**Proiect PDS: Tema 1**

**Frunza Vladimir, 431C**

**30/04/2024**

**Cerinta generala:**

**I. Semnale în timp discret și reprezentarea spectrelor**

Se vor preciza în toate graficele titlul şi mărimile reprezentate pe axele x şi y. Funcțiile necesare cerințelor proiectului se vor implementa în același fișier cu proiectul la sfârșitul fișierului (vezi exemplul demofer.m).

a) Realizați un proiect în Matlab în care se vor realiza și apela funcțiile precizate mai jos. Fiecare student are repartizat un tip de semnal în   
tabelul 1.

- O funcție pentru generarea și reprezentarea grafică a semnalului analogic xa(t) de tipul și cu parametrii conform tabelelor 1 și 2 de la sfârșitul fișierului. Funcția generază și reprezintă grafic (cu funcția plot - axa timp în milisecunde) semnalul analogic.

- O funcție pentru generarea și reprezentarea grafică a semnalul discret x(n) obținut prin eșantionarea semnalului analogic xa(t) cu frecvența de eșantionare Fs. Se va genera direct semnalul discret fără a se folosi semnalul analogic xa(t). Funcția generază semnalul discret și reprezintă grafic (cu funcția stem - axa timp în funcție de n) semnalul discret. Din calcule și verificând în graficele obținute pentru semnalul discret se determină:

* numărul total de eșantioane Nes pentru x(n) corespunzător TMAX,
* numărul de eșantioane N pentru care semnalul x(n) este periodic,
* câte perioade P sunt cuprinse în durata de achiziție TMAX.

b) Calculați și reprezentați spectrele semnalului x(n) folosind funcția implementata conform specificațiilor de mai jos.

- O funcție în care se calculează Transformata Fourier Discretă (TFD), cu ajutorul funcției Matlab fft, pentru semnalul discret x(n). Funcția primește parametrii: semnalul discret x(n), frecvența de eșantionare Fs și numărul de puncte NFFT. Dacă NFFT nu este transmis ca parametru, implicit el se va lua egal cu lungimea semnalului de intrare Nes. Funcția calculează TFD în NFFT puncte și reprezintă următoarele spectre:

* Spectrul de amplitudine |X(k)| în funcție de indicele k al TFD.
* Spectrul de amplitudine și de fază în funcție de frecvența normată f.
* Spectrul de amplitudine în funcție de frecvența nenormată F [Hz].

Pentru cazul (implicit) NFFT egal cu Nes:

* Determinați pe grafic (folosind cursorul) indexul k corespunzător frecvenței fundamentale F0. Ce relație este între k și F0?
* Determinați pe grafic (folosind cursorul) frecvența normată f0 corespunzătoare frecvenței fundamentale F0 și frecvențele normate corespunzătoare armonicelor.
* Determinați pe grafic (folosind cursorul) frecvențele în Hz și amplitudinea corespunzătoare componentei continue, fundamentalei F0 și armonicelor. La ce frecvențe apar componentele armonice?

Pentru cazul NFFT dat conform tabelului 1:

* Determinați pe graficul spectrului de amplitudine |X(k)| în funcție de indicele k (folosind cursorul) noul index k0 corespunzător frecvenței fundamentale F0.
* Determinați pe graficul spectrului de amplitudine în frecvente nenormate [Hz] (folosind cursorul) frecvența fundamentală F0 Calculați eroarea de determinare a frecvenței fundamentale F0 ca urmare a modificării rezoluției spectrale.
* Calculați rezoluția spectrală (în Hz) corespunzătoare NFFT.

**II. Sisteme în timp discret**

a) Se citește un semnal audio din fișierul wav aflat în folderul [www.comm.pub.ro/pds\_proiect/files/](http://www.comm.pub.ro/pds_proiect/files/). Fișierul wav corespunzător fiecărui student se alege din tabelul de la sfârșitul temei. Verificați că frecvența de eșantionare Fs citită din fișierul audio corespunde cu Fs din tabel.

* Se determină lungimea L a semnalului citit din fișier.
* Reprezentați grafic semnalul audio în timp absolut și spectrograma semnalului.

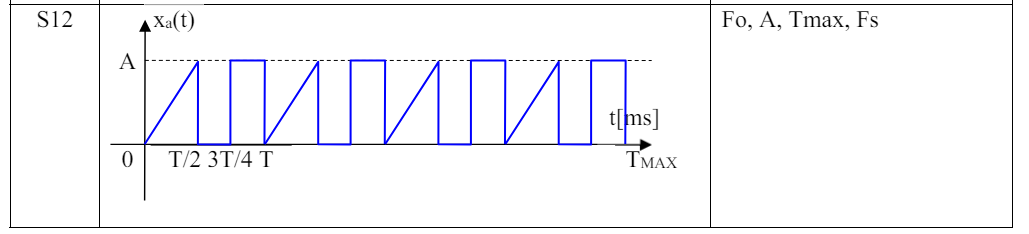
b) Să se genereze următorul semnal discret: 𝑠(𝑛) = 𝑤(𝑛) + 𝑥(𝑛) unde w(n) este semnalul citit din fișierul wav și x(n) este semnalul de la punctul a). Semnalul x(n) va fi regenerat astfel încât să aibă aceeași lungime L ca semnalul w(n), iar amplitudinea A va fi redusă la valori de aprox. 0.1 - 0.2.

* Reprezentați spectrograma semnalului s(n).
* Calculați TFD a semnalului s(n) în NFFT puncte și reprezentați spectrul de amplitudine în frecvență nenormată.
* Determinați (folosind cursorii) din spectrul semnalului s(n) frecvența componentelor spectrale (maximele spectrale) corespunzătoare frecvenței fundamentale F0 și a armonicelor (primele 3 armonice).

c) Proiectați un filtru notch RII care să elimine din spectrul semnalului s(n) componentele spectrale corespunzătoare frecvenței fundamentale F0 și a armonicelor determinate la punctul anterior.

* Determinați toate zerourile și toți polii funcției H(z) (alegeți raza polilor 0.8<r<1).
* Determinați coeficienții funcției de transfer H(z) a filtrului RII.
* Reprezentați grafic diagrama poli-zerouri.
* Reprezentați grafic răspunsul în frecvență al sistemului în dB și liniar. Determinați din grafic, folosind cursorii, banda de oprire la 3dB a filtrului notch (pentru caracteristica de anulare a spectrului pe frecvența fundamentală).
* Reprezentați grafic răspunsul la impuls al sistemului.
* Calculați ieșirea sistemului y(n) dacă la intrare se aplică semnalul s(n).
* Reprezentați spectrograma semnalului de ieșire.
* **Cerintele particulare:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nume** | **Grupa** | **Tip  semnal** | **Frecvența  fundamentală  F0 [Hz]** | **Ampl.  A** | **TMAX [ms]** | **Frecvența  de  eșantionare Fs** | **NFFT** |
| FRUNZĂ C. Vladimir | 431C | S12 | 400 | 2.5 | 20 | 16000 | 1024 |

****

**NOTA: Pentru semnalul audio, link-ul primit nu merge, asa ca am ales eu 2 semnale audio potrivite de pe Youtube.**

**PARTEA 1**

1. **Generarea Semnalelor**

**Functie Ajutatoare**

function y = gate(t, duty)

y = 0.5 \* (square(t, duty) + 1);

end

**Semnalul Analog**

function [t,y] = semnal\_an(F0,A,Tmax,Fs)

% Vectorul timp

t = 0:1/Fs:Tmax-1/Fs;

% Semnalul Analog

y = A\*(sawtooth(2\*pi\*F0\*t).\*gate(2\*pi\*F0\*t - pi, 50) + gate(2\*pi\*F0\*t - pi/2, 25));

end

**Semnalul Discret**

function [n,y] = semnal\_dig(F0,A,Tmax,Fs)

% Vectorul de esantioane

n = 1:Tmax\*Fs;

% Semnalul Digital

y = A\*(sawtooth(2\*pi\*F0\*n/Fs).\*gate(2\*pi\*F0\*n/Fs - 3.14, 50) + gate(2\*pi\*F0\*n/Fs - pi/2, 25));

end

1. **Importarea si Reprezentarea Celor 2 Semnale, Analog si Discret**

%% PARTEA 1

clear;

clc;

close all;

