Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**Raport**

Disciplina: Prelucrarea semnalelor

Lucrarea de laborator nr.4

Tema: **Eșantionarea și cuantizarea semnalelor. Interpolarea semnalelor eșantionate**.

A efectuat: Zavorot Daniel, gr. TI-194

A verificat: asis. univ. Dubac Serghei

**Chişinău 2022**

**Exercițiul 1.1**

Reprezentați grafic 1024 de eșantioane ale unui semnal alcătuit din 2 sinusoide (una cu frecvența de 50 Hz, defazajul 0 și amplitudinea 0.5 V, iar cealaltă cu frecvența de 230 Hz, defazajul π/3 și amplitudinea 0.2 V), folosind o frecvență de eșantionare de 8 kHz.

fs = 8000; //frecventa de esantionare

%se va genera în prealabil un vector de timp care reprezintă momentele de timp la care se va realiza practic eșantionarea:

t = 0:1/fs:1;

x = 0.5\*sin(2\*pi\*50\*t-0)+0.2\*sin(2\*pi\*230\*t-pi/3);//2 semnale sinusoidale

//1 cu ampl 0.5, frec 50, def 0; 2 cu amp 0.2, frecv 230, def, pi/3

stem(t(1:1024),x(1:1024)); //desenarea semnalului

xlabel('timpul');

ylabel('amplitudinea');

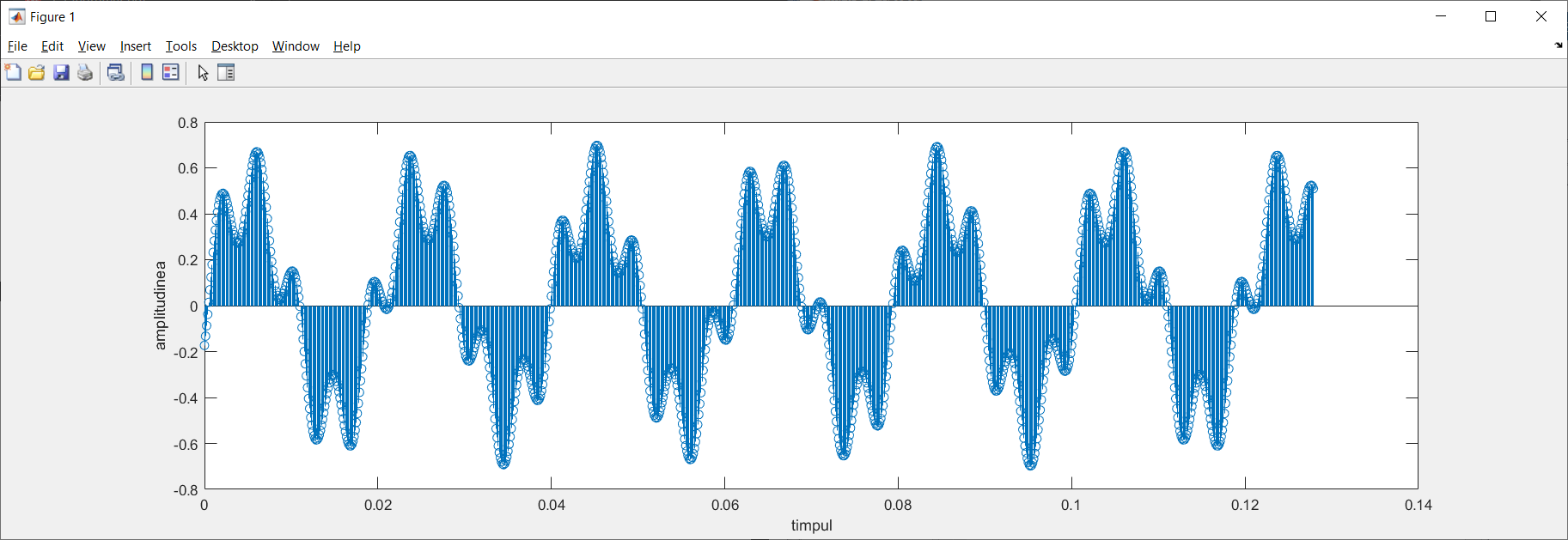
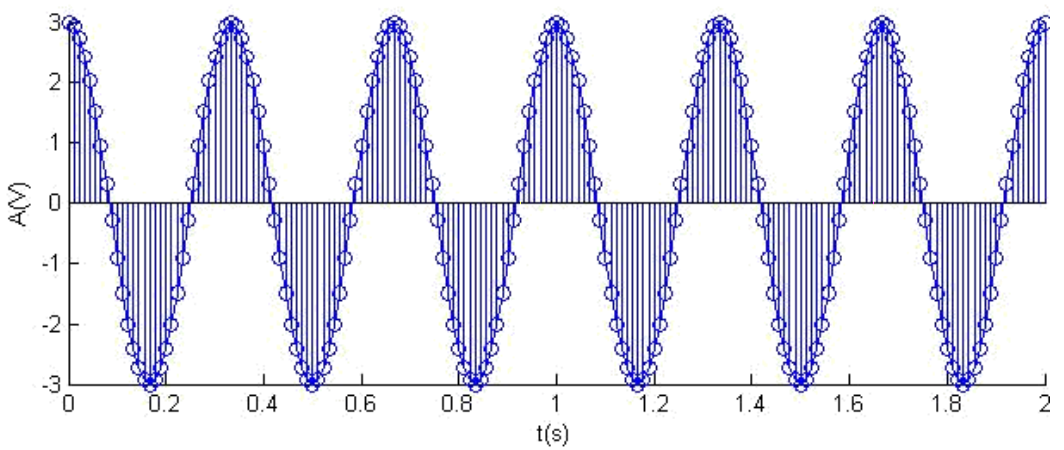


Figura 1 - Semnalul eșantionat ex 1.1

**Exercițiul 1.2**

Să se genereze în MATLAB semnalul din figura de mai jos. Indiciu: pornind de la figură, trebuie să se identifice toți parametrii sinusoidei (amplitudine, frecvență, frecvență de eșantionare, durată, fază inițială). Frecvența fs a fost aleasă astfel, încât să fie 30 de eșantioane într-o perioadă.



fs = 90; //frecventa de esantionare

t = 0:1/fs:2; //timpul

x = 3\*sin(2\*pi\*3\*t-3\*pi/2); //formula semnalului sinusoidal amp 3, frec 3, def 3pi/2

stem(t,x); //desenam semnalul

xlabel('timpul');

ylabel('amplitudinea');

