

Assignment 4

Reconstruction, DFT, FFT, STFT

Information about assignments and in particular on Moodle submission modalities can be found in DSP-Tutorial_2024S_CourseInfo.pdf. Submission deadline is Tue, **June 11, 2024, 08:00**.

Exercise 1 *Reconstruction (15%)*

Consider the analog signal

$$x(t) = 1 + 0.5 \cos(2\pi f_1 t) + 2 \sin(2\pi f_2 t) + \sin(2\pi f_3 t),$$

with $f_1 = 2\text{kHz}$, $f_2 = 4\text{kHz}$ and $f_3 = 6\text{kHz}$.

- Sketch the Fourier transform of $x(t)$ and plot the analog signal $x(t)$ in Matlab using a time-vector $t = 0:1\text{e-}6:1\text{e-}3$.
- Sample the analog signal $x(t)$ with sampling frequencies $f_{s1} = 9\text{kHz}$ and $f_{s2} = 14\text{kHz}$, which yield $x_1[n]$ and $x_2[n]$, respectively. Sketch the corresponding DTFT spectra.
- After ideal reconstruction you end up with the analog signals $x_1(t)$ and $x_2(t)$. Sketch the reconstructed analog spectra. Plot the reconstructed time-domain signal. Compare your results with point a).

Exercise 2 *DFT Theory (20%)*

100 values of an analog signal $x(t)$ were measured with a sampling time of 1 ms, leading to the discrete-time signal $x[n]$. This time domain signal $x[n]$ is transformed to frequency domain using the DFT/FFT, i.e., a 100-point DFT/FFT is calculated.

- What is the frequency spacing between two neighboring spectral points in the DFT spectrum, i.e., what is the frequency resolution?
- What is the “period” of the DFT spectrum in terms of samples, in terms of frequency, and in terms of normalized angular frequency?

We now append zeros to the discrete-time signal $x[n]$ to obtain a signal that has a length of 128 samples.

- Why do we append exactly so many zeros, so that a total signal of length $128 = 2^7$ results?
- What is the frequency spacing between two neighboring spectral points in the DFT spectrum now?
- How can the changed distance between two neighboring spectral points be interpreted?

Exercise 3 FFT in Image Processing (15%)

In image processing, besides other methods, also the FFT can be utilized to conduct edge detection. To this end, a two-dimensional FFT (MATLAB command: `fft2`) is calculated from an image to transform it from spatial domain to frequency domain. Here, edges in an image produce high frequencies in the corresponding spectrum.

In this exercise, you should make the edges of an image visible. Start by making yourself familiar with the MATLAB script `fft_edge_detection`. Now filter the image in frequency domain, such that the edges become visible after a transformation back to spatial domain. Describe your approach and justify why it works. A solution could look the one depicted in Fig. 1.

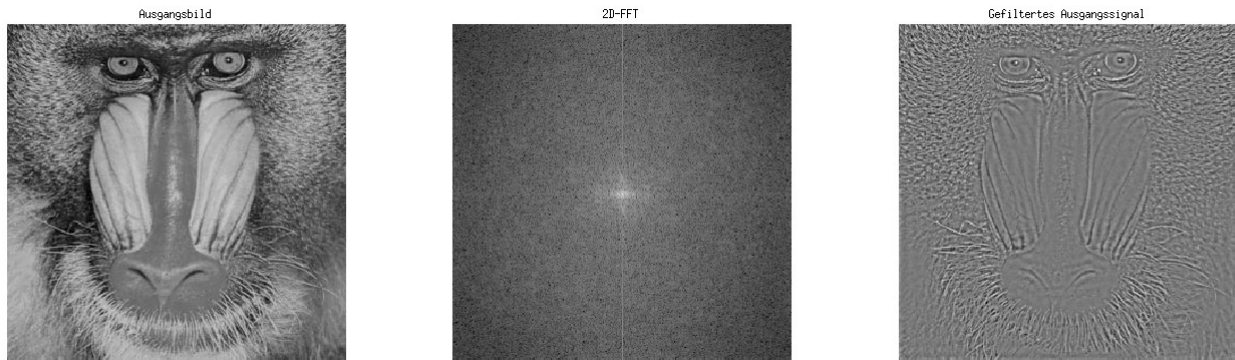


Figure 1 Example for edge detection with the FFT.

Exercise 4 FFT in Audio Signal Processing – Short Time Fourier Transform (30%)

In the Dual Tone Multiple Frequency (DTMF) method, each of the 16 possible information symbols $q \in \{0,1,2,3,4,5,6,7,8,9, A, B, C, D, *, \#\}$ is represented by a superposition of two sinusoidal audio signals with different frequencies. The assignment of every symbols to the two frequencies contained in the corresponding audio signal is given in Tab. 1. The duration of a single symbol is 70 ms, and the audio signals of two symbols are separated by a pause of 30 ms. The file `dtmf.wav` contains a signal consisting of a sequence DTMF signals corresponding to a sequence of randomly chosen symbols, where the signal has been generated with a sampling frequency of $f_s = 8$ kHz.

