FNM - FFT

Felix Svedberg - fesv0011

February 2026

Compilation guide

The c-files used to solve the task of this lab are called: fft_gauss.c, fft_AM.c, fft_decode.c and signal_tools.c, where the first three contain the main functions and task-specific functions, while signal_tools.c contains functions that are used in all tasks. signal_tools.c also comes with a header file which is included in all task-specific files. The files are compiled using a makefile. Make sure that the c-files and makefile are all in the same directory. The c-files are compiled by simply using the command "make" in the terminal, which creates executables fft_gauss, fft_AM, fft_decode. The first two are ran by using the commands "./fft_gauss" and "./fft_AM". However, the executable fft_decode takes the filename of the signal file as user input and is therefore ran by using the command "./fft_decode amxx.dat". The GNU scientific library must be installed in order to run the programs.

1 Fourier Transform of a Gaussian

In the first part of this lab, I was tasked with writing a program to calculate the Fourier transform of the Gaussian signal

$$x(t) = \frac{1}{\sqrt{\pi\sigma^2}} e^{-t^2/\sigma^2}, \tag{1}$$

with $\sigma = 1/64$, and with 2048 samples from times $t = -1/3$ and $t = 1/3$. In Figure 1 the transformed data produced by the program is visualized.

It is worth mentioning that to obtain a result close to the analytical transform, the discrete transform had to be scaled by Δ, which is the time between samples, and also time corrected. The routine DFT routine used in the program assumes time samples from the signal to be from $t = 0, ..., (N-1)\Delta$. The Gaussian signal used in this exercise was sampled from $t = -1/3$ to $t = 1/3$. Therefore, the signal is shifted by -1/3, which has to be corrected. This is done by multiplying every element i of the transform by

$$e^{-2\pi i f_i t_0} \tag{2}$$

where t_0 is the size of the time shift and f_i the frequency corresponding to element i of the transform.

The frequency resolution of the FFT, which is the smallest spacing between distinguishable frequencies, is given by

$$f_{res} = \frac{1}{T}, \tag{3}$$

was 1.5 Hz.

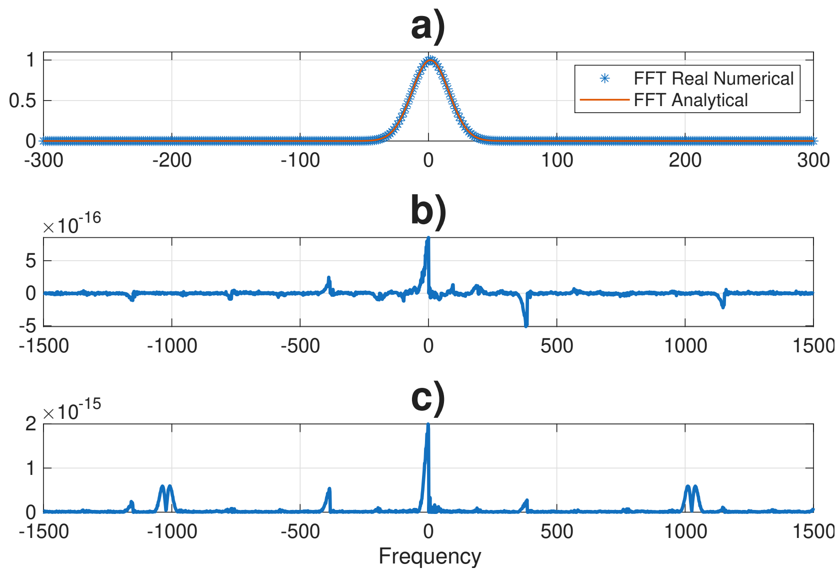

Figure 1: Fourier Transform, plots over: a) Real part of the discrete transform together with analytical transform, b) The complex part of the discrete transform and c) The difference between real part of the discrete transform and the analytical transform.

2 The spectrum of a simple AM Wave

An amplitude-modulated wave has the form:

$$y(t) = [A + B\sin(2\pi f t)]\sin(2\pi f t), \tag{4}$$

where f_c is the carrier wave frequency and f the frequency of the wave that modulates the amplitude of the carrier wave.

$y(t)$ can be rewritten as a sum of three waves of frequencies f_c, $f_c - f$ and $f_c + f$:

$$y(t) = A\sin(2\pi f_c t) + B\sin(2\pi f t)\sin(2\pi f t) \tag{5}$$

$$= A\sin(2\pi f_c t) + B\frac{1}{2}[\cos(2\pi(f_c - f)t - \cos(2\pi(f_c + f)] \tag{6}$$

$$= A\sin(2\pi f_c t) + \frac{B}{2}\cos(2\pi(f_c - f)t) - \frac{B}{2}\cos(2\pi(f_c + f)t). \tag{7}$$

2.1 Bandwidth

The bandwidth of a signal is given by:

$$\Delta f = 2f_m, \tag{8}$$

where f_m is the frequency of the modulating wave, referred to as f in the last section. The amplitude modulation goes through an entire period $u_0 \to u_1 \to u_0$ during $t = 256\Delta$. i.e $T_m = 256\Delta$. This means that

$$f_m = \frac{1}{T_m} = \frac{1}{256\Delta} = 4\text{Hz}. \tag{9}$$

Using Eq. (8) the theoretical prediction of the bandwidth is $\Delta f = 8\text{Hz}$.

