Semnale și sisteme

Lucrare de laborator nr. 13

Aplicatii ale Transformatei Fourier Discrete

<u>FFT</u> (*Transformata Fourier Rapida - Fast Fourier Transform*) este un algoritm eficient de calcul a transformatei Fourier discrete (DFT)

$$X[n] = X\left(\frac{2\pi n}{N}\right) = \sum_{k=0}^{N-1} x[k]\cdot W_N^{kn}, \quad n = \overline{0, N-1}, \quad W_N = e^{-j\frac{2\pi}{N}}$$

si a transformatei Fourier discrete inverse (IDFT):

$$x[k] = \frac{1}{N}\sum_{n=0}^{N-1} X[n]\cdot W_N^{-kn}, \qquad k = \overline{0, N-1}$$

In Matlab aceste transformate pot fi calculate utilizand functiile *fft* si *ifft*. Functiile *fft* si *ifft* implementeaza conventii usor diferite decat cele prezentate anterior. Daca secventa de intrare x are lungimea N, atunci $X = fft(x)$ este un vector cu N elemente:

$$X[n] = \sum_{k=1}^{N} x[k]\cdot e^{-j2\pi\frac{(k-1)(n-1)}{N}}, \quad n = \overline{1, N}$$

si daca X are lungimea N atunci $x = ifft(X)$ este un vector x cu N elemente:

$$x[k] = \frac{1}{N}\sum_{n=1}^{N} X[n]\cdot e^{j2\pi\frac{(k-1)(n-1)}{N}}, \qquad k = \overline{1, N}$$

Algoritmul Cooley-Tukey pentru calculul DFT

Daca $N = 2M$ atunci pentru $k = \overline{0, N-1}$,

$$X_k = x_0 + x_1 W_N^{-k} + x_2 W_N^{-2k} + x_3 W_N^{-3k} + \cdots + x_{N-1} W_N^{-(N-1)k}$$
$$= x_0 + x_2 W_N^{-2k} + \cdots + x_{N-2} W_N^{-(N-2)k} + x_1 W_N^{-k} + x_3 W_N^{-3k} + \cdots + x_{N-1} W_N^{-(N-1)k}$$
$$= x_0 + x_2 W_M^{-k} + \cdots + x_{N-2} W_M^{-(M-1)k} + W_N^{-k}\left[x_1 + x_3 W_M^{-k} + \cdots + x_{N-1} W_M^{-(M-1)k}\right]$$

$W_N^2 = W_M$

Definind secventele (x_{en}) si (x_{on}) de lungime M prin:

$(x_{e0}, x_{e1}, \dots, x_{e(M-1)}) = (x_0, x_2, \dots, x_{N-2})$, $(x_{o0}, x_{o1}, \dots, x_{o(M-1)}) = (x_1, x_3, \dots, x_{N-1})$

si notand

$\{x_{en}\} \xleftrightarrow[M]{\text{DFT}} \{X_{en}\}$ $\qquad$ $\{x_{on}\} \xleftrightarrow[M]{\text{DFT}} \{X_{on}\}$

atunci pentru $k = \overline{0, M-1}$, $X_k = X_{ek} + W_N^{-k} X_{ok}$ si prin periodicitatea DFT

$$X_{k+M} = X_{ek} + W_N^{-(k+M)} X_{ok} = X_{ek} - W_N^{-k} X_{ok}$$

Prin urmare, X_k poate fi determinat pentru $k = \overline{0, N-1}$, de urmatoarele doua ecuatii:

$X_k = X_{ek} + W_N^{-k} X_{ok}$,

$X_{k+M} = X_{ek} - W_N^{-k} X_{ok}$

pentru $k = \overline{0, M-1}$.

Prin urmare *transformata Fourier discreta* ***de ordinul 2M*** *poate fi calculata utilizand* ***doua transformate DFT de ordin M***, *M* inmultiri complexe si 2*M* adunari complexe.

Daca *M* este par acest algoritm poate fi aplicat din nou, iar daca *M* este o putere a lui 2 poate fi utilizat recursiv pentru a calcula integral transformata.

Convolutia circulara, convolutia liniara si DFT

Pentru doua secvente de aceeasi lungime N,

convolutia circulara este

$$z[k]=\{x_1 \otimes x_2\}[k]=\sum_{n=0}^{N-1} x_1[n]\cdot x_2[(k-n)]_N \ , \quad k=\overline{0,N-1} \qquad x[k-n,\bmod N]\equiv x[(k-n)]_N$$

convolutia liniara este

$$z[k]=\{x_1 * x_2\}[k]=\sum_{n=0}^{N-1} x_1[n]\cdot x_2[k-n] \quad , \quad k=\overline{0,2N-2}$$

Multiplicarea DFT-urilor a două secvenţe este echivalentă cu convoluţia circulară a celor două secvenţe în domeniul timp.

Calculul convolutiei circulare utilizand DFT implica 3 pasi:

1. calculul DFT pentru fiecare semnal,
2. inmultirea secventelor transformate,
3. calculul IDFT.

