Necula Madalina Andreea
422D

Tema 4:Transformata Fourier și Laplace

Tema_4_1

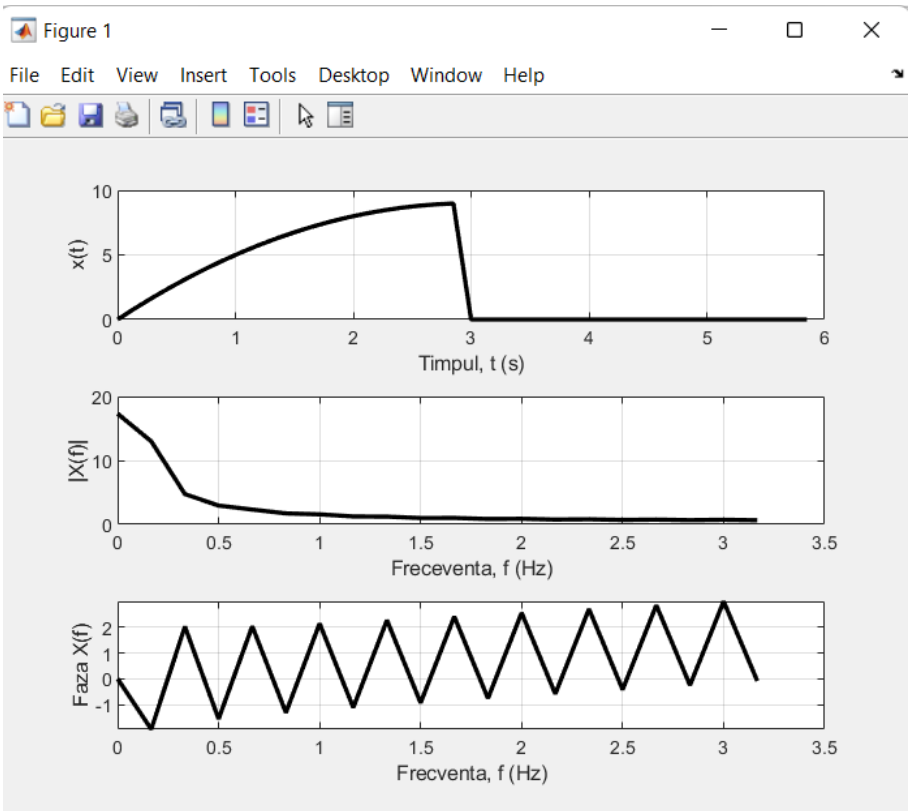
```
%% Calculul transformatei Fourier folosind 40 de esantioane in intervalul
%% 0-2 secunde

clear all ; clf

N = 40 ;   % nr de esantioane pt esantionarea semnalului pt cele 6 secunde
Ts = 6/N ; % perioada de esantionare raportata la 6 secunde

fs = 1/Ts ; % frecventa de esantionare
df = fs/N ; % rezolutia în domeniul frecventa
n = [0:N-1]' ; % vectorul coloana indicilor esantionati de timp
t = Ts*n ; % vectorul valorilor timpului la momentele de esantionare
x = (9-(t-3).^2).*(t>0).*(t<3); % vectorul valorilor functiei în momentele de esantionare.
X = Ts*fft(x) ; % transformata fourieri discret
k = [0:N/2-1]' ; % valorile indicelui freceventelor
%X = fftshift(Ts*fft(x)) ; %folosit pentru a afisa tot spectrul semnalului
%k = [-N/2:N/2-1]'; % folosit impreuna cu fftshift

%Graficele
subplot(3,1,1) ;
p = plot(t,x,"k") ; set(p,"LineWidth",2) ; grid on ;
xlabel("Timpul, t (s)") ; ylabel("x(t)") ;
subplot(3,1,2) ;
p = plot(k*df,abs(X(1:N/2)),"k") ; set(p, "LineWidth",2) ; grid on;
%p = plot(k*df,abs(X(1:N/2)),"k") ; set(p, "LineWidth",2) ; grid on; % folosit impreuna cu
fftshift
xlabel("Freceventa, f (Hz)") ; ylabel("|X(f)|") ;
subplot(3,1,3) ;
p = plot(k*df,angle(X(1:N/2)),"k") ; set(p,"LineWidth",2) ; grid on ;
%p = plot(k*df,angle(X(1:N/2)),"k") ; set(p,"LineWidth",2) ; grid on ; % folosit impreuna cu
fftshift
xlabel( "Frecventa, f (Hz)") ; ylabel("Faza X(f)") ;
```