% Parametrii semnalului analogic

F0 = 400;

A = 2.5;

% Parametrii discretizarii

Fs = 16000;

FsA = Fs\*10;

Tmax = 0.02;

% Parametrii FFT

Nfft = 1024;

% Obtinerea Semnalului Analogic

[t, semnal\_an] = semnal\_an(F0,A,Tmax,FsA);

% Reprezentarea Semnalului Analogic

figure();

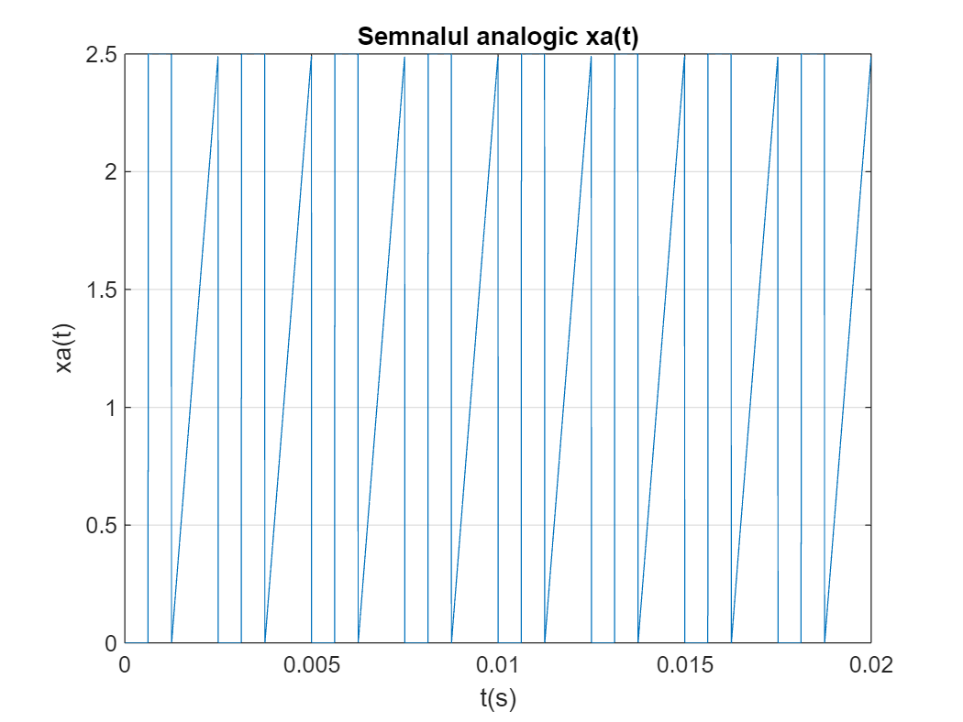
plot(t,semnal\_an);

grid on;

title("Semnalul analogic xa(t)");

xlabel("t(s)");

ylabel("xa(t)");



% Obtinerea Semnalului Discret

[n, semnal\_dig] = semnal\_dig(F0,A,Tmax,Fs);

% Reprezentarea Semnalului Discret

figure();

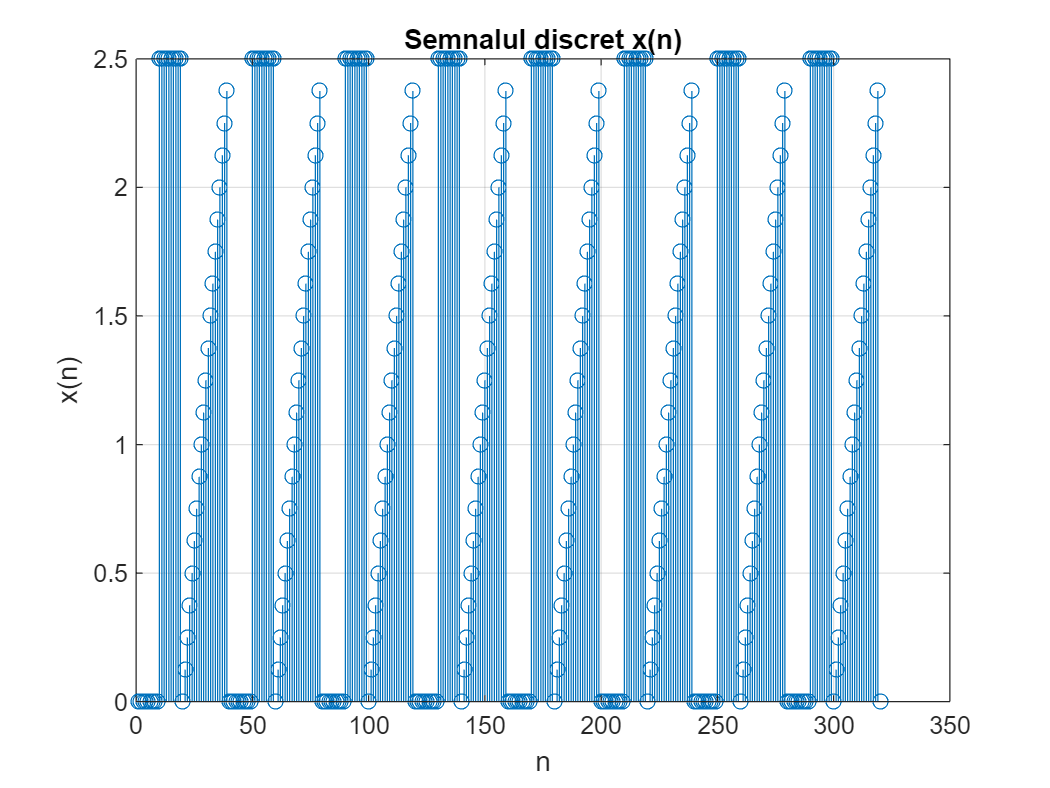
stem(n,semnal\_dig);

grid on;

title("Semnalul discret x(n)");

xlabel("n");

ylabel("x(n)");



1. **Analiza Fourier pe cele 2 semnale, in 2 cazuri distincte pentru Nfft**

function y = Fourier\_Analysis(x, Fs, Nfft)

% Verificam daca stim Nfft

if nargin < 3

Nfft = length(x);

end

% Transformata Fourier Discreta

y = fft(x,Nfft);

% TFD1 Spectru Amplitudine in functie de indicele k

k = 1:Nfft;

figure();

plot(k, abs(y));

title('Spectrul de amplitudine |X(k)|');

xlabel('k');

ylabel('|X(k)|');

% Spectrele de Amplitudine si Faza in functie de frecventa normata

Fnorm = ((k-1)/Nfft) - 0.5;

figure();

plot(Fnorm, fftshift(abs(y)));

title('Spectrul de amplitudine in frecvente normate');

xlabel('f');

ylabel('|X|');

figure();

plot(Fnorm, angle(y))

title('Spectrul de faza in frecvente normate');

xlabel('f');

ylabel('arg(X) [rad]');

% Spectrul de Amplitudine in functie de frecventa nenormata

Fnenorm = Fnorm\*Fs;

figure();

plot(Fnenorm, fftshift(abs(y)));

title('Spectrul de amplitudine in frecvente nenormate');

xlabel('F(Hz)');

ylabel('|X|');

end

1. **Reprezentarea Spectrelor pentru cazul implicit, Nfft = 512**

% Aflarea parametrilor ceruti la I. a)

