

## Úkol 1 - Synchronizace:

### Korelace lokální repliky

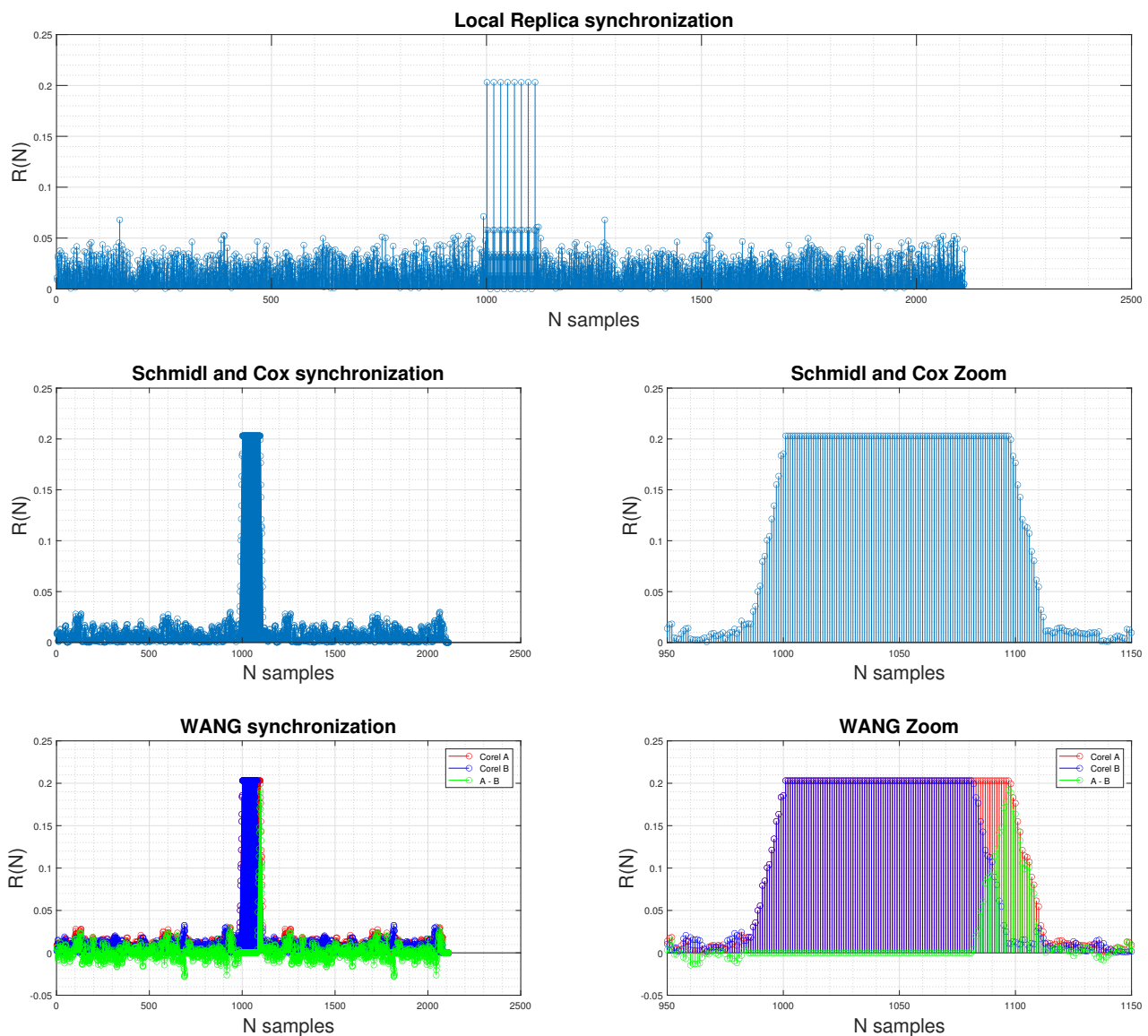
Na obrázku níže je patrných 8 korelačních špiček, to je způsobeno tím, že preamble má délku  $8 \cdot 16$  symbolů, přičemž lokální replika má délku 16 symbolů. Problémem této metody je, že koreluje signál zkreslený průchodem kanálu ze signálem bez zkreslení. To v případě zašumělého kanálu může zhoršit korelační vlastnosti.