Transformata Fourier discreta poate fi utilizata si pentru calculul convolutiei liniare daca secventele sunt completate cu valori nule

Exemplu: Sa se calculeze convolutia circulara dintre secventele x = [2 0 1 0] si y = [0 2 2 0].

a) Calculul convolutiei circulare utilizand DFT

- calcul DFT pentru cele doua secvente

$X[n]=\sum_{k=0}^{N-1} x[k]e^{-j\frac{2\pi}{N}kn},\ n=\overline{0,N-1}, N=4$

$X[n]=x[0]+x[1]e^{-j\frac{\pi}{2}n}+x[2]e^{-j\pi n}+x[3]e^{-j\frac{\pi}{2}3n}$

$n=0,\ X[0]=2+e^0=3$

$n=1,\ X[1]=2+e^{-j\pi}=1$

$n=2,\ X[2]=2+e^{-j2\pi}=3$

$n=3,\ X[3]=2+e^{-j3\pi}=1$

$X=\{3,1,3,1\}$

$Y[n]=y[0]+y[1]e^{-j\frac{\pi}{2}n}+y[2]e^{-j\pi n}+y[3]e^{-j\frac{\pi}{2}3n}$

$n=0,\ Y[0]=2e^0+2e^0=4$

$n=1,\ Y[1]=2e^{-j\frac{\pi}{2}}+2e^{-j\pi}=-2j-2$

$n=2,\ Y[2]=2e^{-j\pi}+2e^{-j2\pi}=0$

$n=3,\ Y[3]=2e^{-j\frac{\pi}{2}3}+2e^{-j3\pi}=2j-2$

$Y=\{4,-2j-2,0,2j-2\}$

- inmultire transformate

Z = {12, -2-2j, 0, -2 + 2j}

- calcul IDFT

$z[k]=\frac{1}{N}\sum_{n=0}^{N-1} Z[n]e^{j\frac{2\pi}{N}kn},\ k=\overline{0,N-1}$

$z[k]=\frac{1}{4}(Z[0]+Z[1]e^{j\frac{\pi}{2}k}+Z[2]e^{j\pi k}+Z[3]e^{j\frac{\pi}{2}3k})$

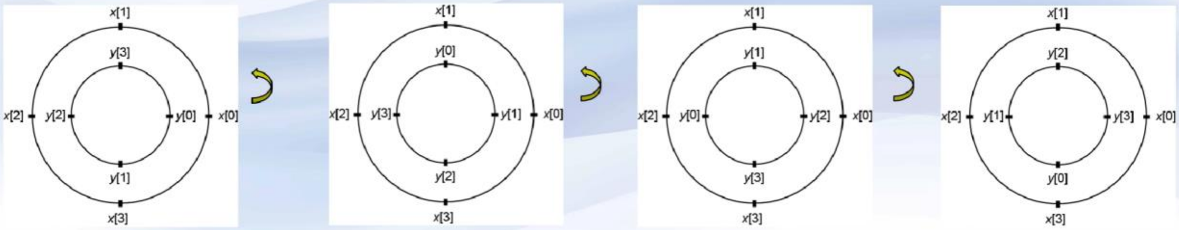
$z=\{2,\ \ 4,\ \ 4,\ \ 2\}$

Secventa de cod pentru calculul convolutiei circulare utilizand DFT in Matlab

```
x = [2 0 1 0], y = [0 2 2 0]
X = fft(x)
Y = fft(y)
Z = X.*Y
z = ifft(Z)
```

b) Calculul convolutiei circulare - metoda cercurilor concentrice

1. se considera doua cercuri concentrice
2. se reprezinta valorile secventelor pe cele doua cercuri (la distanta egala) (valorile $y[n]$ se reprezinta pe cercul interior in directia acelor de ceasornic; valorile $x[n]$ in directie opusa acelor de ceasornic pe cercul exterior, prima valoare a celor doua secvente fiind in acelasi punct)
3. se inmultesc valorile corespunzatoare de pe cele doua cercuri si se insumeaza
4. se roteste cercul interior in directie opusa acelor de ceasornic
5. se repeta pasii 3-4 pana cand cercul interior ajunge in pozitia initiala



$z[0] = x[0]y[0] + x[1]y[3] + x[2]y[2] + x[3]y[1] = 2$

$z[1] = x[0]y[1] + x[1]y[0] + x[2]y[3] + x[3]y[2] = 4$

$z[2] = x[0]y[2] + x[1]y[1] + x[2]y[0] + x[3]y[3] = 4$

$z[3] = x[0]y[3] + x[1]y[2] + x[2]y[1] + x[3]y[0] = 2$

Secventa de cod pentru calculul convolutiei circulare in Matlab

```
x = [2 0 1 0], y = [0 2 2 0], N = 4,
z = cconv(x, y, N)
```

<u>Exemplu</u>: Se considera doua secventele discrete *x* si *y*. Sa se calculeze convolutia liniara in Matlab utilizand *conv* si *fft*.

