UNIVERSITATEA "POLITEHNICA" DIN TIMIŞOARA
FACULTATEA DE MECANICĂ

TEZĂ DE DOCTORAT

INFLUENȚA INTRODUCERII VIBROŞOCURILOR ASUPRA CALITĂȚII LA PRELUCRĂRI PRIN AŞCHIERE

ANEXE

CONDUCĂTOR ŞTIINȚIFIC
Prof. univ.dr. ing. LIVIU BRÎNDEU

DOCTORAND
Şef lucr. ing. IOAN MOGA

2005

Cuprins

Anexa 1.

Program sursă „tra_vibr" pentru introducerea relației (5.28) în cod MATLAB

```
%Traiectorie prelucrare cu vibratii
grid on;
axis([-10 10 0 200]);
hold on;
s = input('Avans [mm/rot]       :');
n = input('Turatie [rot/min]    :');
a = input('Amplitudine [mm]     :');
f = input('Frecventa [Hz]       :');
d = input('Diametru piesa [mm]  :');
nr_cicluri = input('Numar de perioade :');
nr_rot = nr_cicluri*n/(60*f);

lung_desf = pi*d;
delta_y = 1;
nr_val_y = floor(lung_desf/delta_y);
delta_t = (delta_y*(60/n))/(pi*d);
nr_val = floor(nr_val_y*nr_rot);

x = zeros(1,nr_val);
y = zeros(1,nr_val);
t = 0;
for j = 1:nr_rot
   for i = 1:nr_val_y
      ind = (j-1)*nr_val_y+i;
      y(ind) = i*delta_y;
      t = t + delta_t;
      x(ind) = (s*n/60*t+a*sin(2*pi*f*t));
      if i>2
         if abs(y(ind)-y(ind-1)) == delta_y
            line([x(ind-1) x(ind)],[y(ind-1) y(ind)]);
         end
      end
   end
end
```

Anexa 2.

Codurile sursă ale progamului „tra_vibr" și „rug_vibr5"

```
%Rugozitate prelucrare cu vibratii
s = input('Avans [mm/rot]            :');
n = input('Turatie [rot/min]         :');
a = input('Amplitudine [mm]          :');
f = input('Frecventa [Hz]            :');
d = input('Diametru piesa [mm]       :');
kappa = input('Unghiul de atac [grade]    :');
kappa_rad = (90-kappa)*pi/180;
nr_cicluri = input('Numar de perioade          :');

% Valori constante calculate

nr_rot = nr_cicluri*n/(60*f);
lung_desf = pi*d;
delta_y = 1;
nr_val_y = floor(lung_desf/delta_y);
delta_t = (delta_y*(60/n))/(pi*d);
nr_val = floor(nr_val_y*nr_rot);


% Matric pentru x si y

x = zeros(1,nr_val);
y = zeros(1,nr_val);

% Calcul traiectorie cutit

t = 0;
for j = 1:nr_rot
   for i = 1:nr_val_y
      y((j-1)*nr_val_y+i) = i*delta_y;
      t = t + delta_t;
      x((j-1)*nr_val_y+i) = (s*n/60*t+a*sin(2*pi*f*t));
      ind = (j-1)*nr_val_y+i;
      if i>2
         if abs(y(ind)-y(ind-1)) == delta_y
            line([x(ind-1) x(ind)],[y(ind-1) y(ind)]);
         end
      end
   end
end


% Sorteaza

date1 = [x' y'];
date2 = floor(date1*10000000000);
[date, indecsi] = sortrows(date2,[2 1]);
```

```
date = date/10000000000;
x = zeros(1,nr_val);
y = zeros(1,nr_val);
x = date(1:nr_val,1)';
y = date(1:nr_val,2)';
```

%Ciclul de calcul a valorilor rugozitatii

```
nr_pct = nr_val;
supr = zeros(nr_pct,3);
k = 0;
for i = 1:nr_val-1
   if y(i) == y(i+1)
      delta_x = x(i+1)-x(i);
      xm = x(i) + delta_x/2;
      ym= y(i);
      zm = abs(delta_x)*tan(kappa_rad)/2;
      k = k+1;
      supr(k,1) = xm;
      supr(k,2) = ym;
      supr(k,3) = zm;
      k = k+1;
      supr(k,1) = x(i+1);
      supr(k,2) = y(i+1);
      supr(k,3) = 0;
   end
   % Pentru o simulare in timp se activeaza pauza
   %pause(1);
end
[nezerox,nezeroy,nezeroz] = find(supr);
limita_mesh_min = min(nezerox);
limita_mesh_max = max(nezerox);

x_inter = supr(limita_mesh_min:limita_mesh_max,1);
y_inter = supr(limita_mesh_min:limita_mesh_max,2);
z_inter = supr(limita_mesh_min:limita_mesh_max,3);
```

%Generare retea 3D

```
xlin = linspace(min(x_inter),max(x_inter),50);
ylin = linspace(min(y_inter),max(y_inter),50);
[X,Y] = meshgrid(xlin,ylin);

%Interpolare 3D

Z = griddata(x_inter,y_inter,z_inter,X,Y,'linear');

%Afisare grafica 3D

colormap('pink');
figure; meshz(X,Y,Z)
```

Anexa 3.

Program sursă pentru prelucrare fără vibrații și de gestionare a programelor rug_vibr5 și rug_vibr6

```
%Rugozitate prelucrare fara vibratii
s = input('Avans [mm/rot]              :');
n = input('Turatie [rot/min]           :');
d = input('Diametru piesa [mm]         :');
kappa = input('Unghiul de atac [grade]     :');
kappa_rad = (90-kappa)*pi/180;
nr_rot = input('Numar de rotatii           :');


% Valori constante calculate

lung_desf = pi*d;
delta_y = 1;
nr_val_y = floor(lung_desf/delta_y);
delta_t = (delta_y*(60/n))/(pi*d);
nr_val = floor(nr_val_y*nr_rot);


% Matrice pentru x si y

x = zeros(1,nr_val);
y = zeros(1,nr_val);
```

```
% Calcul traiectorie cutit

t = 0;
for j = 1:nr_rot
   for i = 1:nr_val_y
      y((j-1)*nr_val_y+i) = i*delta_y;
      t = t + delta_t;
      x((j-1)*nr_val_y+i) = (s*n/60*t);
      ind = (j-1)*nr_val_y+i;
      if i>2
         if abs(y(ind)-y(ind-1)) == delta_y
            line([x(ind-1) x(ind)],[y(ind-1) y(ind)]);
         end
      end
   end
end


% Sorteaza

date1 = [x' y'];
date2 = floor(date1*10000000000);
[date, indecsi] = sortrows(date2,[2 1]);
date = date/10000000000;
x = zeros(1,nr_val);
```

```
y = zeros(1,nr_val);
x = date(1:nr_val,1)';
y = date(1:nr_val,2)';
```