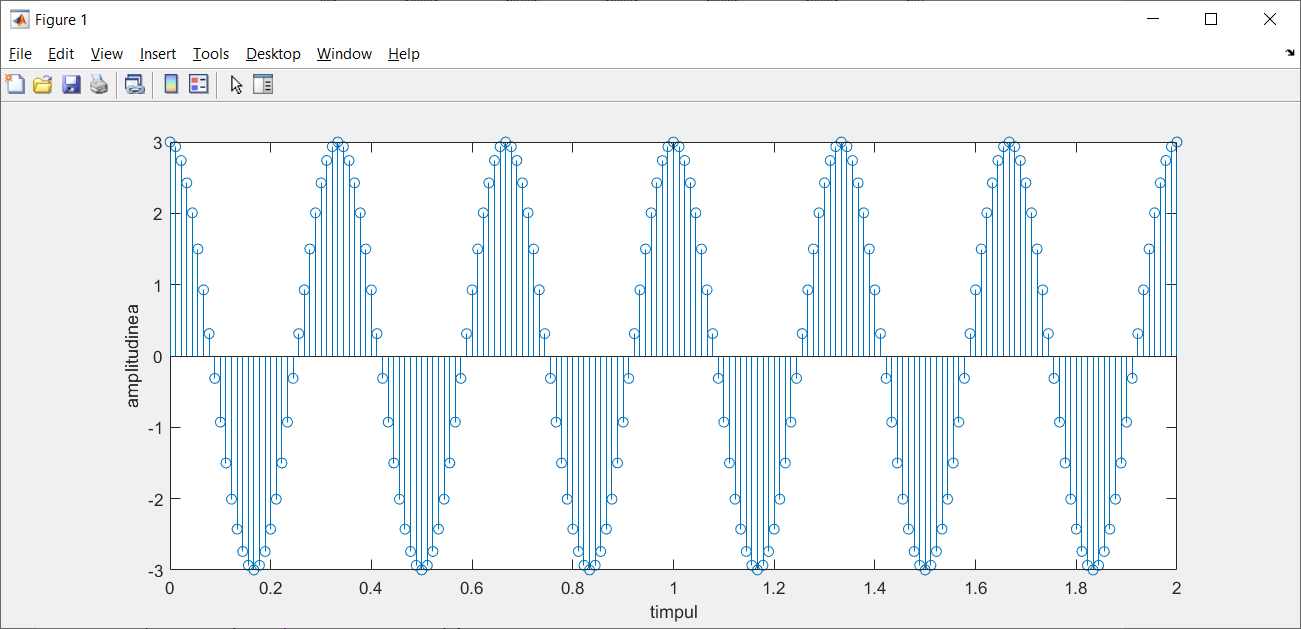


Figura 2 - Semnal eșantionat ex. 1.2

**Exercițiul 1.3**

Fie semnalul: 𝑥(𝑡) = 10 sin (200𝜋𝑡 + 𝜋/ 2 ) + 20 sin(100𝜋𝑡) − 40sin (300𝜋𝑡 – 𝜋/ 4 ).

Cerințe:

* care este frecvența minimă de eșantionare astfel încât să se respecte teorema eșantionării;
* alegând o frecvență de eșantionare de 10 ori mai mare decât cea determinată la punctul anterior, să se eșantioneze semnalul x(t) și să se reprezinte graphic;
* care este frecvența de repetiție a semnalului x(t)?

fs = 2\*13\*50; //perioada de 0.02, 13 puncte intermediare

t = 0:1/fs:1; //timpul

x = 10\*sin(2\*pi\*100\*t+pi/2)+20\*sin(2\*pi\*50\*t)-40\*sin(2\*pi\*150\*t-pi/4);

//1 semn: amplitudinea:10,frecventa:100, defazaj:pi/2

//2 semn: amplitudinea:20; frecventa:50 fara defazaj

//3 semn amplitudinea:40, frecventa:150, defazaj:pi/4

figure

stem(t(1:100),x(1:100)); //desenam semnalul

xlabel('timpul');

ylabel('amplitudinea');

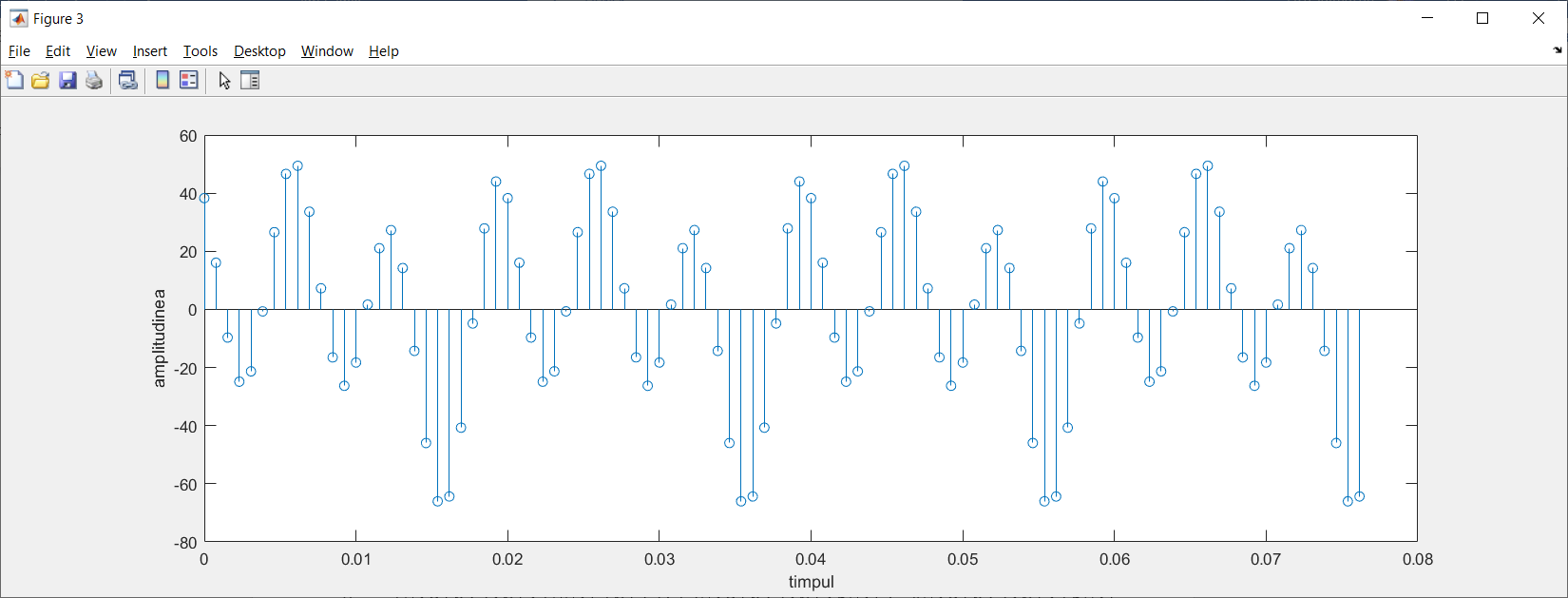


Figura 3 - Semnal eșantionat utilizând minim de frecvență de eșantionare

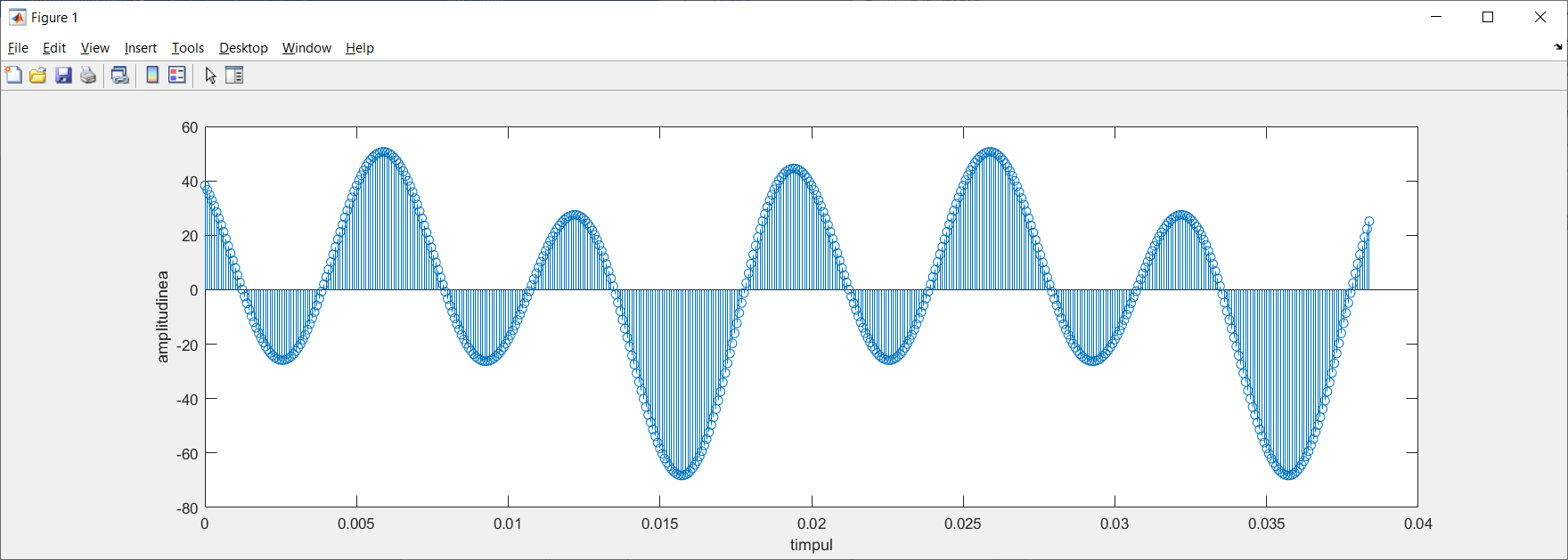


Figura 4 - Semnal eșantionat utilizând frecvență de eșantionare de 10 ori mai mare