### Schmidl a Cox:

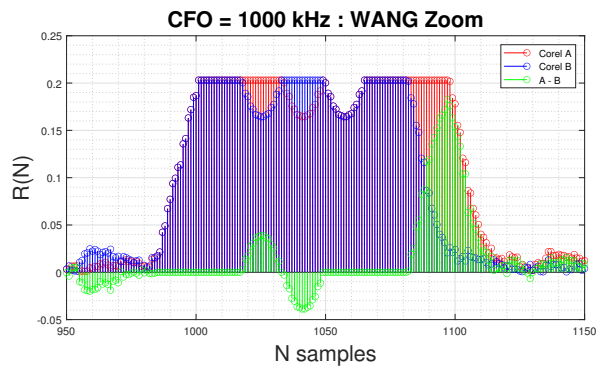
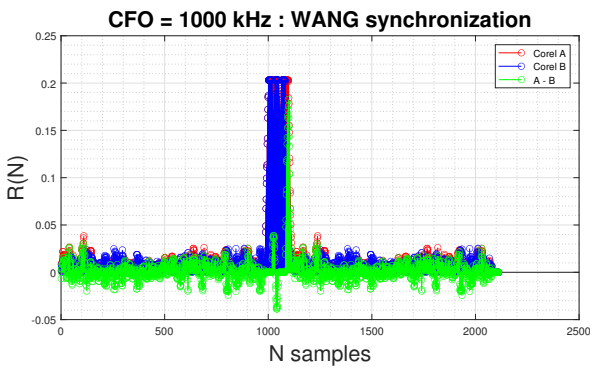
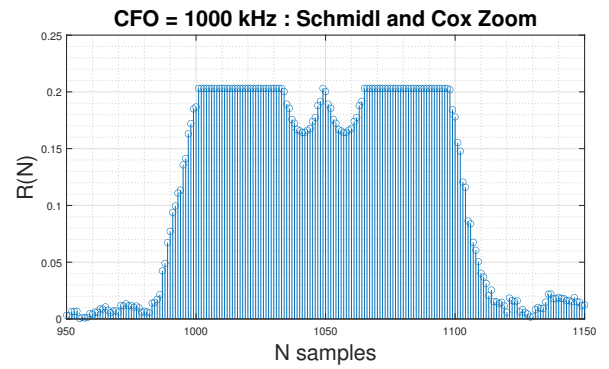
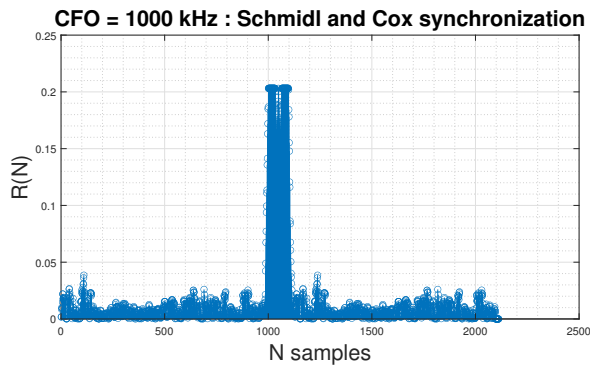
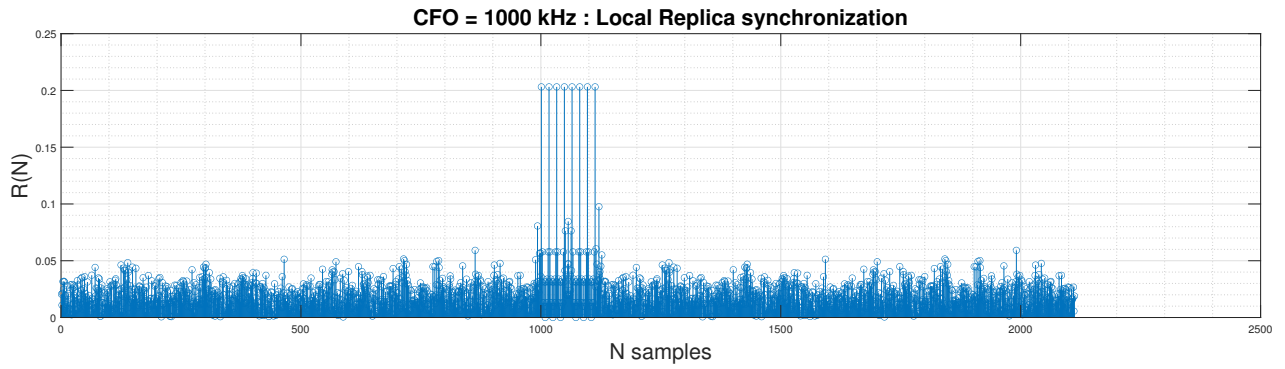
Metoda Schmidl a Cox, koreluje přijaté části signálů mezi sebou. Vzhledem k tomu, že všechny korelované části signálů prochází kanálem v krátkém časovém úseku mají mezi sebou lepší korelační vlastnosti než v případě metody pomocí lokální repliky. Nevýhodou této metody je, že vzniká v časové oblasti "plato" a nelze tak jednoznačně určit čas synchronizace. Pro jednoznačné určení, je potřeba zavést detekci sestupné hrany s tresholdem takovým, aby setupná hrana byla zachycena po ukončení oblasti s "platem".

