Oscilloscope and spectrum analyzer (Lab 2)

In this lab, an oscilloscope and spectrum analyzer is to be constructed in Labview. Most of the components have been finished except a few points that are to be designed by you. Throughout this lab we assume that the sampling rate is 44100Hz. All files are found by opening "lab_02_signal_analyzer.lvproj".

The oscilloscope module

First of all, please open "lab_02_simple_osc.vi". In this VI, you just have to finish the trigger function in the "False" case of the VI and to generate the values for the time-axis (x-axis) in the "True" case of this VI.

In the "True" case, the time values are the time stamps that mark the position of each sample along the time- or x-axis of the oscilloscope display. Since the sampling frequency is 44.1kHz, the time between two samples is 1/44100 second. In the FOR loop, there are 10000 samples that are generated, from 1 to 10000 and the distance between two samples is to be configured as 1/44100 second. Scale the output of this FOR loop in the "true" case of "lab_02_simple_osc.vi" to produce the correct time axis values, i.e., multiply each ample with 1/44100 second.

In the "False" case, the trigger function shall, as in a traditional oscilloscope, ensure that the displayed signal will begin at the desired position of the analyzed sequence. The correct starting position is defined by the trigger level and trigger flank controls (raising flank or falling flank).

In the Trigger function where new program is to be added, there is a "True/False" case. The True case corresponds to the scenario where the falling flack is to be detected. To detect a falling flack, we must compare the current sample and the next sample (time index +1) with the trigger level. If the current sample is larger than the trigger level and at the same time, the next sample is equal or smaller than the trigger level, we can certain that the signal can be triggered. To make sure that the sequence starts from a position that is not far away from the trigger level, it is better to guarantee that the value of the next sample is not much smaller than the trigger level.

The logic of a rising flank is opposite.

Use the "lab_02_simple_osc_test.vi" for testing "lab_02_simple_osc.vi". The "lab_02_ simple_osc_test.vi" produces a 1 V sine test signal (sampling rate 44.1 kHz). Please input a signal frequency and select sufficient number of samples to observe the signal. For example, you can put Trigger level as 0.5V, number of samples as 1000, and signal frequency as 100. Try to click Trigger flank and see how it changes from "NEG" to "POS".

The spectrum module

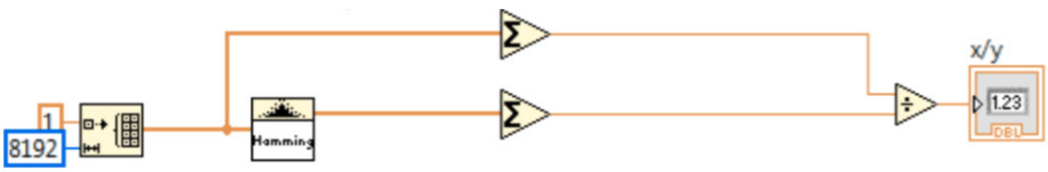
The "lab_02_spectrum.vi" analyzes an input signal in a 8192 points FFT. The input signal is weighted by different window functions and the result is presented in linear or dB mode. The following modifications have to be done to this VI before it can be used:

The frequency values are the frequencies of each component along the frequency- or x-axis of the spectrum display. The frequency range goes from 0 to (22050-$\triangle f$) Hz. Scale the output of the

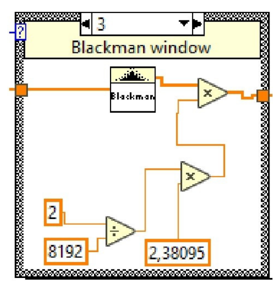
FOR loop in the “true” case to produce the correct values. (Hints, each step in the frequency axis is 22050/4096, or equivalently, 44100/8192.)

The Window case structure is empty. This structure shall do the windowing and scaling. The scaling shall be done so that the output to the display is 1 in linear mode when the input amplitude is 1 and the analyzed sequence covers a whole number of periods.

You can create the VI below in a separate VI[1] in order to find the scaling factor for the windows.



When the scaling factors for different types of windows are calculated, we can add different window functions in the VI. The figure below shows an example when we use blackman window. Note that we need to add a scaling factor for DFT because the output of DFT is N times larger. The scaling factor 2 above N is because half of the frequency components, i.e., the positive components, are shown in the spectrum window.