**Exercițiul 1.4**

Fie semnalul sinusoidal 𝑥(𝑡) = 3sin(2𝜋 ∙ 5𝑡). Să se eșantioneze acest semnal cu fs1 = 4Hz și cu fs2 = 50Hz și să se reprezinte grafic. În care dintre cele două cazuri poate fi reconstituit semnalul x(t) din eșantioanele sale?

fs = 50; %frecventa de esantionare

t = 0:1/fs:1; %timpul

x = 3\*sin(2\*pi\*5\*t); %formula semnalului

figure

stem(t,x); %desenam figura

xlabel('timpul');

ylabel('amplitudinea');

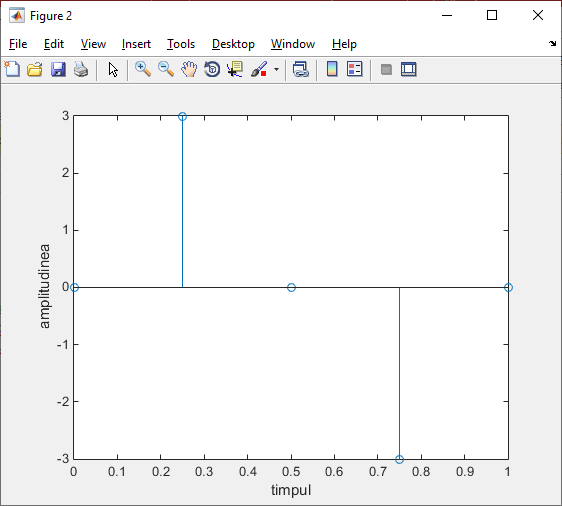


Figura 5 - Frecventa de eșantionare = 4 Hz

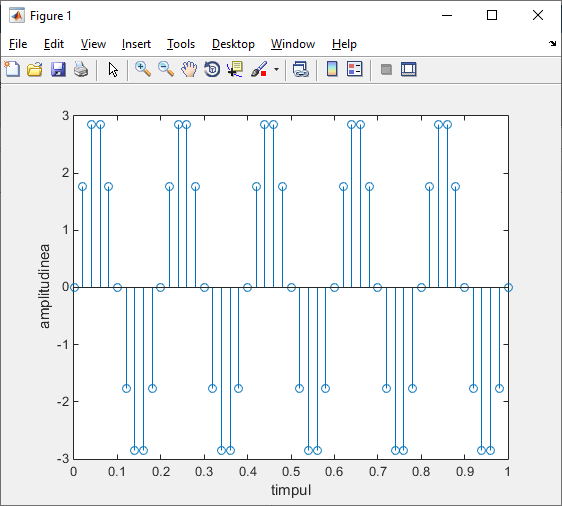
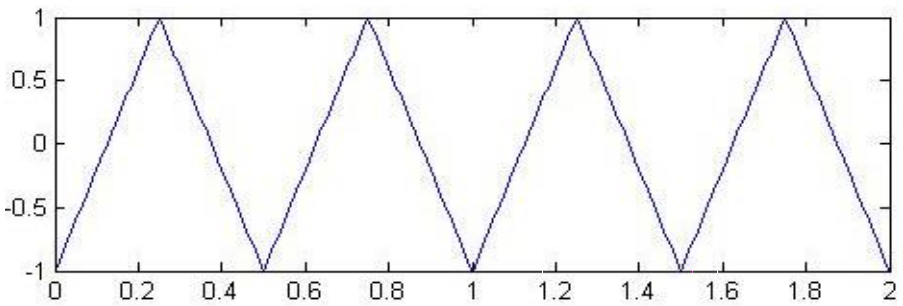


Figura 6 - Frecventa de eșantionare = 50 Hz

**Exercițiul 1.5**

Să se genereze și să se prezinte în MATLAB, în două ferestre separate, semnalul din figura următoare și semnalul discretizat. Indiciu: pornind de la figură, trebuie să se identifice toți parametrii semnalului triunghiular (amplitudine, frecvență, frecvență de eșantionare, durată). Frecvența fs se alege astfel încât să fie 30 de eșantioane într-o perioadă.



t=0:0.01:2; %timpul

x=1\*sawtooth(2\*pi\*2\*t,1/2); %formula semnalului triunghiular

subplot(2,1,1) % primul grafic din figura

plot(t,x); %desenam semnalul

grid; %trasarea liniilor

xlabel('timpul');

ylabel('amplitudinea');

fs=4\*15; %frecventa pentru al doilea semnal discretizat

t=0:1/fs:2; %timpul

x=1\*sawtooth(2\*pi\*2\*t,1/2); %formula semnalului triunghiular discretizat

subplot(2,1,2) %al doilea grafic din figura

stem(t,x); %desenarea figurii

grid;

xlabel('timpul');

ylabel('amplitudinea');

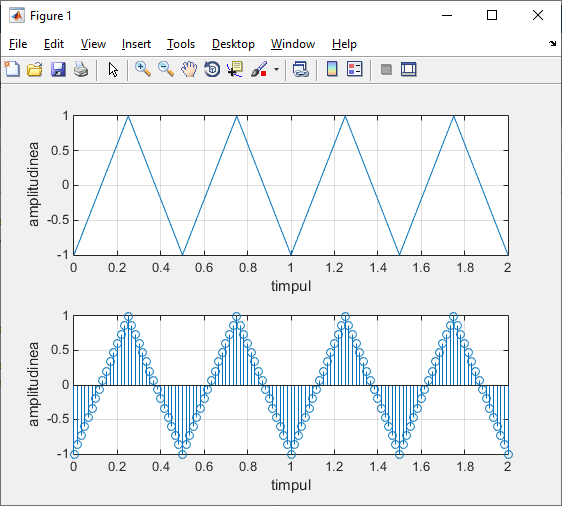


Figura 7 - Semnal original și cel eșantionat

**Exercițiul 2.1**

Un semnal cu durata de 2 minute este eșantionat cu frecvența de eșantionare de 4kHz. Câte eșantioane vor rezulta? Dacă fiecare eșantion este stocat pe 2 octeți, ce memorie vor ocupa toate eșantionale generate?