%Ciclul de calcul a valorilor rugozitatii

```
nr_pct = nr_val;
supr = zeros(nr_pct,3);
k = 0;
for i = 1:nr_val-1
   if y(i) == y(i+1)
      delta_x = x(i+1)-x(i);
      xm = x(i) + delta_x/2;
      ym= y(i);
      zm = abs(delta_x)*tan(kappa_rad)/2;
      k = k+1;
      supr(k,1) = xm;
      supr(k,2) = ym;
      supr(k,3) = zm;
      k = k+1;
      supr(k,1) = x(i+1);
      supr(k,2) = y(i+1);
      supr(k,3) = 0;
   end
   % Pentru o simulare in timp se activeaza pauza
   %pause(1);
end
[nezerox,nezeroy,nezeroz] = find(supr);
limita_mesh_min = min(nezerox);
limita_mesh_max = max(nezerox);

x_inter = supr(limita_mesh_min:limita_mesh_max,1);
y_inter = supr(limita_mesh_min:limita_mesh_max,2);
z_inter = supr(limita_mesh_min:limita_mesh_max,3);
```

%Generare retea 3D

```
xlin = linspace(min(x_inter),max(x_inter),400);
ylin = linspace(min(y_inter),max(y_inter),400);
[X,Y] = meshgrid(xlin,ylin);
```

%Interpolare 3D

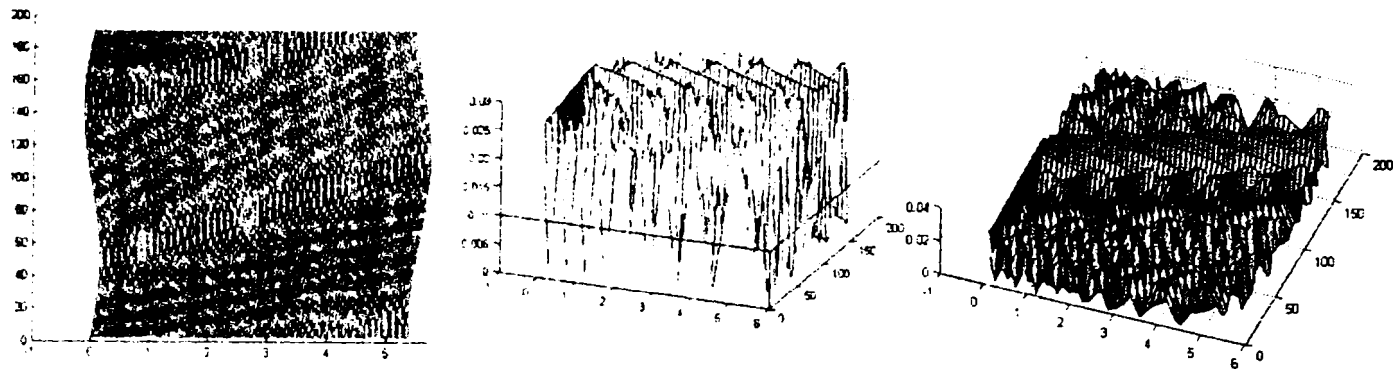
```
Z = griddata(x_inter,y_inter,z_inter,X,Y,'linear');
```

%Afisare grafica 3D

```
colormap('pink');
figure; meshz(X,Y,Z)
%axis('equal');
```

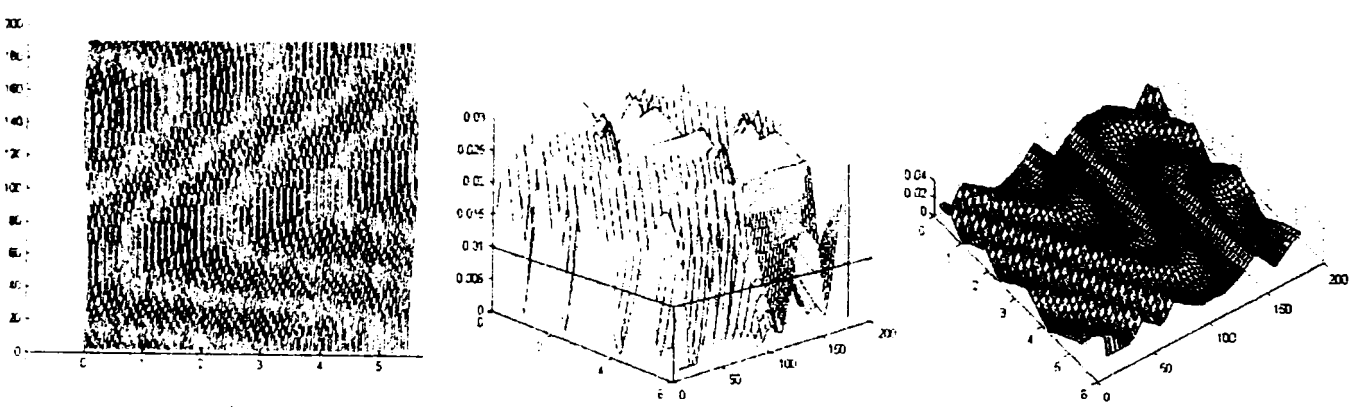
Anexa 4.

Simularea așezării de k_λ perioade pe circumferința piesei
k_λ +i=60*f/n; i=0; k_λ =60*f/n, rezultă f=n* k_λ /60



