Exercise 4: Discrete Fourier Transform (DFT)

For a successful conduction of the Laboratory Exercise an accurate **preparation** at home is **mandatory**. Otherwise the **participation** in the lab will be **refused** and a repetition of the exercise is required!

Preparation of the exercise (homework)

- Study the given PDF-document "*DFT_Examples.pdf*" in detail. In the practical course you will get the opportunity to test the corresponding Matlab file "*DFT_Relations.m*". However it is required that you examine accurately the code of both files illustrated in the PDF before. Try to understand the operation of code without testing it in Matlab.

Conduction of the exercise (must be done in the lab to grade contribution)

Now copy the Matlab template (see folder "*Templates*") and unzip all files to your local Matlab workspace

```
C:\Users\...\Projects\Matlab_2015a
```

DFT-Relations

- Firstly run the m-function *„DFT_Relations.m"* in Matlab to understand exactly the operation of the Matlab code. The m-function expects the following **3 parameters**: Sampling frequency **Fs**, Number of Samples **N** and Number of all samples including zero-padding **Nzp**.

  Note:
  Please do not **modify** the m-function "*DFT-Relations.m*". You can call the m-function with different parameters directly from the Matlab Command Window, for example "***DFT_Relations(30e3, 8, 8)***".

- Observe the output for various parameters assigned as function parameters. Begin with the default parameters as stated below:

Sampling rate:	Fs = 30e3 [Hz]
Number of samples:	N = 8
Number of samples with zero-padding:	N_zp = 8

Please note that both **conditions $N_{ZP} \geq$ N and N_zp even always need to be met** because of implementation issues.

By use of the default parameters above all typical errors discussed in the lecture are visible:

- „Aliasing error": controlled by the sampling rate Fs
- „Smearing or leakage error": affected by the number of samples N
- „Grid error": affected by „zero padding" N_zp

Print the result and interpret it.

In the following the relations between the different transformations are analyzed.
Try to find and understand the underlying systematic.

- Increase now N_zp up to 16, 64 respectively 128.
  Print each result and interpret it.
- Set now the parameter N_zp = N and increase the sample number N beginning from N = 8 step-by-step up to N = 10, 12, 16.
  Print each result and interpret it.
- Assume now the parameters N = 64 and N_zp = 128 and increase the sampling rate Fs beginning from Fs = 30e3 [Hz] step-by-step up to Fs = 40e3, 50e3, 100e3 [Hz].
  Print each result and interpret it.
- Subsequently discuss inside the group the conditions associated with faithful approximation of the spectrum of the analog signal. Which practical constraints could lead to agreements?

Spectral audio analysis with FFT

- Firstly develop a new Matlab script (m-file) which **records audio signals** from the computer sound card and stores the data to the workspace or a Matlab formatted file (mat-file). Create a plot of the captured audio data. Use the provided (dynamic) microphone as audio input device and choose the following recoding settings in Matlab:

Sampling frequency:	44 100 Hz
Resolution:	16 Bits
Record period:	3 seconds

  Note: For the data acquisition you can use the prepared m-File *"AudioRecord.m"*.
  Try to understand the Matlab code using the Matlab help.
- Record now the given **3 audio test signals** played on a loudspeaker in the lab.
  After each recording save the captured data as separate variable in the workspace or in a mat-file (optional) for later analysis.

  The test signals have a duration of 10 seconds, so begin your recoding not before the sound is playing. Some details of the test signals are already known:

  Signal 1: 1 sinusoidal component
  Signal 2: 2 sinusoidal components with different frequencies
  Signal 3: 2 sinusoidal components with close by frequencies

  Note: We do not consider here any amplitudes, because they can vary on each place in the room.
- Then develop an own Matlab script which **calculates** the **FFT** of the recorded audio signals.
  Define the parameters N and Nzp at the beginning of your Matlab script (initial values are N = 1024 and N_zp = 1024). Plot the truncated time signal and the only magnitude of the FFT. Choose the unit "Hertz" for frequency axis!

  Examine the spectrum of the audio signals and try to identify the frequencies of the spectral components by modifying the parameters N and N_zp.

  Note: Do not use the first 0.5 seconds of each record, because there is a short transient effect after the initialization of the sound card.

Save your data from the local workspace to an external drive (e. g. Pendrive) or your Home-Drive after each lab training!