', 'C=T/10^5'); legend(
    'Location', 'NorthEast'); grid off;
124 title('A.3 Spectral Energy Density Nf=1024'); ylabel('|Phi(F)|^2 in log'); xlabel('
    F(Hz)');
125
126 % Nf = 2048
127 subplot(2,1,2);
128 p1 = semilogy(F_2, abs(Phi_F2(1,:)).^2); hold on;
129 p2 = semilogy(F_2, abs(Phi_F2(2,:)).^2); hold on;

```

```

130 p3 = semilogy(F_2, abs(Phi_F2(3,:)).^2); hold on;
131 p4 = plot(xlim ,[c(1) c(1)]); hold on;
132 p5 = plot(xlim ,[c(2) c(2)]); hold off
133 legend([p1, p2, p3, p4, p5], 'a=0', 'a=0.5', 'a=1' , 'C=T/10^3', 'C=T/10^5'); legend(
    'Location','NorthEast'); grid off;
134 title('A.3 Spectral Energy Density Nf=2048'); ylabel('|Phi(F)|^2 in log'); xlabel(
    'F(Hz)');
135 if ~DEBUG ; saveas(f, strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end

```

Listing 5: part_b.m

```

1 % -----
2 %   Exercise 1, part B
3 %
4 %   Authors : Spyridakis Christos
5 %   Created Date : 27/10/2019
6 %   Last Updated : 28/10/2019
7 %
8 %   Description:
9 %               Code created for Exercises of Communication Systems Course
10 %              in Technical Unoversity of Crete
11 % -----
12
13 clear all; close all; clc;
14
15 % Just for saving in a separate folder figures as images
16 DEBUG = true ; part = 'B.' ; dirpath = '../doc/photos' ; ext = '.jpg' ; if ~DEBUG && ~
    exist(dirpath,'dir') ; mkdir(dirpath); end
17 % Auxiliary variables for plots, semilogy, etc...
18 colors = ['r' 'g' 'b' 'c' 'm' 'y' 'k'] ;
19 valueStyles = ['o' 's' '+' '*' 'd' '.' 'x' ] ;
20
21 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22 % B
23 %
24 % Init mantatory variable
25 T=10^-2 ; over=10 ; Ts=T/over ; A=4 ; a=[0 0.5 1] ; phi_t = [] ; t=[] ;
26
27 % Create srrc pulses
28 for i=1:length(a)
29     [phi_tmp t_tmp] = srrc_pulse(T, Ts, A, a(i));
30     phi_t = [phi_t; phi_tmp];
31     t = t_tmp;
32 end
33
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35 % B.1.1
36 stepName='1.1 Plot \phi(t) and \phi(t-kT)'; %TODO
37 % for a = 0, 0.5, 1 and k=0,1,2,4 plot phi(t) and phi(t-kT)
38 f=figure(); extraInfo='';
39     for i=1:length(a)                % for a = 0, 0.5, 1
40         subplot(3,1,i) ; col=2;
41         for k=[0 1 2 4]              % for k=0,1,2,4

```

```

42 %Create signals
43 t_s=[-A*Ts:(A+k)*T]; % time vector with needed extra
    time added for shifting
44 phi_t_zs=[phi_t(i,:) zeros(1,k*over)]; % phi(t) (with zeros added)
45 phi_kt_zs=[zeros(1,k*over) phi_t(i,:)]; % phi(t-kT) (with zeros added)
46
47 % Plot once initial signal then plot others
48 if k==0
49     plot(t_s, phi_t_zs, strcat(colors(1),'-'), 'DisplayName','\phi(t)') ; hold on
    ;
50     plot(t_s, phi_kt_zs, strcat(colors(col),valueStyles(col)), 'DisplayName','\phi
        (t-kT), k=0') ; hold on ; col=col+1;
51 else
52     plot(t_s, phi_kt_zs, strcat(colors(col),'-',valueStyles(col)), 'DisplayName',
        strcat('\phi(t-kT), k=',num2str(k))) ; hold on ; col=col+1;
53 end
54 end
55 hold off; legend('Location','NorthEast'); grid on;
56 title(strcat(part,stepName, ' for a = ',num2str(a(i)))); ylabel(''); xlabel('T(sec
    )');
57 end
58 %suptitle('All');
59 if ~DEBUG ; saveas(f,strcat(dirpath, '/', part, '1.1', extraInfo, ext)) ; end
60
61
62 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
63 % B.1.2
64 stepName='1.2 Products';
65 % for a = 0, 0.5, 1 and k=0,1,2,4 plot phi(t)*phi(t-kT)
66 f=figure(); p_num=1;
67 for i=1:length(a) % for a = 0, 0.5, 1
68     for k=[0 1 2 4] % for k = 0,1,2,4
69         subplot(3,4,p_num) ; p_num=p_num+1 ; col=2;
70         %Create signals
71         t_s=[-A*Ts:(A+k)*T]; % time vector with needed extra
            time added for shifting
72         phi_t_zs=[phi_t(i,:) zeros(1,k*over)]; % phi(t) (with zeros added)
73         phi_kt_zs=[zeros(1,k*over) phi_t(i,:)]; % phi(t-kT) (with zeros added)
74
75         plot(t_s, phi_t_zs, 'r-', 'DisplayName','\phi(t)') ; hold on ;
            % phi(t)
76         plot(t_s, phi_kt_zs, 'b-.', 'DisplayName', '\phi(t-kT)') ; hold on
            % phi(t-kT)
77         plot(t_s, phi_t_zs.*phi_kt_zs,'m', 'DisplayName','\phi(t)*\phi(t-kT)') ; hold
            off; % phi(t)*phi(t-kT)
78
79         legend('Location','NorthEast'); grid on;
80         title(strcat(' a=',num2str(a(i)), ', k=', num2str(k))); ylabel(''); xlabel('T(
            sec)');
81     end
82 end
83 % suptitle('Text');
84 if ~DEBUG ; saveas(f,strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end

```

