Appendix till kapitel 5

Amplitudspektrum för tidsserier

Om vi samplar den kontinuerliga signalen $x(t)$ vid tidpunkterna $t = 0, \Delta t, 2\cdot\Delta t, \ldots, (N-1)\Delta t$ erhålls tidsserien $x_0, x_1, x_2, \ldots, x_{N-1}$.

Amplitudspektrum för tidsserien definieras som $|X_k|$, där

$$X_k = \frac{1}{N}\cdot\sum_{j=0}^{N-1} x_j \cdot e^{-2\pi i jk/N}, \qquad (1)$$

för $k = 0, 1, \ldots, N-1$. Vi kan med andra ord beräkna amplitudspektrum med FFT.

$|X_k|$ svarar mot frekvensen

$$f_k = \frac{k}{N\cdot\Delta t}, \quad k = 0, \ldots, N-1. \qquad (2)$$

Tidsseriens totala längd $N\cdot\Delta t$ bestämmer frekvensupplösningen:

$$\Delta f = \frac{1}{N\Delta t}, \qquad (3)$$

som anger hur nära i frekvens två signalkomponenter kan ligga för att ännu upptäckas.

Samplingsintervallet Δt ger samplingsfrekvensen $1/\Delta t$ och nyquistfrekvensen:

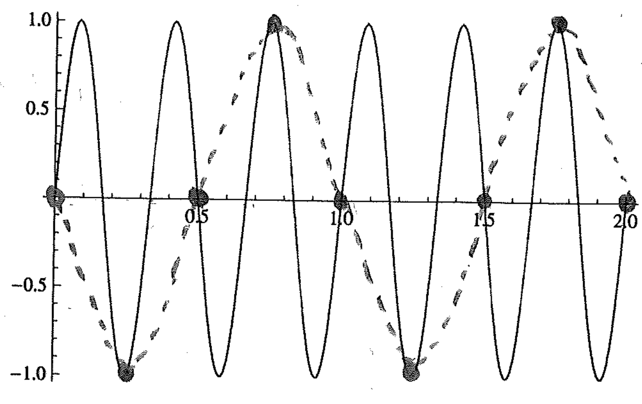
$$f_{ny} = \frac{1}{2\Delta t}, \qquad (4)$$

som är halva samplingsfrekvensen.

Samplingsteoremet. Om en kontinuerlig signal $x(t)$ inte innehåller frekvenser högre än f_{ny}, så förloras ingen information vid samplingen.

Om signalen $x(t)$ innehåller frekvenser över f_{ny} kan man inte skilja dessa från frekvenser under f_{ny}. ~~Fenomenet~~ kallas aliasing:

```
f[t_] := Sin[6 Pi t]
Plot[f[t], {t, 0, 2}]
```



$$\text{Samplingsfrekvens} = \frac{1}{\Delta t} = \frac{1}{1/4} = 4$$

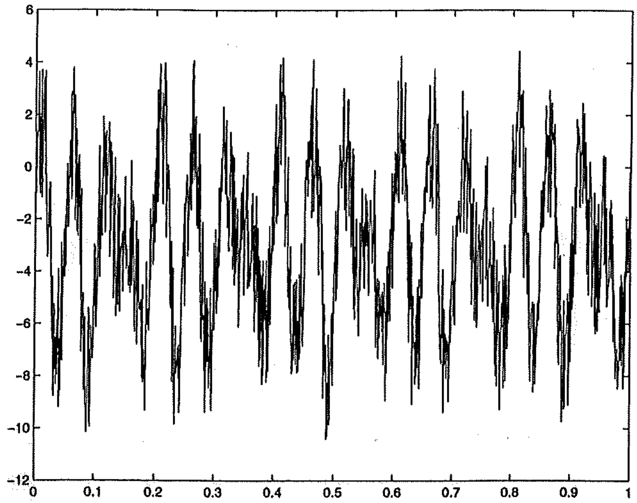
$$\text{nyquistfrekvens} = \frac{1}{2\Delta t} = 2 = f_{ny}$$

$$\text{Signalens } f(t) \text{ frekvens } f = 3 > f_{ny}$$

Få den "falska frekvensen" $2 \cdot f_{ny} - f = 4 - 3 = 1$.

