Signals and Communication Lab
DII University of Brescia - ITALY

Digital Audio Processing

Lab.2 - DFT and DFT's applications - 2014

Alessio Degani, [alessio.degani@ing.unibs.it]
http://www.ing.unibs.it/alessio.degani

1 FFT

The DFT of a sequence $x[n] \in L^2(\mathbb{Z})$ of length N is a linear transform $\mathcal{F} : \mathbb{C} \rightarrow \mathbb{C}$ defined as

$$X[k] = \mathcal{F}\{x[n]\} = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \frac{k}{N} n} \tag{1}$$

and the inverse (IDFT) is defined as

$$x[n] = \mathcal{F}^{-1}\{X[k]\} = \frac{1}{N} \sum_{n=0}^{N-1} X[k] \cdot e^{j2\pi \frac{k}{N} n}. \tag{2}$$

The FFT is a fast implementation of DFT. In Matlab, the built-in command *fft* uses *libfftw*[1] for the calculation of FFT.
Generate a $N = 20$ sequence $s(n) = 0.5^n$ (unilateral exponential) and calculate the FFT. Try to see what happens in the following cases:

- Frequency decimation (eg. 10-point FFT of $s(n)$)
- Frequency interpolation (eg. 40-point FFT of zeropadded $s(n)$)
- Temporal zero-interleaving (adding a zero between each sample)
- Temporal interpolation by a factor of 2. (using command *interpolate*)

Plot the magnitude $|X[k]|$ versus the normalized frequency $\bar{f} \in [0, 1]$.

2 Filtering using DFT

1. Consider the high-pass (derivative) digital FIR filter with impulse response $h[n] = \frac{1}{2}(\delta(n) - \delta(n-1))$. Calculate the filtered version of a signal of your choice using the multiplication in the frequency domain (indirect method). Listen the filtered signal.

2. Verify the correctness of the obtained result, comparing with the convolution (direct method) evaluated in the time domain.

[1] http://www.fftw.org/

⚠ **<u>NOTE</u>: Multiply two spectra in the DFT domain implies a circular convolution in the time domain. Modify the signals to obtain a linear convolution.**

3 Sound Effect: simple echo

Apply a 5-tap echo effect to a signal of your choice using direct or indirect method. A 5-tap echo impulse response can be for example:

$$e[n] = \sum_{t=0}^{5} \alpha^t \cdot \delta(n - t\tau) \tag{3}$$

where, for example, $\alpha = 0.7$ and the time lag in samples between two echoes $\tau = f_s \cdot m$ (f_s is the sampling frequency and m is the time lag in seconds. Try $m = 200$ ms)

4 Spectrogram: Short Time Fourier Transform

The spectrogram of a digital sequence $x[n]$ is calculated by discrete Short Time Fourier Transform. Basically, STFT is a DFT of a windowed version of $x[n]$.

$$STFT\{X[n]\}(k,m) = X(k,m) = \sum_{n \in \mathrm{supp}(w[n-Tm])} x[n]w[n-Tm] \cdot e^{-j2\pi \frac{k}{L_w} n}, \tag{4}$$

where $w[n]$ is the sliding window function of length L_w and T is the time-hop of the sliding window.
In other words

$$X(k,m) = \mathcal{F}\{x_{w_m}[n]\}, \tag{5}$$

where $x_{w_m}[n] = x[n] \cdot w[n - Tm]$.

Calculate the spectrogram of a signal of your choice with $L_w = 1024$, $T = 256$ (75% of overlap) and $w[n] = rect$. Plot the magnitude of STFT as $20\log(|X(k,m)|)$ in a time (seconds) vs. frequency (kHz) graph (use the command *imagesc*).

TIPS: to convert N-point DFT bin number k into frequency: $f(k) = \frac{k \cdot f_s}{N}$. Pay attention of Matlab indexing scheme!

Try different STFT visualization with different parameters and windows (eg. Blackman, Hamming, ...see the Matlab command *window*).