# Lecture 2 assignments

## Exercise 1

Below is the plot of different sample ratings of a signal . The highest frequency component is 3Hz and therefore the Nyquist-Shannon sampling frequency is

The sample plot with (1) gives an approximate plot of the original sample, but with slightly lower amplitudes than the original. The oversampled one (2) (higher sampling frequency than 6Hz), gives more correct representation of the signal. The worst is the under sampled (3), which has significantly lower amplitude and wrong phase. Theoretically speaking, the sampling which gave the best representation here would be option (2). It seems it can capture the original signal in its entirety. ‘

close all, clear all

%Nyquist - Shannon sampling frequency: 6Hz >= 3Hz

dt1 = 1/100; %Original sampling rate

dt2 = 1/6; %Nyquist sampling rate

dt3 = 1/50; %Higher sampling than Nyquist

dt4 = 1/2; %Less sampling than Nyquist

st = 0;

et = 8;

t1 = st:dt1:et;

t2 = st:dt2:et;

t3 = st:dt3:et;

t4 = st:dt4:et;

%Original signal

y1 = 4\*sin(2\*pi\*t1)-2\*cos(6\*pi\*t1)-3\*sin(4\*pi\*t1);

%Nyquist sample signal

y2 = 4\*sin(2\*pi\*t2)-2\*cos(6\*pi\*t2)-3\*sin(4\*pi\*t2);

%Oversampled signal

y3 = 4\*sin(2\*pi\*t3)-2\*cos(6\*pi\*t3)-3\*sin(4\*pi\*t3);

%Undersampled signal

y4 = 4\*sin(2\*pi\*t4)-2\*cos(6\*pi\*t4)-3\*sin(4\*pi\*t4);

subplot(2,2,1)

plot(t1,y1), grid on, title('Original signal')

subplot(2,2,2)

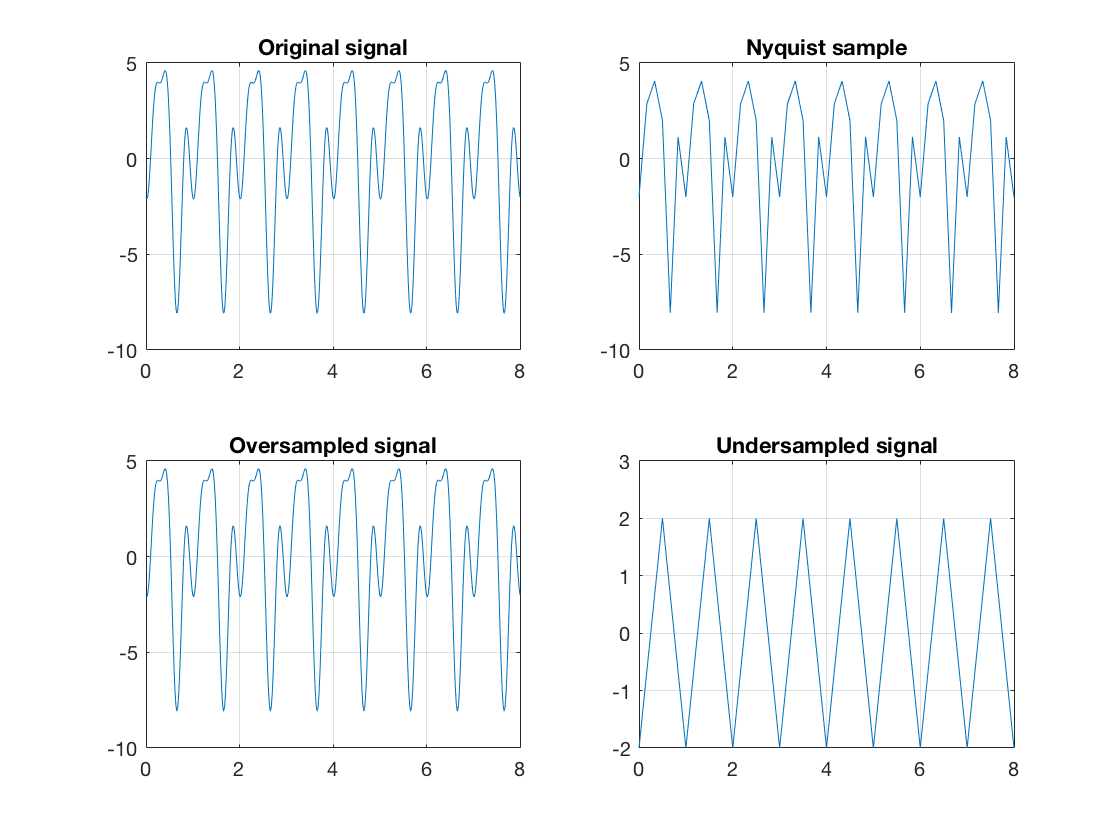
plot(t2,y2), grid on, title('Nyquist sample')

subplot(2,2,3)

plot(t3,y3), grid on, title('Oversampled signal')

subplot(2,2,4)

plot(t4,y4), grid on, title('Undersampled signal')



## Exercise 2

Like exercise 1, only with a more complex signal,

.

Here the Nyquist-Shannon sampling frequency is

The Nyquist-Shannon sample (1) makes a rather good representation of the original signal. It gets its shape and with some noise reduction, but with a reduction in amplitude and different phase (as it doesn’t pick up every signal value during the sampling). The overall shape is pretty good. The oversampled one (2) captures most of the original signal, but perhaps it is not so useful to keep all the noise from it. The under sampled (2) has the shape, but with a phase shift and somewhat lower amplitude. It has however a lot of the original noise taken away from it.