### Schmidl a Cox:

Problém metody Schmidl a Cox řeší metoda WANG. Zde vzniká po korelaci špička a synchronizace je tak určena v bodě (čase) s nejvyšší korelací.

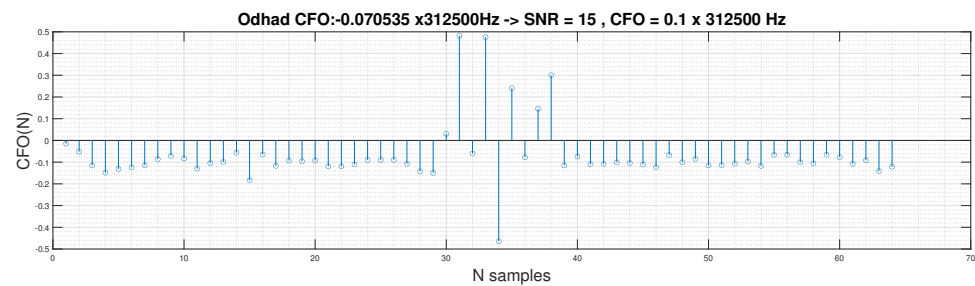
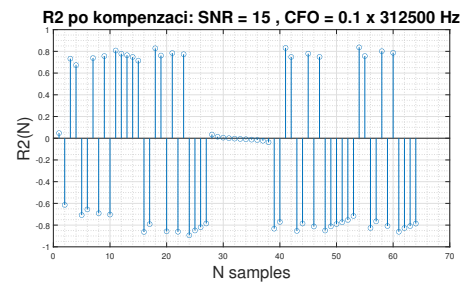
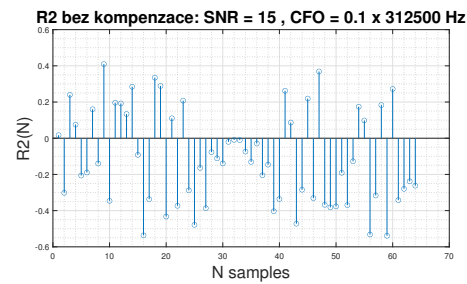
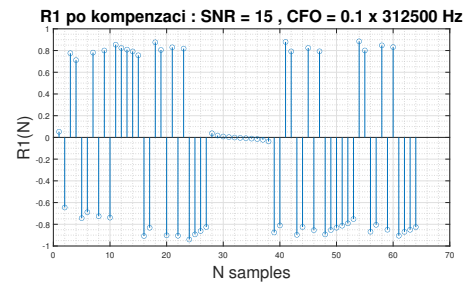
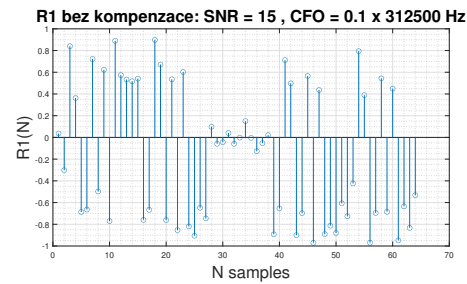
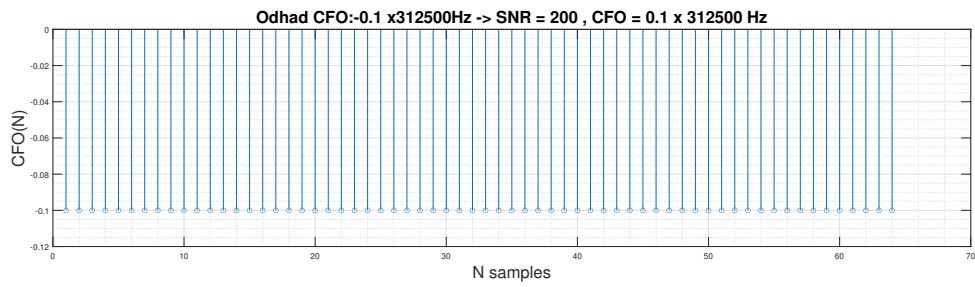
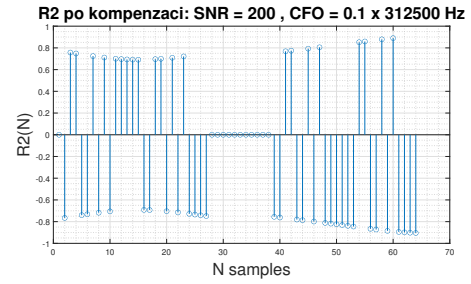
