CERINȚĂ

Să se proiecteze un simulator pentru semnale cu modulaţie analogică care va conţine un modulator şi demodulatorul aferent, ce pot fi conectate între ele printr-un canal ideal, de caracteristică unitară, afectat de un zgomot aditiv gaussian alb, de medie nulă şi densitate spectrală de putere $N_0/2$ (prin intermediul unui raport semnal zgomot ce va fi dată de intrare, in dB). Simulatorul va permite totodată simularea transmisiunii şi în absenţa zgomotului. Simulatorul va fi deschis, permiţând conectarea şi altor tipuri de canale de comunicaţie, trebuind să se specifice formatul în care acestea pot fi simulate.

Simulatorul va permite aplicarea la intrare a unui semnal analogic format din suma a doua componente sinusoidale de amplitudini A şi B si frecvente f1 si f2, furnizate într-un format adecvat, sau va permite aplicarea unui semnal de intrare predefinit într-un format care se va preciza. Frecvenţa f0 si amplitudinea P a purtătoarei constituie de asemnea date de intrare. Aceşti parametri se vor preciza printr-un interval de valori, simulatorul trebuind să funcţioneze pentru orice valoare din acest interval. De asemenea se va preciza tipul de modulaţie şi anumite particularităţi ale semnalului de intrare (dacă este cazul).

Semnalul modulat generat va putea fi reprezentat grafic pe un interval de timp ce constituie dată de intrare, cuprins între perioada minimă a purtătoarei şi perioada sinusoidei de intrare de frecvenţă minimă. De asemenea se vor reprezenta grafic semnalul modulat şi semnalul demodulat (pe acelaşi grafic).

Demodulatorul va putea lucra cu informaţii de sincronizare ideale (parametrii utilizaţi la emisie) sau va include un bloc de sincronizare de purtătoare, dacă este cazul, cu o structură cât mai simplă.

Se va analiza si se va propune explicit o metoda de a transpune modulatorul într-un FPGA.

Se recomandă ca simulatorul să prezinte o interfaţă grafică pentru introducerea datelor şi afişarea/prezentarea rezultatelor simulării.

Simulatorul va fi realizat într-un limbaj de programare / mediu de simulare care este la latitudinea celui care realizează tema.

La predarea proiectului se va preda simulatorul în format electronic şi un memoriu (format electronic şi pe hârtie, îndosariat, având anexat un CD). Memoriul va contine:

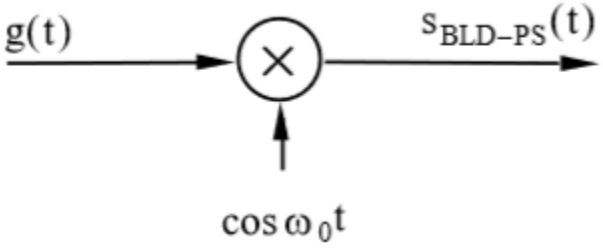
- formularea completă a temei (inclusiv datele de intrare);
- descrierea resurselor necesare pentru utlizarea simulatorului;
- descrierea simulatorului şi a soluţiei alese, incluzând comparaţii ale soluţiei alese cu alte soluţii posibile (motivaţia alegerii solutiei);
- determinarea caracteristicii de modulare / demodulare şi reprezentarea grafică corespunzătoare pentru domeniul de valori indicat în temă;
- soluția propusa pentru transpunerea într-un FPGA a modulatorului;
- bibliografia consultată;
- listing-ul funcţiilor ce compun programul de simulare;
- programul de simulare (în format electronic).

	Date de intrare:		Mod BLU-I, metoda conversiilor succesive, RSZ=20...30 dB, A=1...2, B=1...2, P=1...2, f0=1000...5000, f1=1...10, f2=5...20

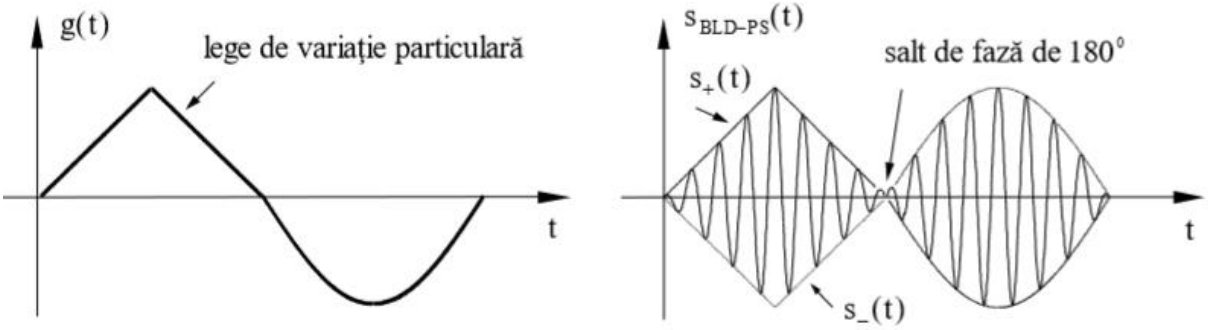
NOȚIUNI TEORETICE

Semnale BLD

Un semnal BLD-PS se obține prin schema următoare:



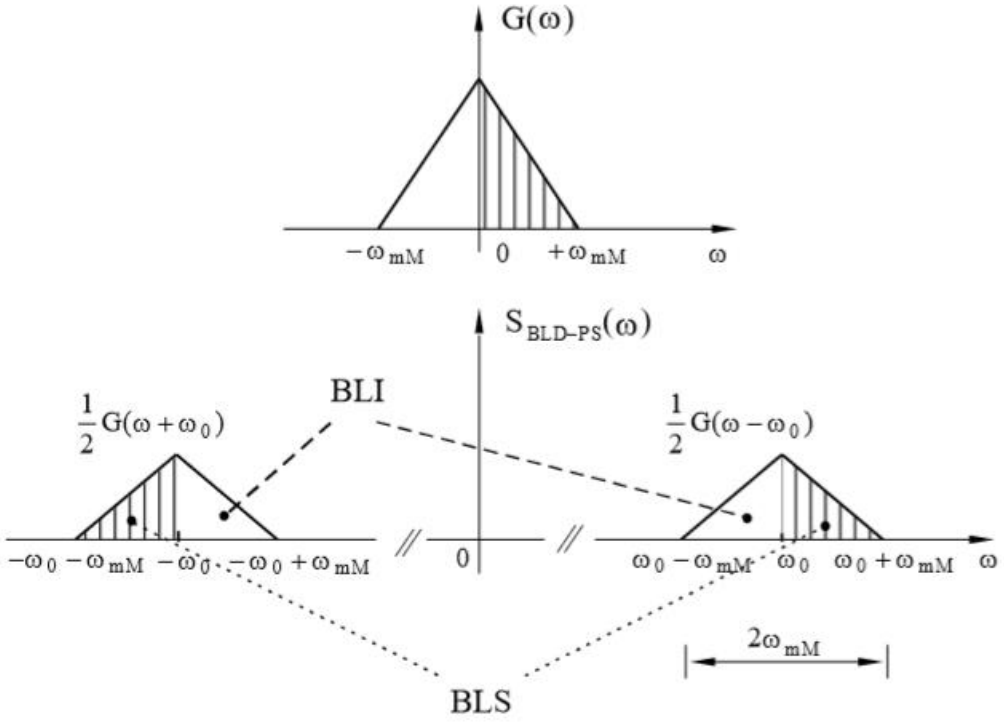
Pentru un semnal modulator oarecare g(t), un semnal BLD-PS are forma