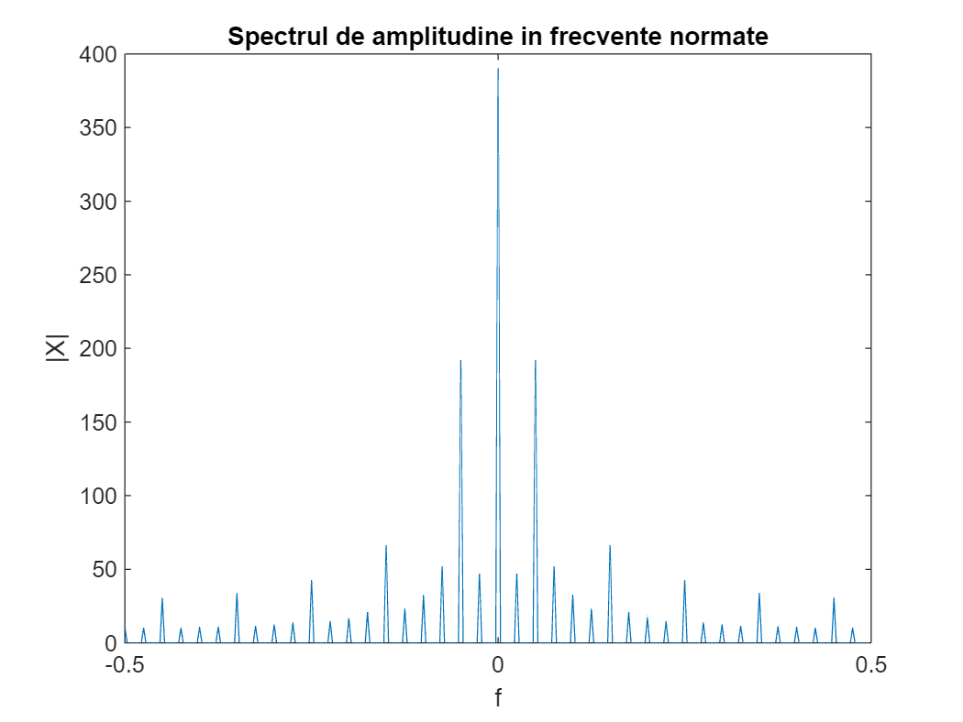
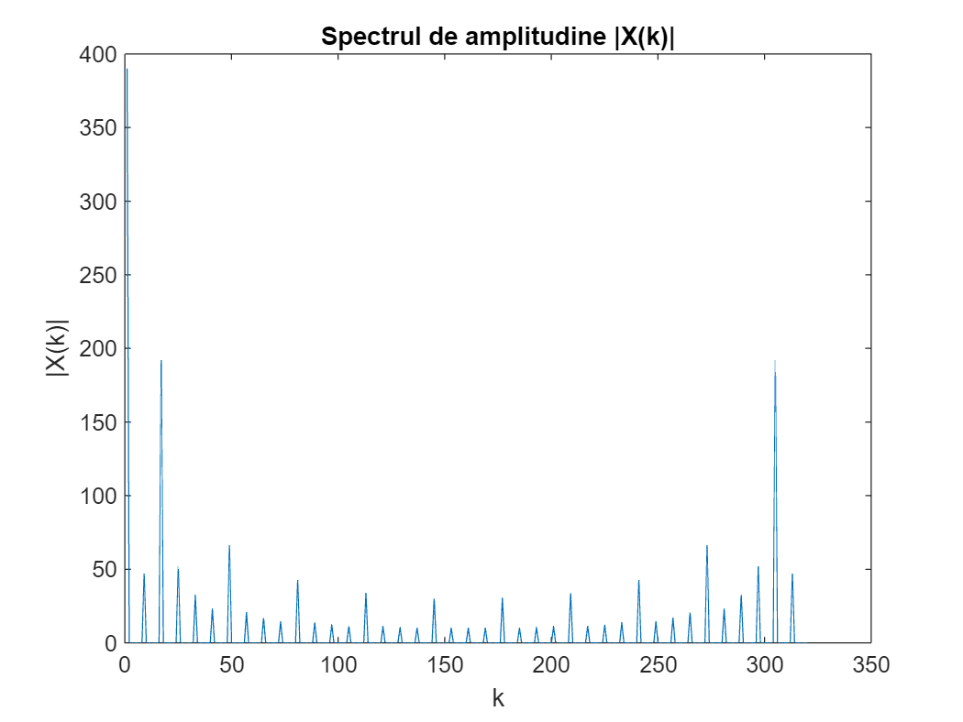
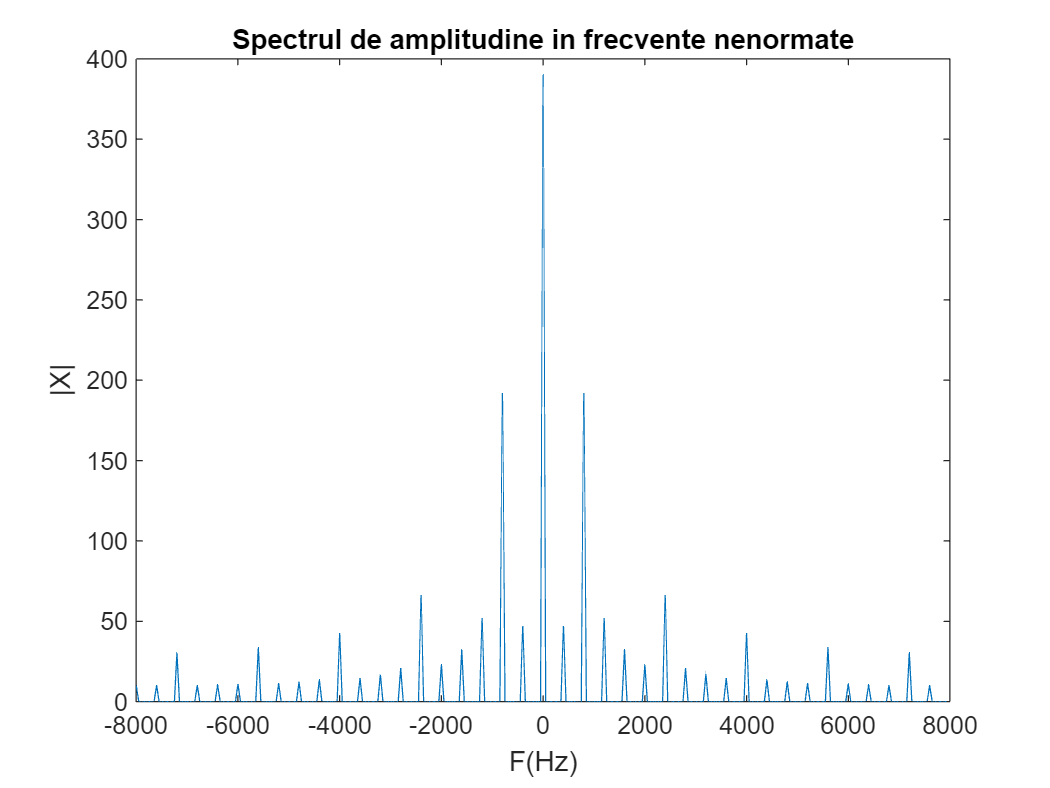
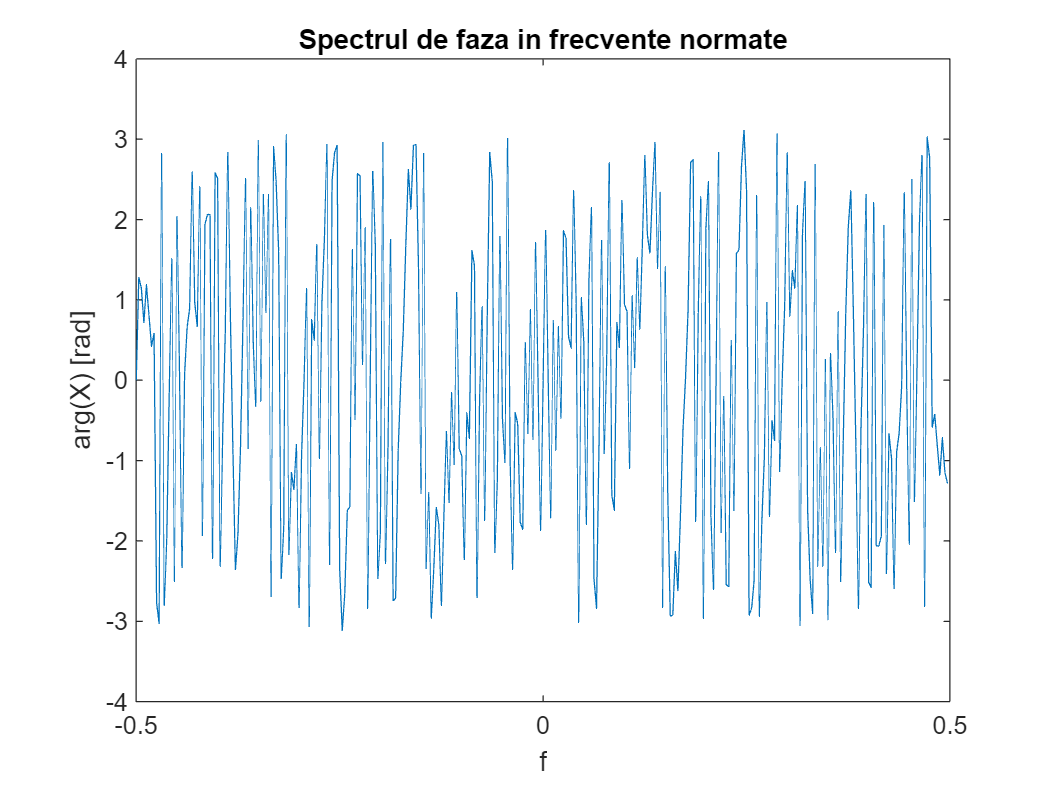
Nes = length(n);