Exempel. I matlab genereras en signal med frekvenserna 10, 15 och 20 och slumpmässiga störningar adderas till

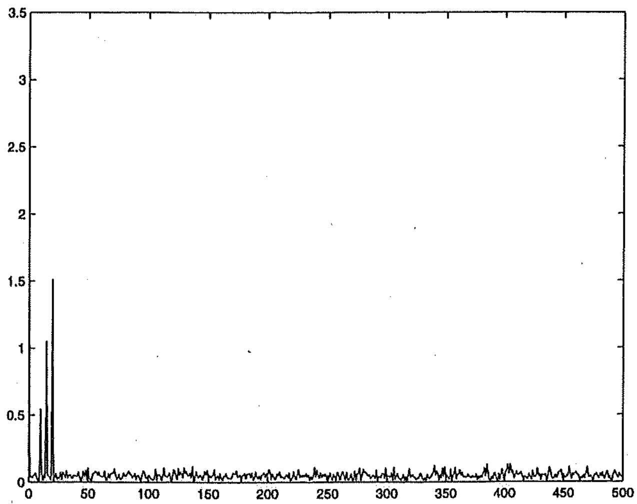
```
>> N=1000;t=[0:1/N:1-1/N];
>> x=sin(10*2*pi*t)+2*cos(15*2*pi*t)+3*sin(20*2*pi*t);
>> x=x+6*(rand(1,N)-1);
>> plot(t,x)
```



Vi beräknar amplitudspektrum med FFT och plottar ut det till nyquistfrekvensen = 500. (Amplitud-spektret är symmetriskt kring fny).

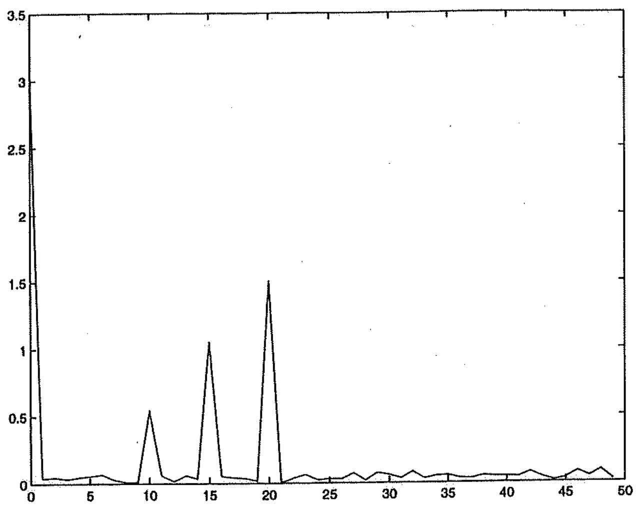
```
>> X=fft(x)/N;

>> plot((0:499),abs(X(1:500)))
```



Vi granskar närmare frekvensintervallet [0, 50]:

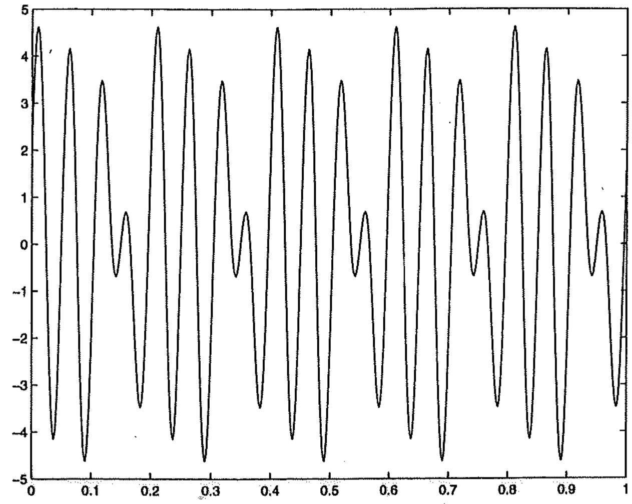
```
>> plot((0:49),abs(X(1:50)))
```



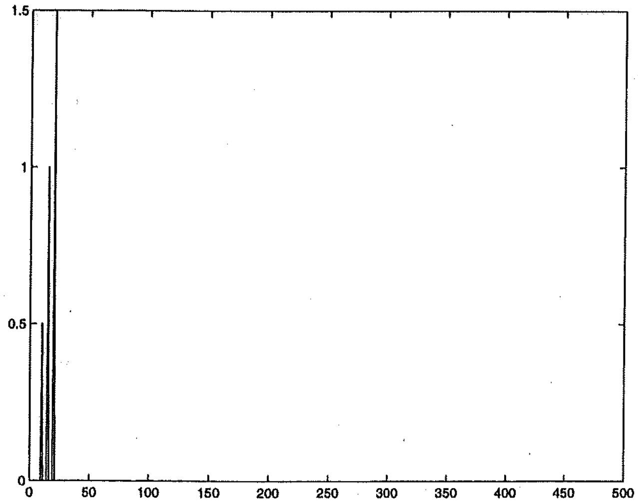
Slutsatsen blir att signalen innehåller störningar och tre periodiciteter med frekvenserna 10, 15 och 20.