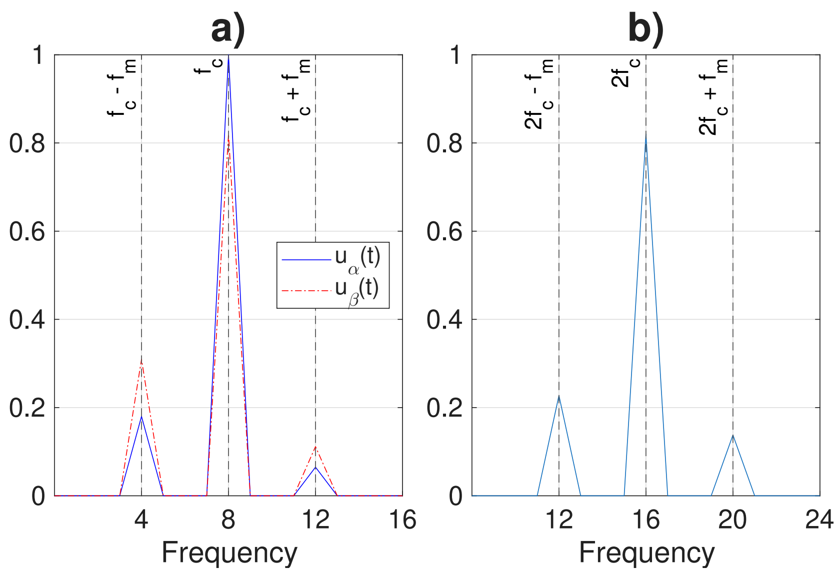

Figure 2: Plots over the frequency spectrum for a) amplitude modulations $\bar{u_\alpha}$ and $\bar{u_\beta}$ with carrier frequency f_c and b) amplitude modulation $\bar{u_\beta}$ with carrier frequency $2f_c$.

2.2 Frequency spectrum

In this part of the assignment I was supposed to investigate the frequency spectrum when using amplidtude modulations: $\bar{u_\alpha}$ and $\bar{u_\beta}$. $\bar{u}_\alpha$ uses $\bar{u}_{\alpha 0} = 1$ and $\bar{u}_{\alpha 1} = 3$, and $\bar{u}_\beta$ uses $\bar{u}_{\beta 0} = 0.5$ and $\bar{u}_{\beta 1} = 3.1225$.

In Figure 2 we see that the bandwidth when using an amplitude modulation of $T_m = 256\Delta$ was indeed 8 Hz. Switching between $\bar{u_\alpha}$ and $\bar{u_\beta}$ does not change the bandwidth, it only changes how much of the signal is made up of the amplitude modulating wave since we are increasing its magnitude.

What is the effect of increasing the carrier frequency but keeping the same bandwidth regarding how large binary message we can encode? Well, nothing happens. We see that the frequency spectrum looks about the same, only shifted to be centered around the new carrier frequency $2f_c$. What actually determines how many bits we can encode is the frequency of the modulating wave. Since we have the same frequency, we can only encode the same number of bits in the given timespan. However, the doubling of the frequency could enable us to encode twice as many bits since we can change the magnitude once every cycle of the carrier, but this would mean increasing the bandwidth.

3 Extracting information from a noisy signal

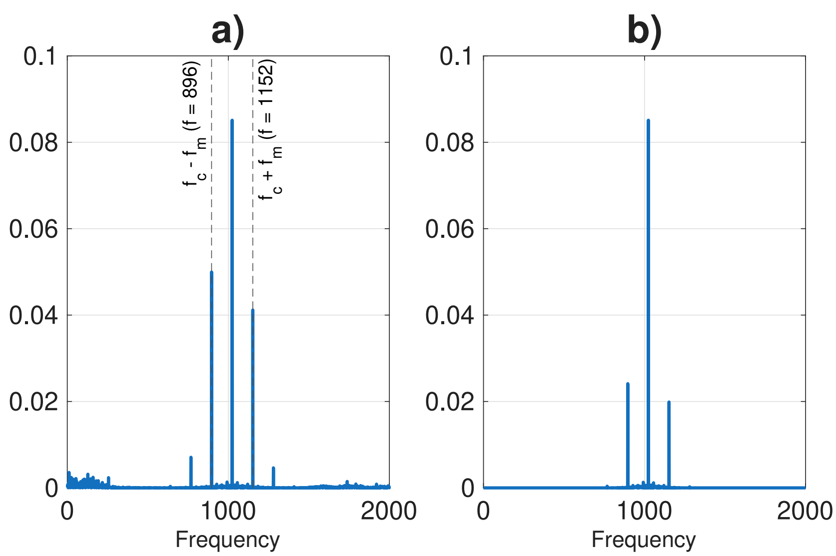

Figure 3: Frequency spectrum for a) Unfiltered signal and b) Filtered signal.

In the last part of the assignment, I was tasked to decode a message in a noisy file. This was achieved by first extracting the frequency spectrum (seen in Figure 3 a)) to find the modulating frequency. I found that it was 128 Hz, meaning that the bandwidth $\Delta f = 256$ Hz. A Gaussian filter, of width $\sigma = 150$ just above the modulating frequency, was then applied to the transformation. This filter gave a good result, seen in Figure 3 b). The signal was then transformed back into the time domain. The modulating frequency was 128 Hz, meaning that the amplitude changes every 64 samples. The signal was therefore examined in windows of 64 samples to find changes in amplitude, and then translated these amplitudes into bits.

The resulting message, when treating the signal in my individual file (am20.dat) was:

> FNM26 VD=VIn*'+*