

Ο κώδικας για όλα τα ερωτήματα σε συνολο:

```
1 %TELECOMMUNICATION SYSTEMS I
2 %exercise 2
3 %authors—Panagiotis Sklabos/Giannis Peridis
4
5 %MATLAB CODE
6 clear all;
7 close all;
8 clc;
9 format compact;
10 %=====
11 %A.1
12 %intialization of symbol period T,oversampling factor over,
13 %parameter A(half duration of the pulse) and roll-off factor a
14 T=10^-2;
15 over=10;
16 A=4;
17 a=0.5;
18
19 %creating srcc pulses
20 [phi,t]=srcc_pulse(T,over,A,a);
21
22 %intialization of sampling period Ts,sampling frequency Fs,
23 %parameter Nf(number of equidistant points) and F axis
24 Ts=T/over;
25 Fs=1/Ts;
26 Nf=4096;
27 F=[-Fs/2:Ff/Nf:Ff/2-Ff/Nf];
28
29 %creating fast fourier transformation
30 PHI=fftshift(fft(phi,Nf)*Ts);
31
32 %creating the energy density spectrum
33 edsPHI=abs(PHI).^2;
34
35 %creating graph of the energy density spectrums
36 %in the same plot
37 disp('press any key to see the energy density spectrum with logarithmic yaxis')
38 pause
39 figure();
```

```

40 semilogy(F,edsPHI);
41 title('energy density spectrum with logarithmic yaxis');
42 xlabel('F=frequency in Hz');
43 ylabel('edsPhi =energy density spectrum of srcc pulses');
44 legend('a = 0.5');
45
46 %=====
47 %A.2
48 %N number of bits
49 N=100;
50
51 %call of the created function that transfers bits to 2PAM
52 disp('a random sequence of independent and equally possible bits')
53 b=(sign(randn(N,1))+1)/2;
54 o=transpose(b)
55
56 %call of the created function that transfers bits to 2PAM
57 disp('the transformed 2PAM sequence')
58 Xn=bits_to_2PAM(b)
59 t1=(0:Ts:N*T-Ts);
60
61 %creation of X_delta
62 X_delta=1/Ts*upsample(Xn,over);
63
64 %creation of X(t)=sum(Xn*phi(t-n*T))
65 %convolution of phi and X_delta
66 tconv=[t(1)+t1(1):Ts:t(end)+t1(end)];
67 X_t=conv(phi,X_delta)*Ts;
68 %creation of the graph of X(t)
69 disp('press any key to see X(t) waveform ')
70 pause
71 figure()
72 plot(tconv,X_t);
73 title(' X(t)=sum(Xn*phi(t-n*T))')
74 ylabel(' X(t)')
75 xlabel('t=time in seconds')
76
77 %=====
78 %A3
79 %creating fast fourier transformation X(F) of X(t)
80 X_F=fftshift(fft(X_t,Nf)*Ts);
81 %the total duration time
82 Ttotal=length(tconv)*Ts;

```

```

83
84 %creation of a periodogram of one of the implementations of X(t)
85 P_x=abs(X_F).^2/Ttotal;
86
87 %creation of periodogram graph in logarithmic y axis
88 disp('press any key to see the periodogram')
89 pause
90 figure();
91 semilogy(F,P_x);
92 title('Periodogram with logarithmic y axis');
93 xlabel('F=frequency in Hz');
94 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
95 legend('a = 0.5');
96
97 %creation of periodogram graph in plot
98 disp('press any key to see the periodogram with logarithmic y axis')
99 pause
100 figure();
101 plot(F,P_x);
102 title('Periodogram of X');
103 xlabel('F=frequency in Hz');
104 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
105 legend('a = 0.5');
106
107 %initializing K the number of periodograms Px(F) of the of implemenations of
108 %X(t) and sum ,helps to find the summary
109 K =500;
110 sum=0;
111 %every loop we create a new implementation of X(t) and a new periodogram
112 %Px(F) for it, we repeat K times
113 for k=0:K-1
114     b=(sign(randn(N,1))+1)/2;
115     Xn=bits_to_2PAM(b);
116     X_delta=1/Ts*upsample(Xn,over);
117     X_t=conv(phi,X_delta)*Ts;
118     X_F=fftshift(fft(X_t,Nf)*Ts);
119     P_x=abs(X_F).^2/Ttotal;
120     sum=sum+P_x;
121 end
122 %creation of the estimated power density spectrum, we divide the summary by K times
123 %to find the arithmetic mean
124 S_x=sum/K;
125

