Laborationsrapport i TSKS10 *Signaler, Information och Kommunikation*

Matti Lundgren
matlu703, xxyyzz-åååä

19 maj, 2017

1 Inledning

Laborationen går ut på att demodulera en smalbandig, I/Q-modulerad signal som består av hörbart ljud. Det är givet att den utsända signalen har utseendet $x(t) = x_I(t)cos(2\pi f_c t) - x_Q(t)sin(2\pi f_c t) + w(t) + z(t)$ där $w(t) = 0.001(cos(2\pi f_1 t) + cos(2\pi f_2 t))$ och $z(t)$ är en summa av signaler som sänds på andra bärfrekvenser.

På grund av ekoeffekter blir den mottagna signalen $y(t) = x(t - \tau_1) + 0,9x(t - \tau_2)$. Termerna är förskjutna med τ_1 sekunder då det är tiden det tar för signalen att nå mottagaren. Den andra termen är resultatet av ekot som orsakar en energiförlust på 10% och som bidrar med en tidsförskjutning på τ_2 - τ_1 sekunder.

Annan relevant information given av laborationshanledningen listas nedan:

- Bärfrekvensen f_c är någon av följande frekvenser: 18, 37, 56, 75, 94, 113, 132, 151 kHz.
- f_1 och f_2 är multiplar av 1 Hz och överlappar inte med någon av de burna signalerna.
- $\tau_2 > \tau_1$, $\tau_2 - \tau_1 < 500$ ms och $\tau_2 - \tau_1$ är en multipel av 1 ms.
- Då signalerna ska bestå av hörbart ljud sätts deras intressanta bandbredd till $B = 20$ kHz.

2 Metod

För att genomföra laborationen användes, utöver laborationshandledningen, kursboken *Signals, Information and Communications* (utkast 14273) och Matlab. Programkoden bifogas på sida 4 och de plottade figurerna på sida 3.

$y(t)$s amplitudspektrum plottas i figur 1. Ur spektrumet identiferas tre möjliga bärfrekvenser; 56, 94 och 151 kHz. De burna signalerna benämns hädanefter som $y_1(t)$, $y_2(t)$ respektive $y_3(t)$. Signalerna erhålls genom att bandpassfiltrera $y(t)$ runt respektive bärfrekvens med bandbredden B. De plottas i figur 2 där $y_1(t)$ ser ut att vara nyttosignalen.

$w(t)$ ser ut att existera runt 46,5 kHz, därför plottas $y(t)$s amplitudspektrum runt denna frekvens, se figur 3. Då det är känt att f_1 och f_2 är multiplar av 1 Hz går det att dra slutsatsen att $f_1 = 46500$ Hz och att $f_2 = 46501$ Hz.

För att ta reda på tidsfördröjningen $\tau_2 - \tau_1$ används tidsestimering med korskorrelation som beskrivs av kursbokens fjärde kapitel. $y(t)$ korskorrelerat med sig själv benämns som $r(\tau)$ och plottas i figur 4. Tidsfördröjningen ges av det τ som bortsett från den första toppen ger $r(\tau)$ sitt största värde. Med hjälp av figur 4 identifieras det sökta τ till att vara 430 ms vilket är troligt då det är känt att $\tau_2 - \tau_1 < 500$ ms och att differensen är en multipel av 1 ms. Slutsatsen blir att τ_0 är tidsfördröjningen orsakat av ekot, det vill säga $\tau_0 = \tau_2 - \tau_1 = 430$ ms.

Ekot annulleras med hjälp av ekvationen $\hat{y}(t) = y(t) - 0,9\hat{y}(t - \tau_0)$ för $t > \tau_0$. Ekvationen för $\hat{y}(t)$ härleds ur ekvationen för $x(t)$ genom att bortse från tidsfördröjningen τ_1 som orsakas av avståndet mellan sändare och mottagare. Notera att $\hat{y}(t)$ är den ekofria signalen och att $y(t)$ är fri från eko då $t < \tau_0$. Då den sökta signalen tros vara $y_1(t)$ behandlas endast den.