N = Fs/F0;

P = Nes/N;

% Analiza Fourier in discret pentru Nfft = Nes

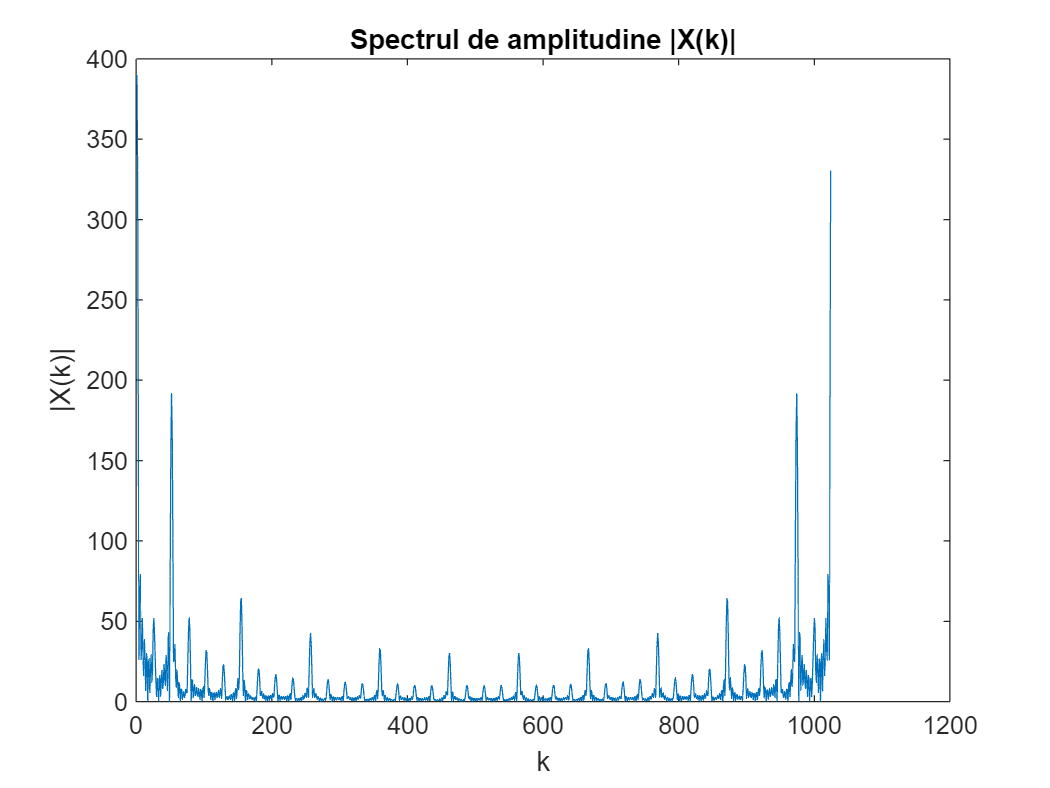
TFD1 = Fourier\_Analysis(semnal\_dig,Fs);

