4. The discrete Fourier transform (DFT)

1. Application goal

We study the discrete Fourier transform (DFT) and its applications: spectral analysis and linear operations as convolution and correlation. We use Matlab as a simulation environment.

2. Definitions and properties of the DFT

The DFT is a Fourier transform applied to a input signal that is discrete. Unlike the discrete-time Fourier transform (DTFT), it only evaluates enough frequency components to reconstruct the analyzed signal of finite support.

Consider a sequence $x:\mathbb{Z}\to\mathbb{C}$ such that $x[n]=0$ for $n<N_1$ and $n>N_2$. We define its DFT of the sequence $x:\mathbb{Z}\to\mathbb{C}$ by:

$$X[k]=\begin{cases}\sum_{n=N_1}^{N_2} x[n]e^{-jk\frac{2\pi}{N}n}, & N_1\le k\le N_2\\ 0, & \text{otherwise}\end{cases} \quad (1)$$

where

$$N=N_2-N_1+1 \quad (2)$$

The periodic signal $\tilde{x}[n]$ of period N is obtained from $x[n]$ extended by periodicity:

$$\tilde{x}[n]=\sum_{m=-\infty}^{\infty} x[n-mN] \quad (3)$$

Denote:

$$e^{-j\frac{2\pi}{N}}=w_N \quad (4)$$

Consider the signal:

$$\tilde{X}[k]=\sum_{n=\langle N\rangle}\tilde{x}[n]w_N^{kn} \quad (5)$$

Here $n=\langle N\rangle$ means the sum is made for N consecutive values of the signal $x[n]$ (one period). $\tilde{X}[k]$ is periodic, of period N, and:

$$\tilde{X}[k]=X[k],\quad N_1\le k\le N_2 \quad (6)$$

The original signal is reconstructed from its DFT using:

$$x[n]=\frac{1}{N}\sum_{k=N_1}^{N_2} X[k]w_N^{-kn} \quad (7)$$

$$\tilde{x}[n]=\frac{1}{N}\sum_{k=\langle N\rangle}\tilde{X}[k]w_N^{-kn} \quad (8)$$

When we use (8), it results that:

$$x[n]=\tilde{x}[n]\left\{\sigma[n-N_1]-\sigma[n-N_2]\right\} \quad (9)$$

($\sigma[n]$ is the discrete-time unit step).

Consider the sequences $x_i[n]$ with the corresponding DFT's $X_i[k]$, i=1,2. Main properties of the DFT, $\tilde{X}_i[k]$, for the corresponding periodic signals $\tilde{x}_i[n]$, are:

a) <u>Linearity</u>

$$a\tilde{x}_1[n]+b\tilde{x}_2[n]\leftrightarrow a\tilde{X}_1[k]+b\tilde{X}_2[k],\quad \text{a,b}\in \text{C} \tag{10}$$

b) <u>Time shifting</u>

$$\tilde{x}[n-n_0]\leftrightarrow w^{kn_0}\tilde{X}[k],\quad n_0\in \text{Z} \tag{11}$$

c) <u>Modulation in time</u>

$$w^{-k_0 n}\tilde{x}[n]\leftrightarrow \tilde{X}[k-k_0],\quad k_0\in \text{Z} \tag{12}$$

d) <u>Circular convolution</u>

Circular convolution is denoted with $\circledast$. It is convolution for periodic signals, with the same period *N*, whose result is another sequence, periodic of period *N*:

$$\tilde{x}[n]=\sum_{m=\langle N\rangle}\tilde{x}_1[m]\tilde{x}_2[n-m] \tag{13}$$

The DFT is:

$$\tilde{x}_1[n]\circledast\tilde{x}_2[n]\leftrightarrow \tilde{X}_1[k]\cdot\tilde{X}_2[k] \tag{14}$$

e) <u>Modulation in time</u>

$$\tilde{x}_1[n]\cdot\tilde{x}_2[n]\leftrightarrow \frac{1}{N}\tilde{X}_1[k]\circledast\tilde{X}_2[k] \tag{15}$$

f) <u>Parseval's theorem</u>

$$\sum_{n=\langle N\rangle}\left|\tilde{x}[n]\right|^2=\frac{1}{N}\sum_{k=\langle N\rangle}\left|\tilde{X}[k]\right|^2 \tag{16}$$

3. Fast Fourier algorithm for DFT computation

The expansion of digital signal processing began with the development of the algorithm for the computation of the DFT: the Fast Fourier transform (FFT). This exploits some properties of the complex exponentials. The FFT implementation is efficient when the length of the sequence in time or the period is a power of 2. If this requirement is not fulfilled, the interval can be extended until it becomes a power of 2, by adding null samples.