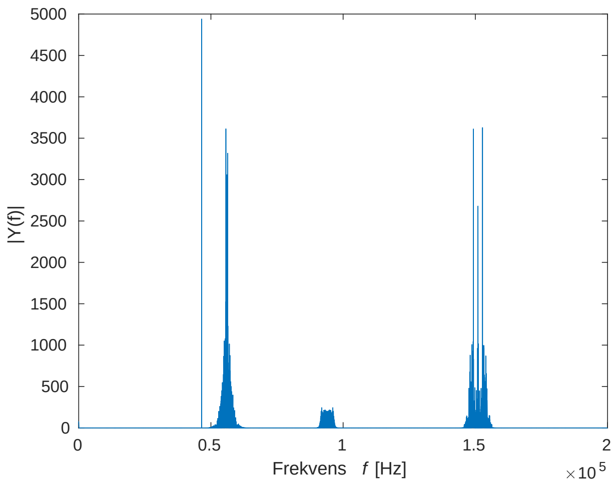
Nu återstår demoduleringen. Ur kursbokens andra kapitel framgår det att $\hat{y}_I(t) = H^{LP}_{B/2}\{2\hat{y}(t)cos(2\pi f_c t + \delta)\}$ och att $\hat{y}_Q(t) = H^{LP}_{B/2}\{-2\hat{y}(t)sin(2\pi f_c t + \delta)\}$. Då $\hat{y}(t)$ i det här fallet är en ekofri variant av $y_1(t)$ är bärfrekvensen $f_c = 56$ kHz. Signalen lågpassfiltreras med lägre bandbredd än $B/2$ för att minimera oönskat ljud. En fasvridning kan behöva kompenseras för, den representeras av δ i demoduleringsekvationerna. Efter prövning sätts $\delta = 1.1$ radianer vilket verkar minimera de talade ordspråkens överlapp.

3 Resultat

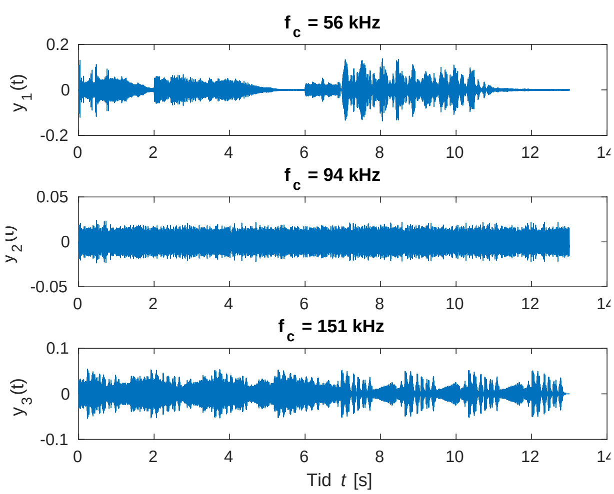
Den sökta informationen är:

- Bärfrekvensen för nyttosignalen är $f_c = 56$ kHz.
- Sinusoidernas frekvenser i $w(t)$ är $f_1 = 46500$ Hz och $f_2 = 46501$ Hz.
- Ekots bidrag till tidsfördröjningen är $\tau_2 - \tau_1 = 430$ ms.
- Ordspråket i $x_I(t)$ är “Inget ont som inte har något gott med sig”.
- Ordspråket i $x_Q(t)$ är “Även den mest skröpliga mussla kan innehålla en pärla”.