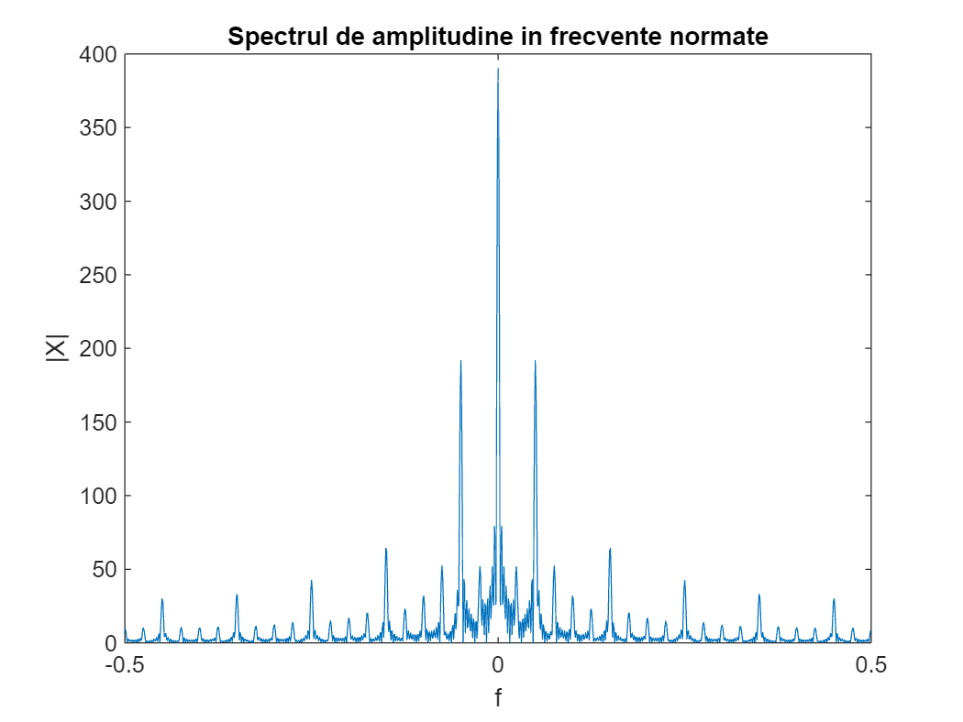


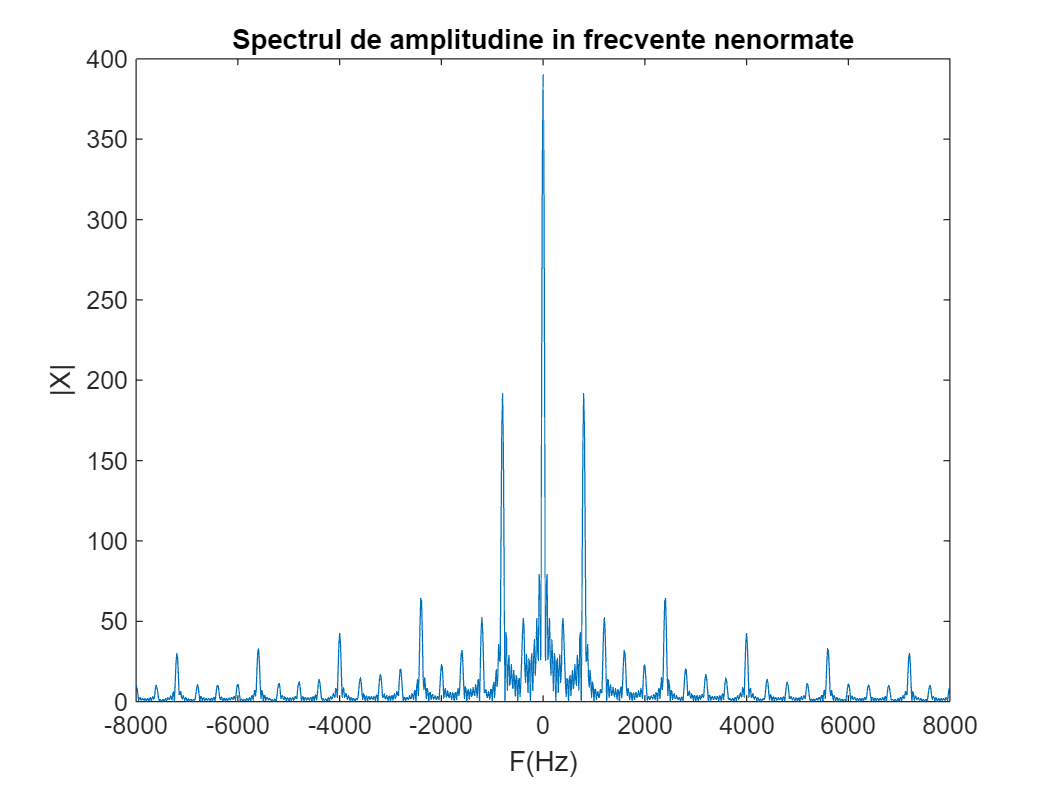
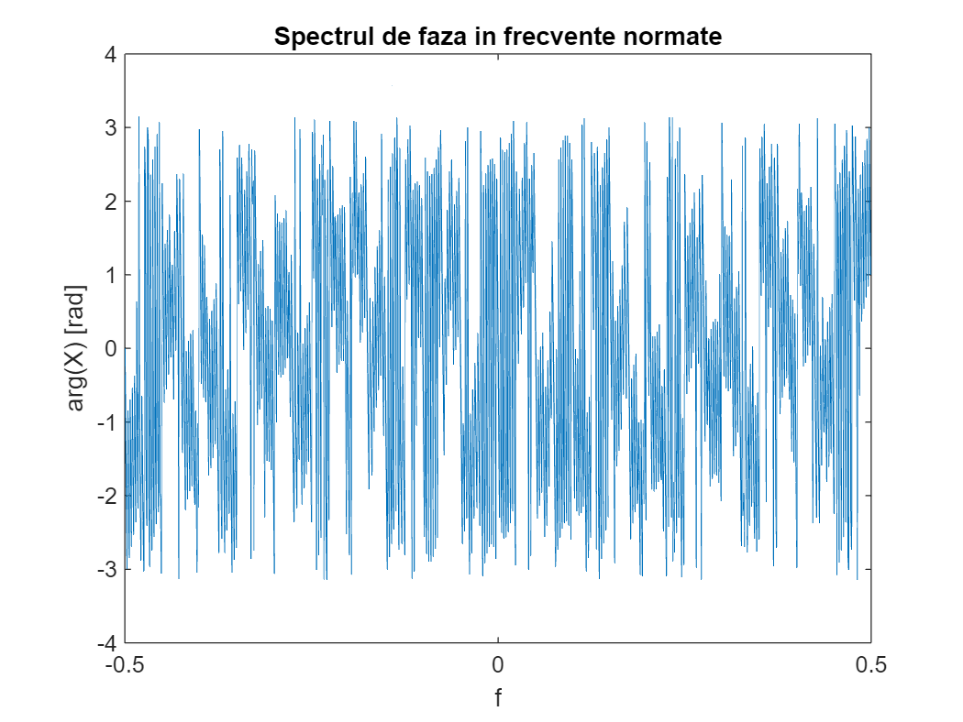
1. **Reprezentarea Spectrelor pentru cazul explicit, Nfft din tabel**

% Analiza Fourier in discret pentru Nfft = 1024 (din tabel)

TFD2 = Fourier\_Analysis(semnal\_dig, Fs, Nfft);







**PARTEA 2**

1. **Alegerea, importarea si reprezentarea semnalului audio**

%% PARTEA 2

clear;

clc;

close all;

% Redefinire parametri semnal de la a)

F0 = 400;

A = 0.15;

Nfft = 1024;

% Accesare semnal audio - Selecteaza semnalul dorit

[w, Fss] = audioread("Goodnight\_Kiss\_Converted.wav");

L = length(w);

ts = 0:1/Fss:(L-1)/Fss;

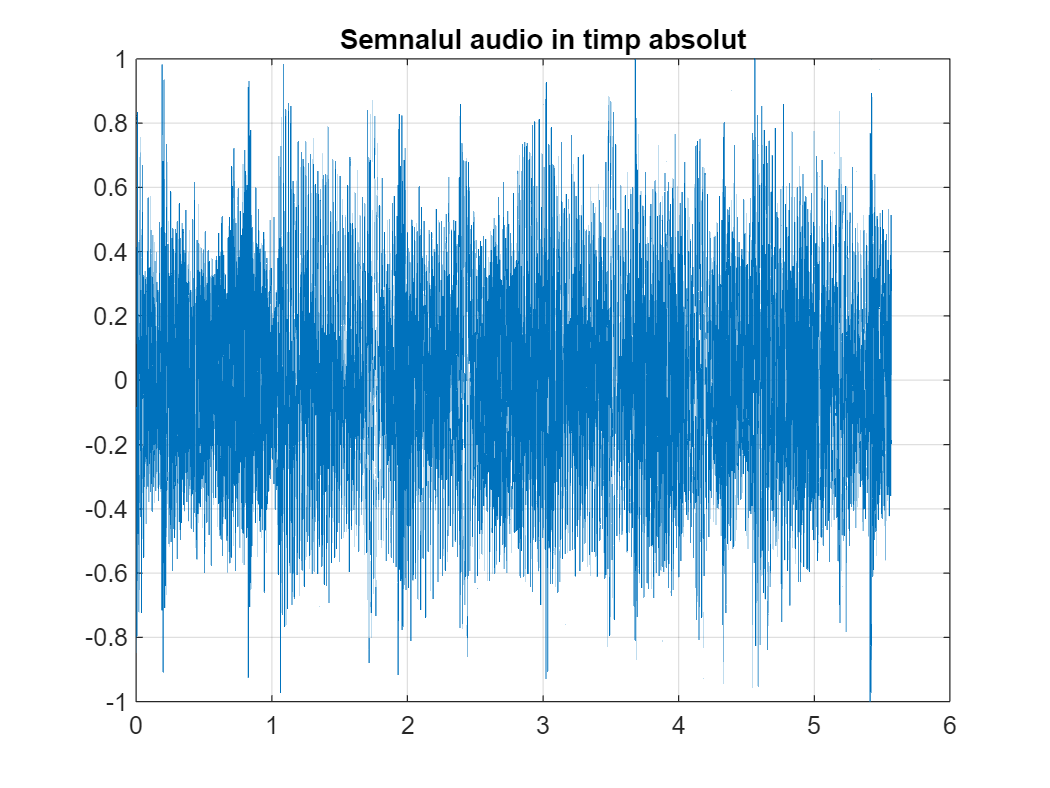
% Reprezentare semnal audio in timp absolut

figure();

plot(ts, w);

grid on;

title("Semnalul audio in timp absolut");



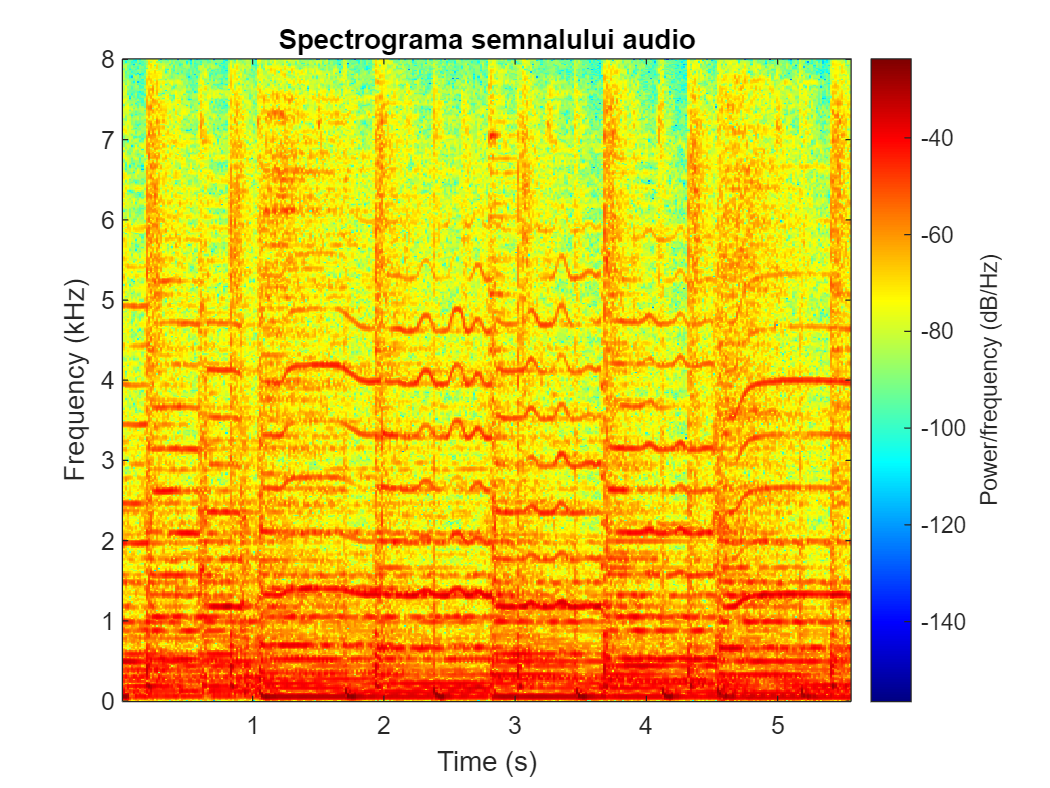
% Reprezentare spectrograma semnal audio

figure();

spectrogram(w, 512, 256, Nfft, Fss, 'yaxis');

colormap("jet");

title("Spectrograma semnalului audio");