2 min \* 60 sec = 120 sec

120 sec \* 4000 Hz = 480000 eșantioane

480000 eșantioane \* 2 octeți = 960000 bytes = 937,5 Kb

**Exercițiul 2.2**

Să se cuantizeze semnalele 𝑥[𝑛] = 17/ 𝑛 și 𝑦[𝑛] = − 17/ 𝑛 , 𝑛 = 1 … 40, folosind metodele floor și round. Se cunosc nivelurile de cuantizare {0, ±1, ±2, … , ±17}. Să se reprezinte grafic semnalele originale 𝑥[𝑛] și 𝑦[𝑛], precum și semnalele obținute în urma cuantizării. Să se calculeze și să se reprezinte grafic zgomotul de cuantizare.

n=1:0.01:40; %definim intervalul cu pasul 0.01

x=17./n; %functia x

y=-17./n; % functia y

x1 = round(x); %cuantizarea lui x prin round

y1 = round(y); %cuantizarea lui y prin round

x2 = floor(x); %cuantizarea lui x prin floor

y2 = floor(y); %cuantizarea lui y prin floor

figure %generarea figurii

subplot(231); %primul grafic din figura

plot(n,x); %generarea semnalului x

title('semnal original 17/n');

grid

subplot(232); %al doilea grafic din figura

plot(n,x1); %generarea semnalului x1

title('round');

grid

subplot(233); %al treilea grafic din figura

plot(n,x2); %generarea semnalului x2

title('floor');

grid

subplot(234); %al patrulea grafic din fig

plot(n,y); %generarea semnalului y

title('semnal original -17/n');

grid

subplot(235); %al cincilea grafic din figura

plot(n,y1); %generarea semnaluluiy1

title('round');

grid

subplot(236); %al saselea grafic din fig

plot(n,y2); %generarea semnalului y 2

title('floor');

grid

x1eror=x1-x; %calcularea zgomotului pt semnalul x prin round

x2eror=x2-x; %calcularea zgomotului pt semnalul x prin floor

y1eror=y1-y; %calcularea zgomotului pt semnalul y prin round

y2eror=y2-y; %calcularea zgomotului pt semnalul y prin floor

figure %generarea figurii

subplot(221); %primul grafic din figura

plot(n,x1eror); %generarea graficului ptu zgomotul lui x prin round

title('round');

grid

subplot(222);%al doilea grafic din figura

plot(n,x2eror); %generarea graficului ptu zgomotul lui x prin floor

title('floor');

grid

subplot(223);%al treilea grafic din figura

plot(n,y1eror); %generarea graficului ptu zgomotul lui y prin round

title('round');

grid

subplot(224);% al patrulea grafic din figura

plot(n,y2eror); %generarea graficului ptu zgomotul lui y prin floor

title('floor');

grid

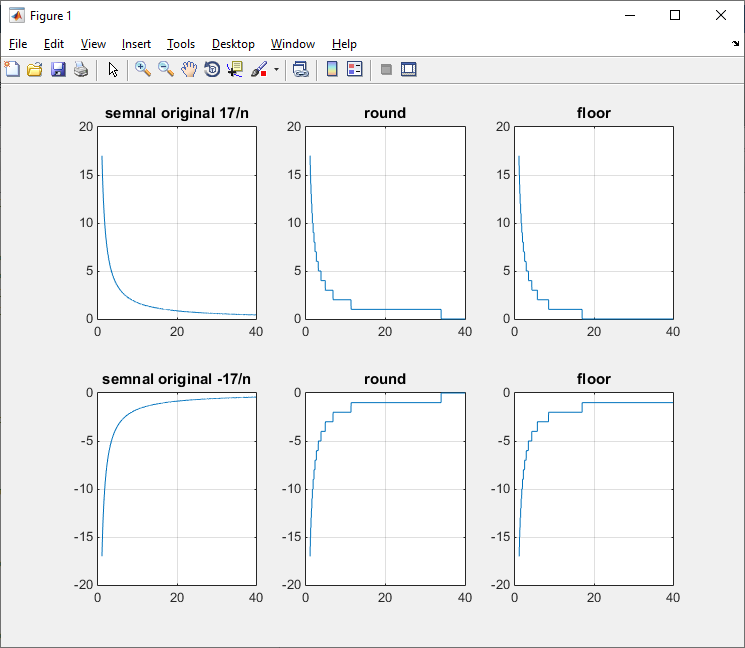


Figura 8 - Semnalele cuantizate

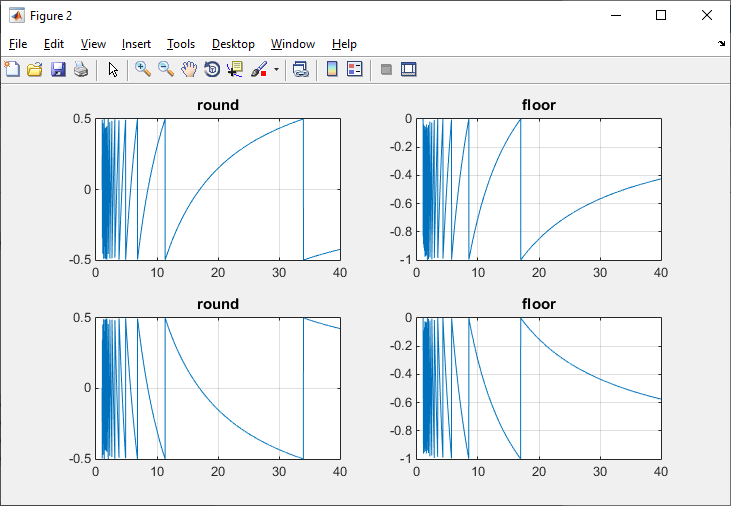


Figura 9 - Zgomotul de cuantizare

**Exercițiul 2.3**

Să se genereze 15 eșantioane de zgomot alb gaussian cu media zero și dispersia 0,2. Indiciu: pentru a genera zgomot alb gaussian se poate folosi funcția randn: z = 0.2\*randn(1,15). Să se cuantizeze semnalul z folosind un cuantizor uniform cu nivelurile {0, ± 1/ 4 , ± 2 /4 , ± 3/ 4 , ±1}.

nr=15; %nr de esantioane

z = 0.2\*randn(1,nr); %functia de calcul

nivele=-1:0.25:1; %intervalul nivelelor

for j = 1 : nr %verificam daca j e de la 1 la nr de esantioane (15)

for i = 1 : (length(nivele)-1) %verificam daca i e de la 1 pana la lung niv - 1

if z(j) >= nivele(i) && z(j) <= nivele(i+1)

if z(j)<= (nivele(i) + nivele(i+1))/2

eta(j) = nivele(i); %nivelul este cel curent

else

eta(j) = nivele(i+1); %nivelul e urmatorul

end

end

end

end

figure %desenam figura

subplot(211);%primul grafic din figura

plot(1:nr, z);%desenarea semn original

title('semnal original');

subplot(212);%al doilea grafic din figura

plot(1:nr, eta); %desenarea semnalului cuantizat

title('semnal cuantizat');

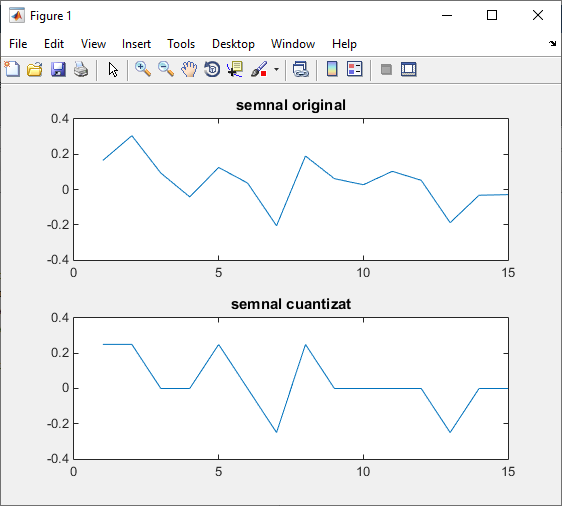


Figura 10 - Cuantizarea semnalului

**Exercițiul 2.4**

Cuantizați semnalul din Exercițiul 4.1.1 pe 8 respectiv, 16 biți. Reprezentați semnalul cuantizat alături de zgomotul de cuantizare.

fs = 8000;%frecventa de esantionare

t = 0:1/fs:1;%timpul

x = 0.5\*sin(2\*pi\*50\*t-0)+0.2\*sin(2\*pi\*230\*t-pi/3);%formula sinus. pct precedent

b = 16; %nr de biti

L = b^2;

