Digital Signal Processing — DSP
(PDS — Processamento Digital de Sinais)
Instituto Superior Técnico - 2. Semester 2009/2010
João Pedro Gomes and Jorge Salvador Marques

The goal of this lab project is to introduce some of the tools for frequency-domain analysis of signals in MATLAB based on the discrete Fourier transform and the short-time Fourier transform.

Discrete Fourier Transform (DFT): A discrete-time signal $x(n)$ may be represented by a weighted sum of complex exponentials as

$$x(n) = \frac{1}{2\pi} \int_0^{2\pi} X(e^{j\omega}) e^{j\omega n} d\omega \,,$$

where $X(e^{j\omega})$ is the 2π-periodic Fourier transform of $x(n)$, and ω belongs to any interval of length 2π. Usually this interval is taken as $[0,\, 2\pi]$ or $[-\pi,\, \pi]$. When $x(n)$ is a finite-length sequence, such that $x(n) = 0$ outside the range $0 \leq n \leq N-1$, the continuous periodic function $X(e^{j\omega})$ may be fully regenerated from N values $X(e^{j\frac{2\pi}{N}0})$, $X(e^{j\frac{2\pi}{N}1})$, $\ldots X(e^{j\frac{2\pi}{N}(N-1)})$. The DFT is a transform that maps an input vector of samples in the time domain into an output vector of samples in the frequency domain

$$\mathbf{x} = \begin{bmatrix} x(0) & x(1) & \ldots & x(N-1) \end{bmatrix} \longleftrightarrow \mathbf{X} = \begin{bmatrix} X(0) & X(1) & \ldots & X(N-1) \end{bmatrix} \,,$$

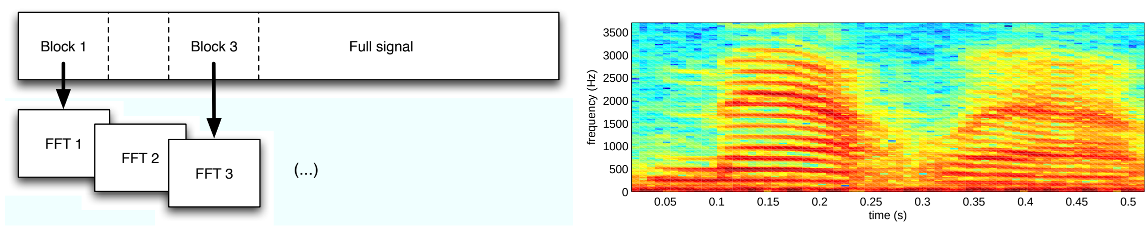
such that $X(k) = X(e^{j\frac{2\pi}{N}k})$.

In MATLAB the DFT is computed by the `fft` command. If the length N is a power of two then the transform is computed with a highly efficient fast Fourier transform (FFT) algorithm. In most practical cases plotting the output vector $\mathbf{X}$ for moderate N is enough to obtain an adequate representation of the spectrum $X(e^{j\omega})$.

Sampling issues in continuous-time and discrete-time spectra: When a discrete-time signal x is obtained by sampling of a continuous-time signal x_c every T seconds, $x(n) = x_c(nT)$, and aliasing effects can be neglected, then their spectra have essentially the same shape.

Many MATLAB functions that display continuous spectra actually plot a discrete spectrum and simply rescale the frequency and magnitude axes based on a sampling frequency $f_s = \frac{1}{T}$ provided by the user.

Short-Time Fourier Transform (STFT): When the frequency content of a signal changes significantly over time calculating a single global spectrum may convey too little information. For example, it is inappropriate to try to regenerate the sequence of phonemes in a speech signal recorded over a few seconds based only on its full spectrum. In such cases, one may resort to the STFT, which breaks a long signal into smaller, possibly overlapping, blocks, and calculates the DFT in each one as illustrated in Fig. 1a. The STFT has numerous applications in speech, sonar,



(a) (b)

Figure 1: Time-frequency representations (a) The concept of STFT with sliding and partially overlapping windows (b) A sample spectrogram

and radar processing. The *spectrogram* of a sequence is the magnitude of the STFT versus time, i.e., a 2D concatenation of the DFTs calculated for the various positions of the sliding analysis window (Fig. 1b). In MATLAB the spectrogram can be easily computed with the `spectrogram` command.

Experimental work

1. Generate a 2 s long segment of a sine wave with frequency 1 kHz, sampled at $f_s = 10$ kHz. Compute its DFT and verify that the peaks are located at the expected discrete frequencies. Play the signal through the speakers using (i) the `sound` command and (ii) the signal viewer in `sptool`.

2. Apply the memoryless transformation $y = \arctan\left(\frac{2x+1}{4}\right)$ to each sample of the previous signal and recompute the DFT. Interpret your results.

3. Generate a segment of the following chirp signal

$$\cos\left[\frac{1}{k}\log(1 + k\omega_0 n)\right] , \qquad\qquad \omega(n) = \frac{1}{\frac{1}{\omega_0} + kn} .$$

   Choose the parameters ω_0 and k such that the instantaneous frequency $\omega(n)$, given above, decreases from $\frac{2\pi}{5}$ at time 0 to $\frac{2\pi}{20}$ at the end of a 10^4 sample block. Plot the spectrum and comment on the usefulness of the DFT to capture the spectral dynamics of this signal.

4. Build a spectrogram of the chirp signal using blocks of length 512 and 25% overlap (128 samples). Plot it using `pcolor` or `surf` (you will find `shading('flat')` useful) and confirm the theoretical evolution of the instantaneous frequency.

5. Invoke `specgramdemo` with no arguments to visualize the default spectrogram of a speech signal. Why does the plot appear to be horizontally layered?