1. **Definirea si reprezentarea semnalului s(n), semnal ce urmeaza   
   sa fie filtrat**

% Definirea semnalului s(n)

Tmax = L/Fss;

[n, x] = semnal\_dig(F0,A,Tmax,Fss);

s = w' + x;

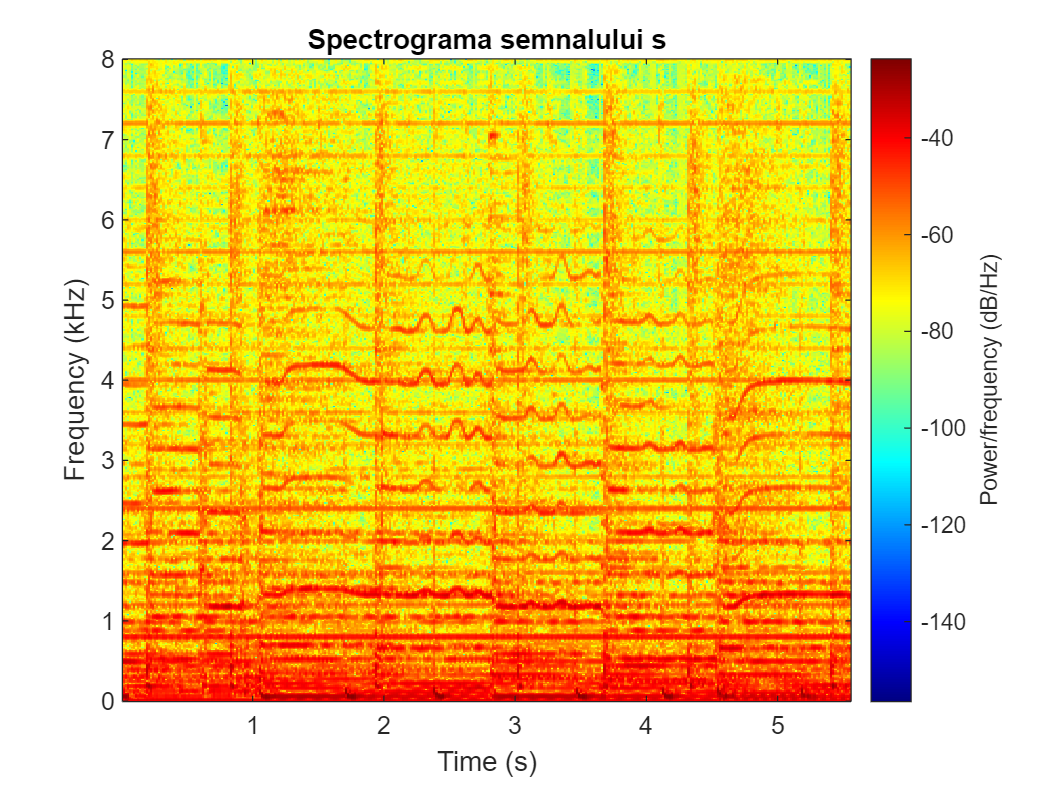
% Reprezentare spectrograma s(n)

figure();

spectrogram(s, 512, 256, Nfft, Fss, 'yaxis');

colormap("jet");

title("Spectrograma semnalului s");



1. **Analiza Fourier si reprezentarea spectrului a semnalului s(n)**

% Transformata Fourier a semnalului s(n)

S = fft(s,Nfft);

% Spectrul de amplitudine in frecvente nenormate

Fs2 = -Fss/2: Fss/Nfft: Fss/2 - Fss/Nfft;

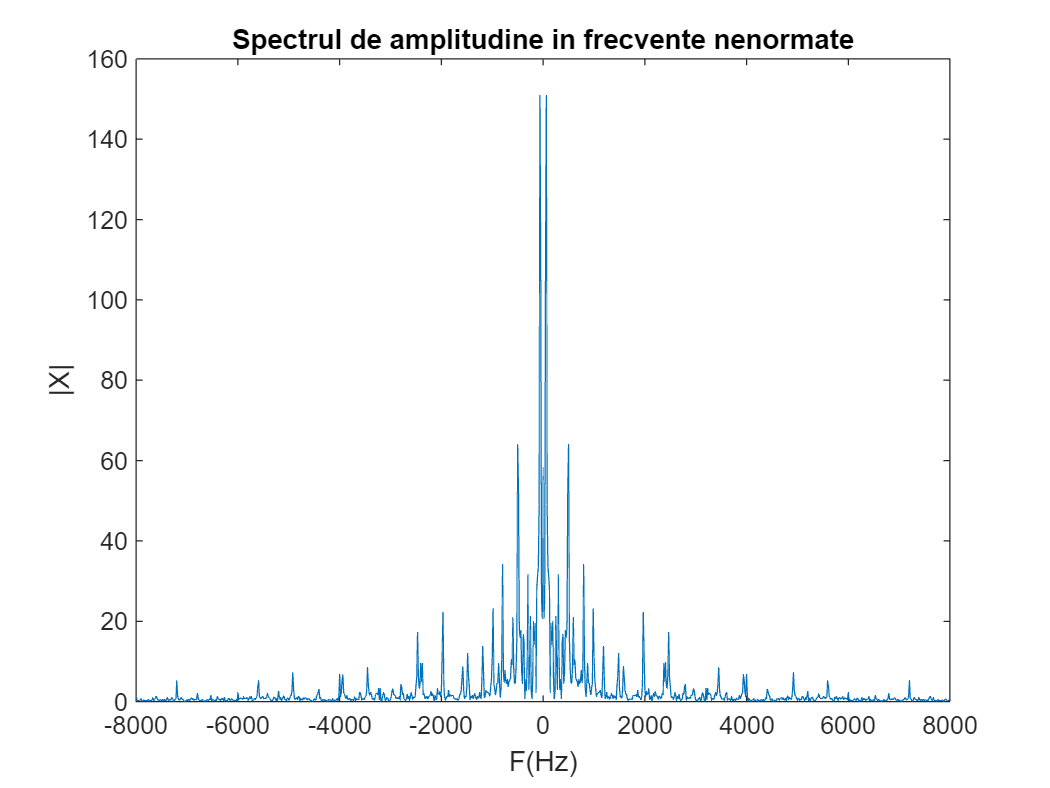
figure();

plot(Fs2, fftshift(abs(S)));

title("Spectrul de amplitudine in frecvente nenormate");

xlabel('F(Hz)');

ylabel('|X|');



1. **Construirea filtrului notch pentru filtrarea semnalului s(n)**

% Determinarea Zerourilor

w0 = 2\*pi\*F0/Fss;

z1 = exp(1j\*w0);

z2 = exp(1j\*2\*w0);

z3 = exp(1j\*3\*w0);

z4 = exp(1j\*4\*w0);

% Determinare Polilor

r = 0.9995;

p1 = r\*z1;

p2 = r\*z2;

p3 = r\*z3;

p4 = r\*z4;

% Vectorii zerourilor si ai polilor

z = [z1, conj(z1), z2, conj(z2), z3, conj(z3), z4, conj(z4)];

p = [p1, conj(p1), p2, conj(p2), p3, conj(p3), p4, conj(p4)];