La trecerea sa prin zero, anvelopa schimbă semnul oscilației purtătoare, deci introduce un salt de fază de 180 de grade.

Spectrul semnalului BLD-PS:

$$s_{BLD-PS}(t) = g(t)\cos\omega_0 t \quad \overset{F}{\longleftrightarrow} \quad S_{BLD-PS}(\omega) = \frac{1}{2}G(\omega-\omega_0)+\frac{1}{2}G(\omega+\omega_0)$$



Semnalul BLD-PS conține 2 benzi laterale:

BLS corespunde lui $G(\omega)$ pentru $\omega \geq 0$

BLI corespunde lui $G(\omega)$ pentru $\omega \leq 0$

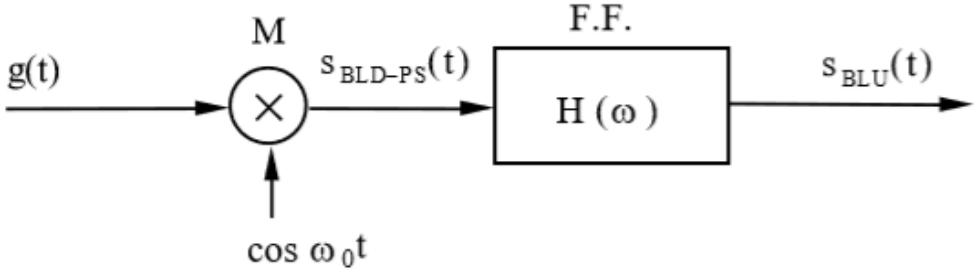
Semnale BLU

Ideea despre semnalele BLU a venit de la faptul că matematic, se poate deduce întreg semnalul dintr-una din benzile laterale cunoscută, deoarece pentru

$$g(t) \in \mathbf{R} \quad \Rightarrow \quad G(-\omega) = G^*(\omega)$$

Deoarece una din BL conține toată informația (cealaltă BL putând fi dedusă din cea cunoscută) este inutil să se transmită ambele benzi.

Obținerea semnalelor BLU-I sau BLU-S se face prin schema:



Unde, în cazul BLU-S avem

$$H_S(\omega) = \begin{cases} 1 & \text{pentru } |\omega| \geq \omega_0 \\ 0 & \text{pentru } |\omega| \leq \omega_0 \end{cases}$$

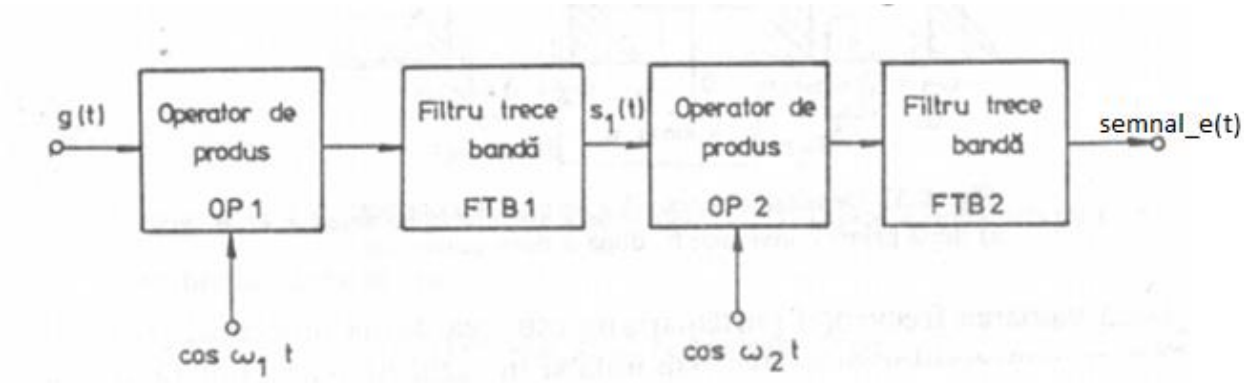
Iar în cazul BLU-I avem

$$H_I(\omega) = 1 - H_S(\omega)$$

Semnale BLU - Metoda conversiilor succesive

Pentru a avea condiții convenabile de realizare a filtrului FF se recurge la metoda conversiilor succesive.

Schema bloc:



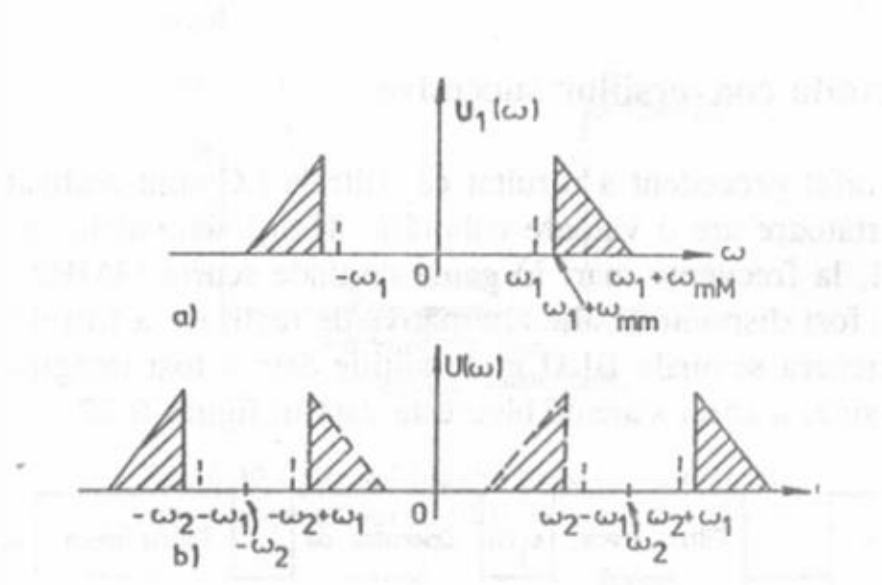
W1=2*pi*fp1

W2=2*pi*fp2

Fp1 se alege 40…60kHz

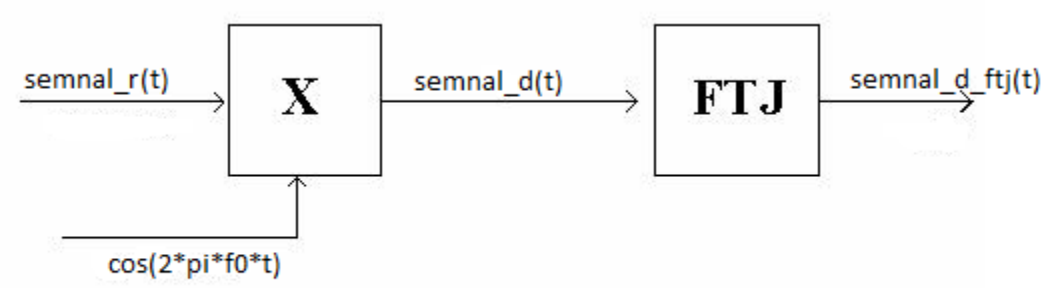
Fp1+Fp2=f0

Spectrele semnalelor după prima, respectiv după a doua conversie:



Demodulare – Metoda demodulării de produs

Schema bloc:



Filtrul trece-jos FTJ de la ieșirea circuitului de multiplicare are rolul de a permite trecerea semnalului de joasa frecvență, blocând componentele cu frecvența 2f0.

ETAPE

1) Generarea celor două sinusuri
   Sinus1(t)=Asin(2*pi*f1*t)
   Sinus2(t)=Bsin(2*pi*f2*t)

2) Generarea semnalului de intrare, prin însumarea sinusurilor
   G(t)=sinus1(t)+sinus2(t)

3) Proiectarea mixerelor și a filtrelor din metoda conversiilor successive
   Primul mixer+FTB:
   P1(t)=sqrt(P)cos(2*pi*fp1*t)
   S1(t)=FTB1(g(t)*p1(t))
   Al doilea mixer+FTB:
   P2(t)=sqrt(P)cos(2*pi*fp2*t)
   S2(t)=FTB2(s1(t)*p2(t))

4) Definirea semnalului BLU emis
   Semnal_e(t)=s2(t)

5) Definirea canalului de comunicație
   Semnalul receptat este afectat de zgomot:
   Semnal_r(t)=semnal_e(t)+ZAGA

6) Proiectarea demodulatorului de produs
   P0(t)=Pcos(2*pi*f0*t)
   Semnal_d(t)=semnal_r(t)*P0(t)

7) Proiectarea filtrului FTJ și definirea semnalului demodulat final
   Semnal_d_ftj(t)=FTJ(semnal_d(t))

SIMULATOR, RESURSE

Simulator: Matlab R2015a

Resurse necesare (Windows):

32-Bit and 64-Bit MATLAB and Simulink Product Families				
Operating Systems	**Processors**	**Disk Space**	**RAM**	**Graphics**
Windows 10 Windows 8.1 Windows 8 Windows 7 Service Pack 1 Windows Vista Service Pack 2 Windows XP Service Pack 3	Any Intel or AMD x86 processor supporting SSE2 instruction set*	1 GB for MATLAB only, 3–4 GB for a typical installation	2 GB	No specific graphics card is required. Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory recommended.