Delta=(max(x)-min(x))/L;

nivele(1) = min(x);

nivele(L) = max(x);

for i=2:L %verificam daca i e de la 2 pana la L

nivele(i)=min(x)+(i-1)\*Delta; %cat e corecta conditia, niv se calculeaza

end

for j = 1 : length(x) %daca j e de la 1 pana la lung lui x

for i = 1 : (length(nivele)-1) %daca i e de la lung niv - 1

if x(j) >= nivele(i) && x(j) <= nivele(i+1)

if x(j)<= (nivele(i) + nivele(i+1))/2

eta(j) = nivele(i); %daca da-> nivelul este cel curent

else

eta(j) = nivele(i+1); %daca nu -> nivelul este cel urmator

end

end

end

end

figure

subplot(311); %primul grafic din figura

plot(t(1:256), x(1:256)); %desenarea semnalului original

title('semnal original');

subplot(312); %al doilea grafic din figura

plot(t(1:256), eta(1:256)); %desenarea semnalului cuantizat

title('semnal cuantizat');

subplot(313); %al treilea grafic din igura

plot(t(1:256), eta(1:256)-x(1:256)); %desenarea zgomotului semnalului cuantizat

title('zgomotul semnalului cuantizat');

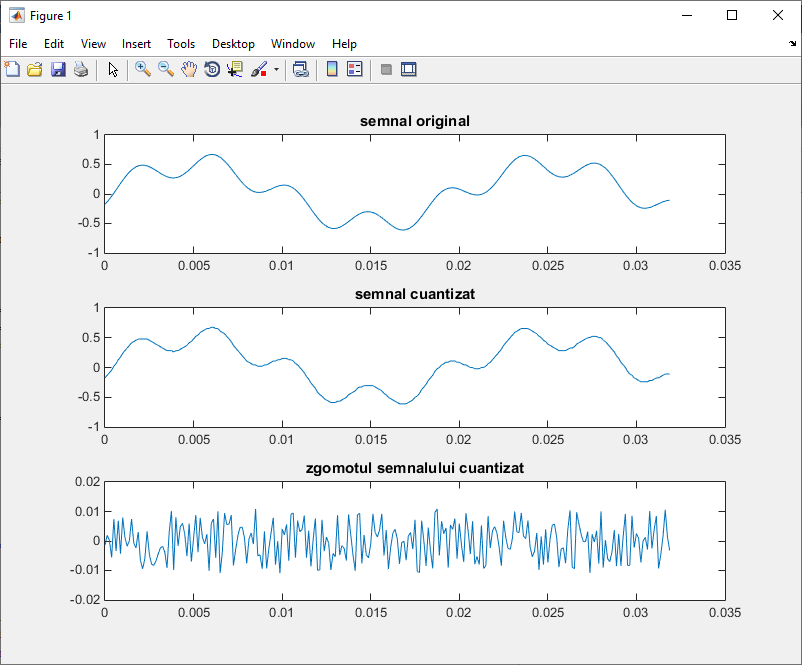


Figura 11 - Cuantizare pe 8 biți

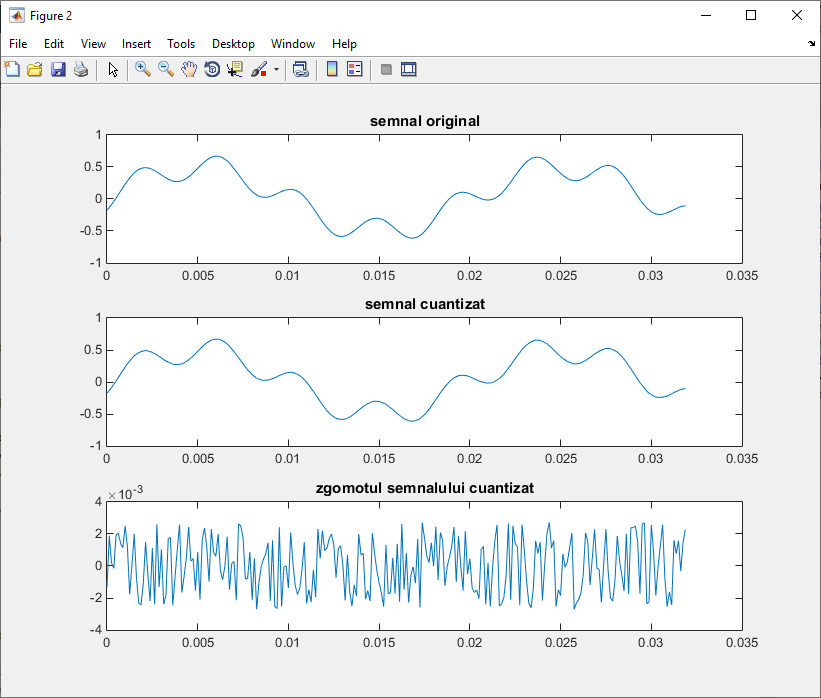


Figura 12 - Cuantizare pe 16 biți

**Exercițiul 3.1**

Executarea calculelor cu valorile recomandate în antetul script-ului, reprezentarea semnalelor în formă originară şi după reconstrucție, precum şi a diferenţelor dintre ele, în spaţii grafice diferite sau în acelaşi spaţiu.

% optiune de reprezentare (rep=0) grafic unic, rep>0) grafice separate

tmax=10; % timpul maxim de reprezentare

fi=pi/2; % faza componentei secundare

f1=1; % frecventa componentei primare

f2=2; % frecventa componentei secundare

% pasul utilizat la reprezentarea grafica

pasmic=0.001;

fes=4.1; % frecventa de esantionare

rep=1;

tes=1/fes; % perioada esantioanelor

a1=2;

a2=1;

t=0:pasmic:tmax;

% pregatirea graficului 1 (semnal original)

y=a1\*cos(2\*pi\*f1\*t)+a2\*cos(2\*pi\*f2\*t-fi);

t1=0:tes:tmax;

n=round(tmax/tes)+1;

% pregatirea graficului 2 (esantionarea)

y1=a1\*cos(2\*pi\*f1\*t1)+a2\*cos(2\*pi\*f2\*t1-fi);

% pregatirea graficului 3 (reconstituire)

y2=y1(1)\*sinc(t/tes);

for k=1:(n-1)

y2=y2+y1(k+1)\*sinc(t/tes-k);

end

if rep>0

subplot(4,1,1)

plot(t,y) % trasarea graficului 1 (semnal)

hold on

ylabel('Original')

title('ESANTIONAREA SEMNALELOR')

plot([0 tmax],[0 0])

for i=1:n

subplot(4,1,2)

% trasarea graficului 2 (esantionarea)

plot([t1(i) t1(i)],[0 y1(i)],'b:')

hold on

ylabel('Esantionat')

plot([0 tmax],[0 0])

end

subplot(4,1,3)

% trasarea graficului 3 (reconstituire)

plot(t,y2,'k')

hold on

plot([0 tmax],[0 0])

for i=1:n

plot([t1(i) t1(i)],[0 y1(i)],'b:')

end

ylabel('Reconstituire')

subplot(4,1,4)

% trasarea graficului 4 (diferente)

plot(t,y-y2,'r')

hold on

plot([0 tmax],[0 0])

ylabel('Diferente')

xlabel('Timp(s)')

else

plot(t,y) % trasarea graficului 1 (semnal)

hold on

plot([0 tmax],[0 0])

for i=1:n

% trasarea graficului 3 (reconstituire)

plot([t1(i) t1(i)],[0 y1(i)],'b:')

end

plot(t,y2,'k')

