**Laboratory 1A**

**Audio Signals**

In this laboratory, you carry out experiments generating and measuring audio signals in the time and in the frequency domain. In addition, you calculate the cross-correlation function of audio signals as a method to implement distance measurements.

**Exercise 1** *Setting up and connecting the External Audio Card*

We will work with the external audio card “Focusrite Scarlett” which are already connected to the PCs in the laboratory.

1. Check in the PC device manager, that the audio card is select as the active device for audio input and output.
2. Connect a microphone to the audio card and set ***direct monitor*** to on.

Hint.: Remember to turn on the microphone. The phantom supply (48V) can be left off.

1. Speak or sing in the microphone and visualize the monitoring output in the oscilloscope in the time domain. The frequency of your voice is probably in the range [50; 2k] Hz. Set and adequate time scale, and check the influence of the amplifiers, varying the gain for microphone and monitor.
2. Imitate the sound of an ambulance “Tiiii-Taaaa” and visualize the output in the oscilloscope in the frequency domain. In order to do that, select: Math Menu > Operation > FFT, and the source channel.

Hint: Which frequency do you expect for your “Tiii-Taaa” sound? Check that the scale in the oscilloscope is adequate to visualize this frequency range.

**Exercise 2** *Recording and plotting an audio signal in Matlab*

Use now the following code snippet to record a sound signal in Matlab using the external audio card:

% Trials with Focusrite external Soundcard

clear all, close all, clc;

% To check the settings of the Audio-Device

% (1) check in device manager that Focusrite is selected as Input-Device

% (2) execute command "asiosettings" to open settings panel for the ASIO driver

asiosettings

% Record your voice for 5 seconds.

recObj = audiorecorder(48e3,24,1);

disp('Start recording...')

recordblocking(recObj, 5);

disp('... end of Recording.');

% Play back the recording.

play(recObj);

% Store data in double-precision array.

myRecording = getaudiodata(recObj);

% Plot the waveform.

plot(myRecording); grid on;

1. Modify the code above to be able to record two channels in parallel.

Hint: you will need to change one input parameter of the ***audiorecorder()*** object.

1. Save each sound track on a separated vector, and generate a plot showing the two-recorded tracks.

**Exercise 3** *Distance Measurement using Correlation*

The following experimental setup is used to acquire simultaneously two audio tracks:

Noise

Generator

Microphone 1

Microphone 2

Oscilloscope

Log File

Matlab

Import & Process

1. Use your script from exercise 2 to record the noise output in parallel over the 2 microphones. A record length of 2 seconds is already enough.   
   Save 2 or 3 different measurements (with different distances between the microphones). Save your measured data with adequate names, e.g. ***audio\_data\_xx\_cm.mat***

Hint: you can save variables from a workspace in a ***\*.mat*** file with the command ***save(FILENAME,VARIABLES)*** and later on load it with ***load(FILENAME)***.   
Check for details with the help function.

1. Extend your Matlab script to calculate the cross-correlation between the signals acquired with Microphone-1 and Microphone-2, and use this result to determine the distance between the microphones.

*Hint-1* : use the cross-correlation function ***xcorr()***. For example: [R12, lags] = xcorr(track\_1,track\_2)

*Hint-2* : use the function max() to find the value and position of the maximum value in a vector.

For example: [peak, id] = max(R12)

*Hint-2 :* Consider the sound velocity as 343m/s

1. While you wait for your time slot to use the experimental setup, you can work with the measured data sample *\*.mat* file in order to develop your script to calculate the distance between the two microphones.
2. What is the resolution of your distance measurement with the current settings? Which parameter is limiting the accuracy? Can you change it?
3. Why a sinusoidal signal (instead of a noise signal) is not practical to do this distance measurement as described above? Justify your answer with an example plot in Matlab.