CODUL, EXPLICAT:

```
clear all;
close all;
clc;

A=1;          %amplitudine sinus1
B=1;          %amplitudine sinus2
P=2;          %amplitudine purtatoare

f1=1000;          %frecventa sinus1
f2=5000;          %frecventa sinus2
f0=1000000;       %frecventa purtatoare

fp1=50000;         %frecventa purtatoare1 (schema)
fp2=f0-fp1;        %frecventa purtatoare2 (schema)

Fs=10*f0;                  %frecventa de esantionare
tstop=max(1/f1,1/f2);      %timpul de afisare
t=0:1/Fs:tstop;            %timpul


%----------------------------%
%----------EMITATOR----------%
%----------------------------%
```

```
%cele 2 sinusuri care formeaza semnalul de la intrare
sinus1=A*sin(2*pi*f1*t);
sinus2=A*sin(2*pi*f2*t);
%semnalul de la intrare
g=sinus1+sinus2;

%cele 2 purtatoare (schema)
p1=sqrt(P)*cos(2*pi*fp1*t);
p2=sqrt(P)*cos(2*pi*fp2*t);

%cele 2 filtre
%f1
wp_f1=2*pi*[fp1-max(f1,f2),fp1];
ws_f1=2*pi*[1,fp1*2];
Rp1=1;
Rs1=40;
[n1,w_f1]=ellipord(wp_f1,ws_f1,Rp1,Rs1,'s');
f_taiere_ftb1=w_f1/(2*pi)
[b1,a1]=ellip(n1,Rp1,Rs1,w_f1,'bandpass','s');
[bd1,ad1]=impinvar(b1,a1,Fs);
%f2
wp_f2=2*pi*[fp1-max(f1,f2),fp1];
ws_f2=2*pi*[(fp1-max(f1,f2))*0.5,fp1*1.5];
Rp2=0.5;
Rs2=50;
[n2,w_f2]=ellipord(wp_f2,ws_f2,Rp2,Rs2,'s');
f_taiere_ftb2=w_f2/(2*pi)
[b2,a2]=ellip(n2,Rp2,Rs2,w_f2,'bandpass','s');
[bd2,ad2]=impinvar(b2,a2,Fs);

%cele 2 semnale (dupa prima conversie si dupa a doua conversie) (schema)
s1=g.*p1;
    %s1=filter(bd1,ad1,s1);
s2=s1.*p2;
    %s2=filter(bd2,ad2,s2);
semnal_e=s2; %semnalul modulat, emis

%-----------------------------------------%
%----------CANALUL DE COMUNICATIE----------%
%-----------------------------------------%

RSZ_ON=0;
%valoarea RSZ, in dB
RSZ=25;
%adaugarea ZAGA peste semnal
if RSZ_ON
semnal_r=awgn(semnal_e,RSZ); %semnalul modulat, receptat
else
semnal_r=semnal_e;
end
```

```
%----------------------------%
%----------RECEPTOR----------%
%----------------------------%

%demodulator produs
p0=cos(2*pi*f0*t);
semnal_d=semnal_r.*p0;


%filtru FTJ
%wp_f3=2*pi*max(f1,f2);
wp_f3=2*pi*1;
ws_f3=2*pi*f0-wp_f3;
Rp3=1;
Rs3=40;
[n3,wt3]=buttord(wp_f3,ws_f3,Rp3,Rs3,'s');
frecv_taiere_f3=wt3/(2*pi)
[b3,a3]=butter(n3,wt3,'s');
[bd3,ad3]=impinvar(b3,a3,Fs);

%filtrarea FTJ
semnal_d_ftj=2*filter(bd3,ad3,semnal_d);


%----------------------------%
%----------ERORI-------------%
%----------------------------%
%eroarea absoluta
e_abs=abs(g-semnal_d_ftj);
%eroarea relativa
e_rel=(e_abs./abs(g))*100;



%----------------------------%
%----------SPECTRE-----------%
%----------------------------%

N=length(t)*10;
f=(0:Fs/N:Fs-1);
fshift=(-Fs/2:Fs/N:Fs/2-1);

%spectrul semnalului dupa prima conversie
spectru_s1=fft(s1,N)*1/N;
%spectrul semnalului dupa a doua conversie (semnal BLU emis)
spectru_e=fft(semnal_e,N)*1/N;
%spectrul semnalului BLU afectat de zgomot (semnal BLU receptat)
spectru_r=fft(semnal_r,N)*1/N;
%spectrul semnalului dupa demodulatorul de produs
spectru_d=fft(semnal_d,N)*1/N;
%spectrul semnalului dupa demodulatorul de produs si FTJ
spectru_d_ftj=fft(semnal_d_ftj,N)*1/N;
```

```
%-----------------------------%
%----------REZULTATE----------%
%-----------------------------%
figure(1);
subplot(2,1,1);
plot(t,sinus1); hold on; plot(t,sinus2);
title('Cele doua sinusuri care formeaza semnalul de la intrare');
xlabel('t[s]'); ylabel('Amplitudine');
legend('sinus1 (f1)','sinus2 (f2)');
subplot(2,1,2);
plot(t,g);
title('Semnalul de la intrare');
xlabel('t[s]'); ylabel('Amplitudine');

figure(2);
subplot(3,1,1);
plot(t,g);
title('Semnalul de la intrare');
xlabel('t[s]'); ylabel('Amplitudine');
subplot(3,1,2);
plot(t,s1);
title('Semnalul modulat, dupa prima conversie');
xlabel('t[s]'); ylabel('Amplitudine');
subplot(3,1,3);
plot(t,semnal_e);
title('Semnalul modulat, dupa a doua conversie = Semnalul BLU');
xlabel('t[s]'); ylabel('Amplitudine');

figure(3);
subplot(2,1,1);
plot(fshift, abs(fftshift(spectru_s1)));
title('Spectrul semnalului modulat - dupa prima conversie');
xlabel('f[Hz]'); ylabel('Amplitudine');
subplot(2,1,2);
plot(fshift, abs(fftshift(spectru_e)));
title('Spectrul semnalului modulat - dupa a 2a conversie');
xlabel('f[Hz]'); ylabel('Amplitudine');

figure(4);
subplot(2,1,1);
plot(t,semnal_e);
title('Semnalul BLU emis, neafectat de zgomot');
xlabel('t[s]'); ylabel('Amplitudine');
subplot(2,1,2);
plot(t,semnal_r);
title('Semnalul BLU receptat, afectat de zgomot');
xlabel('t[s]'); ylabel('Amplitudine');

figure(5);
subplot(2,1,1);
plot(fshift, abs(fftshift(spectru_e)));
title('Spectrul semnalului emis');
```

```
xlabel('f[Hz]'); ylabel('Amplitudine');
subplot(2,1,2);
plot(fshift, abs(fftshift(spectru_r)));
title('Spectrul semnalului receptat');
xlabel('f[Hz]'); ylabel('Amplitudine');

figure(6);
subplot(3,1,1);
plot(fshift, abs(fftshift(spectru_r)));
title('Spectrul semnalului modulat receptat');
xlabel('f[Hz]'); ylabel('Amplitudine');
subplot(3,1,2);
plot(fshift, abs(fftshift(spectru_d)));
title('Spectrul semnalului modulat, dupa demodulare produs');
xlabel('f[Hz]'); ylabel('Amplitudine');
subplot(3,1,3)
plot(fshift, abs(fftshift(spectru_d_ftj)));
title('Spectrul semnalului modulat, dupa demodulare produs si FTJ');
xlabel('f[Hz]'); ylabel('Amplitudine');

figure(7)
subplot(3,1,1)
plot(t,semnal_d_ftj)
title('Semnalul demodulat');
xlabel('t[s]'); ylabel('Amplitudine');
subplot(3,1,2)
plot(t,semnal_d_ftj,'.')
hold on
plot(t,semnal_r)
title('Semnalul demodulat VS semnalul BLU receptat');
xlabel('t[s]'); ylabel('Amplitudine');
legend('semnal demodulat','semnal BLU receptat');
subplot(3,1,3)
plot(t,semnal_d_ftj)
hold on
plot(t,g)
title('Semnalul demodulat VS semnalul initial');
xlabel('t[s]'); ylabel('Amplitudine');
legend('semnal demodulat','semnal initial');

figure(8)
subplot(2,1,1)
plot(t,semnal_d_ftj)
hold on
plot(t,g)
title('Semnalul demodulat VS semnalul initial');
xlabel('f[Hz]'); ylabel('Amplitudine');
legend('semnal demodulat','semnal initial');
subplot(2,1,2)
plot([1:1:length(t)],e_abs,'.')
title('Eroarea absoluta (semnal initial - semnal demodulat)');
xlabel('esantion'); ylabel('eroare');
```

