

Laborationsrapport i TSKS10 *Signaler, Information och Kommunikation*

Martin Söderén
marso329, 9009291098

June 12, 2015

1 Inledning

Den här laborationen gick ut på analysera en delvis okänd signal. En del saker om den var givna men en del som krävdes för att kunna lyssna på signalen behövdes bestämma analytiskt. Signalen ser ut på följande sätt:

$$x(t) = x_I(t) \cos(2\pi f_c t) - x_Q(t) \sin(2\pi f_c t) + z(t)$$

1.1 Givet

- Bärfrekvensen f_c är en multipel av 19 kHz
- Signalen studsar mot ett objekt så ett eko uppstår. Tiden τ_1 är tiden det tar från sändaren till mottagaren direkt. Tiden τ_2 är tiden det tar från sändare till objekt och sedan till mottagare.
- $\tau_2 > \tau_1$ samt $\tau_2 - \tau_1$ är en multipel av 1 ms.
- Första tredjedelen av $x_I(t)$ och $x_Q(t)$ innehåller varsin melodi.
- Andra tredjedelen av $x_I(t)$ och $x_Q(t)$ innehåller var sitt ordspråk.
- Sista tredjedelen av $x_I(t)$ och $x_Q(t)$ innehåller vitt brus.
- Sändaren sänder samtidigt ut andra I/Q-modulerade signaler på andra bärfrekvenser. Dessa betecknas $z(t)$.
- Ekot gör att vi tar emot $y(t) = x(t - \tau_1) + 0,9x(t - \tau_2)$

- Signalen tas emot lågpasfilterad med ett idealt lågpasfilter
- Samplingsfrekvensen är 400 kHz.

2 Metod

Uppgiften delades upp i tre mindre uppgifter:

1. Ta reda på bärfrekvensen f_c
2. Ta reda på tidsfördröjningen $\tau_2 - \tau_1$
3. I/Q modulera signalen

2.1 Bärfrekvens

För att få fram bärfrekvensen så fouriertransformerades signalen och sedan plottades frekvensspektrumet. Detta kan ses i figur 1 under bilagor. Här så ser man tre tydliga toppar. Dessa är vid 39200, 95000 och 133000 Hz.

Signalen filtreras för att få ut respektive signal med varsitt butterworthfilter. Signalerna plottas mot tiden och genom att mäta längden på datasignalen så ser man att det finns 7800000 sampel. Då samplingsfrekvensen var 400 kHz så är signalen 19.5 sekunder lång. Signalen med toppen vid 39200 Hz kan ses i figur 2, signalen med toppen vid 95000 Hz kan ses i figur 3 och signalen med toppen vid 133000 Hz kan ses i figur 4.

Den första signalen ser ut att bara innehålla vitt brus. Den andra innehåller någonting periodiskt som

troligtvis inte är intressant. Den tredje ser ut att innehålla användbar information så bärfrekvensen f_c ansätts till 133 kHz.

2.2 Tidsfördröjning

För att få fram tidsfördröjning i signalen så autokorreleras det vita bruset som var den första signalen med toppen vid 39200 Hz. I figur 5 så kan man se en topp vid 19.5 s och två sidotoppar ± 0.41 s om toppen. Detta ger att $\tau_2 - \tau_1 = 0.41$. Denna metod beskrivs och förklaras i kursboken kapitel 4.

2.3 Filtrera bort eko

Då tidsfördröjning och ekots amplitud nu är kända kan ekot filtreras bort. Detta görs genom att $x(t) = y(t) - 0.9y(t - \tau)$ för alla $t > \tau$. I implementationen sker detta dock i block istället för varje enskilt sampel för att skynda på exekveringen. Efter filtreringen är bara den ursprungliga signalen kvar.

2.4 I/Q-demodulering

Signalen I/Q-demoduleras på följande sätt från kursboken avsnitt 1.4.2:

$$x_I(t) = \mathcal{H}_{B/2}^{LP}\{2x(t) \cos(2\pi f_c t)\}$$

$$x_Q(t) = -\mathcal{H}_{B/2}^{LP}\{2x(t) \sin(2\pi f_c t)\}$$

Där B är signalens bandbredd men det är känt att signalen bara ska innehålla tal så signalen lågpasfiltreras med gränsfrekvensen 20 kHz istället. Bärfrekvensen f_c är känd från avsnitt 2.1 och är 133 kHz.