```

```

126 %creating the theoritical Sx(F)
127 variance=1;
128 S_x_th= (variance*(abs(PHI).^2))/T;
129
130 %creating the graph of the estimated Sx(F) and the theoritical Sx(F) in the
131 %same plot with logarithmic y axis
132 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')
133 pause
134 figure();
135 semilogy(F,S_x,F,S_x_th);
136 title('estimated Sx(F) and theoritical Sx(F) smilogy');
137 xlabel('F=frequency in Hz');
138 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
139 legend('estimated', 'theoritical');
140 figure
141
142 plot(F,S_x,F,S_x_th);
143 title('estimated Sx(F) and theoritical Sx(F) plot');
144 xlabel('F=frequency in Hz');
145 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
146 legend('estimated', 'theoritical');
147 %=====
148 %A4
149 %creating a random sequence of N bits
150 b=(sign(randn(N,1))+1)/2;
151 %call of the created function that transfers bits to 4PAM
152 Xn_4PAM=bits_to_4PAM(b);
153 t_4PAM=(0:Ts:length(Xn_4PAM)*T-Ts);
154
155 %creation of X_delta
156 X_delta_4PAM=1/Ts*upsample(Xn_4PAM,over);
157
158 %creation of X(t)=sum(Xn*phi(t-n*T))
159 %convolution of phi and X_delta
160 tconv_4PAM=[t(1)+t_4PAM(1):Ts:t(end)+t_4PAM(end)];
161 X_t_4PAM=conv(phi,X_delta_4PAM)*Ts;
162 %creation of the graph X(t)
163 disp('press any key to see X(t) waveform ')
164 pause
165 figure()
166 plot(tconv_4PAM,X_t_4PAM);
167 title(' X(t)=sum(Xn*phi(t-n*T))')

```

```

168 ylabel(' X(t)')
169 xlabel('t=time in seconds')
170
171 %creation of a periodogram of one of the implementations of X(t)_4PAM
172 X_F_4PAM=fftshift(fft(X_t_4PAM,Nf)*Ts);
173 Ttotal=length(tconv_4PAM)*Ts;
174 P_x_4PAM=abs(X_F_4PAM).^2/Ttotal;
175
176 %creation of periodogram graph in plot
177 disp('press any key to see the periodogram of X(t)_4PAM ')
178 pause
179 figure();
180 semilogy(F,P_x_4PAM);
181 title('Periodogram with logarithmic y axis');
182 xlabel('F=frequency in Hz');
183 ylabel('Px(F)_4PAM=(|X(F)_4PAM|^2)/Ttotal');
184 legend('a = 0.5');
185
186 %creation of periodogram graph in logarithmic y axis
187 disp('press any key to see the periodogram of X(t)_4PAM with logarithmic y axis')
188 pause
189 figure();
190 plot(F,P_x_4PAM);
191 title('Periodogram of X_4PAM');
192 xlabel('F=frequency in Hz');
193 ylabel('Px(F)_4PAM=(|X(F)_4PAM|^2)/Ttotal');
194 legend('a = 0.5');
195
196 %initializing K the number of periodograms Px(F)_4PAM of the of implemenations of
197 %X(t)_4PAM and sum ,helps to find the summary
198 K =500;
199 sum_4PAM=0;
200 %every loop we create a new implementation of X(t)_4PAM and a new periodogram
201 %Px(F)_4PAM for it, we repeat K times
202 for k=0:K-1
203     b=(sign(randn(N,1))+1)/2;
204     Xn_4PAM=bits_to_4PAM(b);
205     X_delta_4PAM=1/Ts*upsample(Xn_4PAM,over);
206     X_t_4PAM=conv(phi,X_delta_4PAM)*Ts;
207     X_F_4PAM=fftshift(fft(X_t_4PAM,Nf)*Ts);
208     P_x_4PAM=abs(X_F_4PAM).^2/Ttotal;
209     sum_4PAM=sum_4PAM+P_x_4PAM;
210 end