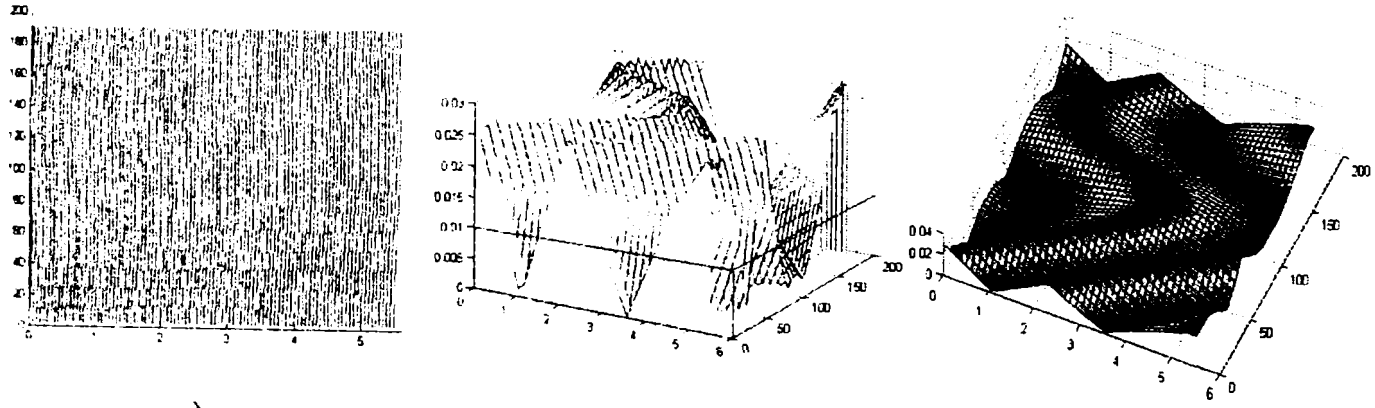
a) b) c)

Fig.A.4.1.1. Regimul mișcării vibratorii: f=10,5 Hz; A_x=0.16 mm; k_λ =1



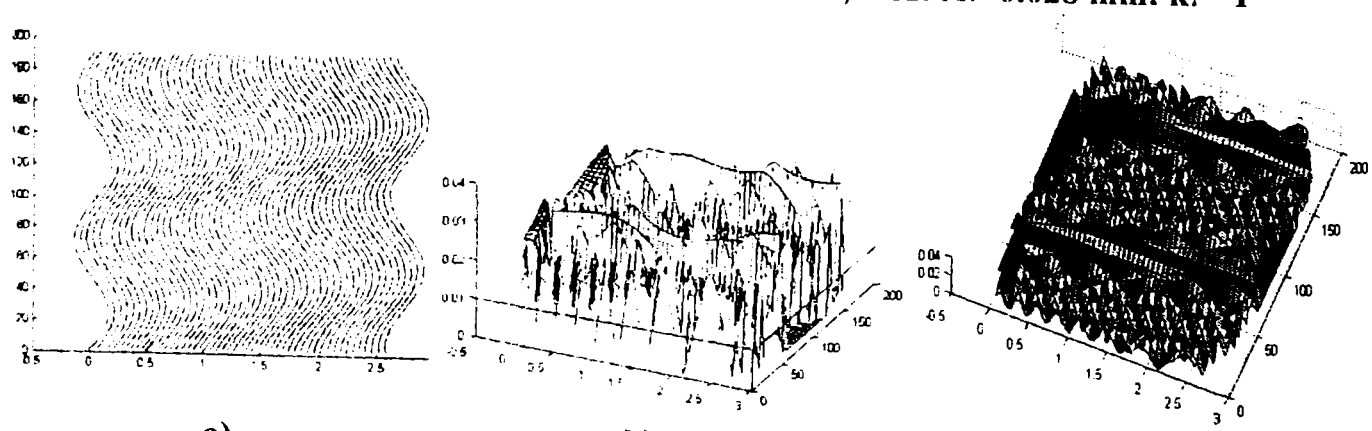
a) b) c)

Fig.A.4.1.2. Regimul mișcării vibratorii: f=10,5 Hz; A_x=0.056 mm; k_λ =1



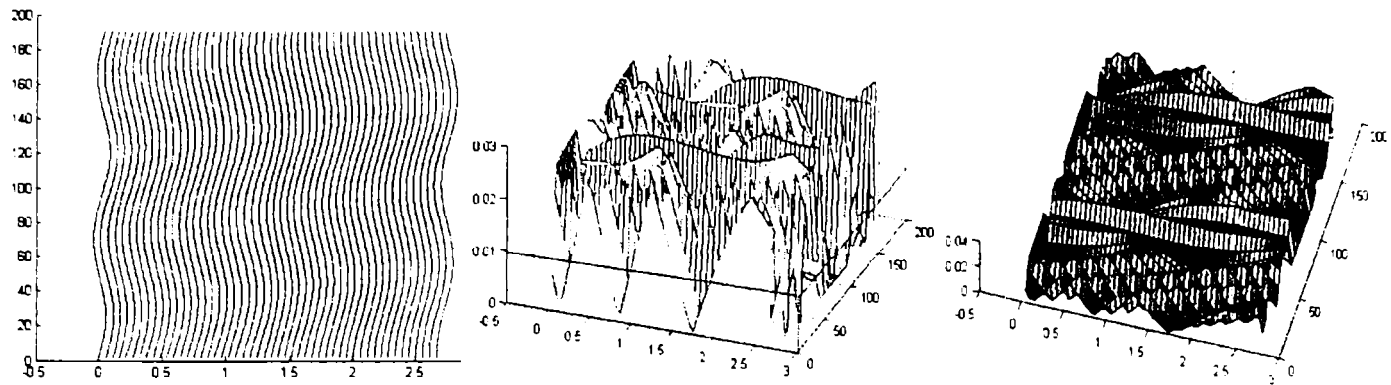
a) b) c)

Fig.A.4.1.3. Regimul mișcării vibratorii: f=10,5 Hz: A_x=0.028 mm: k_λ =1



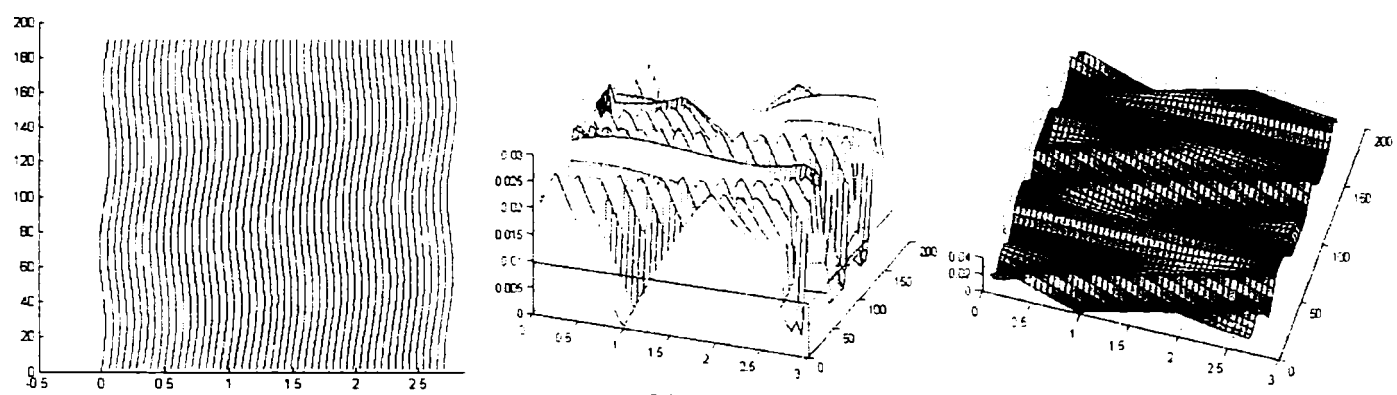
a) b) c)