REZULTATE

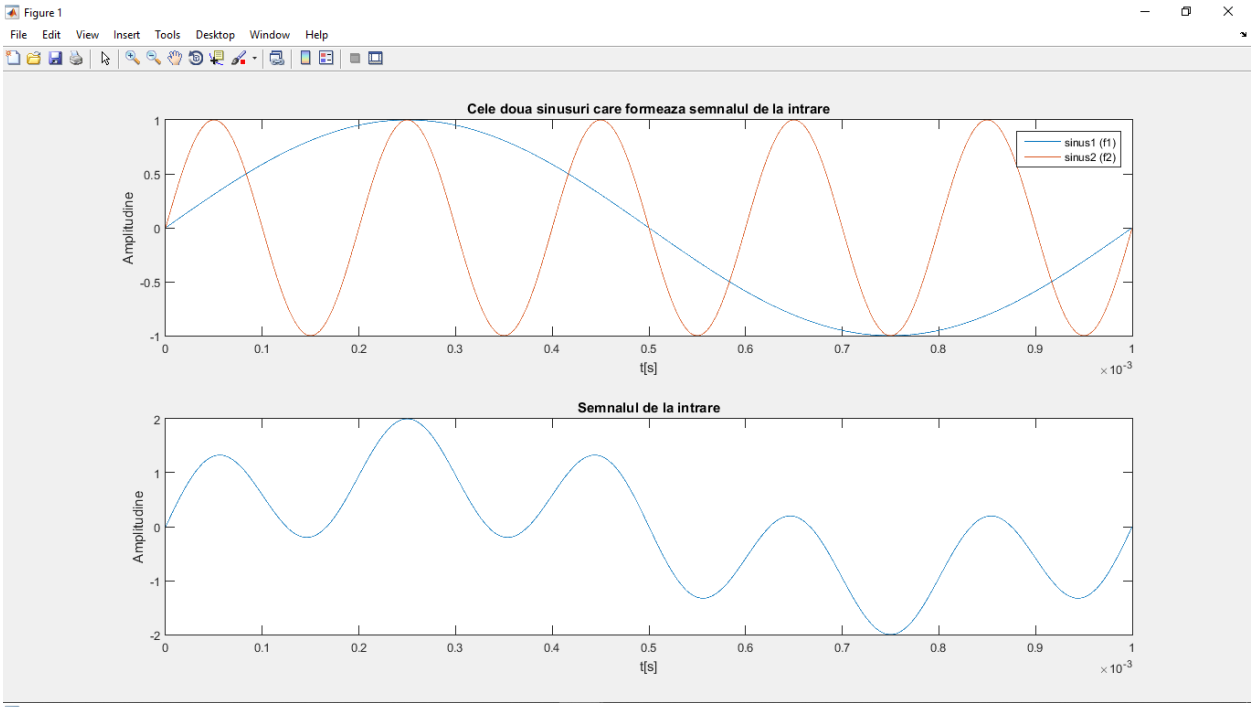


Figure 1
File Edit View Insert Tools Desktop Window Help
Cele doua sinusuri care formeaza semnalul de la intrare
sinus1 (f1)
sinus2 (f2)
Amplitudine
t[s]
×10^-3
Semnalul de la intrare
Amplitudine
t[s]
×10^-3

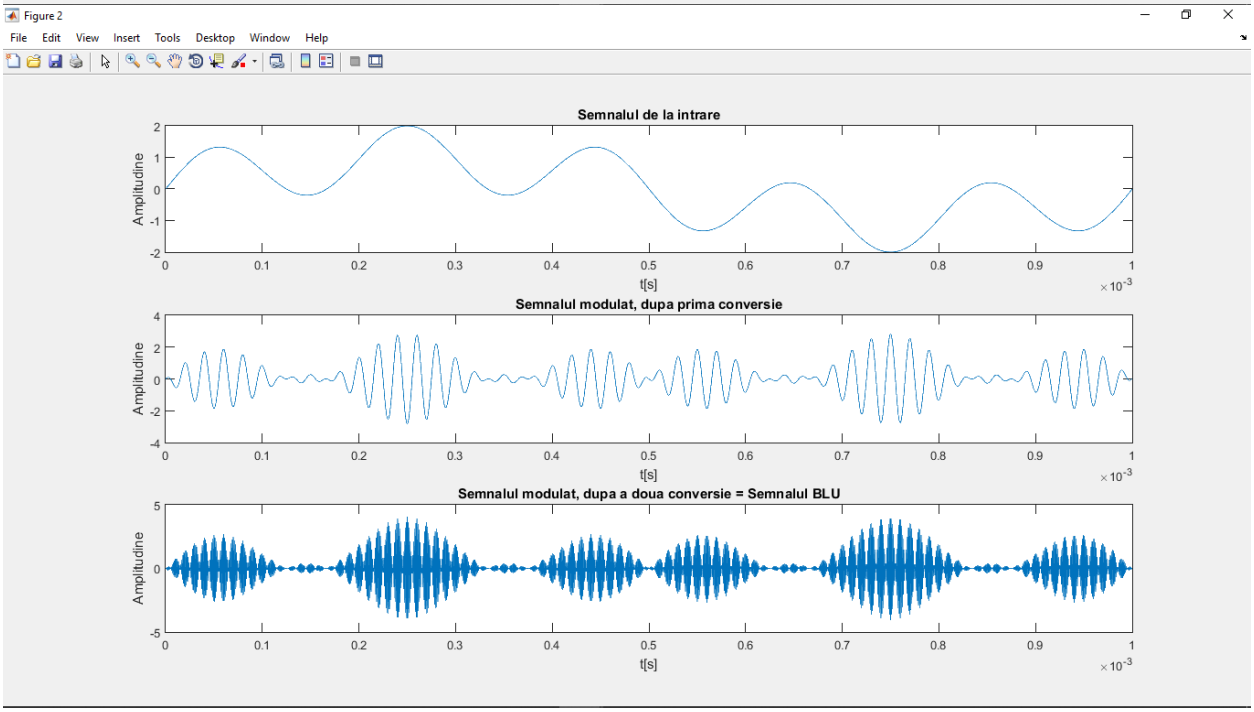


Figure 2
File Edit View Insert Tools Desktop Window Help
Semnalul de la intrare
Amplitudine
t[s]
×10^-3
Semnalul modulat, dupa prima conversie
Amplitudine
t[s]
×10^-3
Semnalul modulat, dupa a doua conversie = Semnalul BLU
Amplitudine
t[s]
×10^-3