```

```

211 %creation of the estimated power density spectrum, we divide the summary by K times
212 %to find the arithmetic mean
213 S_x_4PAM=sum_4PAM/K;
214
215 %creating the theoritical Sx(F)_4PAM
216 variance=5;
217 S_x_th_4PAM= (variance*(abs(PHI).^2))/T;
218
219 %creating the graph of the estimated Sx(F)_4PAM and the theoritical Sx(F)_4PAM in the
220 %same plot with logarithmic y axis
221 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')
222 pause
223 figure();
224 semilogy(F,S_x_4PAM);
225 hold on;
226 semilogy(F,S_x_th_4PAM);
227 title('estimated Sx(F)_4PAM and theoritical Sx(F)_4PAM ');
228 xlabel('F=frequency in Hz');
229 ylabel('Px(F)_4PAM=(|X(F)_4PAM|^2)/Ttotal');
230 legend('estimated', 'theoritical');
231
232 %comparing the S(X) 2PAM and S(X) 4PAM in plot
233 disp('press any key to see the S(X) 2PAM and S(X) 4PAM in the same plot')
234 pause
235 figure()
236 semilogy(F,S_x,F,S_x_4PAM);
237 title('P(X)_2PAM and P(X)_4PAM')
238 xlabel('F=frequency in Hz')
239 ylabel('P(X)_2PAM , P(X)_4PAM')
240 legend('2PAM','4PAM')
241 disp('press any key to see the S(X) 2PAM and S(X) 4PAM in the same plot with logarithmic y axis')
242 figure()
243 plot(F,S_x,F,S_x_4PAM);
244 title('P(X)_2PAM , P(X)_4PAM')
245 xlabel('F=frequency in Hz')
246 ylabel('P(X)_2PAM , P(X)_4PAM')
247 legend('2PAM','4PAM')
248 %=====
249 %A5
250 %initializing the new period to double, that means over is doubled too, and
251 %keeping everything else as it is
252 T=2*T;

```

```

253 over=2*over;
254 %creating srrc pulse
255 [phi,t]=srrc_pulse(T,over,A,a);
256 %creating fast fourier transformations
257 PHI=fftshift(fft(phi,Nf)*Ts);
258 %creating the energy density spectrum
259 edsPHI=abs(PHI).^2;
260
261 %creating a sequence of N bits
262 b=(sign(randn(N,1))+1)/2;
263 %call of the created function that transfers bits to 2PAM
264 Xn=bits_to_2PAM(b);
265 t1=(0:Ts:N*T-Ts);
266 %creating X_delta
267 X_delta=1/Ts*upsample(Xn,over);
268
269 %creating X(t)
270 %convolution of phi and X_delta
271 tconv=[t(1)+t1(1):Ts:t(end)+t1(end)];
272 X_t=conv(phi,X_delta)*Ts;
273 %creation of the graph of X(t)
274 disp('press any key to see X(t) waveform ')
275 pause
276 figure()
277 plot(tconv,X_t);
278 title(' X(t)=sum(Xn*phi(t-n*T)), T_2=2T')
279 ylabel(' X(t)')
280 xlabel('t=time in seconds')
281
282 %creating fast fourier transformation X(F) of X(t)
283 X_F=fftshift(fft(X_t,Nf)*Ts);
284 %the total duration time
285 Ttotal=length(tconv)*Ts;
286 %creation of a periodogram of one of the implementations of X(t)
287 P_x=abs(X_F).^2/Ttotal;
288
289 %creation of periodogram graph in plot
290 disp('press any key to see the periodogram')
291 pause
292 figure();
293 semilogy(F,P_x);
294 title('Periodogram with logarithmic y axis, T_2=2T');
295 xlabel('F=frequency in Hz');