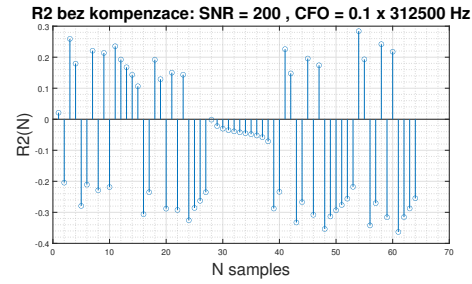
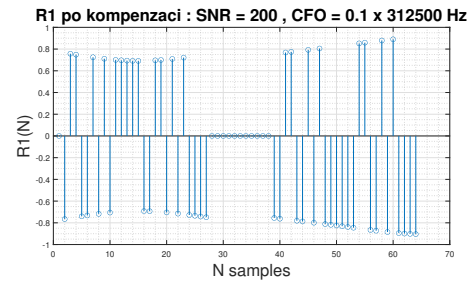
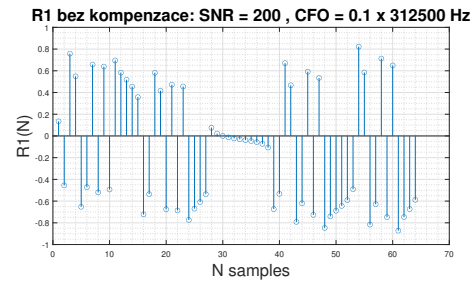


## 1 Úkol 2 - synchronizace s CFO:



## 2 Úkol 3 - Odhad CFO:

Na následujících obrázcích je znázorněn odhad CFO pomocí metody MOOSE. Při simulaci se projevuje i přenosový kanál, který není stejný pro oba zobrazené SNR. Z obrázků je patrné, že při horším SNR se zhorší i odhad CFO, kde dokonce pro některé vzorky má opačné znaménko než by se očekávalo.



