

Explanation of Read_data_CAE

All data recorded by the CAE acoustic camera is saved in a .tdms file. This file-extension cannot be read by Matlab directly, therefore, this code has been constructed. First, the quick data extract and first usage of the code is given. Then the microphone calibration method is explained. Lastly, the outputs are discussed in a bit more detail.

Note that this Read_data_CAE doesn't check whether microphones worked properly. Please perform phase and OSPL coherence checks yourself. Additionally, make sure to read the manual on the directionality and frequency behaviour you can expect from the microphones before you analyse your data. This is important so you know what unexpected levels or sources could be presenting.

Extract data from measurements

- Extract the post_process_CAE_array.zip folder. This folder contains the TDMS-reader.
- Place the TDMS-reader folder in the same folder as the Read_data_CAE code.
- You can now make a Matlab script that starts with:

```
data.dir = 'U:\Measurement Campaign Schiphol 2019\2 - Information on  
acoustic cameras\CAE systems information\Matlab  
code\Meetingroom_outline-cam\measurements\Source_middle.tdms';
```

```
data.start_time = 9;  
data.n_samples = 2048;
```

```
Read_data_CAE
```

- In this script, change the data.dir to the location where you saved the measurement. Also change the data.start_time and data.n_samples to the settings you need. If you use time-domain calibration, never go over 8192 samples!
- In line 15 of Read_data_CAE, specify folder location of the TDMS-reader.
- You should now already be able to run the code and get the necessary structs. In the end the code gives structs:
 - o data – containing time-pressure data as measured by the microphones.
 - o info – containing information on the measurement, such as comments, reference pressure, sampling frequency.
 - o config – containing the microphone coordinates.
 - o video – containing the video footage of the camera, saved in separate frames.

You can directly use “data.full_data” which contains the measured pressure at every timestamp per microphone. **WARNING: this is not calibrated!** The time corresponding to each measurement point can be found in “data.full_time”.

Microphone calibration

The microphones need to be calibrated as they show high offsets at high frequencies. An experiment in the A-tunnel has been performed to determine the offset per frequency. The Read_data_CAE code

contains an example how to do this in the frequency (line 74 through 106) and time domain (line 108 through 121). **ONLY if you need to do time domain beamforming (ROSI), you are allowed to use the time domain correction method.** Otherwise, you have to use the frequency domain calibration, which is more accurate.

In principal the calibration works as follows:

- The middle 7 microphones have a different correction than all other microphones. Therefore in line 77 through 79 the microphones are split in middle and normal.
- The time domain pressure data is transformed to frequency domain data. In the example this is done with a simple pwelch. However, it is strongly advised to construct your own PSD from fft and correcting the fft according to your windowing and zeropadding.
- The frequencies at which the pressure data is available and at which the calibration is available, are matched. When a power of 2 is used for the length in the fft, then always a match can be found. This is done in line 94 which should be an integer number larger than 1. The calibration is then selected at the frequencies `freq_cal(1:match_FD:end)`.
- In the frequency domain data a calibration is applied, which can be done in one of the following ways. The method depends on the usage of the data and has to be determined by the student, although it is advised to calibrate directly on the FFT. The examples are only given for the middle microphones, it works the same for the outer microphones.
 - o `Xdft_cal(middle mics) = Xdft(middle mics) ./ cal_mid_factorFT;`
 - o `PSD_cal(middle mics) = PSD(middle mics) ./ cal_mid_factorPSD;`
 - o `SPL_cal(middle mics) = SPL(middle mics) - cal_mid_dB;`
 - o `cal_mid_factorFT_two` is only used when time-domain calibration is needed. For time-domain calibration the block should be copied exactly.
- You can now use the calibrated frequency domain pressure data! 😊
- If you really needed the time domain calibrated pressure data, use the time domain calibration block. It performs the above steps and then uses ifft to go back to the time domain. Note that this causes round off errors and thus is not preferred! **If you do not use time domain calibration, remove this part of the code to avoid errors! Note: only if you use beamforming methods in the time-domain (ROSI) you are allowed to use time domain calibration!**

To check whether you implemented the calibration correctly, you can always plot SPL spectra of non-calibrated data and your own calibrated data. Then you can see if the difference you see in decibels matches that of `cal_mid_dB` or `cal_out_dB`.

Outputs

The outputs of the info-struct can be found with explanation in line 27 through 37. Note that some are 'set by researcher during recording'. It could very well be that this information was not available during recording, thus it is always best to use the information that is provided about distance and weather conditions.

The outputs of the config-struct are channel number (needed for determining middle/outer mic), x-position, y-position, z-position, and data row. The coordinates (in metres) are with respect to the centre of the array, where the camera is positioned. If (x,y) is plotted, you are looking at the microphone array from behind.

The outputs of the data-struct can be found in line 52 through 72. These include “data” and “time” extensions. The “data” extensions are either the pressure recordings for all microphones for the full file or for a certain time selection of the data which is specified by the user with the start_time and n_samples. For both these data selections, also a time vector is present. Additionally the sample selections are saved in the data-struct. If time-domain calibration is used, also data.selected_data_cal is available. This is the calibrated time domain pressure data for the selected time block.

The outputs of the video-struct include video.images, video.timestamp, and video.frames for the full measurement as well as for the selection of the measurement. The video.images contains the gray-scale footage of the camera.