Necula Madalina Andreea
422D

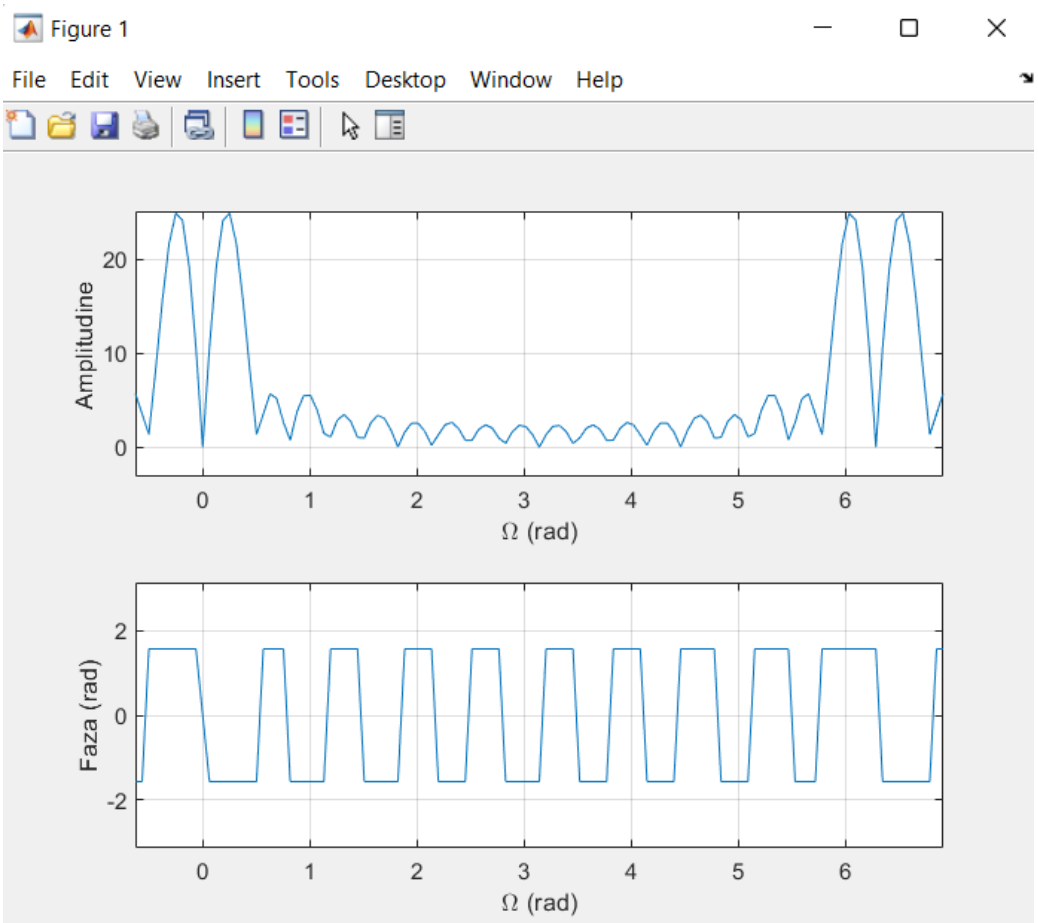
Tema_4_2

```
%% Calculul transformatei Fourier in timp discret
clear all;

n = linspace(0,10,11);
xn1 = (n>0).*(n<10).*(log(n+1));
puteri1=linspace(0,10,11); %puterile exponentialei din formula transformatei
n2 = linspace(-10,0,11);
xn2 = (n2>-10).*(n2<0).*(-1*log(-1*n2+1));
puteri2=linspace(-10,0,11);%puterile exponentialei din formula transformatei
Omega = [-0.1:0.01:1.1]*2*pi; %vectorul 'Omega' necesar calcului
                              %transformatei Fourier în timp discret
TFTD= sum(xn1*exp(-1i*puteri1'*Omega),1)+sum(xn2*exp(-1i*puteri2'*Omega),1); %transformata
Fourier în timp discret

% Grafic
figure(1)
subplot(211);
plot(Omega,abs(TFTD)); grid;
axis([-0.2*pi,2.2*pi,-pi,8*pi]);
xlabel('\Omega (rad)');
ylabel('Amplitudine')

subplot(212)
plot(Omega,angle(TFTD)); grid;
axis([-0.2*pi,2.2*pi,-pi,pi]);
xlabel('\Omega (rad)');
ylabel('Faza (rad)');
```



Necula Madalina Andreea
422D

Tema_4_3

Editor - D:\Users\Madalina\Documents\MATLAB\Tema_4_3.m

Tema_4_1.m | Tema_4_2.m | Tema_4_3.m | Tema_4_4.m

```
1
2   %% Calculul transformatei Laplace
3   %% pkg load symbolic %pentru octave decomentati aceasta linie
4   syms x omega t;
5   x = cos(omega*t);
6   X = laplace(x) %funcție care generează transformata laplace pentru scrierea simbolica
7
```

Command Window

```
>> Tema_4_3

X =

s/(omega^2 + s^2)
```

Tema_4_4

Editor - D:\Users\Madalina\Documents\MATLAB\Tema_4_4.m

Tema_4_1.m | Tema_4_2.m | Tema_4_3.m | Tema_4_4.m

```
1
2   %% Calculul transformatei Laplace inverse
3   syms X s x ; %simbolurile
4   X = (5*s-1)/(s^3-3*s-2); %transformata Laplace
5   x = ilaplace(X) %transformata Laplace inversa
6
7   %% Calculul parametrilor semnalului
8   numaratorul = [5 -1]; %coeficientii numaratorului
9   numitorul = [1 0 -3 -2]; %coeficientii numitorului
10  [r,p] = residue(numaratorul,numitorul); %calculul reziduurilor (r) si polilor(p)
```

Command Window

```
x =

exp(2*t) - exp(-t) + 2*t*exp(-t)

>> r

r =

    1.0000
   -1.0000
    2.0000

>> p

p =

    2.0000
   -1.0000
   -1.0000
```