## Přílohy:

Všechny scripty si můžete zobrazit v mojem github repozitáři

## IKS<sub>sync</sub>empty.m

```

1 clear variables
2 close all
3
4 %% figure settings
5 fontsize_axis = 20;
6 fontsize_ticks = 18;
7 fontsize_legend = 15;
8 fontsize_title = 20;
9 fontsize_linewidth = 3;
10 fontsize_marker = 15;
11
12 %%OFDM symbols
13 SIG_def=sqrt(13/6).*[0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0,...
14                     0, 0,-1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 0, 0,...
15                     0, 0,-1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0,...
16                     0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0];
17
18 %Input for IFFT
19 SIG_IFFT = [SIG_def(27), SIG_def(28:end),zeros(1,11),SIG_def(1:26)];
20 SIG_OUTPUT_IFFT=ifft(SIG_IFFT); % After IFFT
21
22 if 1
23     CFO = 1e6 %kHz
24     Ts = 1/20e6 %sample period
25     t = 0:Ts:length(SIG_OUTPUT_IFFT)*Ts-Ts;
26     SIG_OUTPUT_IFFT = SIG_OUTPUT_IFFT.*exp(1j*2*pi*CFO*t);
27     save_string = "sync_CFO"
28     title_string = sprintf("CFO = %d kHz : ", CFO/1e3)
29 else
30     title_string = ""
31     save_string = "sync"
32 end
33
34
35 %Rx signal [rnd_noise, SIG, SIG, rnd_noise]
36 rnd_noise = 0.05*randn(1,1000);
37 preamble = [SIG_OUTPUT_IFFT, SIG_OUTPUT_IFFT];
38 RX_SIG = [rnd_noise, preamble, rnd_noise];
39
40 %% CFO
41
42 %% Korelace s lok ln replikou
43 replica = preamble(1:16);
44 corel = zeros(1,length(RX_SIG)- length(replica));
45 for i = 1:length(RX_SIG)- length(replica)
46     corel(i) = dot(RX_SIG(i:i+15),replica);
47 end
48
49 %% Local replica Figure:
50 figure('Name','local_Replica','units','normalized','outerposition',[0 0 0.5 0.8]);
51 subplot(3,1,1)
52 stem(abs(corel))
53 title(title_string + "Local Replica synchronization","FontSize",fontsize_title)
54 xlabel("N samples","FontSize",fontsize_axis)
55 ylabel("R(N)","FontSize",fontsize_axis)
56 grid on
57 grid minor
58
59 %% Schmidl and Cox
60 corel = zeros(1,length(RX_SIG)- 16);
61 for i = 1:length(RX_SIG)- 2*16
62     corel(i) = dot(RX_SIG(i:i+15),RX_SIG(i+16: i+16 + 15));
63 end
64
65 %% SCHMIDL and COX Figure:
66 subplot(3,2,3)
67 stem(abs(corel))
68 title(title_string + "Schmidl and Cox synchronization","FontSize",fontsize_title)
69 xlabel("N samples","FontSize",fontsize_axis)
70 ylabel("R(N)","FontSize",fontsize_axis)
71 grid on
72 grid minor
73
74 subplot(3,2,4)

```

```

75 stem(abs(corel))
76 title( title_string + "Schmidl and Cox Zoom","FontSize",fontsize_title)
77 xlabel("N samples","FontSize",fontsize_axis)
78 ylabel("R(N)","FontSize",fontsize_axis)
79 xlim([950,1150])
80 grid on
81 grid minor
82
83 %% WANG
84 corel_A = zeros(1,length(RX_SIG)- 16);
85 corel_B = zeros(1,length(RX_SIG)- 16);
86 for i = 1:length(RX_SIG)- 3*16
87     corel_A(i) = dot(RX_SIG(i:i+15) ,RX_SIG(i+16: i+16 + 15));
88     corel_B(i) = dot(RX_SIG(i:i+15) ,RX_SIG(i+2*16: i+2*16 + 15));
89 end
90
91 subplot(3,2,5)
92 stem(abs(corel_A), 'r')
93 hold on
94 stem(abs(corel_B), 'b')
95 stem(abs(corel_A)-abs(corel_B), 'g')
96 title( title_string + "WANG synchronization","FontSize",fontsize_title)
97 xlabel("N samples","FontSize",fontsize_axis)
98 ylabel("R(N)","FontSize",fontsize_axis)
99 legend(["Corel A", "Corel B", "A - B"])
100 grid on
101 grid minor
102
103 subplot(3,2,6)
104 stem(abs(corel_A), 'r')
105 hold on
106 stem(abs(corel_B), 'b')
107 stem(abs(corel_A) - abs(corel_B), 'g')
108 title( title_string + "WANG Zoom","FontSize",fontsize_title)
109 xlabel("N samples","FontSize",fontsize_axis)
110 ylabel("R(N)","FontSize",fontsize_axis)
111 xlim([950,1150])
112 legend(["Corel A", "Corel B", "A - B"])
113 grid on
114 grid minor
115 saveas(gcf,save_string,"eps")

