

Practical Lab Class: Electronic, Signal and Measurement – Experiment 4

Signals & Filters

Filters play an important role in signal processing and can be used in different forms for different applications. The two most used forms are the low-pass filter, which allows low frequencies to pass and attenuates higher frequencies, and the high-pass filter, which allows only high frequencies to pass. An example of the use of filters can be found in multi-way loudspeakers, where filters allow only the signal components relevant to each loudspeaker to pass through.

**Questions to prepare before the lab.**

1. Explain low pass, high pass, band pass and band stop filters with frequency response diagrams.
2. You have received an electronic filter that is able to remove all the frequencies above 100 Hz and can preserve all the frequencies from 0 to 100 Hz. Recognize the filter and plot the frequency response.
3. In one of your projects, you want to pass only frequencies between 250 Hz to 350 Hz, then what kind of filter will you use?
4. For the complex signal containing frequencies from 10 Hz to 1000 Hz, we want to preserve all the frequencies between 100 to 200 Hz and all the frequencies above 700 Hz. How will you do this?
5. For the signal stated in question 4, now for another application, we want to preserve the frequencies between 400 to 450 Hz and all the frequencies below 100 Hz. How will you do this?

ATTENTION:

1. All the above stated questions must be prepared and written properly before coming to the lab.
2. All the labs must be implemented using Python 3. You can use the built-in function to generate sin, cos, and sawtooth waveform (if required) but for Fourier synthesis (DFT, IDFT, and others) direct use of built-in functions is not allowed.

4.1 Frequency analysis of signals

Given a sampling frequency, and a signal length of 2000 samples. Create the corresponding variables, as well as the resulting sampling interval and the corresponding time vector.

1. Create the following oscillations:

A cosine oscillation with frequency and amplitude 50.

A sine oscillation with frequency and amplitude 100

A sinusoidal oscillation with the frequency and the amplitude 200

1. Add up the oscillations you created and plot them on a graph. Label and format your graph accordingly.
2. Find the frequencies and amplitudes of the signal created in subtask b) using your DFT function and plot your result in the frequency domain in a figure. Label and format your diagram accordingly.

4.2 Creating simple filters and applying them in the frequency domain.

In the following, the respective signals from 4.1b) are to be filtered in the frequency domain.

1. Define a rectangular filter as shown below in PYTHON in the frequency domain to filter out all the oscillation less than and show by means of frequency analysis that the signal is filtered, whereby the higher oscillations are preserved. Graph your result in the frequency domain.

How is this filter form called?

A graph with a line

Description automatically generated

1. In the frequency domain, define a rectangular filter as shown below in PYTHON to filter the oscillations between 250 Hz to 350 Hz and show by frequency analysis that the signal is filtered, while the other oscillations (lower and high frequencies) are preserved. Graph your result in the frequency domain.

How is this filter form called?

A graph with a line

Description automatically generated

1. In the frequency domain, define a rectangular filter as shown below in PYTHON to filter the oscillations having frequency higher than 650 Hz and show by frequency analysis that the signal is filtered, while the other oscillations should be preserved. Graph your result in the frequency domain.

How is this filter form called?

A graph with a line

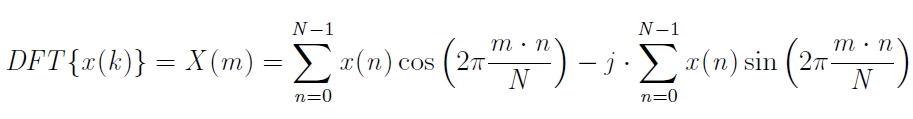
Description automatically generated

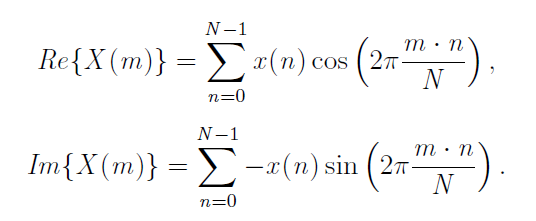
4.3 Back transformation of filtered signals into the time domain

In the following, the respective filtering from task 4.2 are to be examined in the time domain. For this purpose, create a function in PYTHON to calculate the inverse DFT to transform your filtered signal into the time domain.

DFT:

or after inserting:





with

Inverse DFT:

1. Create **your own** function to calculate the inverse DFT in PYTHON. The function should be able to pass any signals, whereby the output of the function should be the IDFT result.
2. Transform your filtered signal from 4.2a) into the time domain using the inverse DFT and show the result in the time domain together with the oscillation from 4.1b).

Check the plausibility of your result.

1. Transform your filtered signal from 4.2b) into the time domain using the inverse DFT and show the result in the time domain together with the oscillation from 4.1b).

Check the plausibility of your result.

1. Transform your filtered signal from 4.2c) into the time domain using the inverse DFT and show the result in the time domain together with the oscillation from 4.1b).

Check the plausibility of your result.