Lab 1: Introduction to MATLAB and basic digital signals

- **Exercise 1:** Create a vector containing 40 points of the **discrete-time** sinusoid

  $$x(n) = \sin(\omega n) \ , \quad n = 1, \dots , 40$$

  with the discrete-time frequencies $\omega = 0.1$, 0.2, 0.4, and 0.8, respectively. Plot the so-obtained discrete-time signals using the function `stem`. Use the option `help stem` to learn how to use this function.

  Now, assume that the sampling interval if $\Delta t = 0.1$ seconds. For each plot, compute the continuous-time frequency f in Hertz using the given value of the sampling interval and the discrete-time frequency ω. Redisplay the results indicating the time axis, and indicate the obtained continuous-time frequency on the top of your plot using the function `title`.

  What happens with your plots when increasing the frequency?

- **Exercise 2:** Assume that you have the **continuous-time** cosine signal

  $$x(t) = A\cos(2\pi f t + \phi) \ , \quad t \in [t_{\text{start}}, t_{\text{end}}]$$

  with the amplitude $A = 5$, frequency $f = 10$ Hz, initial phase $\phi = \pi/3$, starting time $t_{\text{start}} = 0$ seconds, and ending time $t_{\text{end}} = 100$ seconds.

  Sample this signal with the sampling interval $\Delta t = 10^{-2}$ seconds and plot the result. How many points of signal did you get after such a sampling?

  Compute the sampling frequency f_s. Compare it with f. How does the obtained sampling frequency compare with the Nyquist rate? Comment on the Nyquist rate: why is it important to sample the signal with a sampling frequency higher than the Nyquist rate?

- **Exercise 3:** Create the unit impulse signal $\delta(n - 15)$ and the unit step signal $u(n - 10)$ in the interval $n = 1, \dots , 30$. Plot them using the function `stem`.

  Plot the difference signal

  $$x_1(n) = u(n - 10) - u(n - 11)$$

  Interpret this signal as $\delta(n - k)$ and find the value of k from your plot.

  Plot another difference signal

  $$x_2(n) = u(n - 10) - u(n - 15)$$

  Find from the plot the width of the so-obtained impulse signal. Express this signal as a sum of unit impulses.

- **Exercise 4:** Generate a **complex-valued** discrete-time signal

  $$x(n) = e^{j\omega n} \ , \quad n = 1, \dots , 40$$

  with $\omega = 0.2$. Extract the real and imaginary parts of signal using the functions `real` and `imag` and plot them. Use `subplot(211)` and `subplot(212)` commands prior to each `stem` to create two plots of real and imaginary parts placed on the same screen.

Lab 2: Time-domain signal analysis and DTFT

2.1 LTI systems

The purpose of this part of the lab is to get you familiarized with the linear convolution and LTI systems concepts.

- **Exercise 1:** Let the LTI system be defined by its impulse response

$$h(n) = \begin{cases} 0.02\,, & 0 \le n \le 49 \\ 0\,, & \text{otherwise} \end{cases}$$

  Is this system causal and stable? Motivate your answer.

  Create the discrete-time input signal

$$x(n) = 3 + \sin(0.4n)\,, \quad 0 \le n \le 299$$

  Compute and plot the output signal $y(n)$ of this system using the standard MATLAB m-file `conv` performing linear convolution of two arbitrary finite-length vectors.

  Compare the input and output signals by means of displaying them together and describe what happens with the signal after transmitting through the system.

  Change the order of the convolved sequences and compare the output signals. Explain the results of this comparison.

- **Exercise 2:** Create another signal

$$x(n) = \sin(0.01n) + \xi(n)\,, \quad 0 \le n \le 599$$

  where $\xi(n)$ is the nth element of the noise vector. To obtain the noise part of this signal, create the noise vector as `xi = randn(600,1)`. Compute the output signal of the system considered in the previous exercise using the created input signal. Plot the input and output signals and describe how does the signal change when transmitting through the system.

- **Exercise 4:** Create the sequence

$$s(n) = 0.1\,, \quad 0 \le n \le 9$$

  and save it as the vector `vec0`. Compute convolution of `vec0` with itself and save the result as the vector `vec1`. Then, convolve the sequence `vec1` with the sequence `vec0` and save the result as the vector `vec2`. Convolve once again the sequence `vec2` with the sequence `vec0` and save the result as the vector `vec3`. Finally, convolve the sequence `vec3` with `vec0` and save the result as the vector `vec4`. Plot and compare the sequences `vec0`, `vec1`, `vec2`, `vec3`, `vec4` using `stem` representation. What happens with the resulting function at each intermediate step?

2.2 DTFT

The purpose of this part of the lab is to create the m-file computing DTFT and to test it using basic digital signals.

- **Exercise 1:** Create a MATLAB program computing the DTFT $X(f)$ of the discrete signal $x(n)$ having nonzero values in the interval $n = -N/2, \ldots, N/2$ and zero values elsewhere.

