

**Proiect la disciplina Tehnici avansate de măsurare a
câmpului electromagnetic**

**Analiza în timp și frecvență a unui semnal modulat în
amplitudine și frecvență**

Student: Ignat Codrina-Victoria

Master AAIE, anul I

2022 – Iunie

Cuprins

1. Obiectivul proiectului	3
2. Mijloace și metode de realizare	4
3. Etapele realizării proiectului	4
4. Simulări și observații	5
5. Concluzii și interpretări.....	12
6. Îmbunătățiri viitoare.....	13
7. Bibliografie	13
8. Anexe	14

1. Obiectivul proiectului

Proiectul își propune realizarea unei interfețe grafice în mediul de dezvoltare Matlab2021 care să permită alegerea unui semnal modulator și a unui semnal purtător în vederea realizării modulației în amplitudine și frecvență a acestora. De asemenea, un alt obiectiv este vizualizarea semnalelor în diferite stadii (semnal modulat, semnal demodulat, semnal cu zgomot sau fără zgomot) și domenii (timp și frecvență).

Obiective specifice:

- Definirea parametrilor semnalelor modulator și purtător (frecvență, amplitudine);
- Semnal modulator de mai multe tipuri, selectabil de către utilizator:
 - o Sinusoidal
 - o Dreptunghiular
 - o Dinte de fierăstrău
 - o Sumă de sinusoidale
- Vizualizarea mai multor tipuri de semnale, meniu cu opțiuni selectabile de către utilizator:
 - o Semnal modulator
 - o Semnal purtător
 - o Semnal AM
 - o Semnal AM cu zgomot
 - o Spectrul semnalului AM
 - o Semnal demodulat AM
 - o Spectrul semnalului demodulat AM
 - o Semnal FM
 - o Semnal FM cu zgomot
 - o Spectrul semnalului FM
 - o Spectrul semnalului demodulat FM
 - o Spectrul semnalului demodulat FM
- Buton de info, la a cărui apăsare se deschide o mini-interfață grafică cu detalii despre proiect: autor, nume proiect, specializarea, anul creării proiectului.

Scopul proiectului este didactic.

2. Mijloace și metode de realizare

Mijloacele de realizare includ utilizarea unor funcții ale mediului de dezvoltare Matlab pentru îndeplinirea anumitor funcționalități ale proiectului. De asemenea, pentru realizarea proiectului a fost necesară parcurgerea unei bibliografii despre tipurile de modulație, dar și despre implementarea acestora în Matlab.

3. Etapele realizării proiectului

Pentru realizarea proiectului, am stabilit o serie de etape pe care le-am urmărit, după cum urmează:

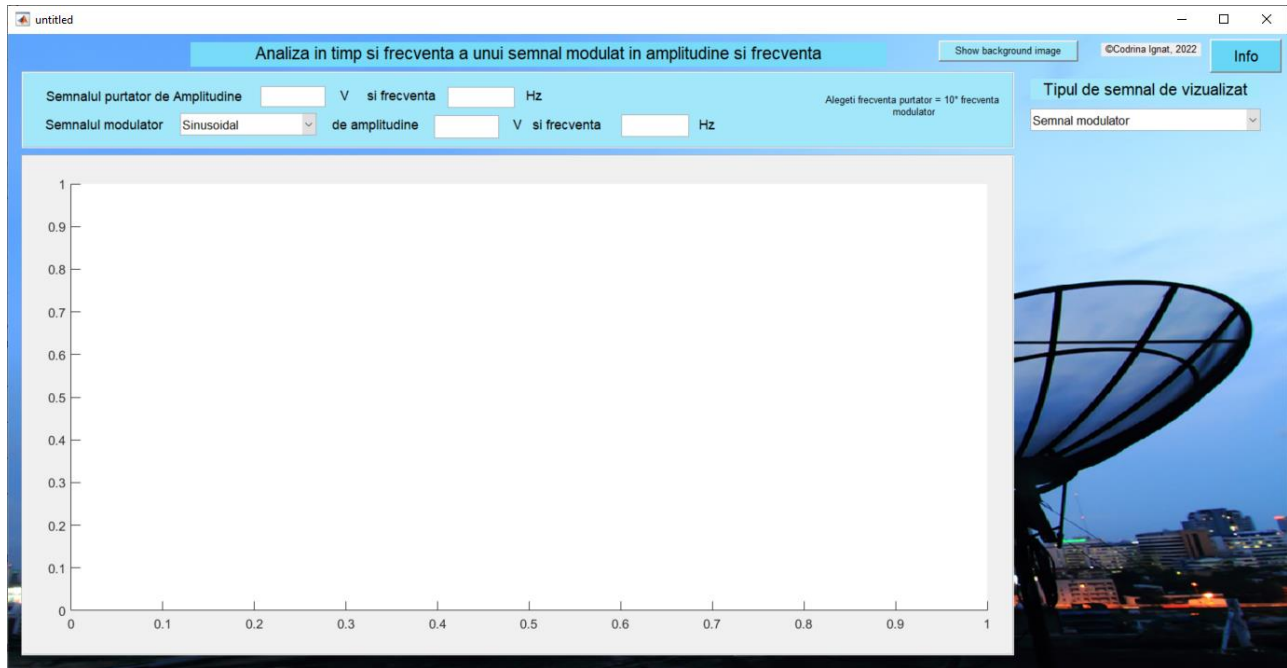
- a. Realizarea interfeței grafice
- b. Așezarea butoanelor, a panourilor, a ferestrei în care vor fi afișate semnalele, a pop-up menu-urilor și a căsuțelor în care se va completa text, pe interfața grafică.
- c. Scrierea codului corespunzător care să descrie semnalele necesare și operațiile de modulare, adăugare de zgomot, extragerea spectrului, demodulare, afișarea semnalelor originale, combinate, modulate și a spectrelor aferente. Implementarea funcțiilor necesare pentru buna funcționare a programului.
- d. Testarea fiecărei funcționalități a proiectului după implementare pentru a descoperi eventuale erori.
- e. Testarea în ansamblu a aplicației pentru a verifica funcționarea corectă între modulele implementate.
- f. Modificări minore legate de estetica aplicației sau bug-uri funcționale.

Rezultatul etapelor implementate în cod se regăsește în secțiunea Anexe, dar și în secțiunea Simulări și observații, sub formă de capturi de ecran ale funcționalităților interfeței grafice/proiectului.