Detta ger två signaler som spelas upp efter nedsampling. Matlab kunde inte spela upp ljud i 400 kHz så signalerna nedsamlades till 100 kHz.

3 Resultat

Den sökta informationen är:

- Bärfrekvensen f_c är 133 kHz
- Tidsfördröjningen $\tau_2 - \tau_1$ är 410 ms.

- Ena delen innehöll först ett pianostycke, sedan ordspråket "Den som ger sig in i leken får leken tåla" och sedan vitt brus
- Andra delen innehöll först ett pianostycke, sedan ordspråket "Även små grytor har öron" och sedan vitt brus.

3.1 Anmärkning

Om det uppkommer en fasförskjutning på $\frac{\pi}{2}$ så kan $x_I(t)$ och $x_Q(t)$ byta plats. Så man kan inte vara säker på att signalen som demodulerades till $x_I(t)$ var signalen som modulerades och sändes som $x_I(t)$.

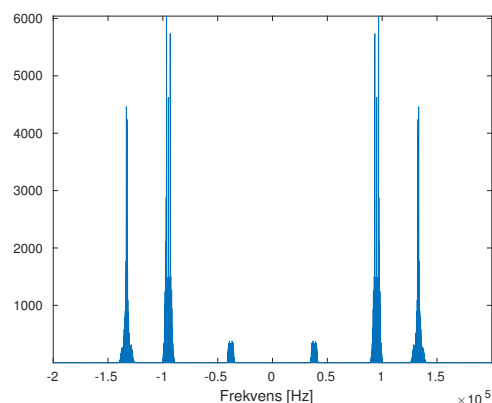


Figure 1: Fouriertransform av signalen.

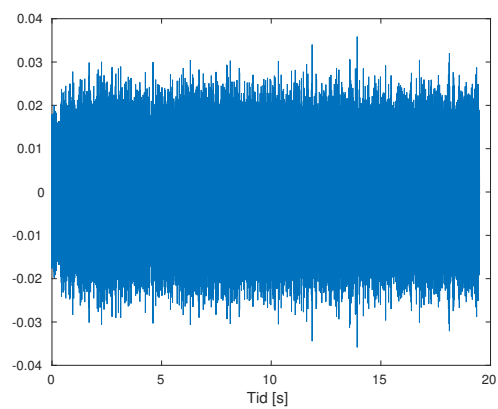


Figure 2: Första signalen.

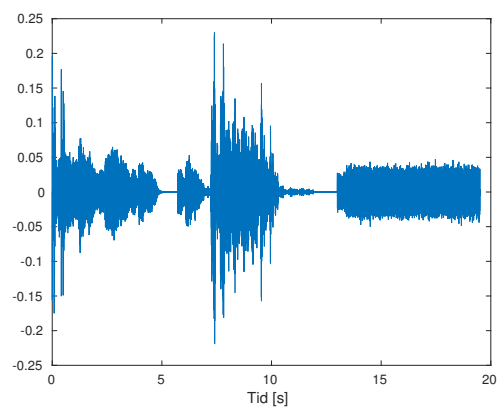


Figure 4: Tredje signalen.

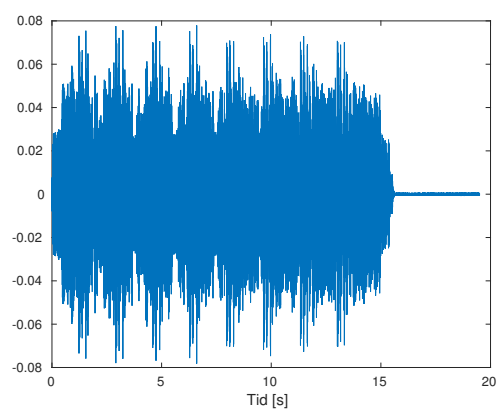


Figure 3: Andra signalen.