```

### *IKS<sub>synchroniz<sub>e</sub>empty.m</sub>*

```

1 clear variables;
2 close all;
3
4 %% figure settings
5 fontsize_axis = 20;
6 fontsize_ticks = 18;
7 fontsize_legend = 15;
8 fontsize_title = 20;
9 fontsize_linewidth = 3;
10 fontsize_marker = 15;
11
12
13 L = [1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0,...
14 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1];
15
16 SIG_ref = [ L(28:end),L(1:26)];
17
18
19 %% Generovani kanalove fce
20 H = ChannelModel;
21 H_vyrez = H(1,1:52);
22 H = H_vyrez;
23
24 SIG = SIG_ref.* H(1,:);
25
26 %Channel estimation
27 H_LS_perfect = SIG./SIG_ref;
28
29 %OFDM modulation
30 sig = ifft([L(27), SIG(1:26),zeros(1,11),SIG(27:end)]);
31
32 s_dlouhy = [sig(end-32+1:end),sig,sig]; % dlouha preamble s CP [GI2 | T1 | T2]

```

```

33
34 %% simulace CF0:
35 Ts = 50e-9;
36 T = length(sig)*Ts;
37 dF = 1/T;
38 cfo_rel = 0.1
39 cfo = cfo_rel*dF;
40 t = 0:Ts:length(s_dlouhy)*Ts-Ts;
41 srec = s_dlouhy.*exp(j*2*pi*cfo*t);
42
43 %SREC AWGN
44 AWGN = 200
45 srec = awgn(srec,AWGN,'measured');
46
47 % FFT
48 R1 = fft( srec(33: 33+length(sig)-1) )
49 R2 = fft( srec(end+1-length(sig) :end) )
50
51 %odhad CF0
52 CF0_est = angle(R1.*conj(R2))/(2*pi)
53 CF0_comp = mean(CF0_est)*dF
54
55 %kompenzace
56 srec_comp = s_dlouhy.*exp(j*2*pi*cfo*t).*exp(j*2*pi*CF0_comp*t)
57
58 %Equalizace kanalu
59 R1_comp = fft( srec_comp(33: 33+length(sig)-1) );
60 R2_comp = fft( srec_comp(end+1-length(sig) :end) );
61
62 title_string = sprintf (" SNR = %d , CF0 = %0.1f x %d Hz",AWGN,cfo_rel,dF)
63 %% jak vypada preamble bez korekce CF0:
64 figure('Name','CF0','units','normalized','outerposition',[0 0 0.5 0.8]);
65 subplot(3,2,1)
66 stem(R1)
67 title( "R1 bez kompenzace:" + title_string,"FontSize",fontsize_title)
68 xlabel("N samples","FontSize",fontsize_axis)
69 ylabel("R1(N)","FontSize",fontsize_axis)
70 grid on
71 grid minor
72
73 subplot(3,2,2)
74 stem(R1_comp)
75 title( "R1 po kompenzaci :" + title_string,"FontSize",fontsize_title)
76 xlabel("N samples","FontSize",fontsize_axis)
77 ylabel("R1(N)","FontSize",fontsize_axis)
78 grid on
79 grid minor
80
81 subplot(3,2,3)
82 stem(R2)
83 title( "R2 bez kompenzace:" + title_string,"FontSize",fontsize_title)
84 xlabel("N samples","FontSize",fontsize_axis)
85 ylabel("R2(N)","FontSize",fontsize_axis)
86 grid on
87 grid minor
88
89 subplot(3,2,4)
90 stem(R2_comp)
91 title( "R2 po kompenzaci:" + title_string,"FontSize",fontsize_title)
92 xlabel("N samples","FontSize",fontsize_axis)
93 ylabel("R2(N)","FontSize",fontsize_axis)
94 grid on
95 grid minor
96
97 %% odhad CF0:
98 subplot(3,1,3)
99 stem(CF0_est)
100 title( "Odhad CF0:" +CF0_comp/dF + " x" +dF+ "Hz ->" + title_string,"FontSize",fontsize_title)
101 xlabel("N samples","FontSize",fontsize_axis)
102 ylabel("CF0(N)","FontSize",fontsize_axis)
103 grid on
104 grid minor
105
106 %saveas(gcf,"H_eq_200", "eps")

```