Fig.A.4.2.1. Regimul mișcării vibratorii: f=21 Hz; A_x=0.16 mm; k_λ =2



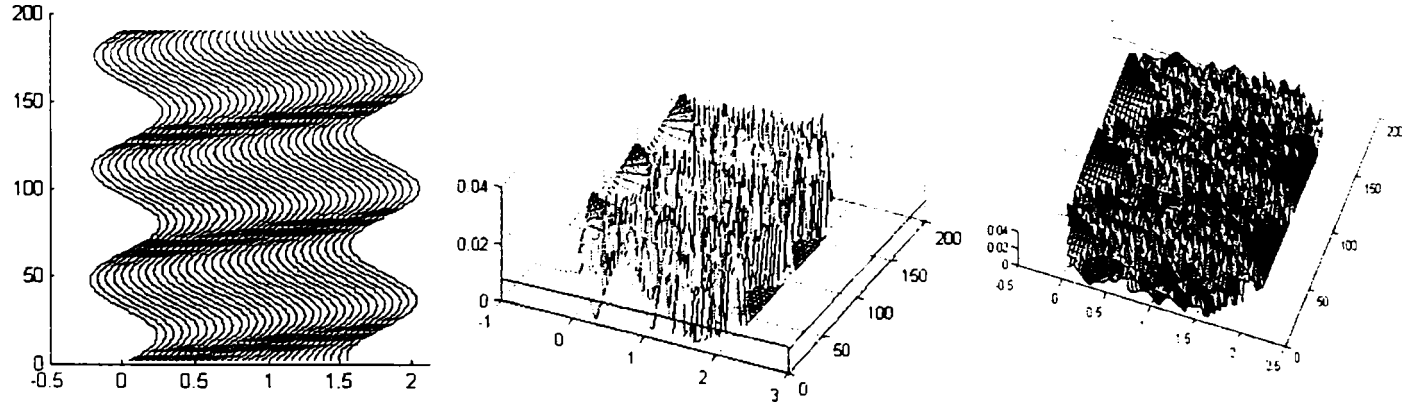
a) b) c)

Fig.A.4.2.2. Regimul mişcării vibratorii: f=21 Hz; A_x=0.056 mm; k_λ =2



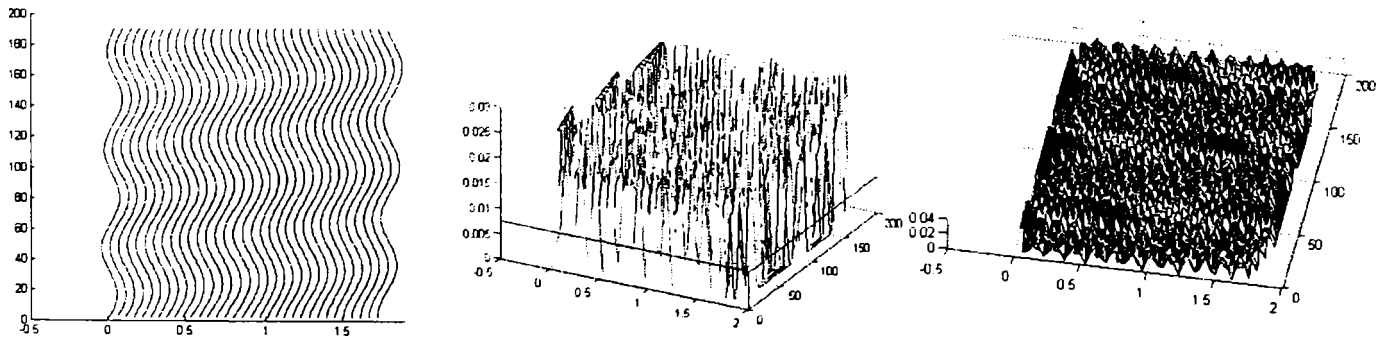
a) b) c)

Fig.A.4.2.3. Regimul mişcării vibratorii: f=21 Hz; A_x=0.028 mm; k_λ =2



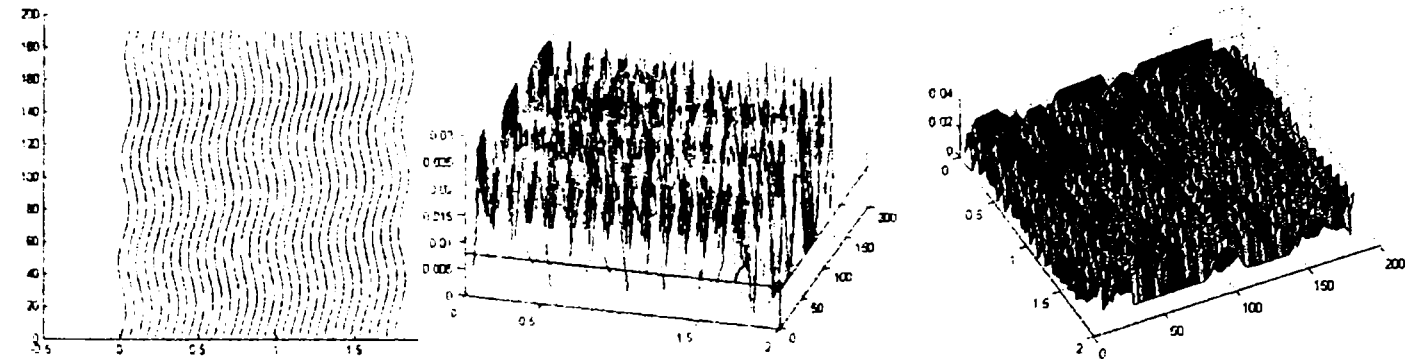
a) b) c)