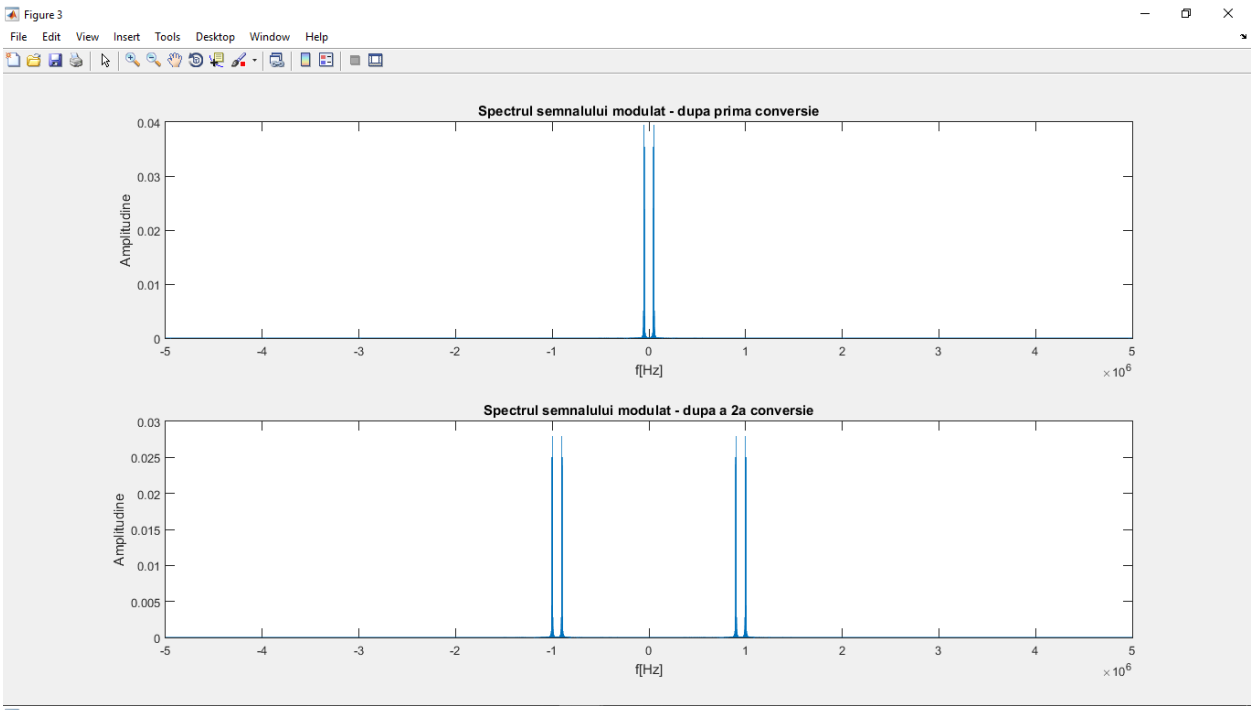


Figure 3
File Edit View Insert Tools Desktop Window Help
Spectrul semnalului modulat - dupa prima conversie
Amplitudine
f[Hz]
×10^6
Spectrul semnalului modulat - dupa a 2a conversie
Amplitudine
f[Hz]
×10^6

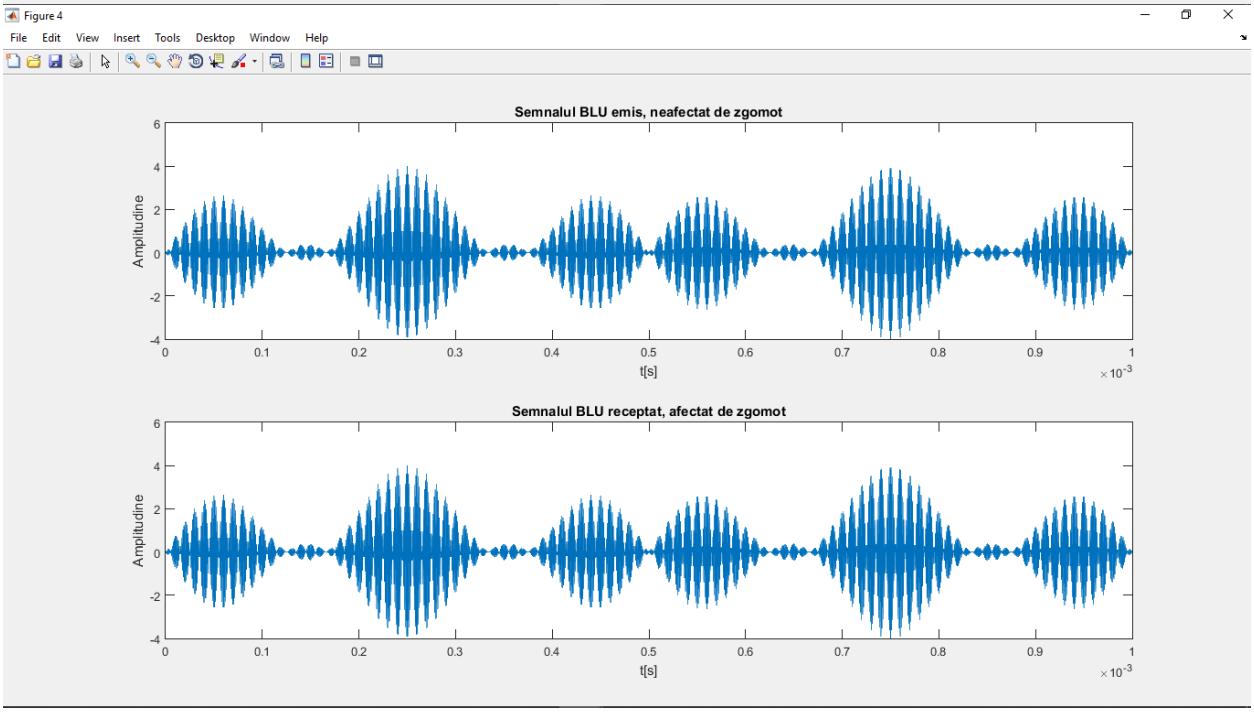


Figure 4
File Edit View Insert Tools Desktop Window Help
Semnalul BLU emis, neafectat de zgomot
Amplitudine
t[s]
×10^-3
Semnalul BLU receptat, afectat de zgomot
Amplitudine
t[s]
×10^-3