All in all, with capturing the original signal, I would say that option 2) looks the best. It might not be however possible to get the exact signal, as it has a lot of noise added to it.

%Exercise 2

%Nyquist-Shannon sampling frequency: 2\*12Hz = 24Hz >= 12Hz;

dt = 1/200; % Sampling rate

dt1 = 1/24; %Nyquist-Shannon sampling rate, double the samples.

dt2 = 1/100; %Oversampling

dt3 = 1/5; %Undersampling

st = 0; % Start time

et = 8; % End time

t = st:dt:et;

t1 = st:dt1:et;

t2 = st:dt2:et;

t3 = st:dt3:et;

%Original signal

y = 5\*sin(24\*pi\*t)+1\*cos(12\*pi\*t)-3\*sin(6\*pi\*t)+2\*cos(8\*pi\*t);

%Nyquist sample signal

y1 = 5\*sin(24\*pi\*t1)+1\*cos(12\*pi\*t1)-3\*sin(6\*pi\*t1)+2\*cos(8\*pi\*t1);

%Oversampled signal

y2 = 5\*sin(24\*pi\*t2)+1\*cos(12\*pi\*t2)-3\*sin(6\*pi\*t2)+2\*cos(8\*pi\*t2);

%Undersampled signal

y3 = 5\*sin(24\*pi\*t3)+1\*cos(12\*pi\*t3)-3\*sin(6\*pi\*t3)+2\*cos(8\*pi\*t3);

subplot(2,2,1)

plot(t,y), grid on, title('Original signal')

axis tight

subplot(2,2,2)

plot(t1,y1), grid on, title('Nyquist-Shannon signal')

axis tight

subplot(2,2,3)

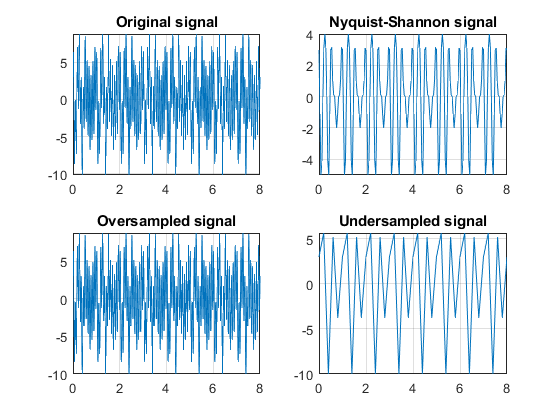
plot(t2,y2), grid on, title('Oversampled signal')

axis tight

subplot(2,2,4)

plot(t3,y3), grid on, title('Undersampled signal')

axis tight



## Exercise 3

The fast Fourier transform (fft) is an algorithm which computes the discrete Fourier transform (dft) of a set of samples, only in a more efficient matter. With a data size ***n***, the computation of dft takes time, while computing a fft takes time. What this means for the fft, is that for a data set of size ***n***, the algorithm must execute amount of operations. The input of a dft is a complex signal – a signal with a real and imaginary component. For computing dft, an amount of complex multiplications are needed.

The fft has a lower runtime than the dft and thus a higher efficiency rate. It is worth mentioning that the fft is not its own operation, but a family of algorithms whose purpose is to calculate the discrete Fourier transform. Examples of fft’s are the Cooley-Turkey algorithm and the Sande-Tukey algorithm.

## Exercise 4

a)

Below are four signals y1, y2, y3 and y4 plotted alongside with their Fourier transformation. For these signals, who each have ***n*** number of frequency components, there seems to be a relationship between frequency components and number of peaks in the fft. In other words, for ***n*** frequency components, the Fourier transformation has 2***n*** number of peaks.

b)

A fourth signal, y4 was also considered. The Fourier transform of this random signal gives several peaks. This may be explained through the rand() function, which assigns pseudo random numbers to every index. The output results in a function that might be expressed by several frequency components, and thus giving several peaks in the Fourier transform.

%Exercise 4

close all

clear

dt = 1/200; %sampling rate

st = 0; %start time

et = 8; %end time

t = st:dt:et; %time steps (vector)

y1 = 2\*sin(10\*pi\*t);

y2 = 2\*sin(10\*pi\*t) + 3\*sin(40\*pi\*t);

y3 = 5\*sin(10\*pi\*t)+2\*cos(40\*pi\*t)+3\*sin(80\*pi\*t);

y4 = rand(1,200);

%Takes the Fourier transform of the signal

%and shifts the zero frequency component to origin.

F1 = fftshift(fft(y1));

F2 = fftshift(fft(y2));

F3 = fftshift(fft(y3));

F4 = fftshift(fft(y4));

%Amount of peaks in the Fft - plot is

%relatable with amount of frequency components \* 2

%Amount of peaks in the Fft plot of y4 is at the most equal

%to the length of y4, as there might be values that are similar

%within the vector.

figure(1)

subplot(3,2,1)

plot(t,y1), title('y1(t)')

xlabel('t [s]')

subplot(3,2,2)

stem(abs(F1)), title('Fourier trans. of y1'), grid on

xlabel('f [s^{-1}]')

subplot(3,2,3)

plot(t,y2), title('y2(t)')

xlabel('t [s]')

subplot(3,2,4)

stem((abs(F2))), title('Fourier trans. of y2'), grid on

xlabel('f [s^{-1}]')

subplot(3,2,5)

plot(t,y3), title('y3(t)')

xlabel('t [s]')

subplot(3,2,6)

stem(abs(F3)), title('Fourier trans. of y3'), grid on

xlabel('f [s^{-1}]')

figure(2)

subplot(1,2,1)

plot(y4), title('y4')

subplot(1,2,2)

stem(abs(F4)), title('Fourier trans. of y4'), grid on