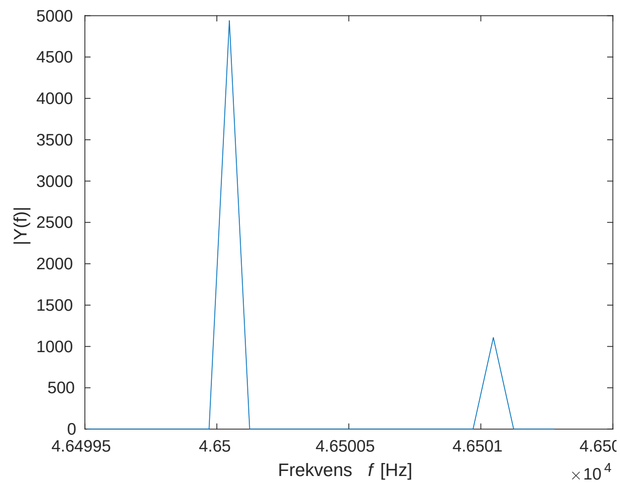
Figurer



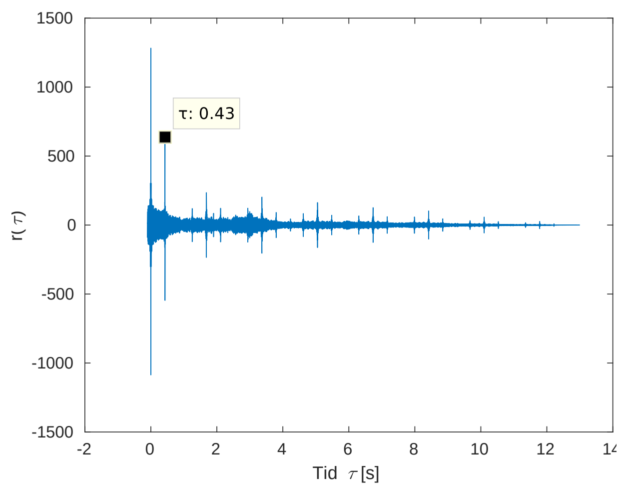
Figur 1: Amplitudspektrum för $y(t)$



Figur 2: De tre identifierade delsignalerna



Figur 3: Amplitudspektrum för $y(t)$ runt 46.5 kHz



Figur 4: $y(t)$ korskorrelerat med sig själv med betoning på τ för den högsta sidotoppen

Matlab-kod

```matlab
clear;

[y, Fs] = audioread('signal-extra012.wav');
T = 1/Fs;         % Sampling period
B = 20000;        % Bandwidth (hearable sound)
N = length(y);   % Number of samples

% 1. Identify fc
absY = abs(fftshift(fft(y)));
faxis = linspace(-0.5, 0.5, N)*Fs;
freqs = (faxis >= 0);
plot(faxis(freqs), absY(freqs));
xlabel('Frekvens {\itf} [Hz]')
ylabel('|Y(f)|')

Fc = [56000, 94000, 151000]; % From the plot

taxis = linspace(0, T*N, N);
[b, a] = butter(10, [Fc(1)-B, Fc(1)+B]/(Fs/2));
y_1 = filter(b, a, y);
[b, a] = butter(10, [Fc(2)-B, Fc(2)+B]/(Fs/2));
y_2 = filter(b, a, y);
[b, a] = butter(10, [Fc(3)-B, Fc(3)+B]/(Fs/2));
y_3 = filter(b, a, y);

figure;
subplot(3, 1, 1);
plot(taxis, y_1);
title('f_{c} = 56 kHz'); % Looks like the sought signal
ylabel('y_{1}(t)');
subplot(3, 1, 2);
plot(taxis, y_2);
title('f_{c} = 94 kHz');
ylabel('y_{2}(t)');
subplot(3, 1, 3);
plot(taxis, y_3);
title('f_{c} = 151 kHz');
ylabel('y_{3}(t)');
xlabel('Tid {\itt} [s]');

% 2. Identify f1 and f2 of w(t)
figure;
freqs = (faxis > 4.64995e4) & (faxis < 4.65013e4);
plot(faxis(freqs), absY(freqs));
ylabel('|Y(f)|');
xlabel('Frekvens {\itf} [Hz]');

% 3. Identify t2-t1
[r, lags] = xcorr(y);
lags = lags/Fs;
time = (lags >= -0.1);
figure;
plot(lags(time), r(time));
ylabel('r(\tau)');
xlabel('Tid {\it\tau} [s]');
tau = 0.430; % From the plot

% 4. Remove echo from y1
N_tau = uint32(tau / T); % Number of samples in time period tau
for i = N_tau + 1 : N
    y_1(i) = y_1(i) - 0.9 * y_1(i - N_tau);
end

% 5. Demodulate y1
[b, a] = butter(10, 5000/(Fs/2), 'low');
d = 1.1; % Deduced by trial and error listening to the sounds
I = 2*cos(2*pi*Fc(1)*taxis + d)';
Q = 2*sin(2*pi*Fc(1)*taxis + d)';
y_1I = filter(b, a, y_1.*I);
y_1Q = -1*filter(b, a, y_1.*Q);
y_1I = decimate(y_1I, 10);
y_1Q = decimate(y_1Q, 10);

pause;
close all;

% 6. Play each sound using the commands below
% soundsc(y_1I, Fs/10);
% soundsc(y_1Q, Fs/10);
```