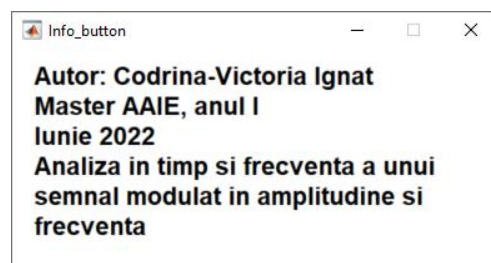
4. Simulări și observații

Am realizat o serie de simulări pentru a evidenția funcționalitățile principale ale aplicației și buna funcționare a acesteia. Capturile de ecran se regăsesc mai jos, împreună cu explicațiile necesare.

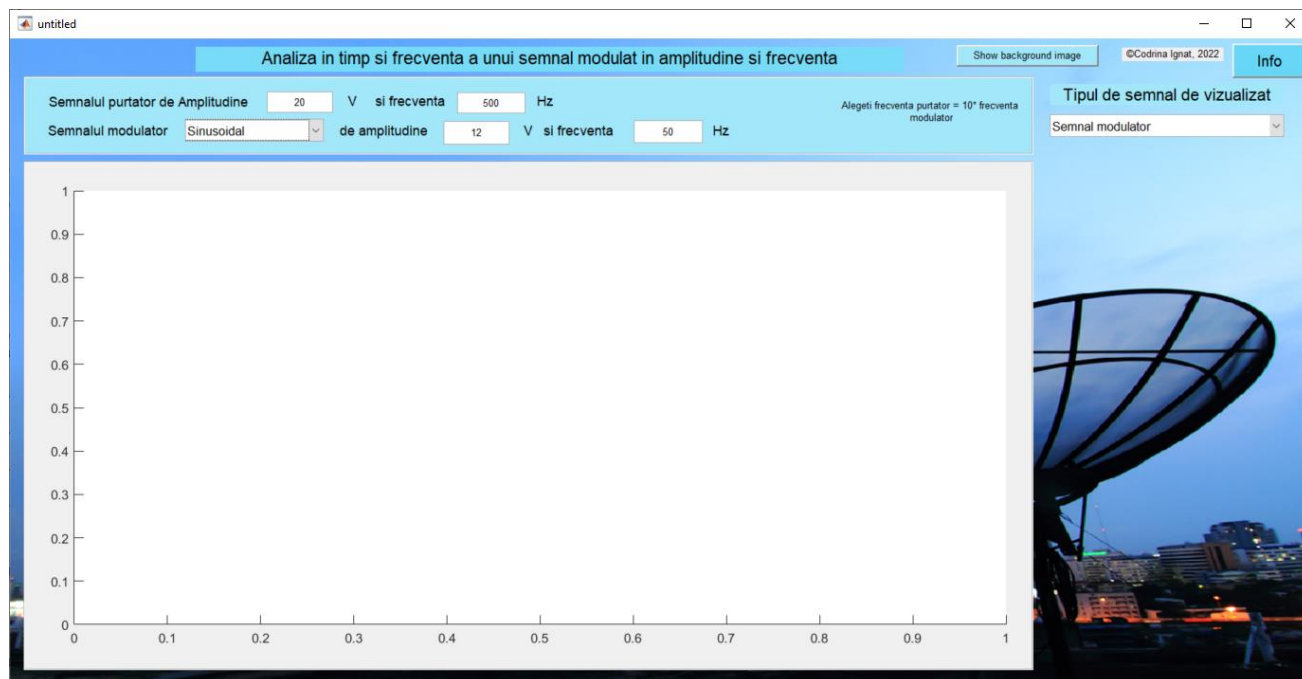
Interfața grafică – vedere generală:



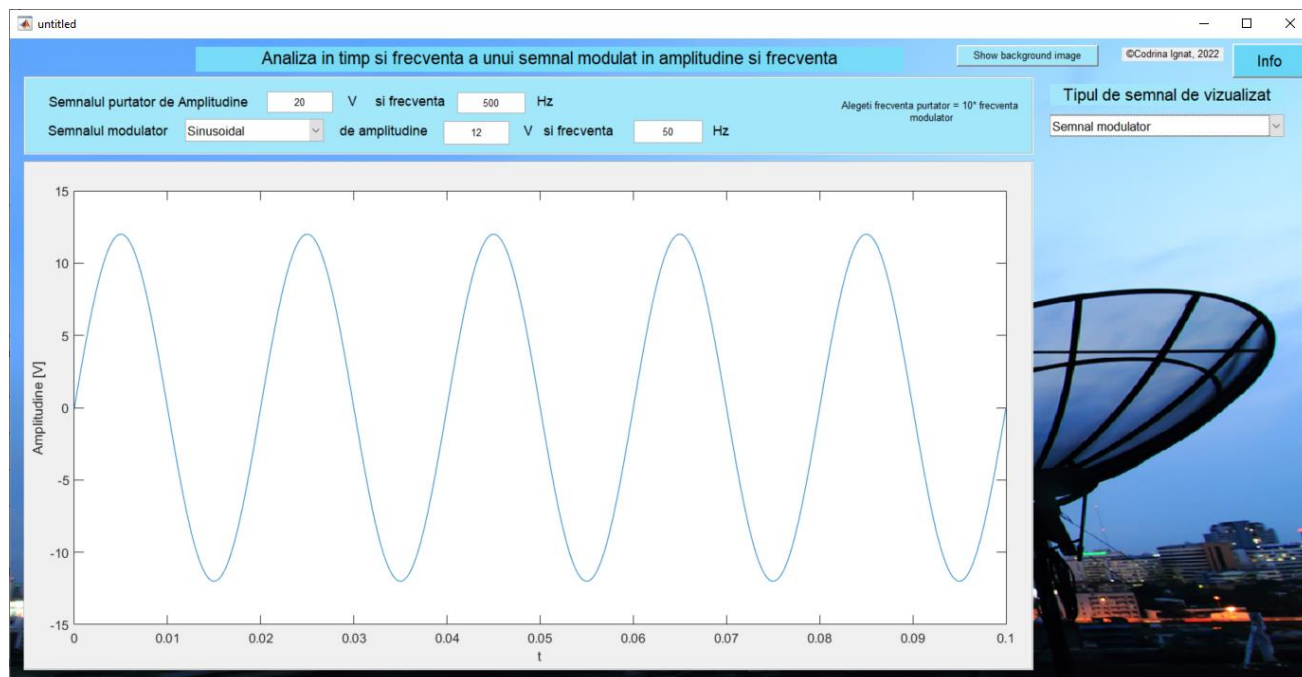
Rezultatul apăsării butonului Info:



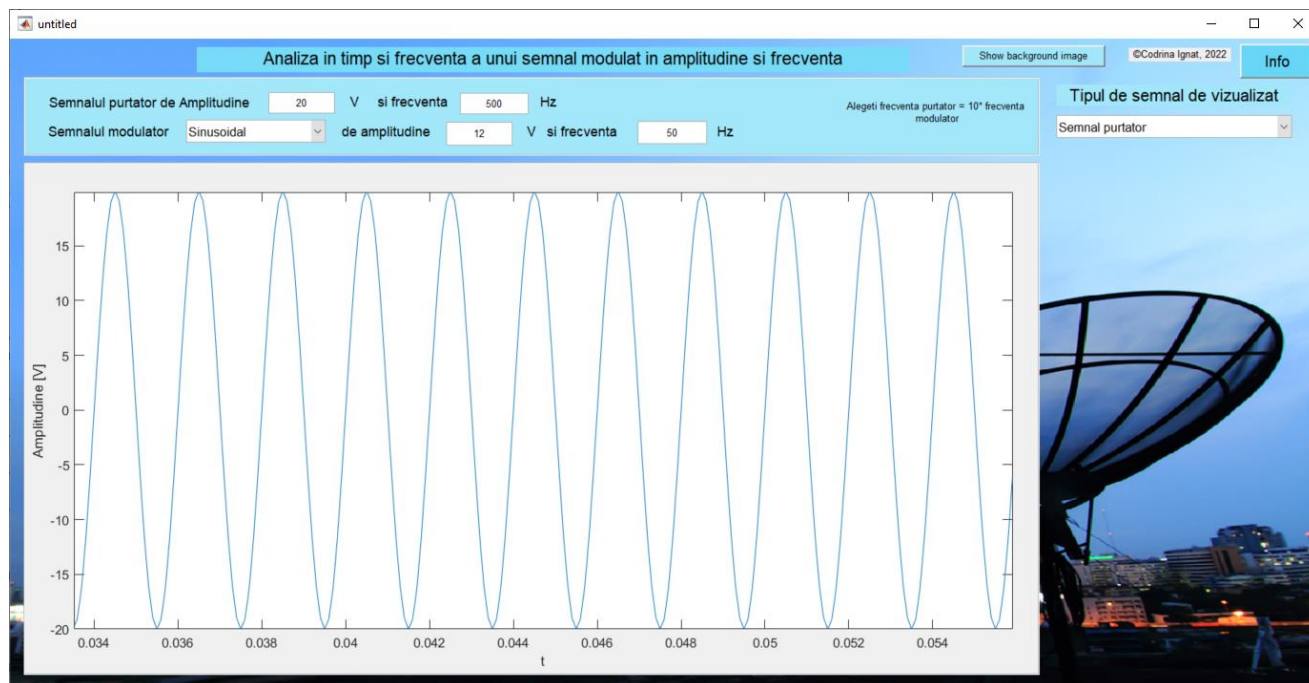
Interfața grafică cu parametrii introduși:



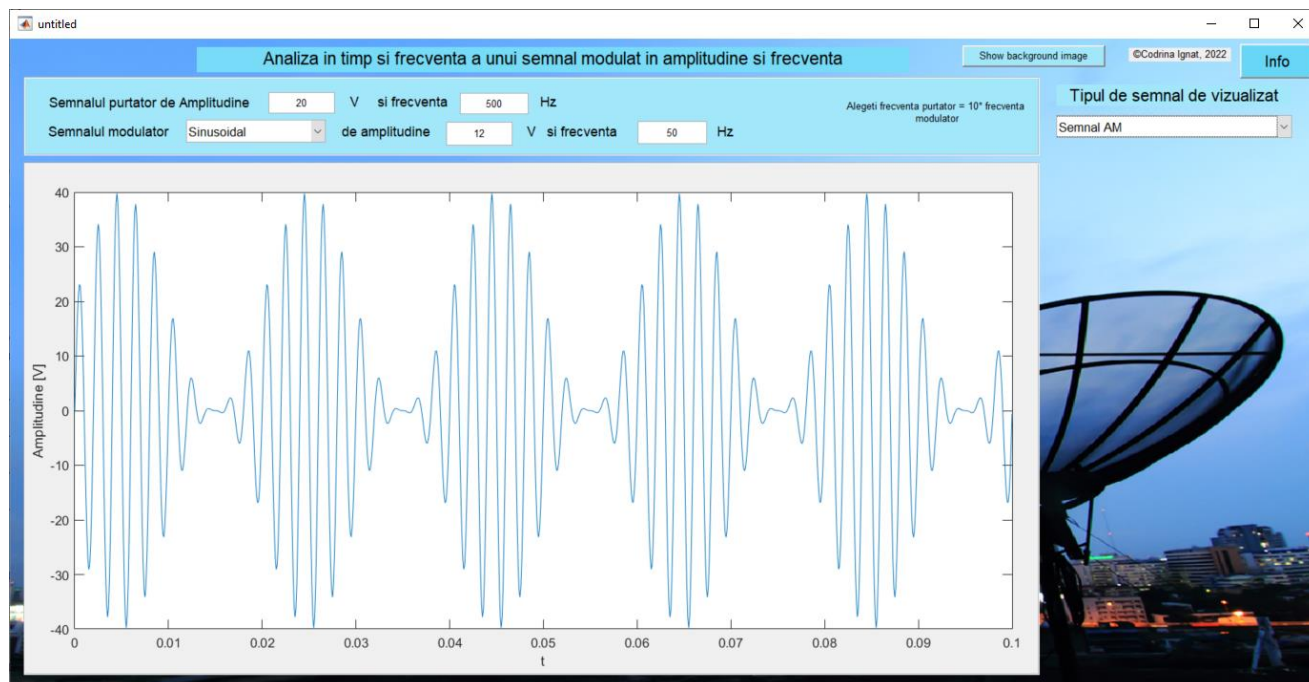
Vizualizarea semnalului modulator:



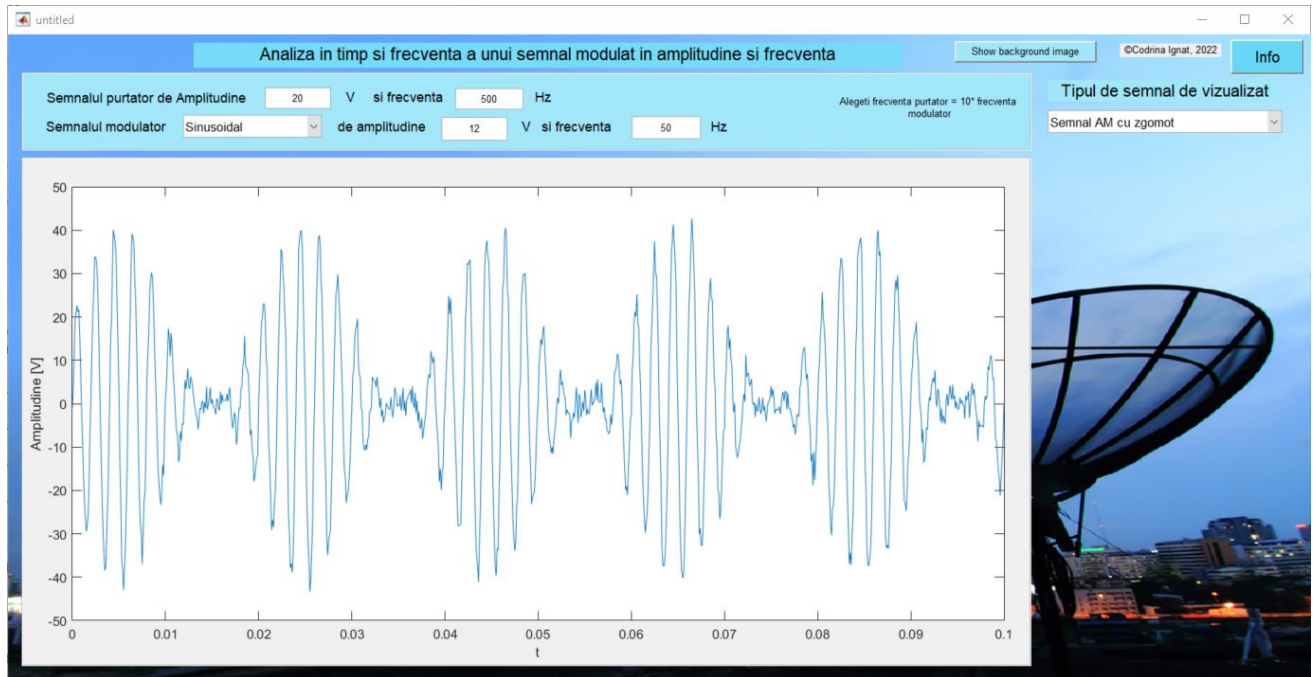
Vizualizarea semnalului purtător:



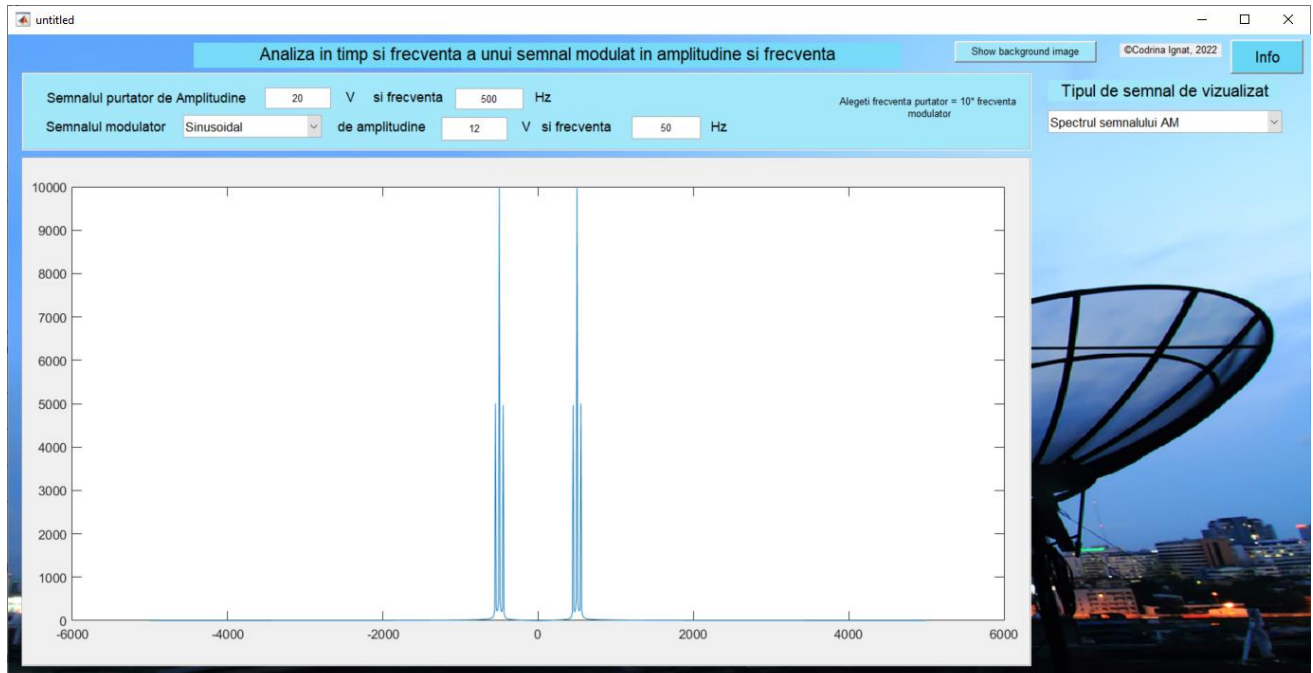
Vizualizarea semnalului modulat în amplitudine:



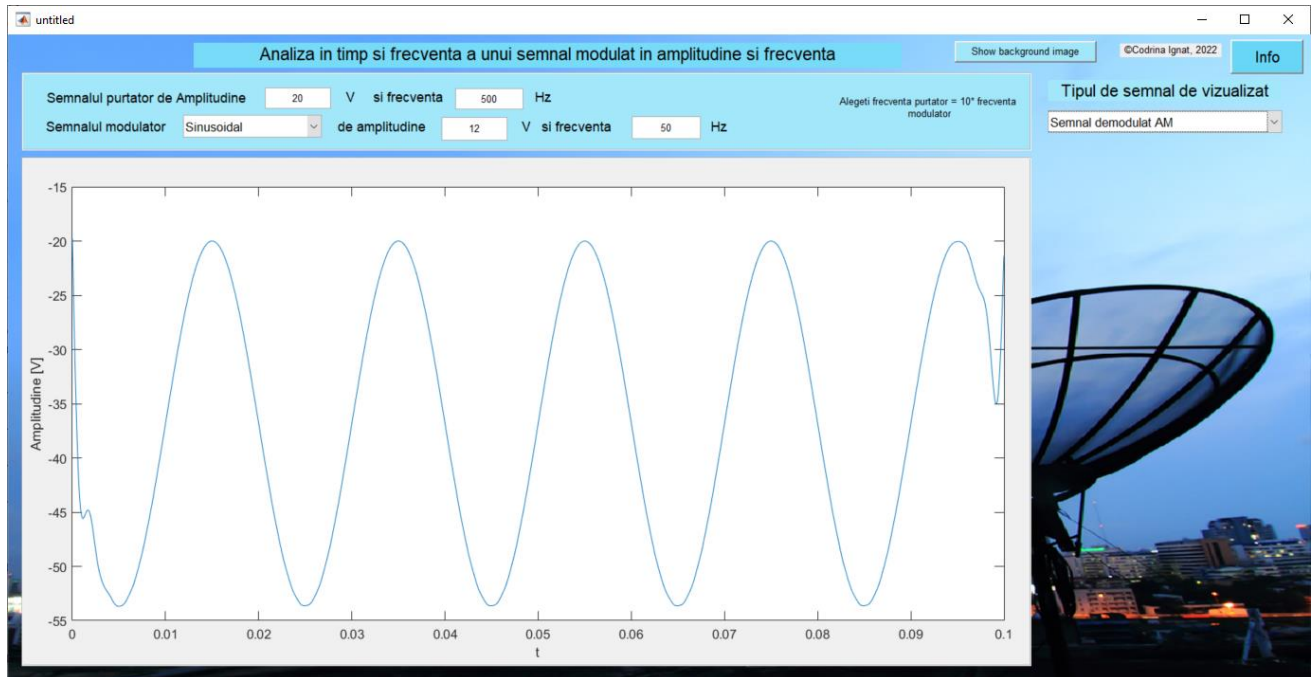
Vizualizarea semnalului modulat în amplitudine însumat cu zgomot:



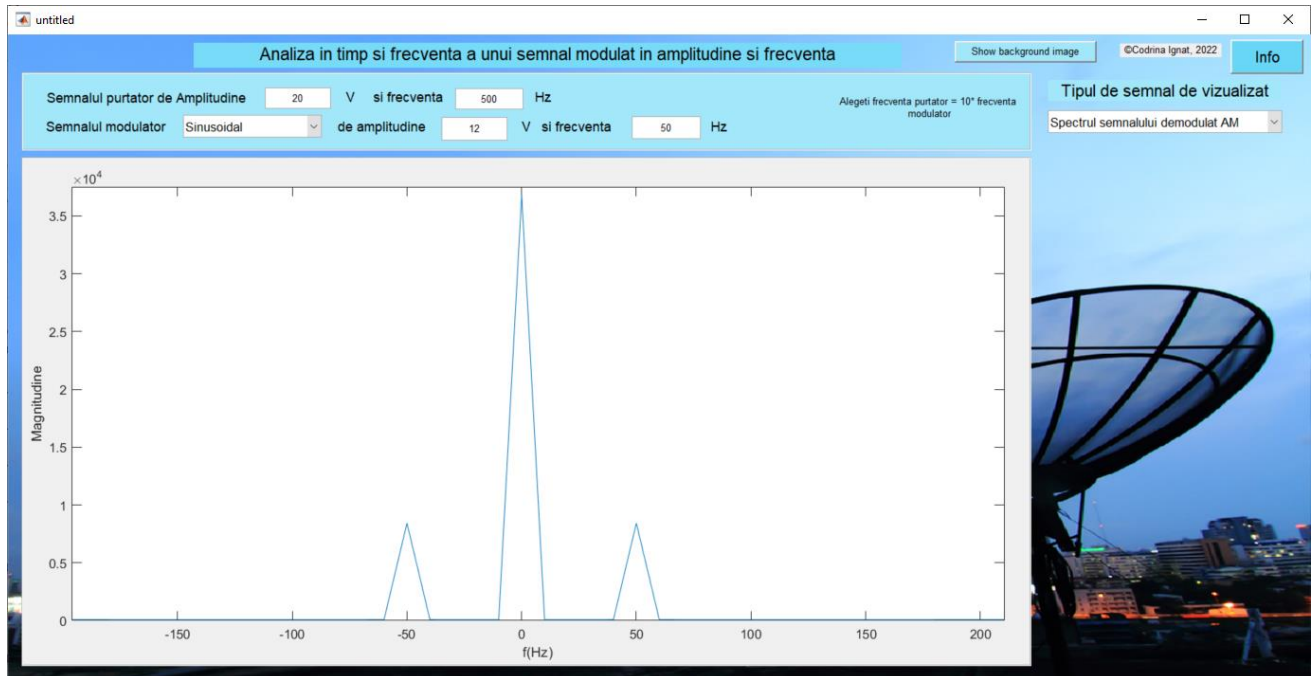
Vizualizarea spectrului semnalului modulat în amplitudine:



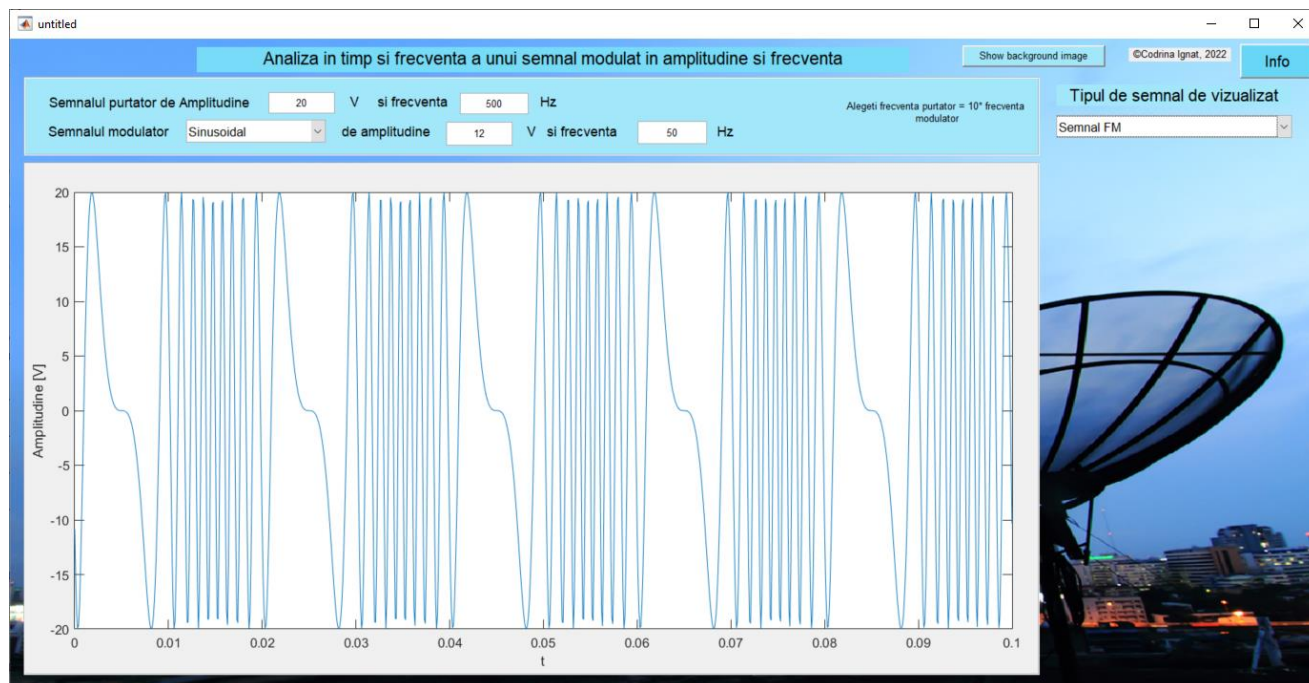
Vizualizarea semnalului demodulat AM:



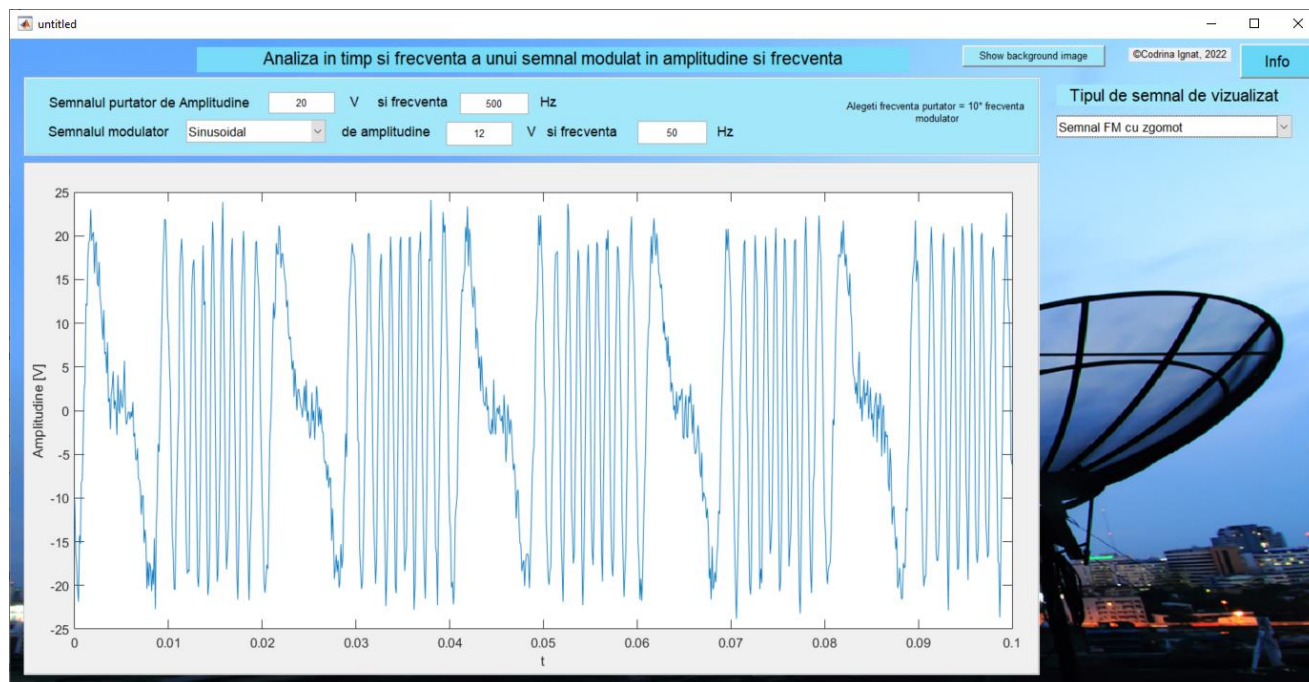
Vizualizarea spectrului semnalului demodulat AM:



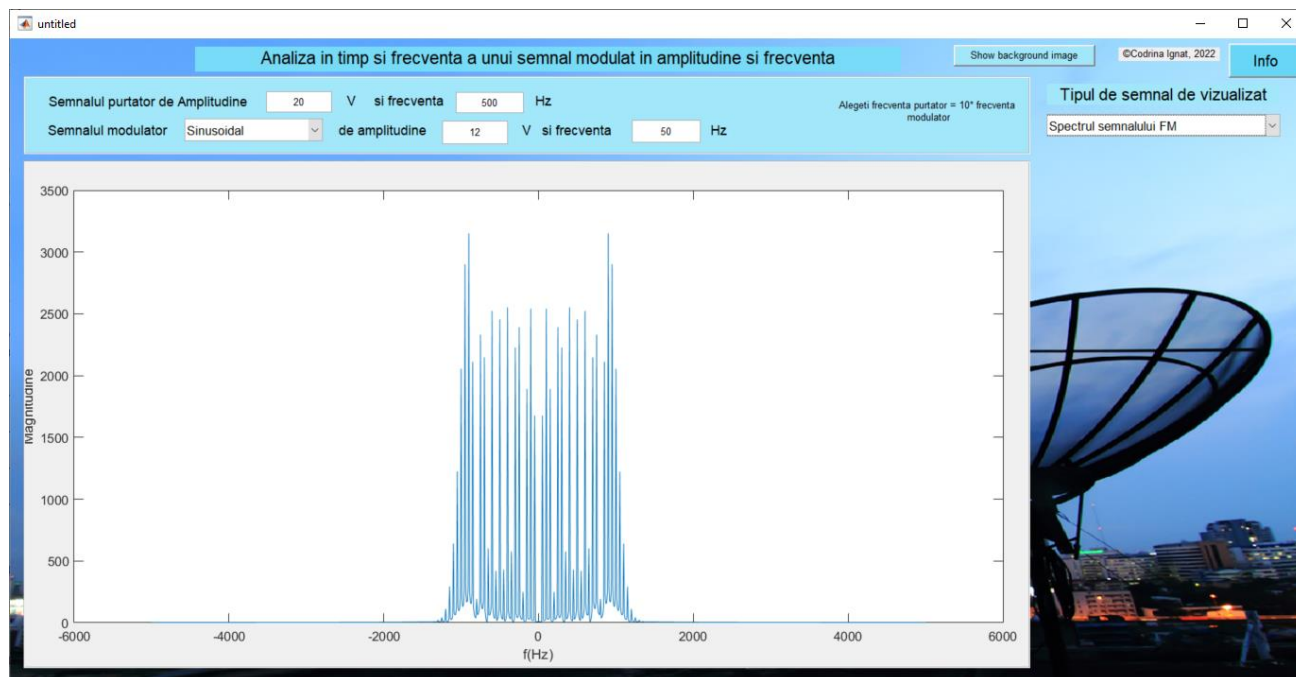
Vizualizarea semnalului modulat în frecvență:



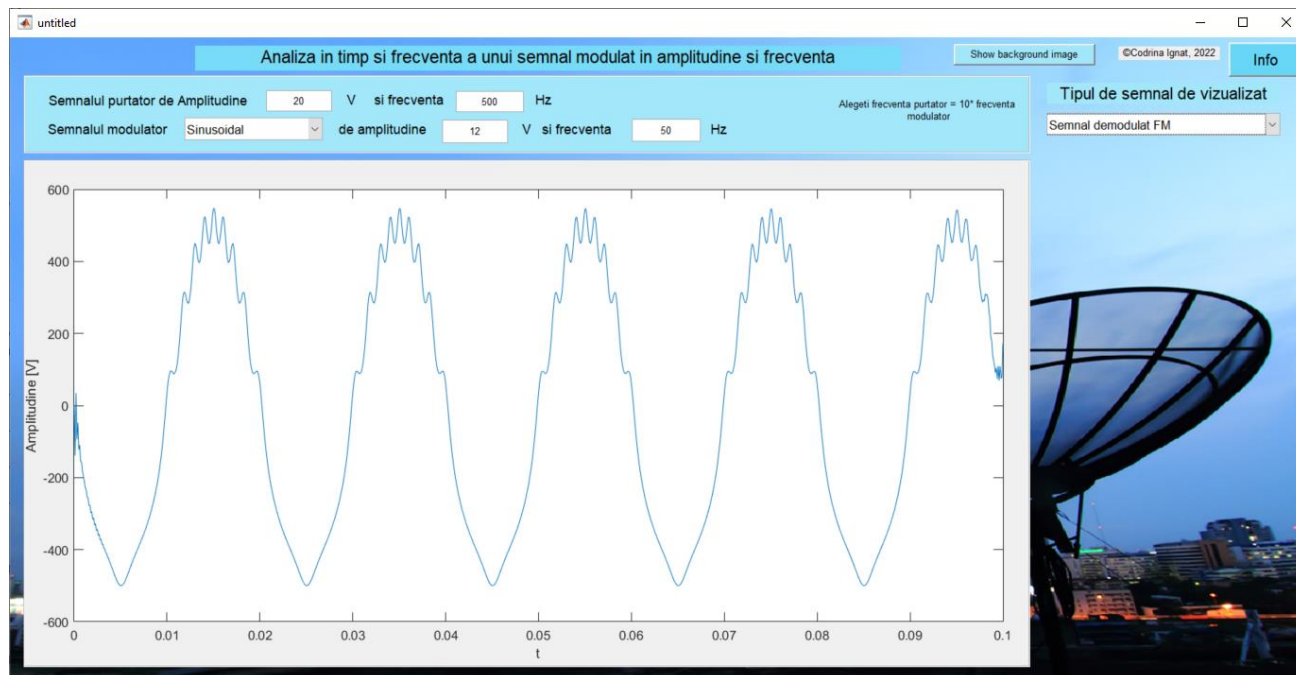
Vizualizarea semnalului modulat în frecvență însumat cu zgomot:



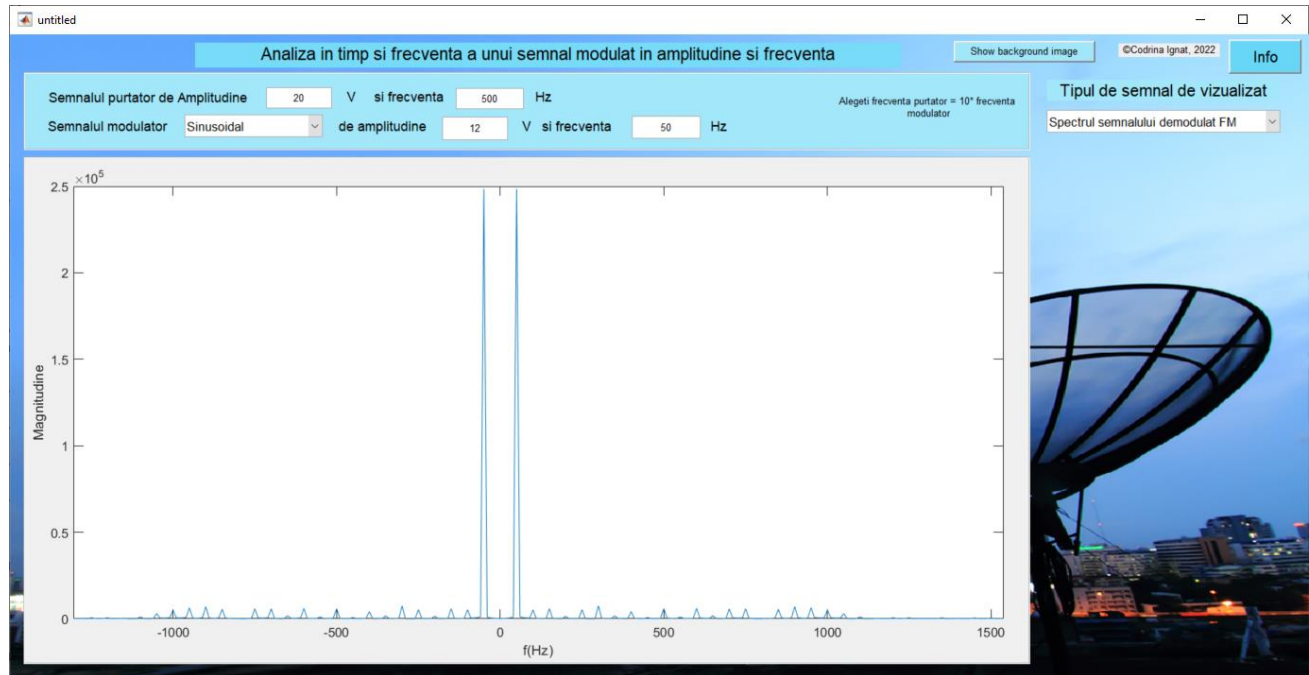
Vizualizarea spectrului semnalului modulat în frecvență:



Vizualizarea semnalului demodulat FM:



Vizualizarea spectrului semnalului demodulat FM:



5. Concluzii și interpretări

Aplicația funcționează conform obiectivelor propuse. Graficele sunt afișate corect, cu parametrii definiți în interfață. Programul realizează cu succes operațiunile de afișare a semnalelor, modulare, demodulare, afișarea spectrului semnalului selectat.