% trasarea graficului 4 (diferente)

plot(t,y-y2,'r')

plot([0 tmax],[0 0])

title('ESANTIONAREA SEMNALELOR')

ylabel('Semnal(albastru)/Reconstituire(negru)/ Diferente(rosu)')

xlabel('Timp(s)')

end

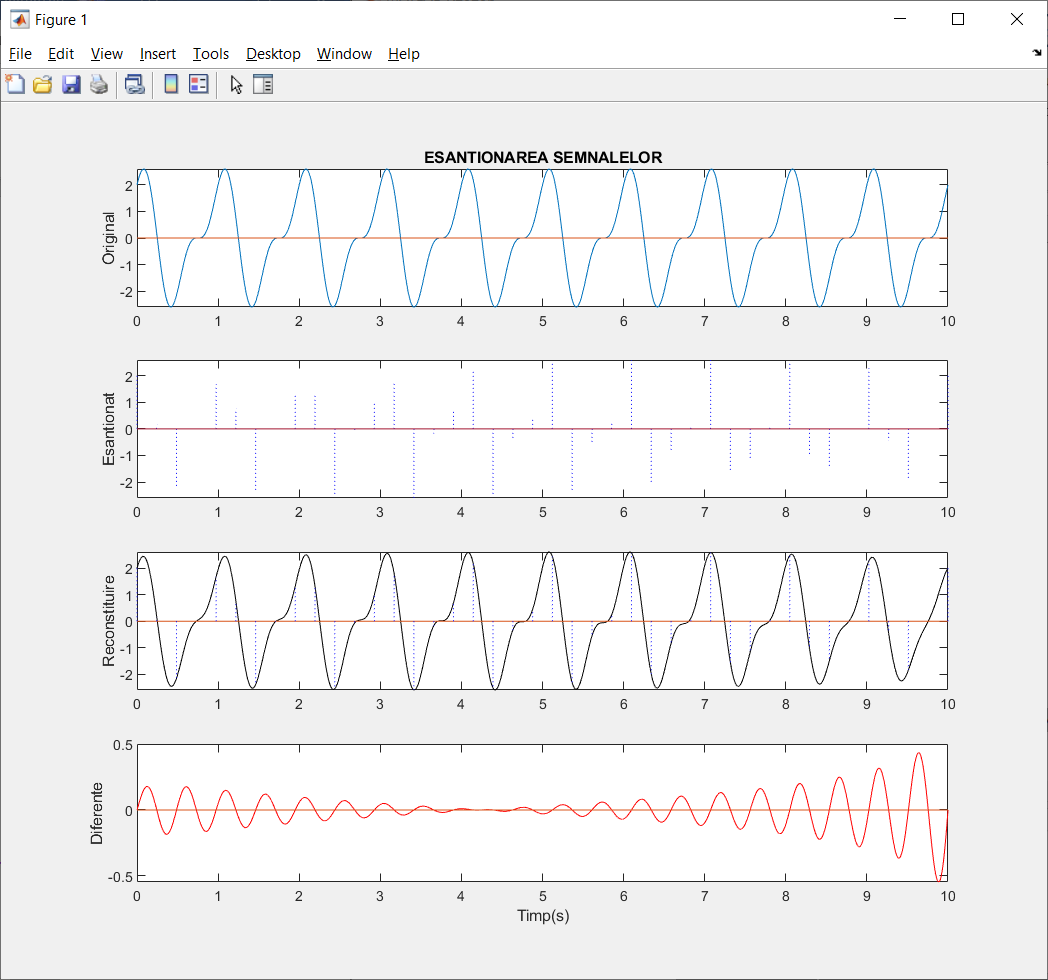


Figura 13 - Scriptul original

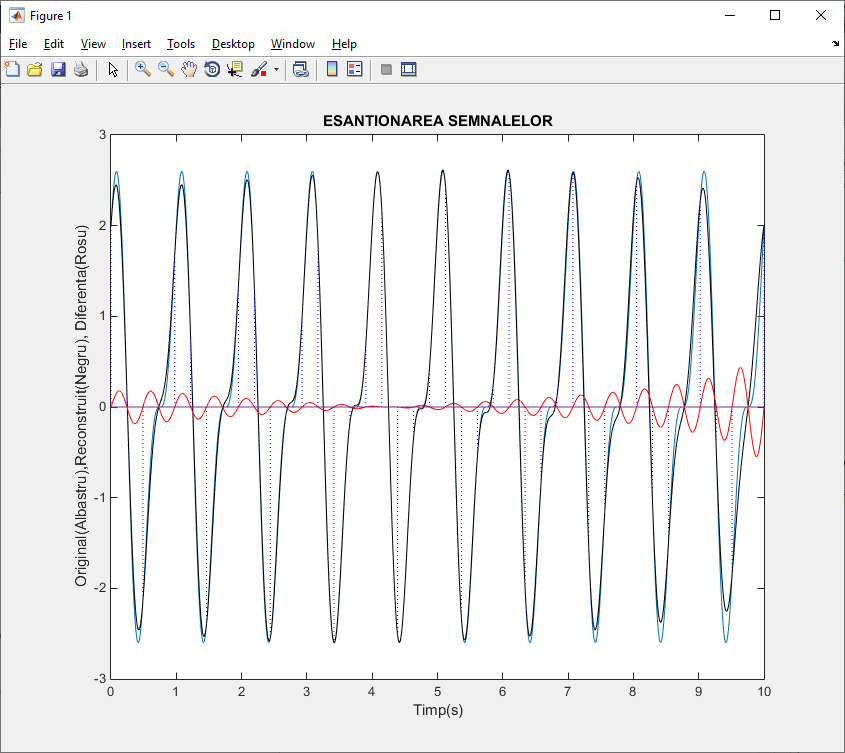


Figura 14 - Scriptul original în același sistem de coordonate

**Exercițiul 3.2** Executarea de calcule cu valorile diferite pentru frecvenţele f1 şi f2 , eventual pentru amplitudini diferite ale sinusoidelor componente, utilizând pentru fiecare caz trei valori ale frecvenţei de eşantionare: mai mică, egală şi mai mare (de 10 ori) ca frecvenţa Nyquist.

**Exercițiul 3.3** Aprecierea vizuală şi/sau cantitativ a modificărilor semnalului reconstituit, calitatea reconstrucției semnalului în zona de început şi de final a intervalului de timp observat.