% Determinarea polinoamelor

b = poly(z);

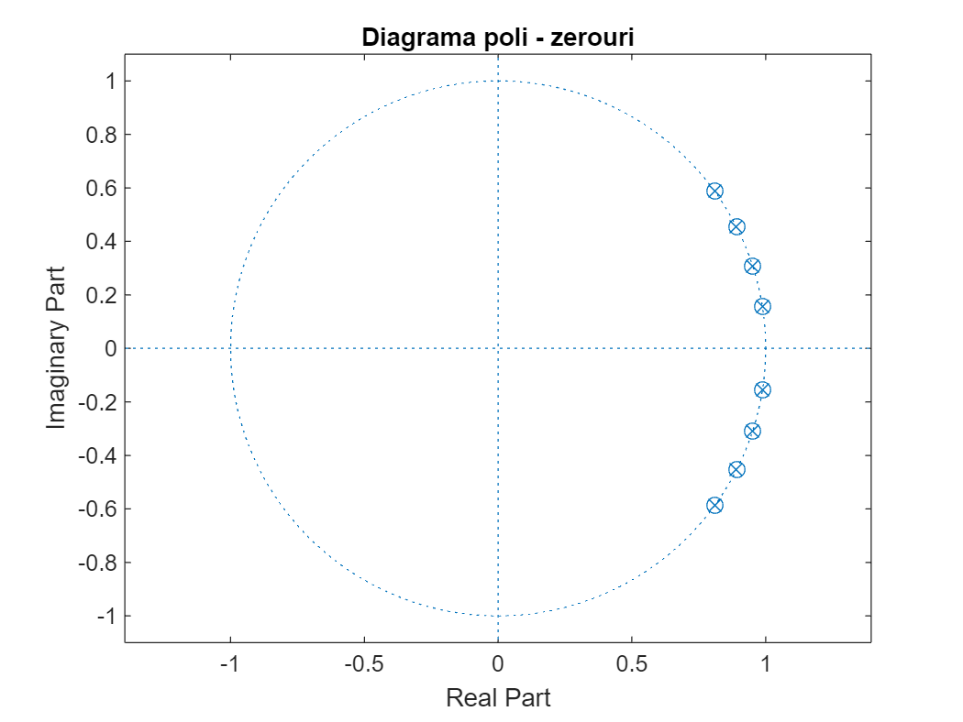
a = poly(p);

% Diagrama poli-zerouri

figure();

zplane(b, a);

title('Diagrama poli - zerouri');



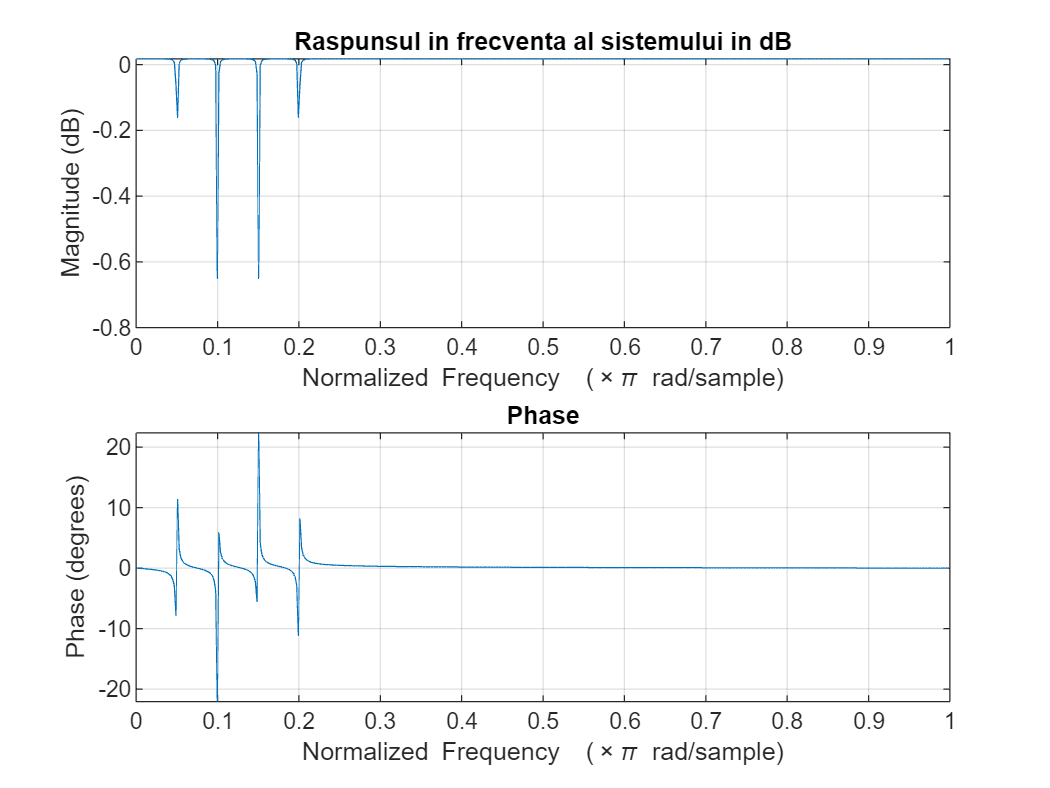
1. **Analiza Fourier a filtrului construit si aflarea raspunsului la impuls**

% Reprezentarea raspunsului in frecventa (dB)

figure();

freqz(b,a);

title('Raspunsul in frecventa al sistemului in dB');



% Aflarea functiei de transfer

[H, W] = freqz(b, a);

% Reprezentarea raspunsului in frecventa (linear)

figure();

subplot(2, 1, 1), plot(W, abs(H)), grid;

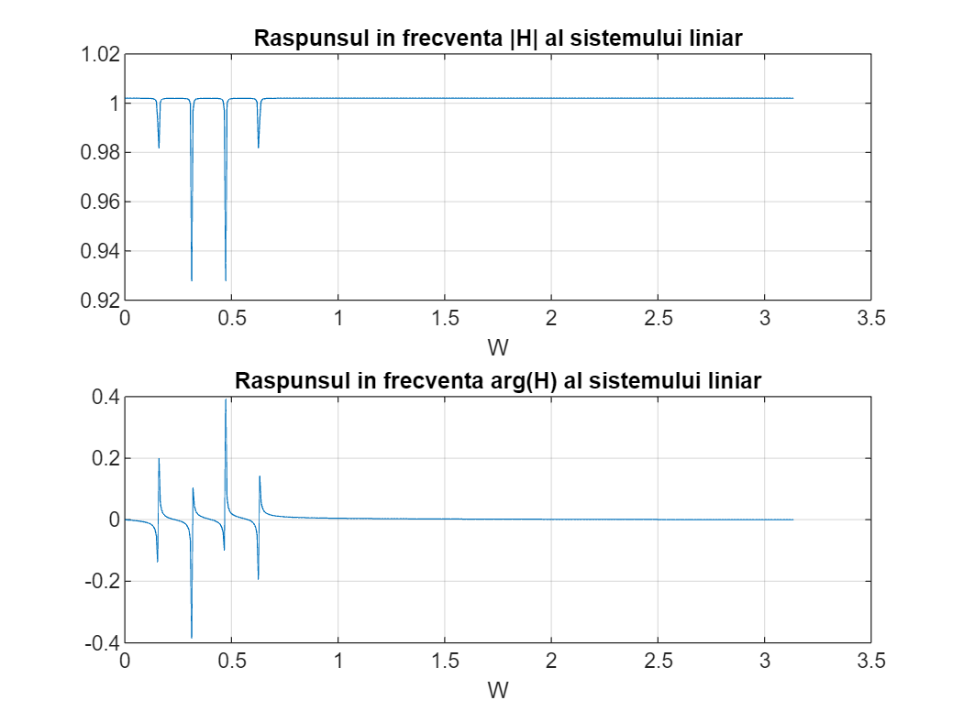
title('Raspunsul in frecventa |H| al sistemului liniar');

xlabel('W');

subplot(2, 1, 2), plot(W, angle(H)), grid;

title('Raspunsul in frecventa arg(H) al sistemului liniar');

xlabel('W');