  Using this program, compute the DTFT $X(f)$ of the rectangular signal

$$x(n) = \begin{cases} 1\,, & -5 \le n \le 5 \\ 0\,, & \text{otherwise} \end{cases}$$

  at equispaced frequencies in the frequency interval $[-1.5; 1.5]$ with the frequency step 0.01. Use the MATLAB function `linspace` to obtain the required frequency grid. The resulting DTFT should be real – explain why. Since after your MATLAB computations the result will be complex-valued, use the aforementioned property and represent $X(f)$ as the real part of the previously obtained complex-valued DTFT. Plot the resulting function versus the discrete-time frequency in the given frequency interval $[-1.5; 1.5]$. Is this function periodic? If it is periodic, evaluate its period and compare it with the period of DTFT following from theoretical considerations.

- **Exercise 2:** Create the harmonic signal

$$x(n) = \begin{cases} \cos(2\pi f_0 n), & -30 \le n \le 30 \\ 0, & \text{otherwise} \end{cases}$$

Compute the DTFT of this signal assuming that:

a) $f_0 = 0.05$,

b) $f_0 = 0.2$,

c) $f_0 = 0.4$,

d) $f_0 = 0.5$,

e) $f_0 = 0.7$,

f) $f_0 = 0.9$.

Similarly to the Exercise 1, exploit the fact that the resulting DTFT's must be real and represent them as real-valued functions. Plot the so-obtained DTFT's $X(f)$ versus f in the interval $[-0.5, 0.5]$ with the frequency step 0.01 for the cases a)-f). Notice the values of f_0 where the aliasing occures and indicate this effect in your plots.

- **Exercise 3:** Create two finite-length sequences

$$x(n) = 1, \quad 0 \le n \le 9$$

and

$$y(n) = 2, \quad 0 \le n \le 14$$

Compute their convolution $z(n)$. Then, compute and plot the absolute value of its DTFT $Z(f)$ in the frequency interval $[-0.5; 0.5]$ with the frequency step 0.01.

Now, compute the DTFT's $X(f)$ and $Y(f)$ of the original sequences $x(n)$ and $y(n)$, respectively, and plot the absolute value of the function $X(f)Y(f)$ in the frequency interval $[-0.5; 0.5]$ with the frequency step 0.01. Are the absolute values of the functions $Z(f)$ and $X(f)Y(f)$ identical? Explain why.

Lab 3: DFT and FFT

The purpose of this lab is to create an m-file for DFT computation, apply it to different signals, study the effect of zero-padding and DFT resolution properties, and compare the computational complexities of DFT and FFT.

- **Exercise 1:** Create a MATLAB program computing the DFT $X(k)$, $k = 0, \dots, N-1$ of an arbitrary discrete finite-length signal $x(n)$, $n = 0, \dots, N-1$.

  Using your program, compute the DFT of the rectangular signal

  $$x(n) = \begin{cases} 1, & 0 \le n \le 15 \\ 0, & \text{otherwise} \end{cases}$$

  with a) $N = 16$, b) $N = 32$, c) $N = 64$, d) $N = 128$, and e) $N = 256$. Assuming that the sampling interval of the signal x is 0.2 sec, find (in terms of continuous-time frequency, Hz) the frequency step between neighboring DFT points in the cases a)-e). Plot the absolute value of the resulting DFT function using the `stem` representation.

  Describe what happens with the absolute value of your DFT when involving more and more zero values in the sequence $x(n)$. Interpret your results in terms of zero-padding.

- **Exercise 2:** Use the standard MATLAB program `fft` and evaluate the absolute values of the FFT for the sequences corresponding to the cases a)-e) of Exercise 1. Compare the results of the direct DFT computation of Exercise 1 and the results of FFT in each of these cases.

- **Exercise 3:** For the cases a)-e) of Exercise 1, evaluate the computational complexities of the DFT and FFT and plot the function $F(N) = C_{\text{DFT}}/C_{\text{FFT}}$ versus N, where C_{DFT} is the number of flops necessary for the DFT computation, whereas C_{FFT} is the number of flops required by FFT. Use the `semilogx` semi-logarithmic representation to plot this function and employ the standard MATLAB operator `flops` to count the number of flops. Compare the obtained dependence $F(N)$ with the order of magnitude of the corresponding theoretical dependence. Describe for what values of N the computational improvement of FFT over DFT becomes most significant.

- **Exercise 4:** Create the finite-length signal

  $$x(n) = e^{j2\pi f_1 n} + e^{j2\pi f_2 n}, \qquad 0 \le n \le M-1$$

  with $f_1 = 0.20$, $f_2 = 0.25$ and different number of points a) $M = 5$, b) $M = 10$, c) $M = 20$, d) $M = 40$, and e) $M = 80$. Using the standard MATLAB program `fft`, obtain the N-point DFT of this signal for $N = 1024$. Before computing the FFT, this function must be padded with $N - M$ zeros to form the N-point input sequence. Plot the curve of the absolute value of DFT for the cases a)-e). Can you explain the evolution of the DFT curves from a) to e)?