**Laboratory 1A**

**Audio Signals**

Sample Solution

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

1. Default setting in lab.
2. Use output in the back from direct monitor. See photo of setup below.

Ein Bild, das Tisch, Elektronik, Schreibtisch, Computer enthält.

Automatisch generierte Beschreibung

1. Time scale in the order of 10ms to 100ms / division.  
   Singing a vowel like “I” shows signal close to sinusoidal
2. **Using trigger + single shot** to observe the “Tiii-Taaa” sound. Frequencies around 50 till 200Hz (depending how deep or high do you sing)

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

1. The difference for single or double channel is in the parameter from the audiorecorder:

Single channel

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

Double channel

recObj = audiorecorder(48e3,24,2);

For double channel you need 2 microphones. Check the output variable (has 2 columns).

1. .

myRecording = getaudiodata(recObj);

track\_1 = myRecording(:,1)';

track\_2 = myRecording(:,2)';

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

Noise

Generator

Microphone 1

Microphone 2

Oscilloscope

Log File

Matlab

Import & Process

1. Record your measurement as ***audio\_data\_xx\_cm.mat***

%% If you are importing data from an existent measurement start here

clear all, close all, clc;

%load audio\_data\_meas\_A.mat

%load audio\_data\_meas\_B.mat

load audio\_data\_meas\_C.mat

Fs = 48e3;

N = length(track\_1);

aux = 0:1:(N-1); % index vector to plot recorded sound tracks

aux\_long = -(N-1):1:(N-1); % long index vector to plot auto-corr output

% long\_index not needed if you use lags

% check autocorrelation of track\_1 to have peak at "middle-point"

[R11, lags11] = xcorr(track\_1);

[peak11, id11] = max(R11)

% comparing 2 sound tracks

[R12, lags12] = xcorr(track\_1,track\_2);

[peak12, id12] = max(R12)

% The distance equals the time-difference between the peaks x speed of sound (343m/s)