4. Applications for DFT

4.1. Spectral analysis of continuous-time signals

This is based on the sampling theorem for finite energy signals. Consider a continuous-time signal, $x(t)$, with the length

$$\text{supp}\{x(t)\}\subset[0,T] \tag{17}$$

and the signal resulted by periodicity (see Fig.1):

$$x_T(t)=\sum_{n=-\infty}^{\infty}x(t-nT) \tag{18}$$

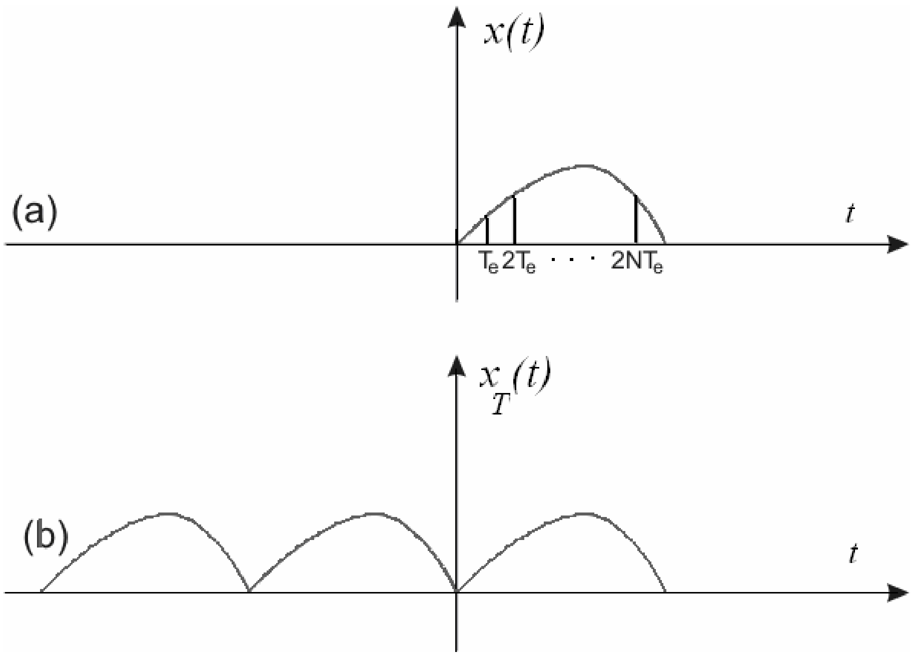

Fig.1 Analyzed signal (a); Periodized signal (b)

Denote by $X(\omega)$ the Fourier transform of the signal $x(t)$ and assume $x_T(t)$ can be expanded in a Fourier series with a finite number of terms:

$$x_T(t) = \sum_{k=-N}^{N} a_k e^{jk\frac{2\pi}{T}t} \tag{19}$$

We sample $x(t)$ with the step:

$$T_S = \frac{T}{2N+1} \tag{20}$$

and consider the discrete-time signal

$$x[n] = x(nT_s) \tag{21}$$

with the DFT $X[k]$.

We can see that,

$$a_k = X[k] + \frac{1}{T} X\left(k\frac{2\pi}{T}\right) \tag{22}$$

meaning the elements of the DFT are proportional with the samples of the continuous-time Fourier transform of signal $x(t)$. The signal $x(t)$ can be reconstructed from its samples (21), if the sampling step satisfies (20).

In reality it is less likely that the sampling fulfills the above mentioned condition, or it is impossible. In this case, the expansion of $x_T(t)$ contains an infinite number of terms:

$$x_T(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\frac{2\pi}{T}t} \tag{23}$$

Thus, aliasing of coefficients appears after sampling:

$$X[k] = \sum_{p=-\infty}^{\infty} c_{k+(2N+1)p} = \frac{1}{T} \sum_{p=-\infty}^{\infty} X\left(\left(k+(2N+1)p\right)\frac{2\pi}{T}\right) \quad (24)$$

If the Fourier transform $x(\omega)$ is small for $|\omega| > N2\pi/T$, then the values $TX[k]$ are accurate estimates of the samples of this transform in $k2\pi/T$, $k=[-N,N]$. In some spectral analyzers with FFT, linear interpolation of samples $X[k]$ is used.

If the signal $x(t)$ has a large support, we apply the technique of the short time Fourier transform (STFT), see Fig. 2. Basically, segments of the signals are analyzed. The segments are obtained by multiplication of the signal with a window. The simplest case is the rectangular pulse, but in order to avoid Gibbs phenomenon, other signals are used such as Hamming, Gauss, etc.