**Exercițiul 3.4** Studierea şi a altor semnale de bandă limitată, recomandate de profesor.

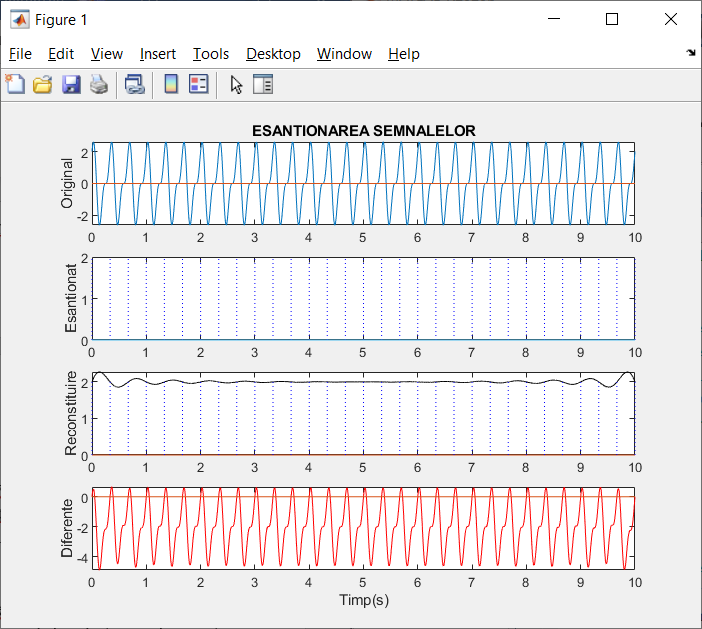


Figura 15 - f1=3, f2=6, fes=3

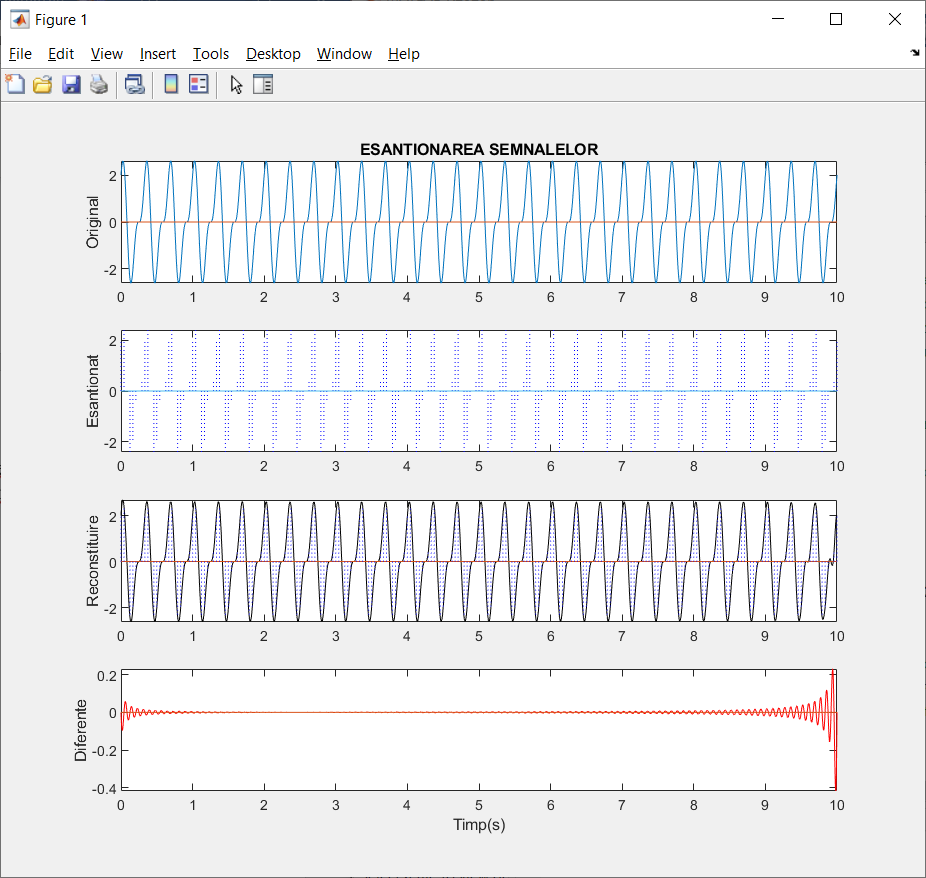


Figura 16 - f1=3, f2=6, fes=2\*12

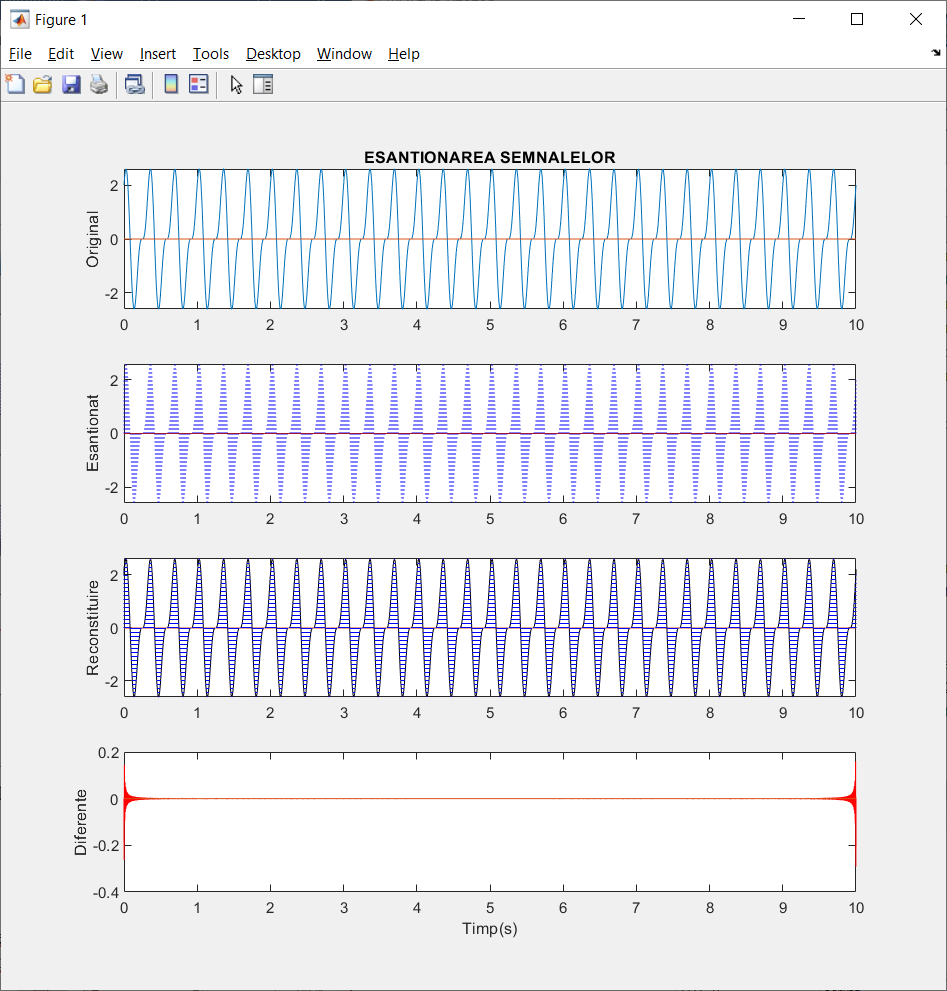


Figura 17 - f1=3, f2=6, fes=2\*12\*10