Alte funcționalități minore implementate:

- Posibilitatea de Zoom On pe fiecare grafic;
- Buton de Show background image, pentru a vizualiza imaginea de background a aplicației;
- Opțiunea de maximize, pentru a mări interfața grafică la dimensiunea ecranului utilizatorului;

6. Îmbunătățiri viitoare

O serie de îmbunătățiri pot fi aduse aplicației, pentru a extinde gama analizei semnalelor oferite de aceasta. Printre îmbunătățiri se pot număra:

- Posibilitatea de a alege dintr-o gamă mai variată de semnale modulatorie;
- Analiza unui semnal audio, cu afișarea analizei aferente, cu posibilitatea de a încărca sunetul respectiv dintr-un folder;
- Posibilitatea de a compara două semnale, prin suprapunerea acestora și evidențierea diferențelor (în semnalul modulat sau în spectru) cu o culoare diferită;
- Posibilitatea de a crea rapoarte pdf despre analiza semnalelor, care să cuprindă date despre semnalele modulator și purtător, modulat, spectre, comentarii ale utilizatorului;
- Posibilitatea de a încărca forme de undă diverse spre a fi analizate;
- Implementarea unor filtre cu frecvențe de tăiere editabile pe aceeași interfață, pentru a putea filtra anumite semnale și a afișa diverse informații (ex. caracteristica filtrului, semnalul înainte și după filtrare).

7. Bibliografie

- https://en.wikipedia.org/wiki/Amplitude_modulation
- https://en.wikipedia.org/wiki/Frequency_modulation
- <https://www.mathworks.com/help/comm/ref/fmdemod.html>
- <https://www.mathworks.com/help/comm/ref/amdemod.html>
- <https://www.mathworks.com/matlabcentral/answers/565580-amplitude-modulation-demodulation-signal>

8. Anexe

Codul aplicației în Matlab:

```
function varargout = untitled(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @untitled_OpeningFcn, ...
                  'gui_OutputFcn',    @untitled_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before untitled is made visible.
function untitled_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
% create an axes that spans the whole gui
ah = axes('unit', 'normalized', 'position', [0 0 1 1]);
% import the background image and show it on the axes
bg = imread('radio-antenna.png'); imagesc(bg);
% prevent plotting over the background and turn the axis off
set(ah, 'handlevisibility', 'off', 'visible', 'off')
% making sure the background is behind all the other uicontrols
uistack(ah, 'bottom');
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = untitled_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```



```

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

function edit4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)

function edit5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

function edit6_CreateFcn(hObject, eventdata, handles)

```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

function edit7_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupmenu1_Callback(hObject, eventdata, handles)

a = get(handles.edit1,'string');
Am = str2num(a);
b = get(handles.edit2,'string');
fm = str2num(b);
c = get(handles.edit4,'string');
Ac = str2num(c);
d = get(handles.edit5,'string');
fc = str2num(d);

fs = 40001;      %frecventa de esantionare
Ts = 1/fs;      %Perioada de esantionare
durata = 0.1;
N = 20000;      %Numar de esantioane
%t = 0: Ts : durata;
t = 0: Ts :N*Ts-Ts;
%f = [-fs/2:fs/N:fs/2-fs/N];

%axa frecventei
m = 1; %indicele de modulatie
Semnal_purtator=Ac*sin(2*pi*fc*t);

Semnal_modulator = get(handles.popupmenu2,'value')
switch Semnal_modulator
    case 1
        Semnal_modulator=sin(2*pi*fm*t);
    case 2
        Semnal_modulator=square(2*pi*fm*t);
    case 3
        Semnal_modulator=sawtooth(2*pi*fm*t);
    case 4

Semnal_modulator=sin(2*pi*fm*t)+10*sin(2*pi*10*fm*t)+100*sin(2*pi*100*fm*t);

```

```

end

ModulatAM = (Ac*(1+m*Semnal_modulator)).*sin(2*pi*fc*t);
SpectruSemnalAM = fft(ModulatAM);

IndiceModFM = 10;
%ModulatFM = Ac*sin(2*pi*fc*t + IndiceModFM.*cos(2*pi*fm*t));
ModulatFM = Ac*sin(2*pi*fc*t + IndiceModFM.*Semnal_modulator);
SpectruSemnalFM = fft(ModulatFM);

DemodulatAM = amdemod(ModulatAM, fc, fs, 1);

DemodulatFM = fmdemod(ModulatFM, fc, fs, 1);

Zgomot = 2*randn(size(ModulatAM));
SemnalAMcuzgomot = ModulatAM + Zgomot;

ZgomotFM = 2*randn(size(ModulatFM));
SemnalFMcuzgomot = ModulatFM + ZgomotFM;

k = get(handles.popupmenu1, 'value')
switch k
    case 1
        %semnal modulator
        plot(t, Am*Semnal_modulator);
        xlabel('t');
        ylabel('Amplitudine [V]');
        zoom on
    case 2
        %Semnal purtator
        plot(t, Semnal_purtator);
        xlabel('t');
        ylabel('Amplitudine [V]');
    case 3
        %Semnal AM
        plot(t, ModulatAM);
        xlabel('t');
        ylabel('Amplitudine [V]');
    case 4
        %Semnal AM cu zgomot
        plot(t, SemnalAMcuzgomot);
        xlabel('t');
        ylabel('Amplitudine [V]');
    case 5
        %Spectrul semnalului AM
        axa1 = linspace(-fs/2, fs/2, length(ModulatAM));
        AM = abs(fftshift(fft(ModulatAM)));
        plot(axa1, AM);
    case 6
        %Semnal demodulat AM
        plot(t, DemodulatAM);

```

```

        xlabel('t');
        ylabel('Amplitudine [V]');
    case 7
        %Spectrul semnalului demodulat AM
        axa2 = linspace(-fs/2, fs/2, length(DemodulatAM));
        DAM = abs(fftshift(fft(DemodulatAM)));
        plot(axa2, DAM);
        xlabel("f(Hz)")
        ylabel("Magnitudine");
    case 8
        %Semnal FM
        plot(t,ModulatFM);
        xlabel('t');
        ylabel('Amplitudine [V]');
    case 9
        %Semnal FM cu zgomot
        plot(t,SemnalFMcuzgomot);
        xlabel('t');
        ylabel('Amplitudine [V]');
    case 10
        %Spectrul semnalului FM
        axa3 = linspace(-fs/2, fs/2, length(ModulatFM));
        FM = abs(fftshift(fft(ModulatFM)));
        plot(axa3, FM);
        xlabel("f(Hz)")
        ylabel("Magnitudine");
    case 11
        %Semnal demodulat FM
        plot(t, DemodulatFM);
        xlabel('t');
        ylabel('Amplitudine [V]');
    case 12
        %Spectrul semnalului demodulat FM
        axa4 = linspace(-fs/2, fs/2, length(DemodulatFM));
        DFM = abs(fftshift(fft(DemodulatFM)));
        plot(axa4, DFM);
        xlabel("f(Hz)")
        ylabel("Magnitudine");
end

```

```

function popupmenu1_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit8_Callback(hObject, eventdata, handles)

```

```

function edit8_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupmenu2_Callback(hObject, eventdata, handles)

function popupmenu2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)

% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
Info_button;

% --- Executes during object creation, after setting all properties.
function text13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
imshow('radio-antenna.png');
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```