% Time difference can be checked with index-difference x Ts

distance = (id12 - id11)\* 343/Fs

% Alternatively use index of R12 peak, and corresponding shift in lags12 vector

distance = lags12(id12)\* 343/Fs

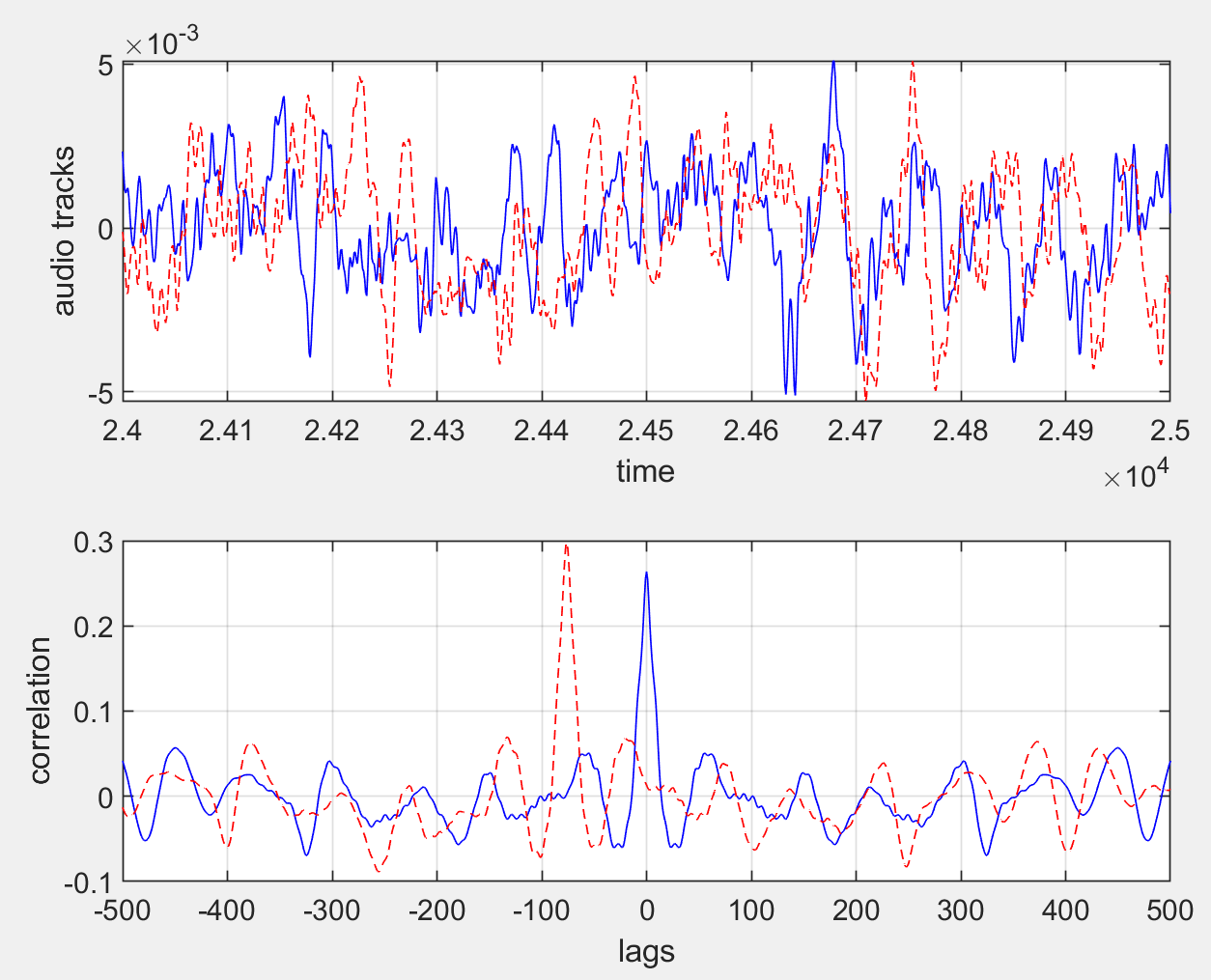
figure();

subplot(211),plot(aux,track\_1,'b',aux,track\_2,'r--'), grid on

pause(1), xlim([2.4e4 2.5e4]), xlabel('time'),ylabel('audio tracks')

subplot(212),plot(lags11,R11,'b', lags12,R12,'r--'), grid on

pause(1), xlim([-500 500]),xlabel('lags'),ylabel('correlation')



Measurements:

A : 0 cm

B : -30.01 cm

C: -54.01 cm

1. Limit for resolution? It is Fs, the sampling frequency. Would need a measurement with finer resolution to increase accuracy of distance measurement. At the moment accuracy equals:

(1)\*343m/s / 48kHz = 0.0071m = 0.71cm

1. The correlation of 2 sinusoidals is also a sinusoidal curve, which have several maximum points (not a single peak like the correlation between the 2 noise signals).

%% Correlation of 2 discrete sine waves, with a shift

clear all, close all, clc;

Fs = 48e3;

N = Fs; % corresponding to record 1 sec

aux = 0:1:N-1;

y1\_n = sin(aux\*2\*pi/50);

y2\_n = sin(aux\*2\*pi/50 + pi/5);

% check autocorrelation of track\_1 to have peak at "middle-point"

[R11, lags11] = xcorr(y1\_n);

[peak11, id11] = max(R11)

% comparing 2 sound tracks

[R12, lags12] = xcorr(y1\_n,y2\_n);

[peak12, id12] = max(R12)

% The distance equals the time-difference between the peaks x speed of sound (343m/s)

% Time difference can be checked with index-difference x Ts

distance = (id12 - id11)\* 343/Fs

% Alternatively use index of R12 peak, and corresponding shift in lags12 vector

distance = lags12(id12)\* 343/Fs

…

figure();

subplot(211),plot(aux,y1\_n,'b',aux,y2\_n,'r--'), grid on

pause(2), xlim([2.4e4 2.5e4])

subplot(212),plot(lags11,R11,'b', lags12,R12,'r--'), grid on

pause(2), xlim([-500 500])