Table 1 DTMF frequencies.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

- (a) Write a MATLAB script, where you read in the signal and the sampling frequency from the file `dtmf.wav`. *Hint:* the MATLAB function `audioread` might be useful. Plot a 300 ms long segment of the signal and put the resulting plot into your protocol (including a correct labeling of the axes, etc.). Given this signal segment in time domain, can you make any statement which symbols are contained in this segment?

- (b) Compute the spectrum for the whole signal length using the MATLAB command `fft`. Plot the magnitude of the spectrum. Label the axes correctly, with the frequency axis scale in Hz (*Hint*: the frequency values given in Tab. 1 should be visible at the correct position on the frequency axis).
- (c) Now the so-called short-time Fourier Transform (STFT) should be implemented. To this end, the total signal is grouped into overlapping blocks of a specified length, and each block is transformed individually into frequency domain. Additionally, the individual blocks should be filtered using a Hamming window (MATLAB command `hamming`) for improving the spectral illustration.
- Implement the generation of the signal blocks given the total signal. Every block should have a length of 256 samples. Use an overlapping factor of 2, i.e., every block is overlapping with its preceding block by half the signal length. The last block should also have a length of 256, i.e., if not enough samples from the total signal are left to obtain a block of length 256, append the appropriate number of zeros.
 - Every block should be multiplied with a Hamming window. The window can be generated with the MATLAB command `hamming`.
 - Transform the windowed blocks to frequency domain using the FFT to obtain the Fourier transformed blocks (FTBs).
 - Plot a 2d diagram showing the FTBs. For this purpose, consider which frequency values have to be displayed on the frequency axis, and which time values have to be plotted on the time axis, i.e., which frequency range has been regarded with the FFTs and at which points in time have the FFTs of the individual blocks been computed? Generate two MATLAB vectors `f_stft` and `t_stft`, which contain the appropriate frequency and time values, respectively. Then, plot the diagram containing the magnitude of the STFT result using the MATLAB command `surface(t_stft, f_stft, abs(stft_mtx))`, where `stft_mtx` is the matrix containing the individual FTBs. Display the amplitude information with the command `colorbar`. The STFT magnitude diagram in your protocol should look similar as in Fig. 2 (the symbol sequence might not be the same), where in Fig. 2 only the frequency range 0-2000 Hz is shown.

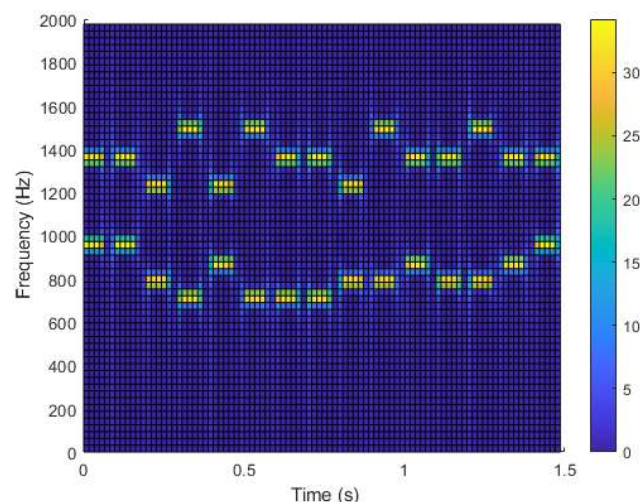


Figure 2 STFT magnitude of a DTMF sequence.

- (d) Perform the same steps as in (c), but without multiplying the signal blocks by a Hamming window. How does the resulting magnitude diagram of the STFT differ to the one computed in (c)? How is the effect called that causes this difference?
- (e) On basis of the plotted diagram in (c), determine the symbol sequence that has been used for generating the total signal.
- (f) Answer the following questions:
 - What is the essential difference between the diagrams plotted in (b) and (c), and what becomes apparent in the diagram in (c) that cannot be observed from the diagram in (b)?
 - Give an example for an application of the STFT and describe it briefly.

Exercise 5 *Window effects of the DFT (20%)*

Let us consider a signal consisting of two cosine oscillations with close frequencies. The actually infinite signal is time limited by windowing it once with a rectangular window and once with a Hamming window of length N . Generate the signal in MATLAB as follows:

```
N = 128;
n = 0:N-1;
w1 = 2*pi*0.1;
w2 = 2*pi*0.15;
x = cos(w1*n) + cos(w2*n);
```

Since x has only finite length, it implicitly has already been windowed with a rectangular window.

- (a) Compute the (discrete) spectrum of x and plot a line plot of its magnitude (MATLAB commands `fft`, `abs`, `stem`). Do not forget to label the axes!
- (b) Generate a Hamming window of length N using the MATLAB command `hamming`. Multiply the Hamming window with the signal x to obtain the signal y . Compute the spectrum of y and display its magnitude like in (a).
- (c) Compare and interpret the results from (a) and (b).
- (d) Experiment with $w1$ and $w2$ (i.e., adjust their values) and find a setting, where the DFT/FFT yields the exact result. Explain why the DFT/FFT result is exact with the selected settings.