Fig.A.4.3.1. Regimul mişcării vibratorii: f=31,5 Hz; A_x=0.24 mm; k_λ =3



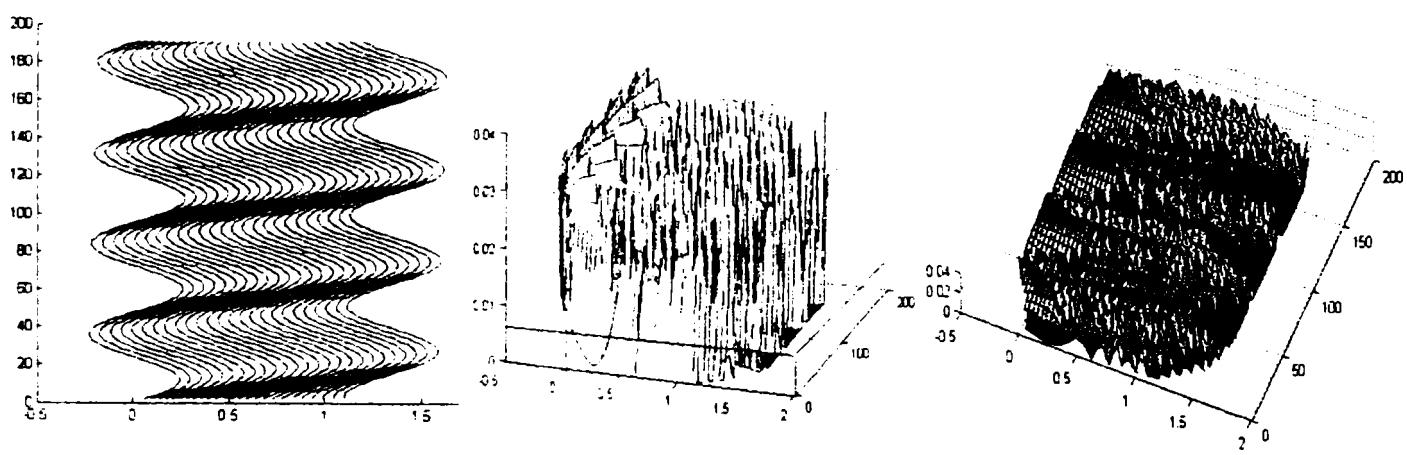
a) b) c)

Fig.A.4.3.2. Regimul mişcării vibratorii: f=31,5 Hz; A_x=0.056 mm; k_λ =3



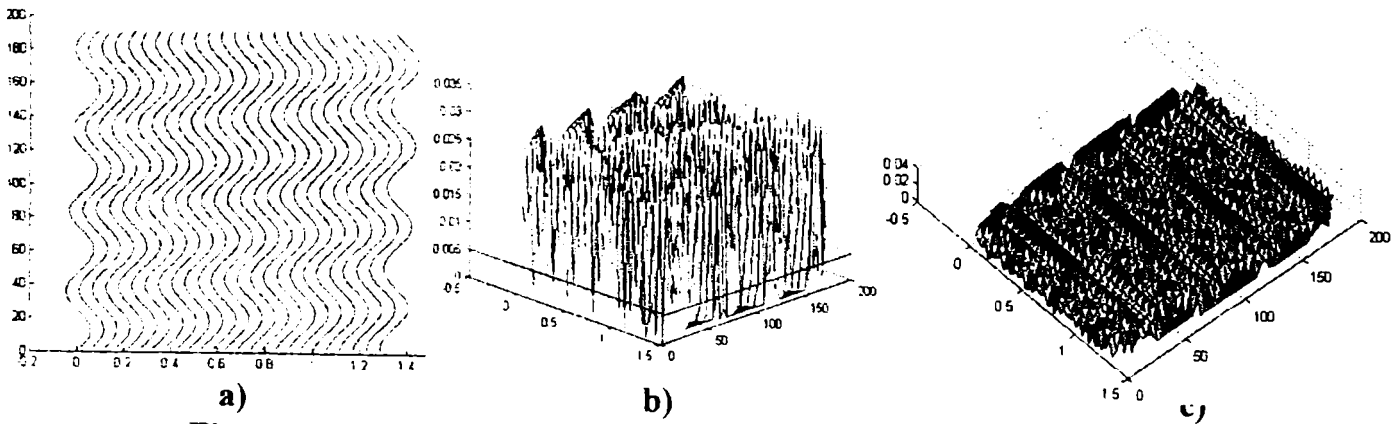
a) b) c)

Fig.A.4.3.3. Regimul mișcării vibratorii: f=31,5 Hz; A_x=0.028 mm; k_λ =3



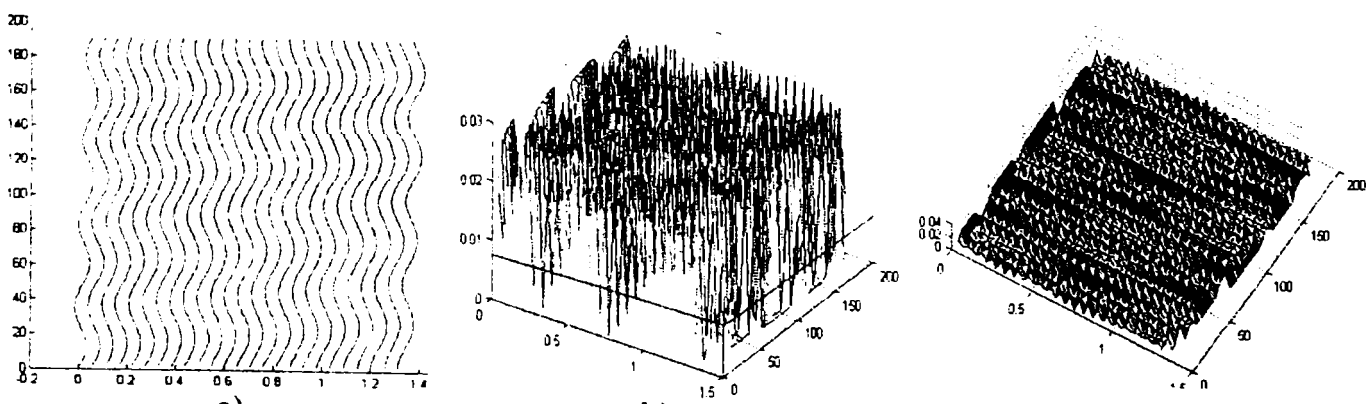
a) b) c)