```

85
86
87 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88 % B.1.3
89 stepName='3'; integrals=[] ; %TODO
90 for i=1:length(a)
91     for k=[0 2 4]
92         %Create signals
93         phi_t_zs=[phi_t(i,:) zeros(1,k*over)];           % phi(t) (with zeros added)
94         phi_kt_zs=[zeros(1,k*over) phi_t(i,:)];          % phi(t-kT) (with zeros added)
95
96         integrals=[integrals sprintf('a=%.1f, k=%d, integral=%f\n',a(i),k,sum(phi_t_zs.*
97             phi_kt_zs)*Ts)];
98     end
99 end
100 disp('Integrals') ; disp(integrals)

```

Listing 6: part_c.m

```

1 % -----
2 %   Exercise 1, part C
3 %
4 %   Authors : Spyridakis Christos
5 %   Created Date : 28/10/2019
6 %   Last Updated : 28/10/2019
7 %
8 %   Description:
9 %               Code created for Exercises of Communication Systems Course
10 %               in Technical University of Crete
11 %
12 % -----
13
14 clear all ; close all ; clc ;
15
16 % Just for saving in a separate folder figures as images
17 DEBUG = false ; part = 'C.' ; dirpath = '../doc/photos' ; ext = '.jpg' ; if ~DEBUG && ~
    exist(dirpath,'dir') ; mkdir(dirpath); end
18 % Auxiliary variables for plots, semilogy, etc...
19 colors = ['r' 'g' 'b' 'c' 'm' 'y' 'k'] ;
20 valueStyles = ['o' 's' '+' '*' 'd' '.' 'x' ];
21
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 % C
24 %
25 % Init mandatory variables
26 T = 0.1 ; over = 10 ; a = 0.5 ; A = 5 ;
27
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 % C.1
30 N = 100;
31 b=(sign(randn(N, 1)) + 1)/2;
32
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

34 % C.2.a
35 X = bits_to_2PAM(b);
36
37 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
38 % C.2.b
39 stepName = '2.b X_d(t)'; extraInfo = '';
40 %Create signal
41 Ts=T/over;
42 X_delta = 1/Ts * upsample(X, over);
43 t_delta = [ 0 : Ts : (N*over-1)*Ts ];
44 %Plot
45 f=figure();
46 stem(t_delta, X_delta);
47 title(strcat(part,stepName)); ylabel('X_delta(t)'); xlabel('t(sec)');
48 if ~DEBUG ; saveas(f,strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end
49
50 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51 % C.2.c
52 stepName = '2.c '; extraInfo = ' X(t)';
53 %Create SRRC signal
54 [phi t_phi] = srrc_pulse(T, Ts, A, a);
55
56 % Convolution
57 t_Xd_conv_phi = [t_delta(1) + t_phi(1) : Ts : t_delta(end) + t_phi(end)];
58 X_t=conv(X_delta,phi)*Ts;
59
60 %Plot X_delta(t) ** phi(t)
61 f=figure();
62 plot(t_Xd_conv_phi, X_t, 'b') ; grid on;
63 title(strcat(part,stepName,' X_delta(t) conv \phi(t), for a=0.5')); ylabel('X(t)');
64 xlabel('T(sec)');
65 if ~DEBUG ; saveas(f,strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end
66
67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68 % C.2.d
69 stepName = '2.d '; extraInfo = ' Z(t) and Xk';
70 phi_rev = phi(end:-1:1);
71 t_phi_rev = t_phi;
72
73 % Convolution
74 t_Xd_conv_phi_rev = [t_Xd_conv_phi(1) + t_phi_rev(1) : Ts : t_Xd_conv_phi(end) +
75     t_phi_rev(end)];
76 Z_t=conv(X_t,phi_rev)*Ts;
77
78 %Plot Xdelta(t) conv phi(-t) and Xk
79 f=figure();
80 p1 = plot(t_Xd_conv_phi_rev, Z_t, 'b') ; hold on;
81 p2 = stem([0:N-1]*T, X,'r') ; hold off;
82 legend([p1,p2],'Z(t)', 'X_k'); legend('Location','NorthEast'); grid on;
83 title(strcat(part,stepName,' X(t) conv \phi(-t), for a=0.5 and X_k')); ylabel('Z(t)');
84 xlabel('T(sec)');
85 if ~DEBUG ; saveas(f,strcat(dirpath, '/', part, stepName, extraInfo, ext)) ; end

```