Lab 2: The discrete Fourier transform (DFT)

Problem 1 (DFT implementation):

(a) Write a Matlab function that takes an arbitrary signal vector $y[k]$ as its only input argument and that returns the DFT vector as computed according to the following equation:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-jkn\frac{2\pi}{N}}. \tag{1}$$

Make sure that your implementation

- does not use Matlab's `fft` function but implements equation (1) directly;
- does not exceed 10 lines of code;
- is as fast as possible.

(b) Compare the runtime of your DFT implementation against Matlab's `fft` function for an input signal with a length of 10000 sampling points.

Hints:

- Organise your code so that your DFT implementation is contained in a function whereas the code that calls and tests the function is contained in a Matlab script.
- You can test the correctness of your function by checking that the output coincides with that of Matlab's `fft` function for a short test signal (`1:4`, for instance).
- To optimise your function
  - follow the links entitled *MATLAB documentation - Preallocating Arrays* and *Vectorization* in EMIL and read the corresponding documentation; and
  - add the following code

    ```
    profile on;
    ... code to be profiled;
    profile viewer;
    ```

    to your script to use the profiler (follow the link entitled *MATLAB documentation - Profiling for Improving Performance* provided on EMIL for more information).
- It is convenient to use the functions `tic` and `toc` for sub-exercise (b).

Problem 2 (Spectrum of a rectangular signal):

In this problem, the amplitude spectrum of the signal

$$x(t) = \begin{cases} 1 & \text{for} \quad 0 \leq t \leq 2\,\text{ms} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

is to be analysed.

(a) Theoretical considerations:

  (i) Calculate the Fourier transform $X(j\omega)$ of $x(t)$ and sketch the amplitude spectrum $|X(j\omega)|$ (using pen and paper).

  (ii) At what frequency does the amplitude spectrum take on its maximum value? What is the maximum value?

  (iii) Where are the zeros of the spectrum located?

(b) Numerical calculation of the spectrum:

  (i) Use Matlab to generate and plot a vector containing the sample values of the rectangular signal defined in (2) sampled at $f_s = 8\text{kHz}$. Choose the number N of sample values so that it is a power of 2 and that the signal duration is at least 4 ms. Do not forget to label the axes of your plot.

  (ii) Use your own DFT implementation from problem 1 or Matlab's `fft` function to plot the amplitude spectrum in the frequency range $[0, f_s/2]$. Make sure that the frequency axis is labelled correctly.

  (iii) Which parameter has to be changed to increase the frequency resolution of the spectrum? Use the zeros of the spectrum to verify that the DFT approaches the expected spectrum from (a) when the frequency resolution is increased.

  (iv) How does the spectrum change when the sampling frequency f_s is increased?

2) a) i) it is a rect function shifted by 1 ms, so

$$X(jw) = 2\,S_i(w) \cdot e^{-jw}$$

Problem 3 (Dual tones):

(a) Read and understand the background section on dual tones below.

(b) Read the documentation about the Matlab function `wavread` and use it to load the file `touchtone1.wav` provided on EMIL. Play the signal with the function `soundsc`.

(c) Plot the signal in the time domain. Make sure that the time axis is scaled correctly.

(d) Plot the amplitude spectrum over the entire frequency range. Make sure the frequency axis is scaled correctly. Can you explain the peaks at large frequencies? Make another plot where the displayed frequencies are restricted to a range relevant for dual tones (`xlim` function). Do you see the expected frequencies?

(e) Write a Matlab function that takes a vector of digits as its input argument and returns a vector containing the corresponding dual tone signal. Use the following parameters:

- Sample rate $f_s = 8\text{kHz}$
- Duration of single digit signal: 75 milliseconds
- Duration of the break in between two digits: 30 milliseconds

(f) Create a dual tone signal of your own telephone number, play it and plot it in both the time and frequency domains.

Background

Dual-tone multi-frequency signaling is a method to transmit the digits $0, 1, \ldots, 9$ across an audio channel (*e.g.*, for transmitting telephone numbers). Each digit is represented as a superposition (*i.e.* as the sum) of two harmonic waves with specific frequencies as shown in the following table:

Frequency (Hz)	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

For example, the digit 1 is the sum of two harmonic waves with frequencies 697Hz and 1209Hz. In order to transmit a sequence of digits, the audio signal for each digit is transmitted only for a limited period of time. The signals for different digits are interrupted by signal breaks (zero-amplitude signal). For more details refer to http://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling for instance.