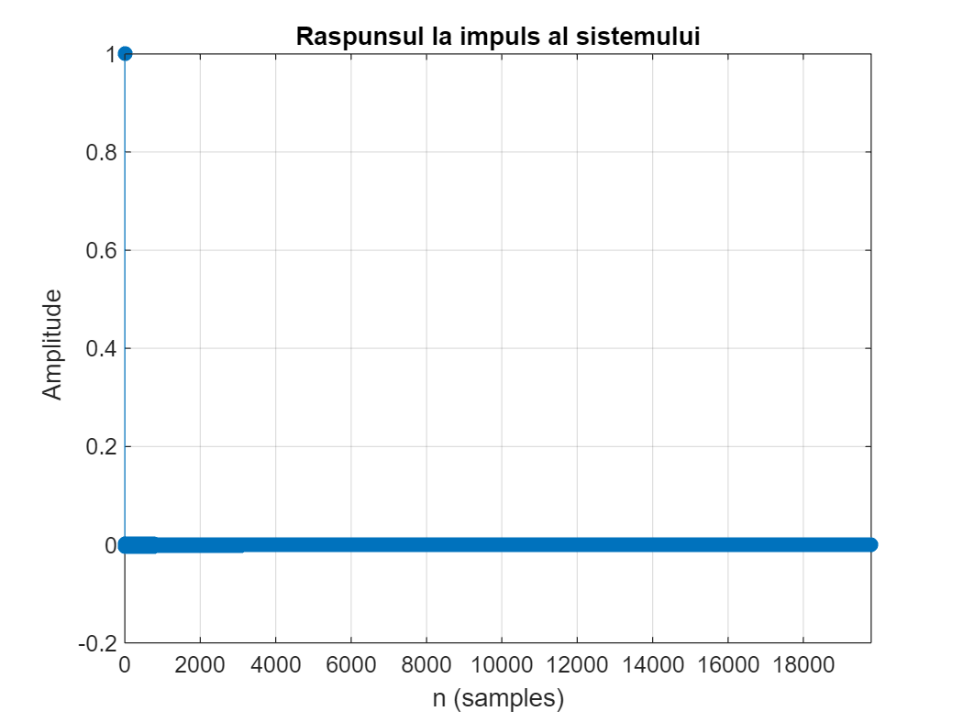
% Raspunsul la impuls al sistemului

figure();

impz(b, a);

grid on;

title('Raspunsul la impuls al sistemului');



1. **Calcularea si reprezentarea iesirii sistemului (semnalul filtrat)**

% Iesirea sistemului cand la intrare este s(n)

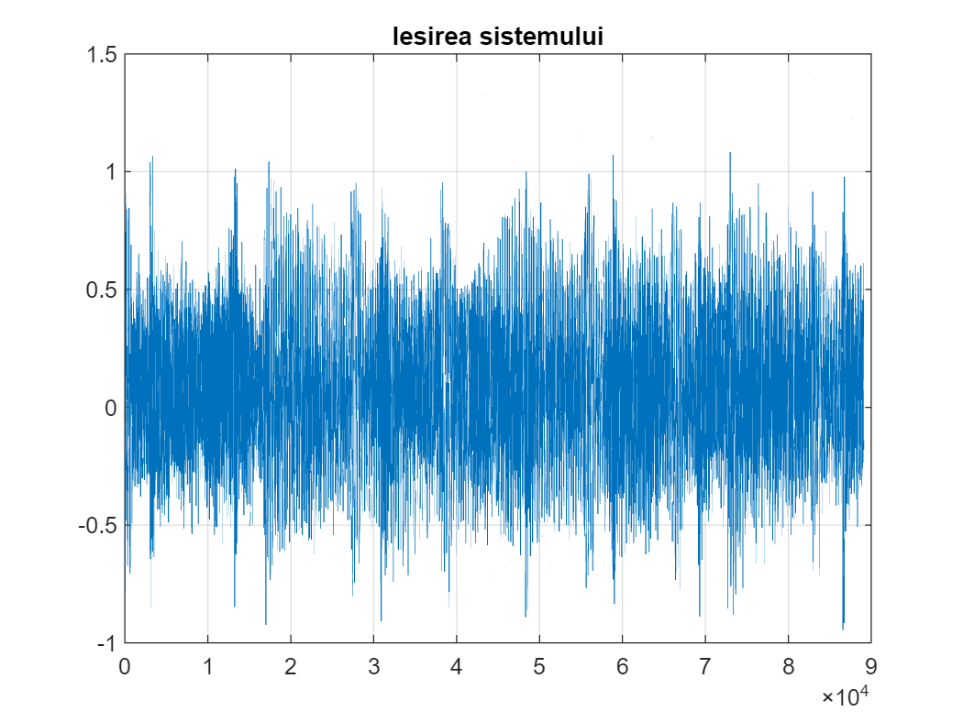
y = filter(b, a, s);

figure();

plot(n, y);

grid on;

title('Iesirea sistemului');



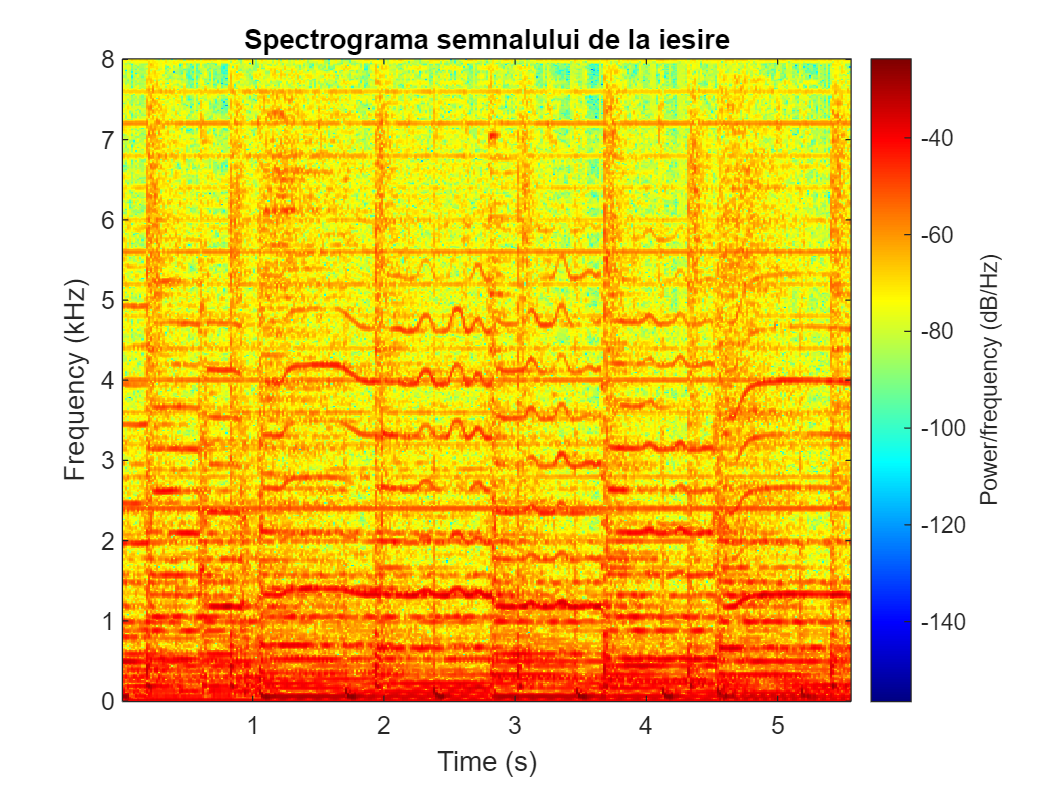
% Spectrograma semnalului de la iesire

figure();

spectrogram(y, 512, 256, Nfft, Fss, 'yaxis');

colormap("jet");

title('Spectrograma semnalului de la iesire');



**BONUS: Apasa cele 2 butoane, pe rand, pentru a asculta semnalul la intrare (initial), respectiv la iesirea din sistem (filtrat).**

%% ASCULTA INTRAREA SISTEMULUI

sunet\_intrare = audioplayer(w, Fss);

play(sunet\_intrare);

% Apasa butonul

%% ASCULTA IESIREA SISTEMULUI

sunet\_iesire = audioplayer(y, Fss);

play(sunet\_iesire);

% Apasa butonul