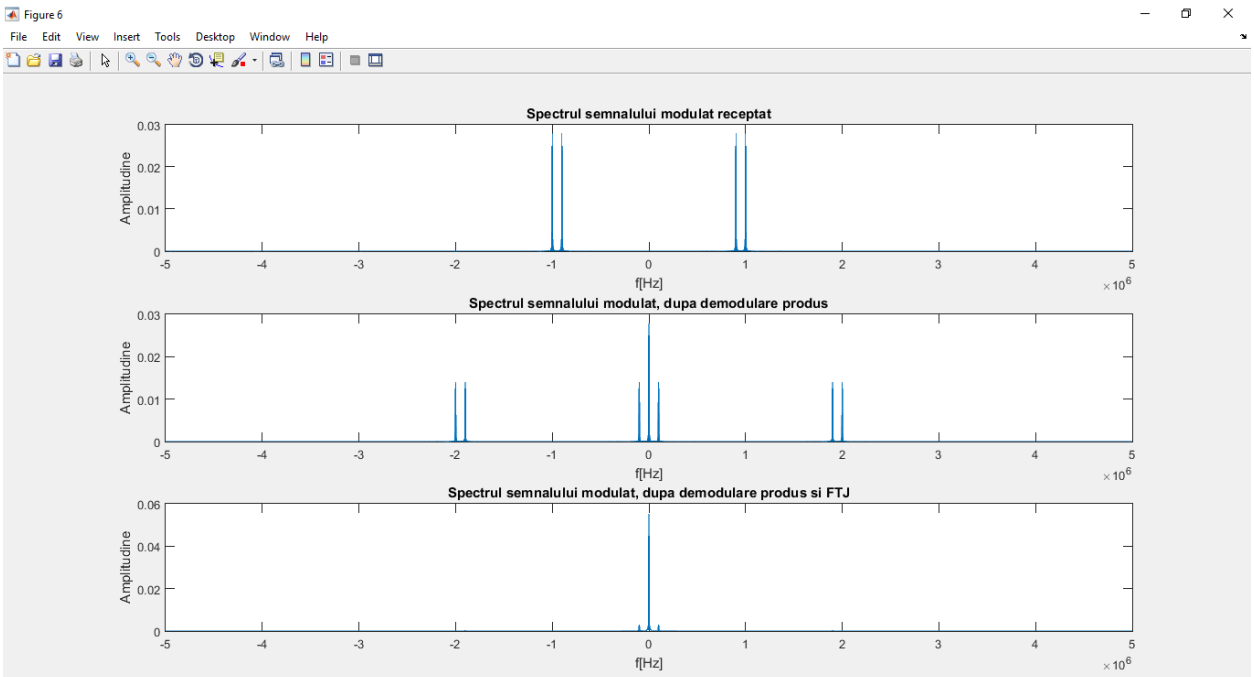


Figure 6
File Edit View Insert Tools Desktop Window Help
Spectrul semnalului modulat receptat
Amplitudine
f[Hz]
×10^6
Spectrul semnalului modulat, dupa demodulare produs
Amplitudine
f[Hz]
×10^6
Spectrul semnalului modulat, dupa demodulare produs si FTJ
Amplitudine
f[Hz]
×10^6

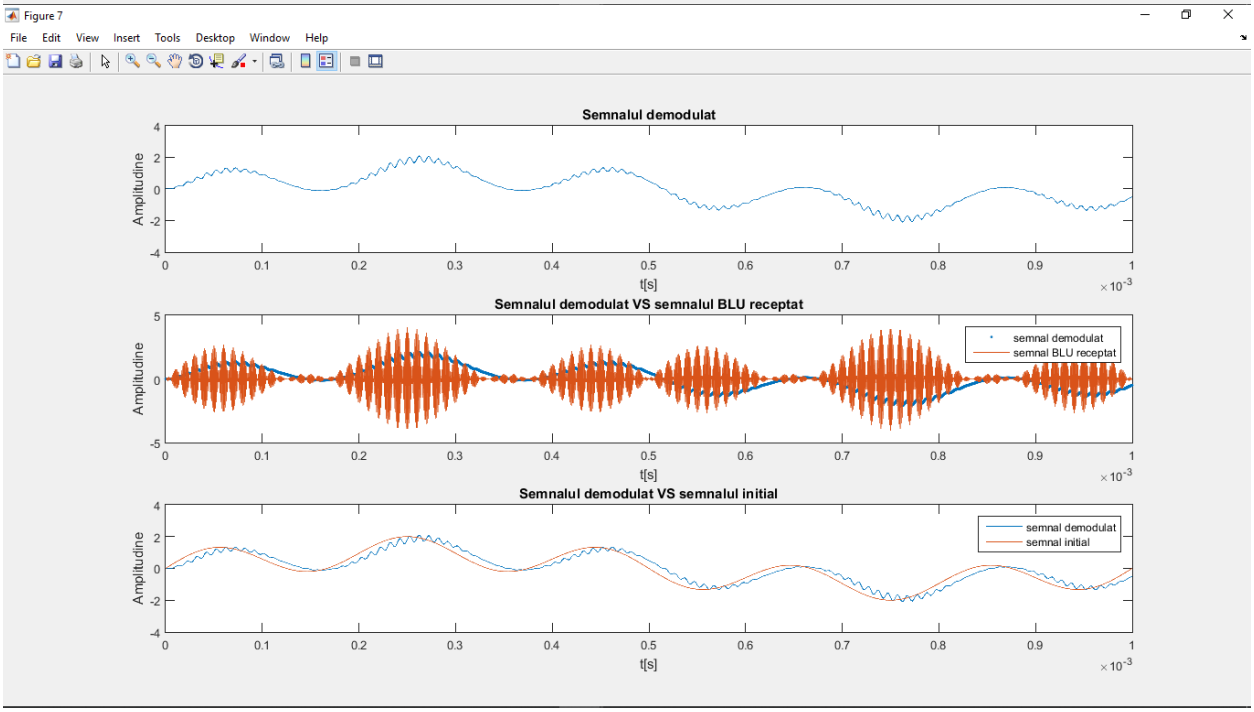


Figure 7
File Edit View Insert Tools Desktop Window Help
Semnalul demodulat
Amplitudine
t[s]
×10^-3
Semnalul demodulat VS semnalul BLU receptat
semnal demodulat
semnal BLU receptat
Amplitudine
t[s]
×10^-3
Semnalul demodulat VS semnalul initial
semnal demodulat
semnal initial
Amplitudine
t[s]
×10^-3