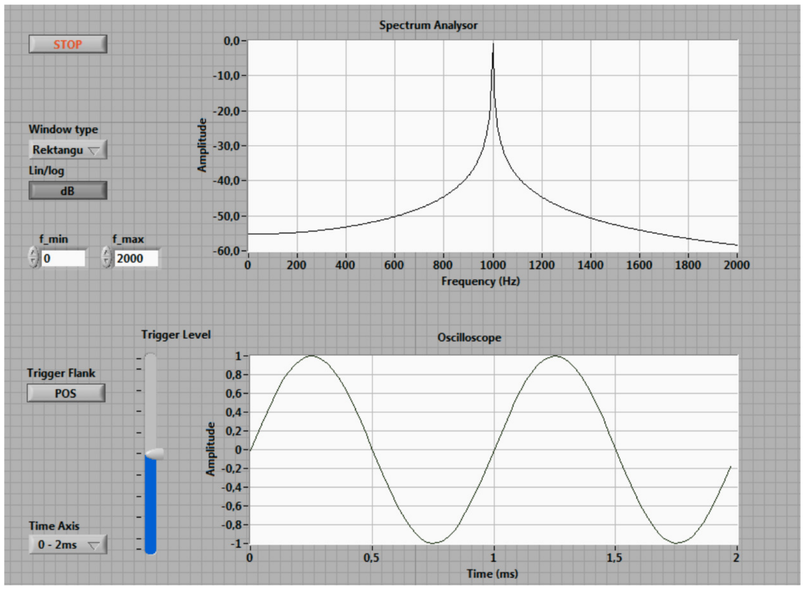
Put the required components into the linear/dB case structure. The constants connected to the Max_y_Value and Min_y_Value outputs shall be so that the Y scale goes from 0 to 1 in linear mode and from -60 to 0 dB in logarithmic mode. (Hints for DB level: As the signal is an amplitude signal, the formula is $20 \lg x$.)

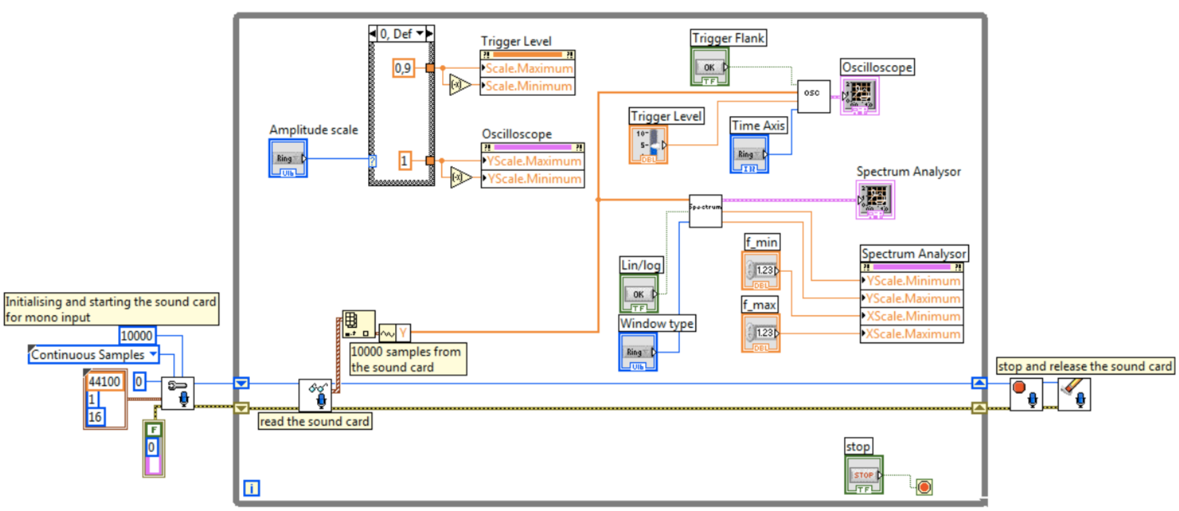
Use “lab_02_spectrum_test.vi” to test “lab_02_spectrum.vi”. The “lab_02_spectrum_ test.vi” produces 8192 samples of a sine signal which is passed to the input of “lab_02_spectrum.vi” and the output from “lab_02_spectrum.vi” is passed to a spectrum graph. Property nodes control the max and min values along the x and y axis. The sine sequence has frequency 1100Hz. You may change this if you find it necessary for testing the scaling.

Constructing a “live” instrument with sound card input.

[1]Note that it is not necessary to add this VI in “lab_02_spectrum.vi”. This VI is utilized to find out the scaling factors for different windows. If one does not want to make this VI, the following scaling factors can be used: Rectangle window: 1; Hanning Window: 2; Hamming window: 1.85185; Blackman window: 2.38095.

A virtual instrument containing an oscilloscope and a spectrum analyzer shall be completed. Open the “lab_02_signal_analyze.vi”, the front panel should look as shown below. The instrument reads the data from the sound card and presents the processed signals on the screens “oscilloscope” and “spectrum analyzer”.





Test the instrument

The instrument is now ready for use. Start the testing by putting a 1 volt sine (1kHz) (via a signal generator in Lab or another PC as input) into the sound card input (Microphone input) and adjust the sensitivity of the sound card (if possible) so that the oscilloscope of the LabView instrument shows the correct amplitude. Test it with different inputs (sine and square waves) from an external signal generator.

The report would document the parts of the program that you have created as well as measurement results. Comment the difference in the spectrum of a sine and a square wave.
Use log scale and analyze a sinus signal. If the resulting frequency consists of more than one frequency component, discuss possible reasons for this. Also comment the effect of the different windows.