En signal med högfrekvent "brus" borde köras igenom ett "lågpassfilter" innan man beräknar amplitudspektrum, för att få en tillförlitligare bild av frekvensinnehållet. I vårt exempel var detta inget problem, jämför med den "brusfria" signalen på nästa sida.

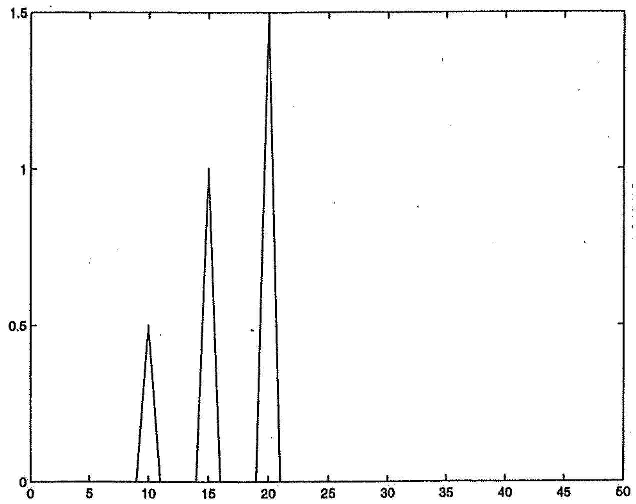
```
>> N=1000;t=[0:1/N:1-1/N];
>> x=sin(10*2*pi*t)+2*cos(15*2*pi*t)+3*sin(20*2*pi*t);
>> plot(t,x)
```




```
>> X=fft(x)/N;
>> plot((0:499),abs(X(1:500)))
```



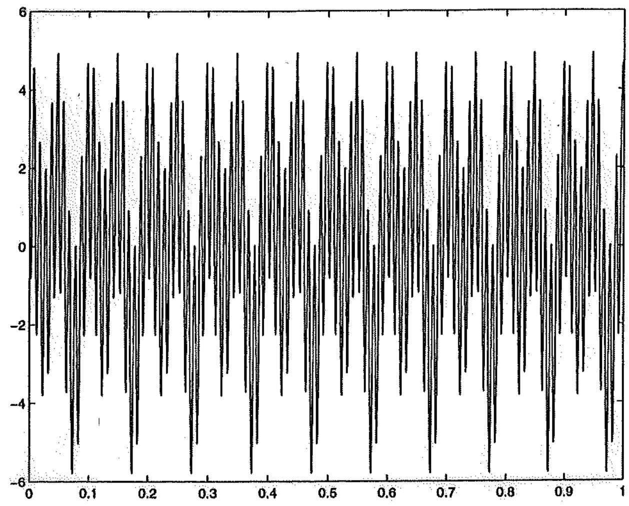

```
>> plot((0:49),abs(X(1:50)))
```



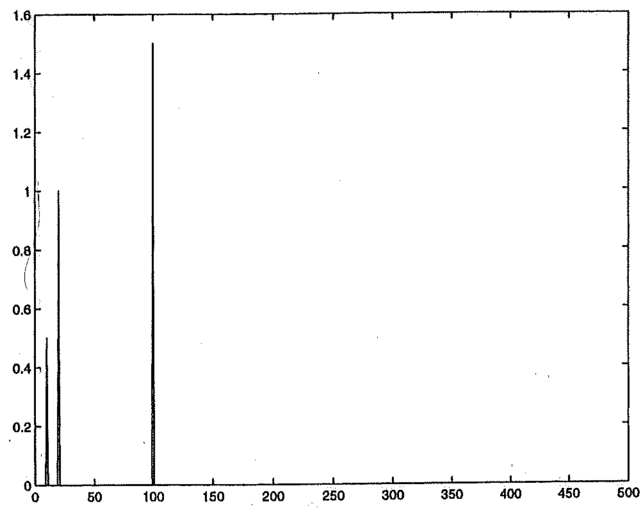
Signal med frekvenserna 10, 20 och 900.

$900 > f_{ny} = 500$.