Fig.A.4.4.1. Regimul mișcării vibratorii: f=42 Hz; A_x=0.24 mm; k_λ =4



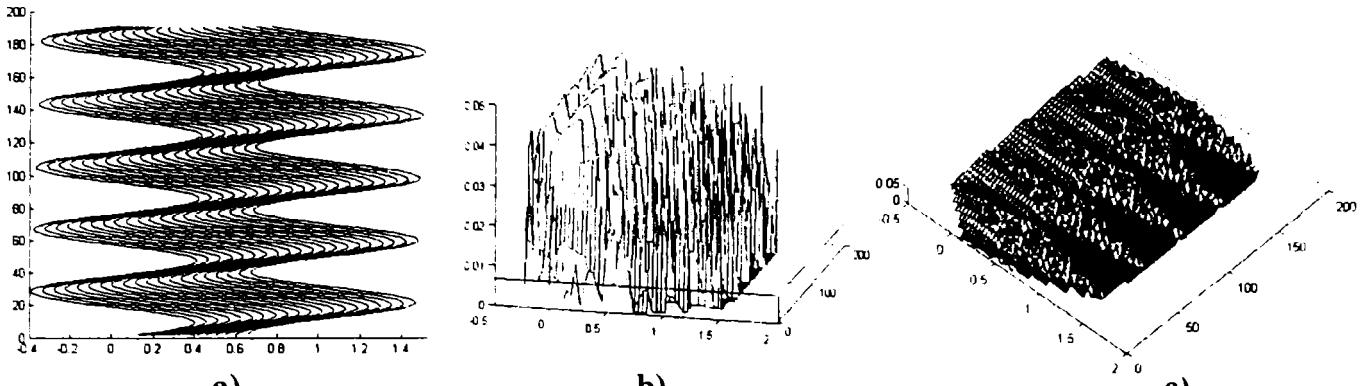
a) b) c)

Fig.A.4.4.2. Regimul mișcării vibratorii: f=42 Hz; A_x=0.056 mm; k_λ =4



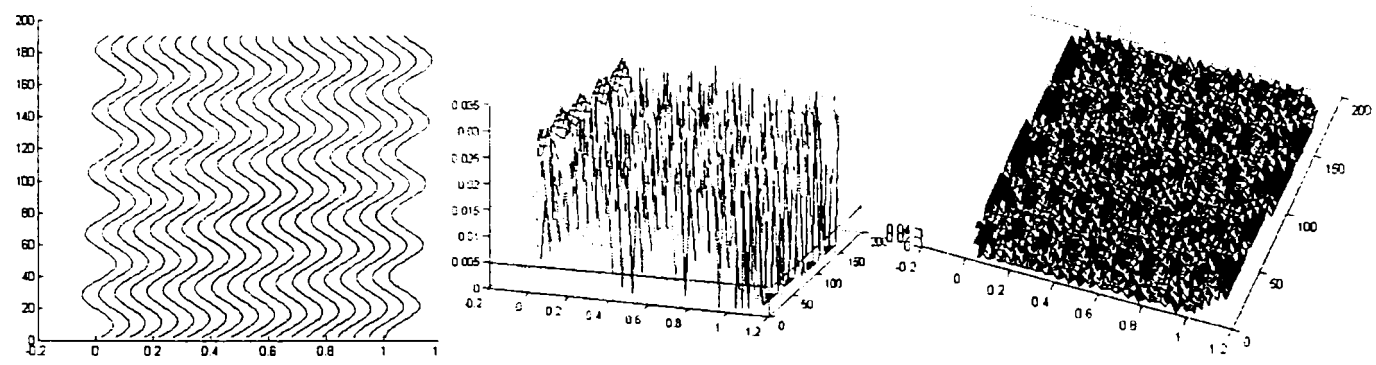
a) b) c)

Fig.A.4.4.3. Regimul mișcării vibratorii: f=42 Hz; A_x=0.028 mm; k_λ =4



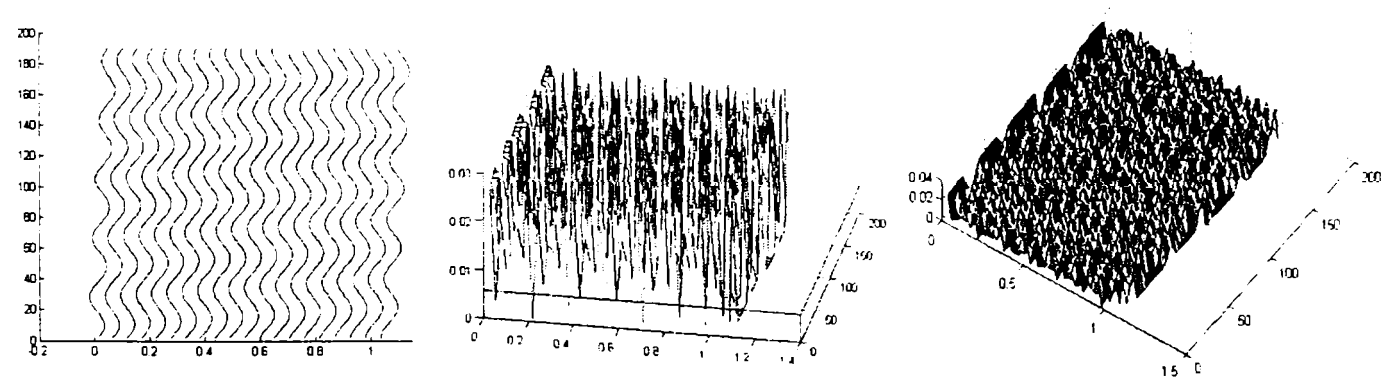
a) b) c)

Fig.A.4.5.1. Regimul mişcării vibratorii: f=52,5 Hz; A_x=0.4 mm; k_λ =5



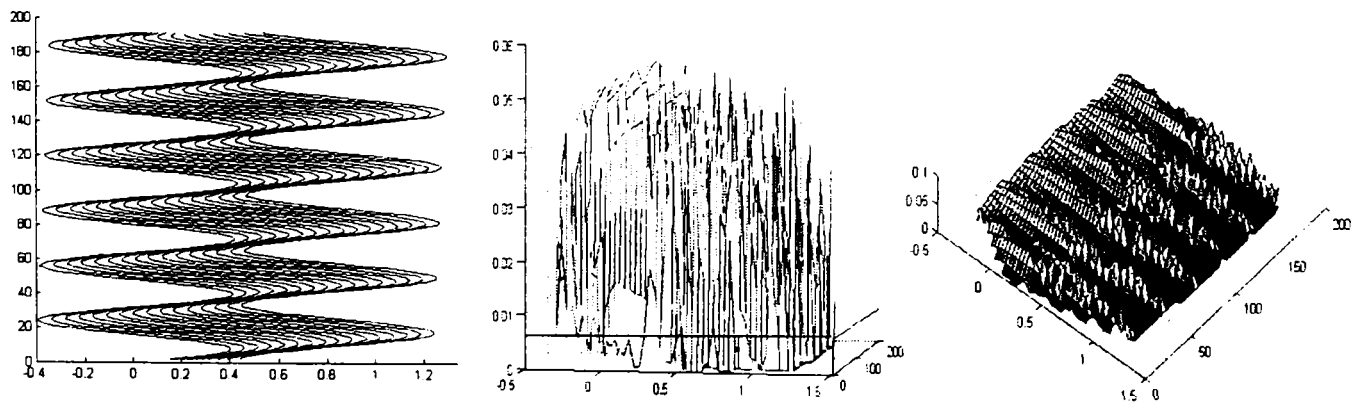
a) b) c)