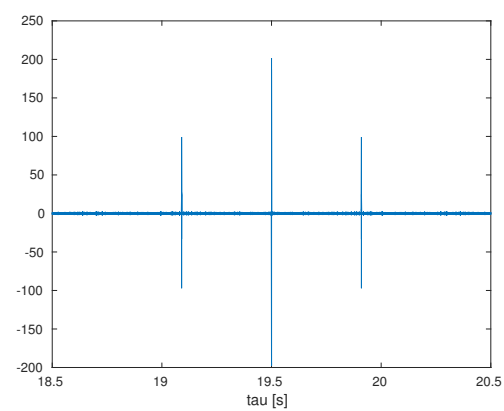


Figure 5: Autokorrelationen av det vita bruset

Min Matlab-kod:

*%Right click on signal-marso329.wav and choose import data. A window will
%appear and click on import in the upper right corner*

%some constants

```
number_of_samples=length(data);  
sampling_frequency=fs;
```

%plot frequency spectrum

```
X_mags = abs(fftshift(fft(data)));  
bin_vals = 0 : number_of_samples-1;  
N_2 = ceil(number_of_samples/2);  
fax_Hz = (bin_vals-N_2)*sampling_frequency/number_of_samples;  
figure(10);  
plot(fax_Hz, X_mags)  
xlabel('Frekvens [Hz]')  
axis tight
```

%peaks at 39200(0,70000), 95000(70000,115000),133000(115000,20000)

%we have 7800000 samples and at samplefreq 400000 we have 19.5 s of data

```
time_axis = 0:19.5/number_of_samples:19.5;  
time_axis=time_axis(1:7800000);
```

% First peak through lowpassfilter

```
[B,A] = butter(10,0.35,'low');  
first_peak_filtered = filter(B,A,data);  
figure(2);  
plot(time_axis, first_peak_filtered);  
xlabel('Tid [s]');
```

% Second peak through bandpassfilter

```
[B,A] = butter(10,[0.35, 0.60],'bandpass');  
second_peak_filtered = filter(B,A,data);  
figure(3);  
plot(time_axis, second_peak_filtered);  
xlabel('Tid [s]');
```

% Third peak through highpassfilter

```
[B,A] = butter(10,0.6,'high');  
third_peak_filtered = filter(B,A,data);  
figure(4);  
plot(time_axis, third_peak_filtered);  
xlabel('Tid [s]');
```

```

%carrier freq is 133k Hz
carrier_frequency=133000;

%cross-correlation in white noise
first_peak_cross_correlation = xcorr(first_peak_filtered , first_peak_filtered);
time_axis_cross = 0:39/15599999:39;
time_axis_cross=time_axis_cross(1:end-1);
figure(5);
plot(time_axis_cross , first_peak_cross_correlation);
xlabel('tau [s]');
axis([18.5,20.5,-200,250]);

%largest value at tau=19.5 and sidetops +-0.41
%0.41*400000=164000 samples
echo_samples=164000;
filtered_data = zeros(size(third_peak_filtered));
filtered_data(1:echo_samples) = third_peak_filtered(1:echo_samples);

%remove echo
for i = 0 : 45
temp1 = filtered_data((1+echo_samples*i):(echo_samples + echo_samples*i));
temp2 = third_peak_filtered((echo_samples+1+echo_samples*i):(i+2)*echo_samples);
filtered_data((echo_samples+1+echo_samples*i):(i+2)*echo_samples) = temp2 - 0.9*temp1;
end

%demodulation
I=((2*cos(2*pi*carrier_frequency*time_axis))').* filtered_data;
Q=(-2*sin(2*pi*carrier_frequency*time_axis))').* filtered_data;

% filter I and Q on the hearable spectrum
[B,A] = butter(10,0.10,'low');
i_filtered = filter(B,A,I);
q_filtered = filter(B,A,Q);

%downsample
q_audio=decimate(q_filtered , 4);
i_audio=decimate(i_filtered , 4);

%Uncomment which one you want to listen to
%soundsc(q_audio,100000)
soundsc(i_audio,100000)

```