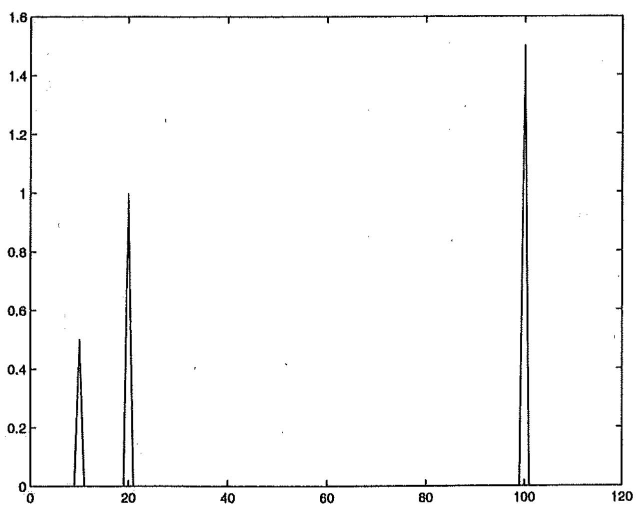
```
>> N=1000;t=[0:1/N:1-1/N];
>> x=sin(10*2*pi*t)+2*cos(20*2*pi*t)+3*sin(900*2*pi*t);
>> plot(t,x)
```



```
>> X=fft(x)/N;
>> plot((0:499),abs(X(1:500)))
```



```
>> plot((0:109),abs(X(1:110)))
```



Erhåller "falsk" frekvens = 100.

$2 \cdot f_{ny} - 900 = 100$.

"aliasing effekten".

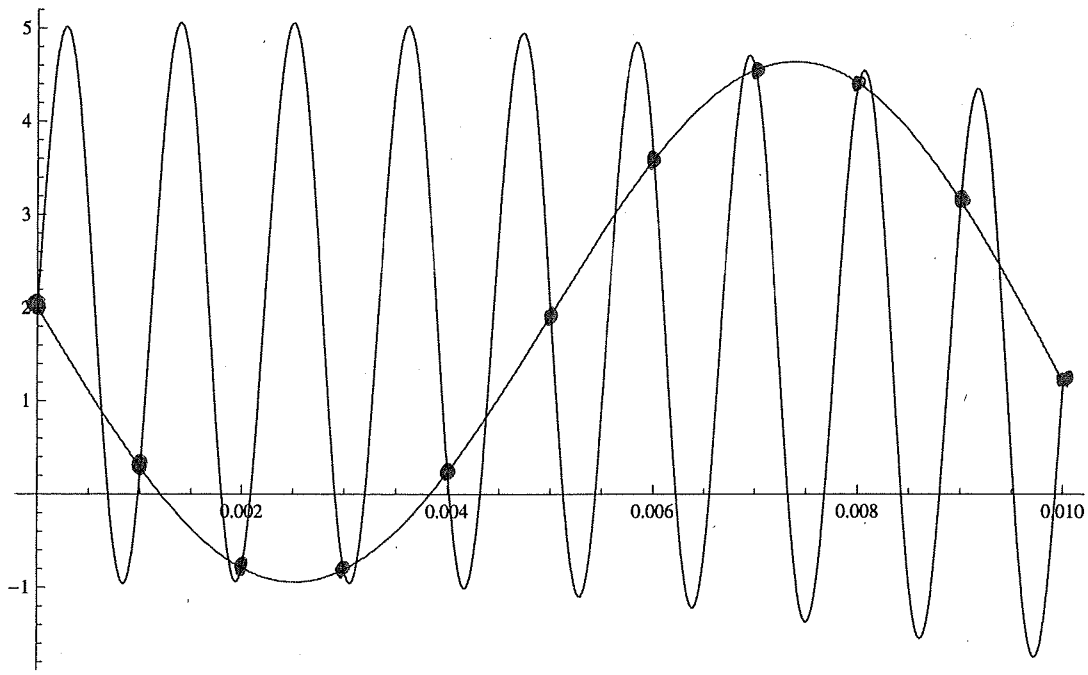
```
In[4]:= x[t_] := Sin[10 × 2 Pi t] + 2 Cos[20 × 2 Pi t] + 3 Sin[900 × 2 Pi t]

In[3]:= Clear[x]

In[10]:= y[t_] := Sin[10 × 2 Pi t] + 2 Cos[20 × 2 Pi t] - 3 Sin[100 × 2 Pi t]

In[11]:= Plot[{x[t], y[t]}, {t, 0, 0.01}]
```

Out[11]=



Aliasing effekten illustrerad på intervallet [0, 0.01].