Fig.A.4.5.2. Regimul mişcării vibratorii: f=52,5 Hz; A_x=0.056 mm; k_λ =5



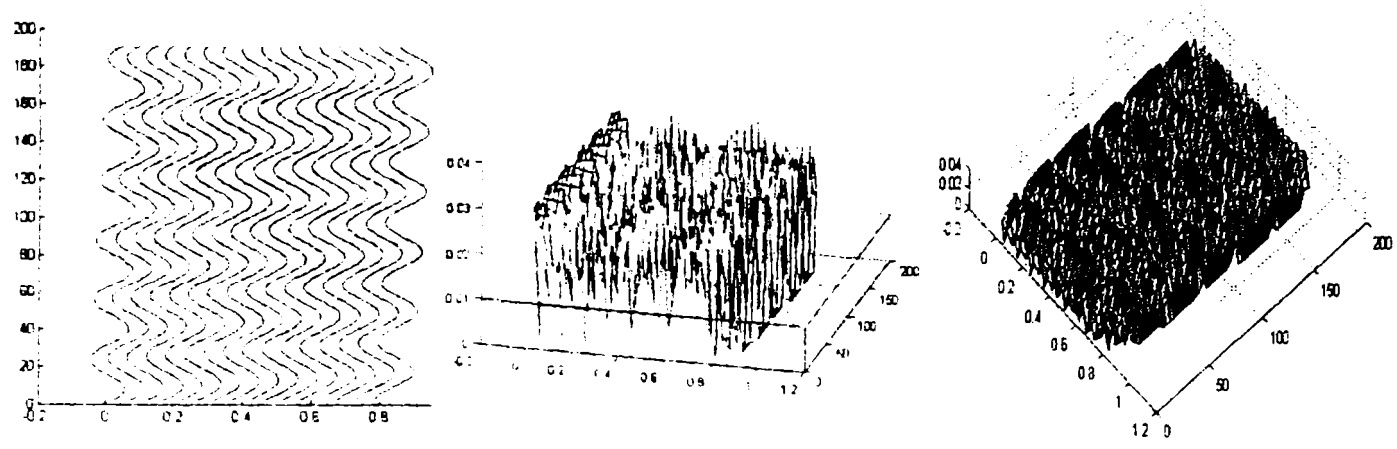
a) b) c)

Fig.A.4.5.3. Regimul mişcării vibratorii: f=52,5 Hz; A_x=0.028 mm; k_λ =5



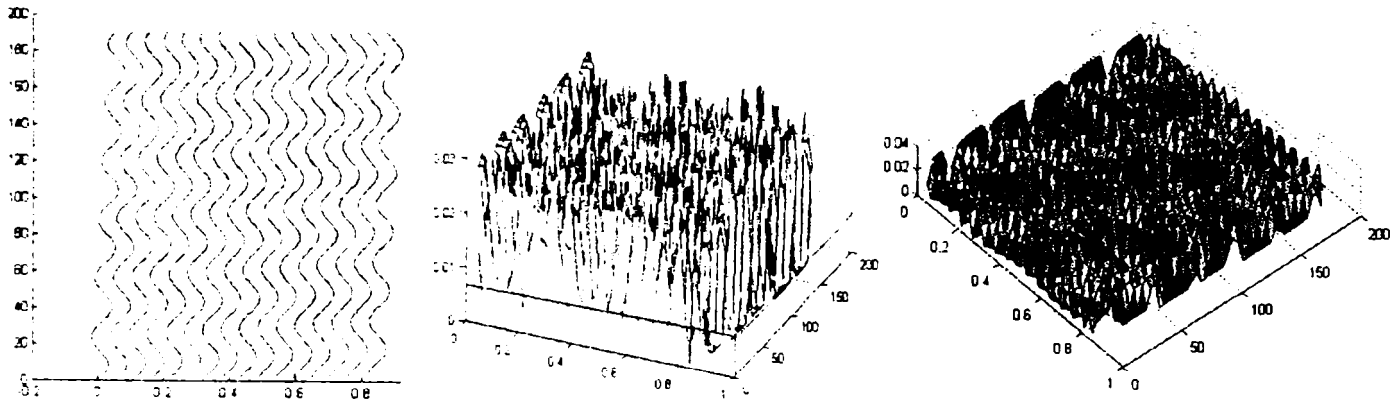
a) b) c)

Fig.A.4.6.1. Regimul mişcării vibratorii: f=63 Hz; A=0.4 mm; k_λ =6



a) b) c)

Fig.A.4.6.2. Regimul mișcării vibratorii: f=63 Hz; A_x=0.056 mm; k_λ =6

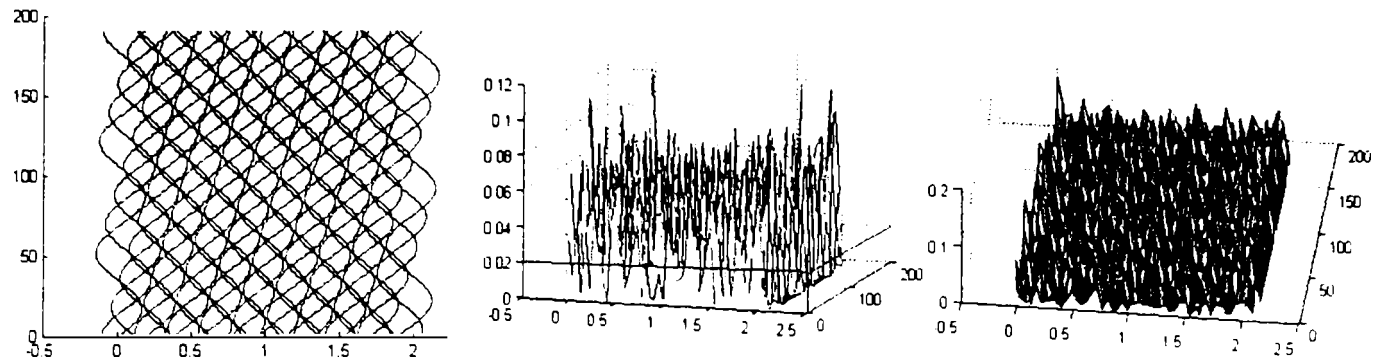


a) b) c)

Fig.A.4.6.3. Regimul mișcării vibratorii: f=63 Hz; A_x=0.028 mm; k_λ =6.