```

```

296 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
297 legend('a = 0.5');
298
299 %creation of periodogram graph in logarithmic y axis
300 disp('press any key to see the periodogram with logarithmic y axis')
301 pause
302 figure();
303 plot(F,P_x);
304 title('Periodogram of X, T_2=2T');
305 xlabel('F=frequency in Hz');
306 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
307 legend('a = 0.5');
308
309 %initializing K the number of periodograms Px(F) of the of implemenations of
310 %X(t) and sum ,helps to find the summary
311
312 K =50;
313 sum=0;
314 %every loop we create a new implementation of X(t) and a new periodogram
315 %Px(F) for it, we repeat K times
316 for k=0:K-1
317     b=(sign(randn(N,1))+1)/2;
318     Xn=bits_to_2PAM(b);
319     X_delta=1/Ts*upsample(Xn,over);
320     X_t=conv(phi,X_delta)*Ts;
321     X_F=fftshift(fft(X_t,Nf)*Ts);
322     P_x=abs(X_F).^2/Ttotal;
323     sum=sum+P_x;
324 end
325 %creation of the estimated power density spectrum, we divide the summary by K times
326 %to find the arithmetic mean
327 S_x=sum/K;
328
329 %creating the theoritical Sx(F)
330 variance=1;
331 S_x_th= (variance*(abs(PHI).^2))/T;
332
333 %creating the graph of the estimated Sx(F) and the theoritical Sx(F) in the
334 %same plot with logarithmic y axis
335 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')
336 pause
337 figure();

```



```

338 semilogy(F,S_x,F,S_x_th);
339 title('estimated Sx(F) and theoritical Sx(F) smilogy');
340 xlabel('F=frequency in Hz');
341 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
342 legend('estimated', 'theoritical');
343 figure
344
345 plot(F,S_x,F,S_x_th);
346 title('estimated Sx(F) and theoritical Sx(F) plot');
347 xlabel('F=frequency in Hz');
348 ylabel('Px(F)=(|X(F)|^2)/Ttotal');
349 legend('estimated', 'theoritical');
350 %making the period and the over back to the staring ones
351 T=T/2;
352 over=over/2;
353 %=====
354 %B4
355 %initializing the modulation frequency f0 between 1/2T and (FS/2-1/2T)=>50<f0<450 and
356 % K number of waveforms
357 f0=300;
358 K = 4;
359
360 %creating srrc pulse
361 [phi,t]=srrc_pulse(T,over,A,a);
362 t1=(0:Ts:N*T-Ts);
363 variance=1;
364 %for every loop we make a different modulated waveform that comes from 2PAM
365 %its different , due to the change of theta and the change of X(t) sequence
366 for k=0:K-1
367     b=(sign(randn(N,1))+1)/2;
368     Xn=bits_to_2PAM(b);
369     X_delta=1/Ts*upsample(Xn,over);
370     X_t=conv(phi,X_delta)*Ts;
371     tconv=[t(1)+t1(1):Ts:t(end)+t1(end)];
372     %theta is a random variable evenly distributed
373     Theta=2*pi*rand;
374     %modulation
375     Y_t=X_t.*cos(2*pi*f0*tconv+Theta);
376     %making fast fourier transformation
377     Y_F=fftshift(fft(Y_t,Nf)*Ts);
378     absY=abs(Y_F);
379     %making the graphs ,we put absY , because Y has a complex part too, and
380     %we just want to see the meter

```

```

381     figure()
382     plot(F,absY)
383     xlabel('F=frequency in Hz')
384     title('the modulated :Y(F)=X(t)*cos(2*pi*f0+\theta)')
385     ylabel('Y(F)')
386 end
387
388 %initializing K number of periodograms , total duration time and the sums
389 %we use to create the Py(F) and Px(F)
390 K=100;
391 Ttotal=length(tconv)*Ts;
392 sumX=0;
393 sumY=0;
394 %every loop we make a different implementation of a periodogramm for the same Y(t)
395 %its different , due to the change of theta
396 for k=0: K-1
397     Theta=2*pi*rand;
398     Y_t=X_t.*cos(2*pi*f0*tconv+Theta);
399     Y_F=fftshift(fft(Y_t,Nf)*Ts);
400     absY=abs(Y_F);
401     P_y=absY.^2/Ttotal;
402     sumY=sumY+P_y;
403 end
404 %divide the summary by K so we can find the power density spectrum
405 S_y=sumY/K;
406
407 %creation of the graph of Sy(F) for plot an dplot with logarithmic y axis
408 disp('press an key to see the power density spectrum :Sy(F)in logarithmic y axis ')
409 pause
410 semilogy(F,S_y)
411 title('power density spectrum :Sy(F)in logarithmic y axis')
412 xlabel('Sy(F)')
413 ylabel('F=frequency in Hz')
414 disp('press any key to see the power density spectrum :Sy(F)')
415 pause
416 figure();
417 plot(F,S_y)
418 title('power density spectrum :Sy(F)')
419 xlabel('Sy(F)')
420 ylabel('F=frequency in Hz')

```