Pentru calculul convolutiei liniare utilizand DFT este necesara completarea cu valori nule a celor doua secvente. Astfel daca

X = *fft*([x zeros(1,length(y)-1)]) si Y = *fft*([y zeros(1,length(x)-1)])

atunci conv(x,y) = ifft(X.*Y)

```
x=[1 2 3 4]
y=[0 1 2 3]
X=fft([x, zeros(1,length(y)-1)])
Y=fft([y, zeros(1,length(x)-1)])
Z=X.*Y
z=ifft(Z)

zlin=conv(x,y)
```

Aproximarea semnalelor audio prin serii Fourier finite

Estimarea coeficientilor seriei Fourier utilizand FFT

$$x(t) = a_0 + 2\sum_{k=1}^{M}\left[B_k \cos k\omega_0 t - C_k \sin k\omega_0 t\right]$$

$$B_k = A_k \cos\theta_k \qquad C_k = -A_k \sin\theta_k$$

$$x(t) = a_0 + 2\sum_{k=1}^{M} A_k \cos(k\omega_0 t + \theta_k)$$

$$A_k = \sqrt{B_k^2 + C_k^2} \qquad \tan(\theta_k) = B_k / C_k$$

(1/N)*real(fft(x))

$[a_0\ B_1\ B_2\ \dots\ B_{N/2-2}\ B_{N/2-1}\ B_{N/2}\ B_{N/2-1}\ B_{N/2-2}\ \dots\ B_2\ B_1]$

(1/N)*imag(fft(x))

$[0\ C_1\ C_2\ \dots\ C_{N/2-2}\ C_{N/2-1}\ -C_{N/2}\ -C_{N/2-1}\ -C_{N/2-2}\ \dots\ -C_2\ -C_1]$

1/N*abs(fft(x))

$[a_0\ A_1\ A_2\ \dots\ A_{N/2-2}\ A_{N/2-1}\ A_{N/2}\ A_{N/2-1}\ A_{N/2-2}\ \dots\ A_2\ A_1]$

angle(fft(x))

$[0\ (\text{sau}\ \pi)\ -\theta_1\ \dots\ -\theta_{N/2-1}\ 0\ (\text{sau}\ \pi)\ \theta_{N/2-1}\ \dots\ \theta_2\ \theta_1]$

Frecventa asociata termenului k (k = index Matlab - 1) este $f = Fs\ k\ /\ N$

```
load train
figure, plot([0:length(y)-1]/Fs, y), xlabel('t'), ylabel('x(t)')
sound(y, Fs)

N = 2000; n = [0:N-1]; x = y(8000+n)';
figure, subplot(311), t=[0:N-1]/Fs; plot(t, x), axis([0 0.03 -1 1])
subplot(312), Xf=fft(x); plot(abs(Xf)/N), ylabel('Spectrul semnalului x')

% 3 varfuri in spectru la indicii Matlab 173 217 286
k1 = [173 217 286]
Ak = abs(Xf(k1))/N
phik = angle(Xf(k1)) % faze (radiani)
fk = (k1-1) / N * Fs % frecvente (Hz)

z =  2*Ak(1)*cos(2*pi*fk(1)*t+phik(1)) …
   + 2*Ak(2)*cos(2*pi*fk(2)*t+phik(2)) …
   + 2*Ak(3)*cos(2*pi*fk(3)*t+phik(3));
subplot(311), hold on, plot(t, z, 'g--'),
subplot(313), Xz=fft(z); plot(abs(Xz)/N, 'g--'), ylabel('Spectrul semnalului z')
```

Exercitii

Pentru un sunet produs de un instrument muzical (se va utiliza un fisier audio la alegere),

a) sa se realizeze analiza Fourier pentru 20 de cadre de timp succesive, considerand ca un cadru de timp contine 1024 de esantioane si suprapunerea dintre cadre este de 50% (a se vedea nota din josul paginii pentru informatii referitoare la durata unui cadru de timp)

b) sa se compare valorile amplitudinilor A_k obtinute cu *fft* cu cele obtinute utilizand metoda trapezelor de integrare numerica utilizata la Laboratorul 8.

c) sa se calculeze centrul de greutate spectrala pentru fiecare cadru $CGS = \frac{\sum_{k=1}^{N} kA_k}{\sum_{k=1}^{N} A_k}$,

A_k sunt amplitudinile armonicilor (se vor lua in considerare primele 24 de armonici).

Sa se reprezinte grafic spectrul de amplitudine si sa se figureze pe grafic pozitia CGS

d) Pentru fiecare cadru de timp sa se calculeze parametrul IRR (gradul de neregularitate al spectrului)

$$IRR = log_{10}\left(\sum_{k=2}^{n-1}\left|20log_{10}(A_k) - \frac{20log_{10}(A_{k+1}) + 20log_{10}(A_k) + 20log_{10}(A_{k-1})}{3}\right|\right)$$

Nota: Spre exemplu, pentru o frecvență de eșantionare 44.100Hz, considerand ca numarul de esantioane dintr-un cadru de timp este 1024, durata unui cadru de timp este 23,22ms.