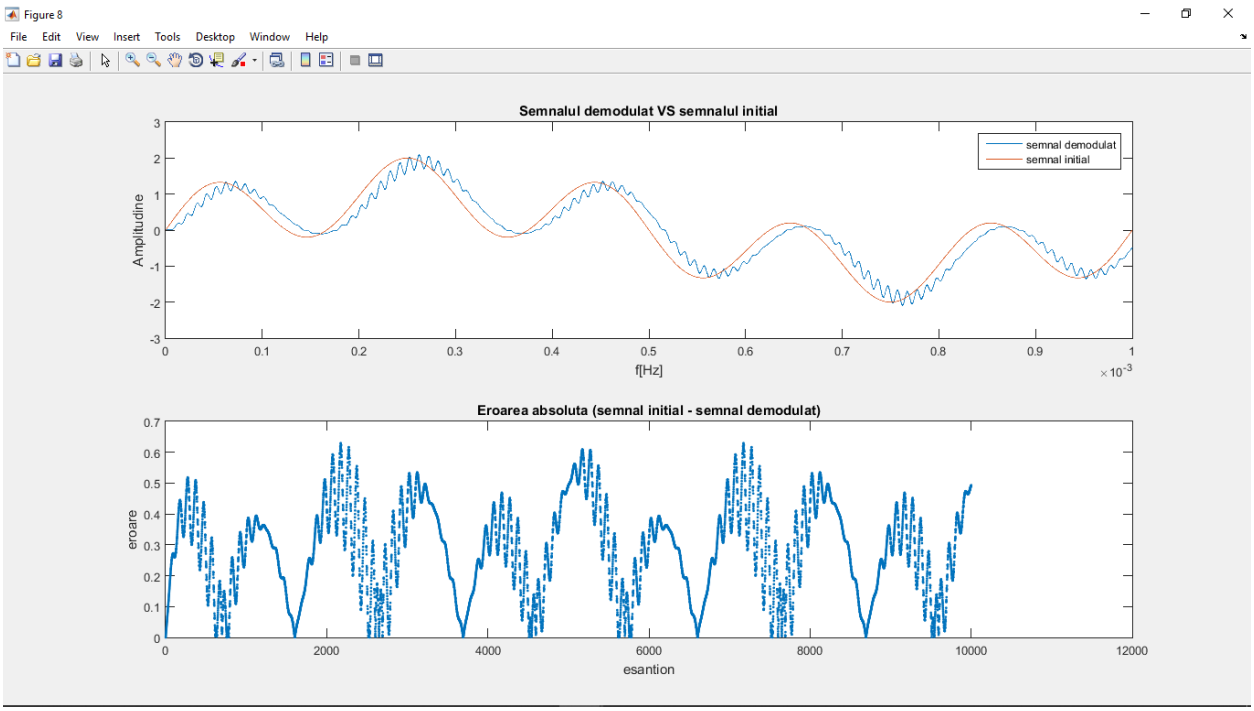
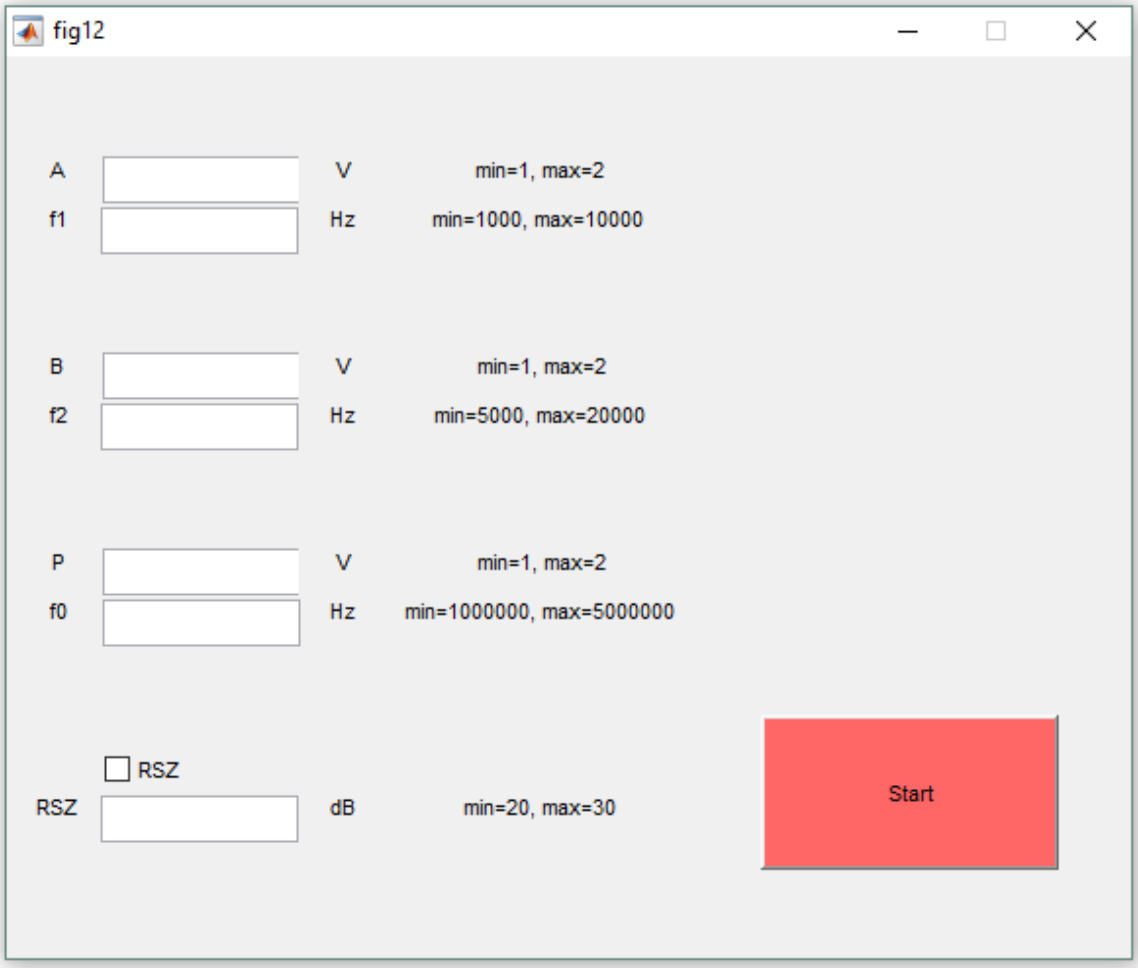


Figure 8
File Edit View Insert Tools Desktop Window Help
Semnalul demodulat VS semnalul initial
semnal demodulat
semnal initial
Amplitudine
f[Hz]
×10^-3
Eroarea absoluta (semnal initial - semnal demodulat)
eroare
esantion

INTERFAȚA GRAFICĂ

Pentru programul scris în MATLAB, am creat o interfață grafică folosind MATLAB GUIDE. Interfața grafică cuprinde:

- 7 câmpuri de completat, pentru valorile parametrilor A, f1, B, f2, P, p0, RSZ. Valorile introduse în aceste câmpuri sunt de tip string, deci vor fi convertite la valori de tip double prin instrucțiunea str2double().
- Un checkbox, pentru adăugarea sau nu a zgomotului în canalul de comunicație. ”Bifat” implică valoarea 1, iar ”nebifat” implică valoarea 0.
- Un buton de tip pushbutton, care startează simularea (programul și afișarea rezultatelor)



BIBLIOGRAFIE

Curs TAD (ETTI, UPB,2019)
Curs CAD (ETTI, UPB,2019)

CUPRINS