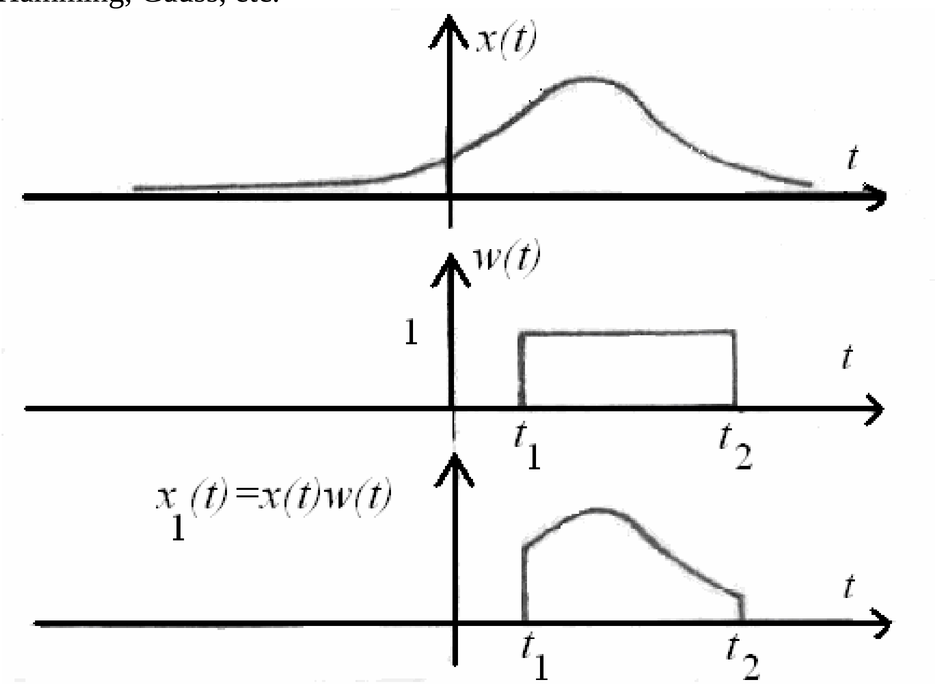

Fig. 2 (a) Original signal; (b) Rectangular pulse; (c) Analyzed signal.

4.2. Computation of convolution

Consider two sequences $x_1[n]$ and $x_2[n]$:

$$\begin{aligned} x_1[n] &= 0 \text{ for } n < N_1 \text{ and } n > N_2 \\ x_2[n] &= 0 \text{ for } n < N_3 \text{ and } n > N_4 \end{aligned} \quad (25)$$

For the computation of convolution of the sequences, the following steps are performed:

- Obtain signals $\tilde{x}_1[n]$ and $\tilde{x}_2[n]$ by periodicity

$$N \geq N_2 - N_1 + N_4 - N_3 + 1 \quad (26)$$

that is a power of 2; thus the convolution of the two signals can be recovered from the circular convolution of the periodic signals.

- Compute $\tilde{X}[k] = \tilde{X}_1[k]\tilde{X}_2[k]$
- Compute inverste DFT, $\tilde{x}[n]$, the circular convolution of $\tilde{x}_1[n]$ and $\tilde{x}_2[n]$

- Compute $x[n]$:

$$\tilde{x}[n] = \{\sigma[n - N_1 - N_3] - \sigma[n - N_2 - N_4 - 1]\} \quad (27)$$

4.3. Computation of correlation

Consider the same signals as before (eq. (25)), but real-valued. Their intercorrelation function is:

$$R_{X_1X_2}[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[m+n] = x_1[-n] * x_2[n] \quad (28)$$

To evaluate this function, same steps are made, using convolution betwen the first signal reversed in time, and the second signal.

5. Practical part

5.1. Consider the following signals:

(a) $x_1(t)$, din fig.3.a (rectangular pulse)

(b) $x_2(t)$, din fig.3.b (triangle pulse)

(c) $x_3(t) = A\sin(\omega_0 t)\left[\sigma(t) - \sigma(t-T)\right]$ (a sine)

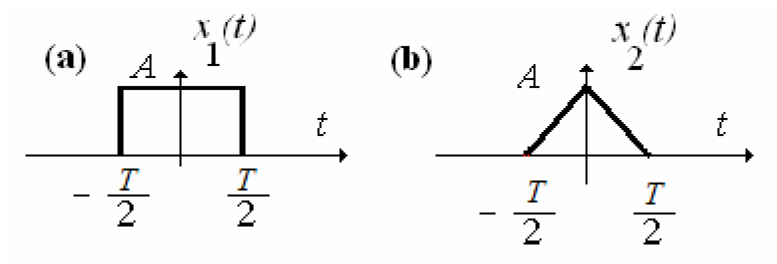

Fig. 3. Signals $x_1(t)$ and $x_2(t)$

Parameters A, T and ω_0 are chosen in the program „*lfft*” by the user.

Compute and plot the CTFT $X_1(\omega)$, $X_1(\omega)$ and $X_3(\omega)$.

Compute the DFT of the discrete-time signals that result by sampling the three signals, with 8, 16 and 32 samples per period T. Use the Matlab program „*lfft*”. Compare the last two members of eq. (28) (see also (24)) for each sample.

Compute the DFT using 256 samples. Compare the plots obtained using the program with the computed CTFT and find the maximum error.

5.2. Compute the convolution of the signals from 5.1:

$$x_1(t) * x_1(t),\ x_1(t) * x_2(t),\ x_2(t) * x_2(t)$$

Plot the results. Use 512 samples and compute their convolution using the program “*lfft*”. Compare the plots obtained by computation and the ones from Matlab.