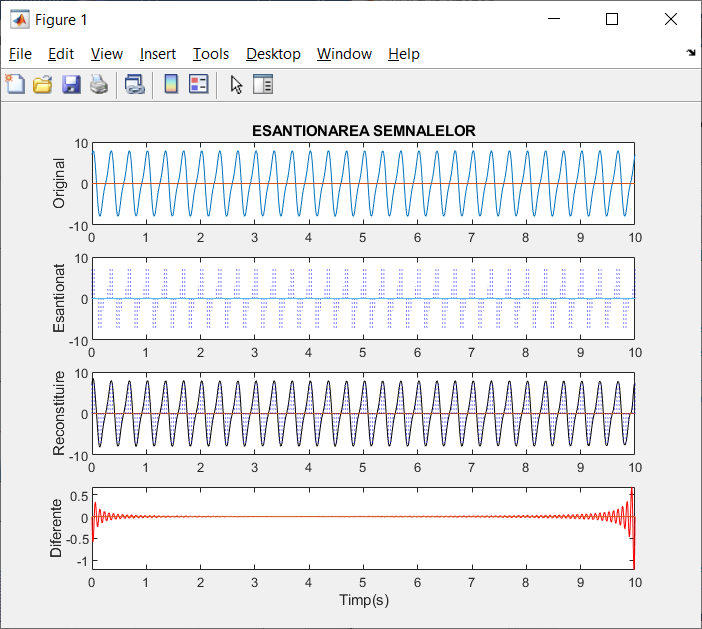


Figura 18 - f1=3, f2=6, fes=2\*12, a1=7, a2=2

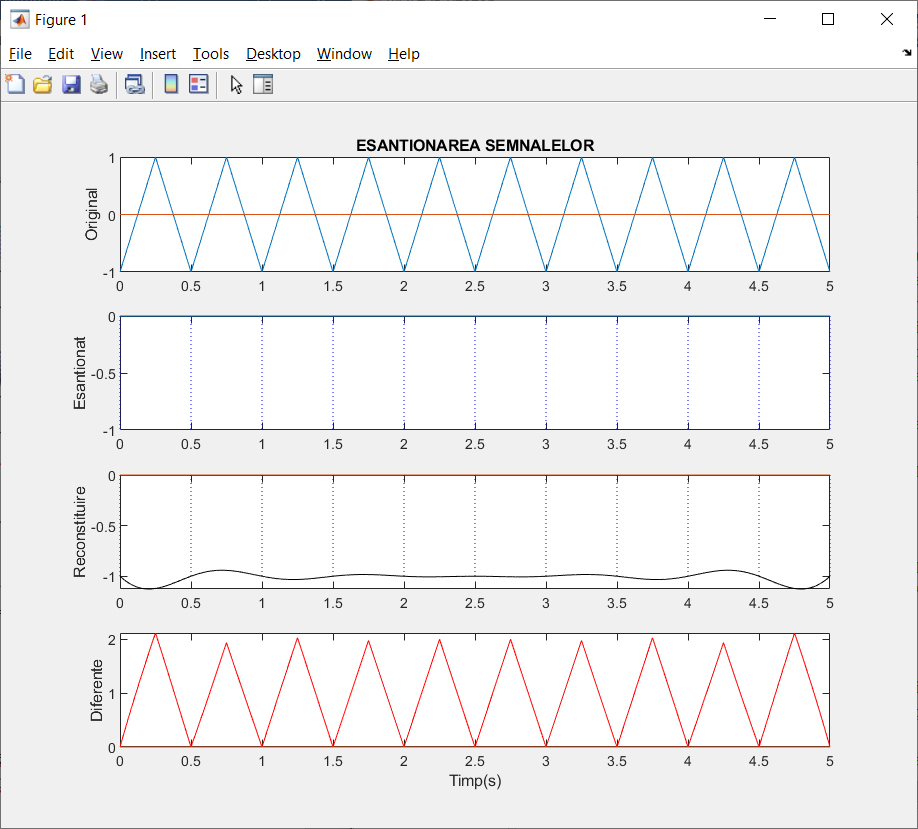


Figura 19 - Sawtooth(2\*pi\*2\*t, 1/2), fes=2

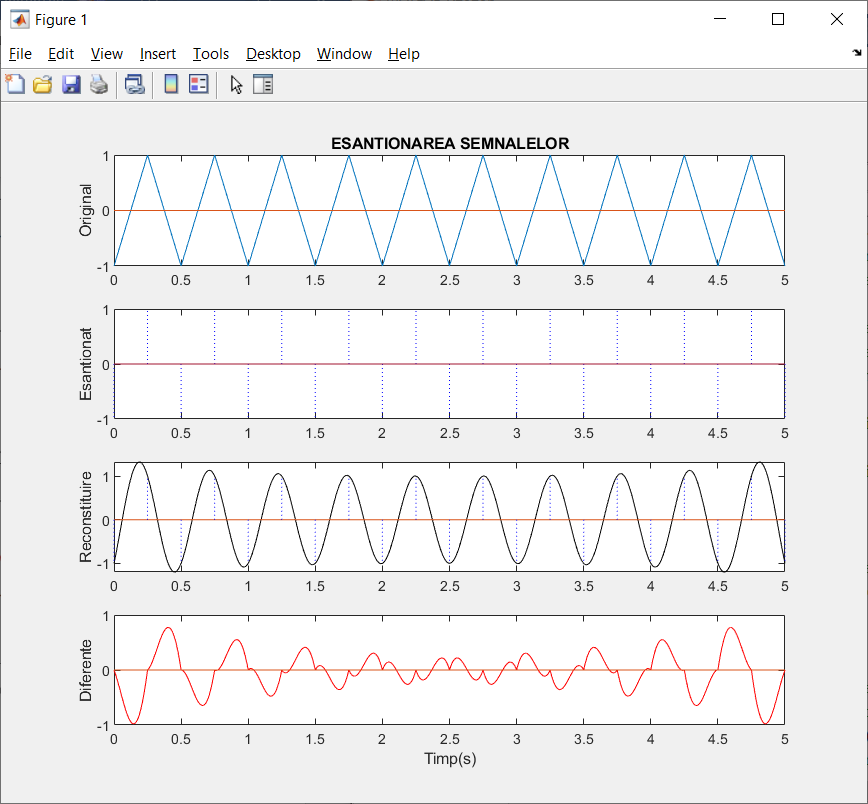


Figura 20 - Sawtooth(2\*pi\*2\*t, 1/2), fes=2\*2

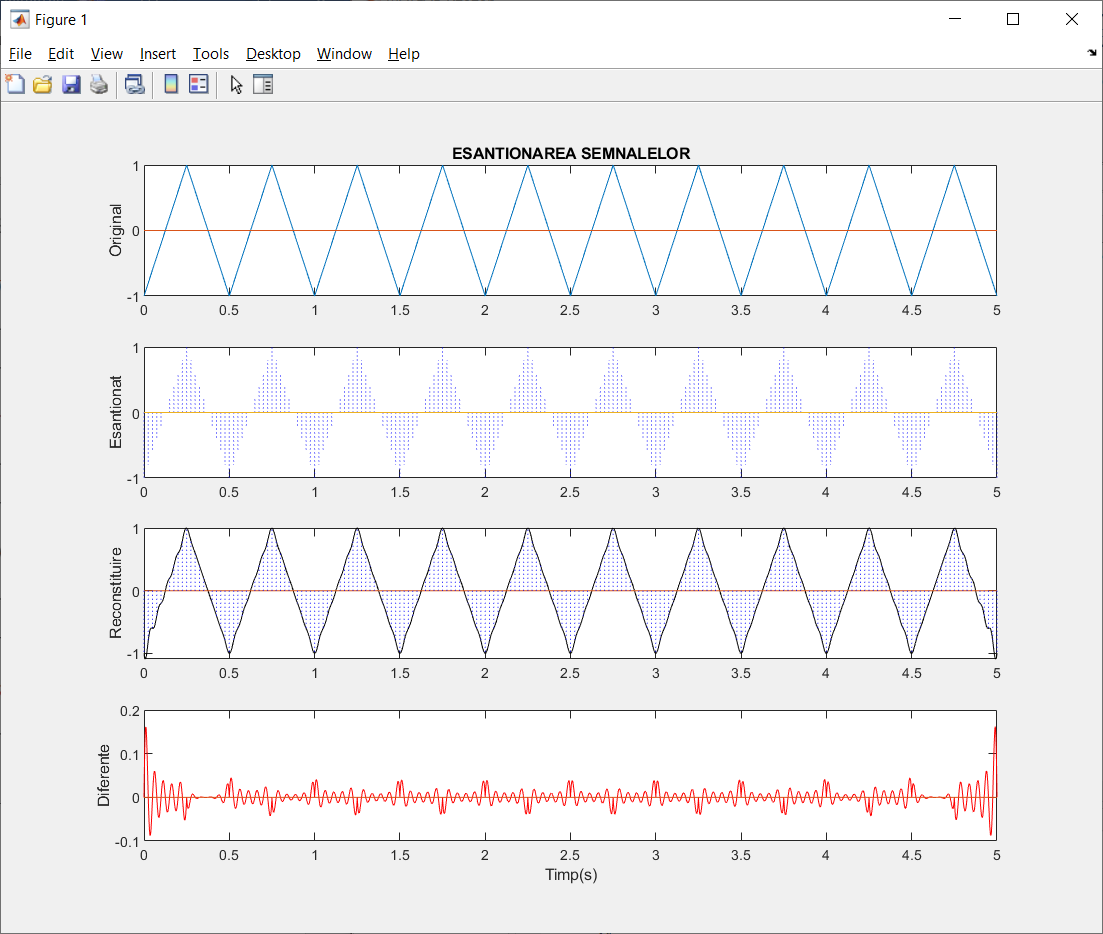


Figura 21 - Sawtooth(2\*pi\*2\*t, 1/2), fes=2\*2\*10