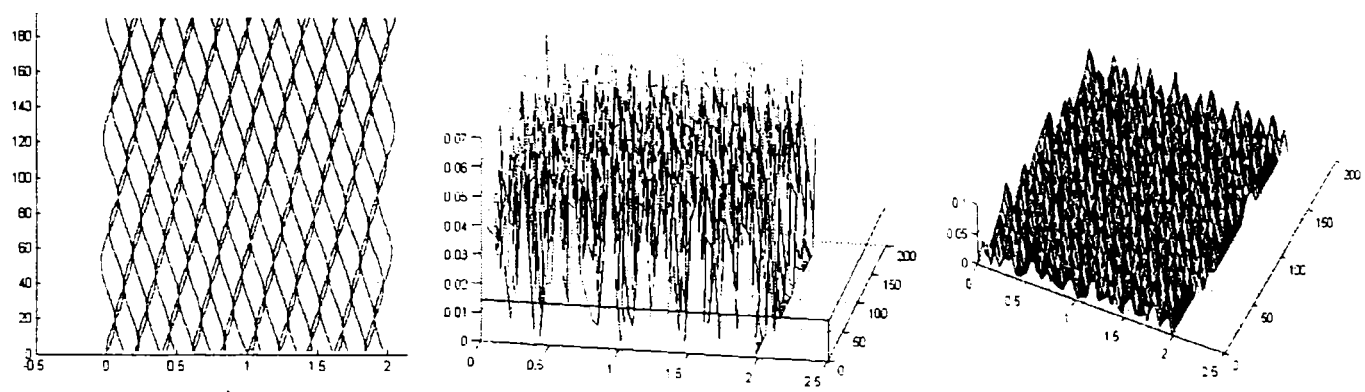
Anexa 5.

Simularea aşezării de k_λ+i perioade pe circumferinţa piesei
k_λ +i=60*f / n; pentru i=1/4; i=1/2; i=3/4 şi i=1, rezultă f=(k_λ +i)*n/60



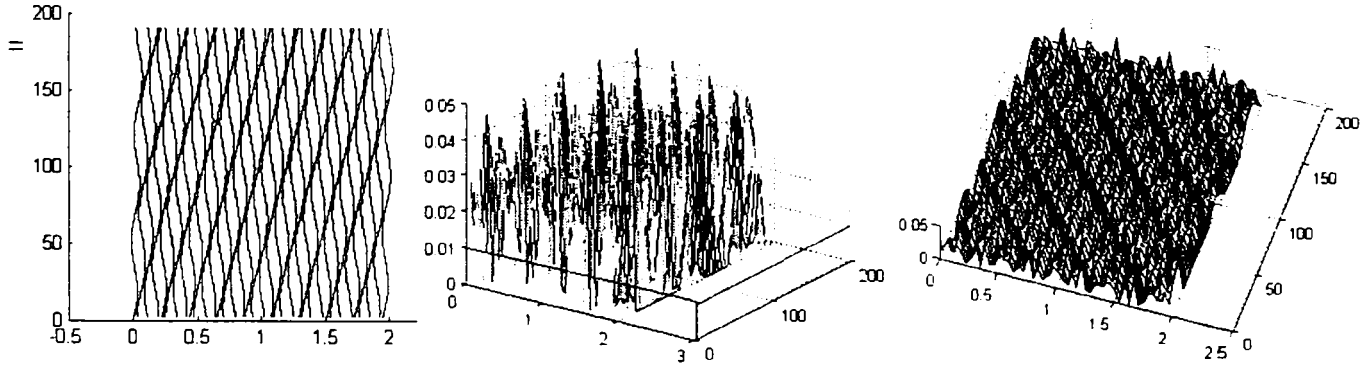
a) **b)** **c)**

Fig.A.5.1.1. Regimul mişcării vibratorii: f=28,875 Hz; A_x=0.16 mm; k_λ =2; i=3/4



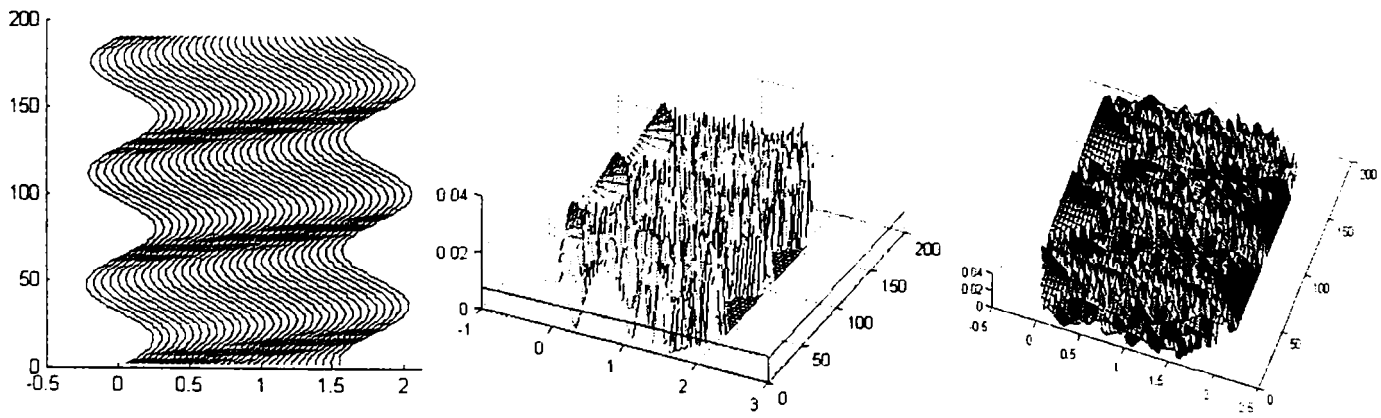
a) **b)** **c**

Fig.A.5.1.2. Regimul mişcării vibratorii: f=28,875 Hz; A_x=0.056 mm; k_λ =2; i=3/4



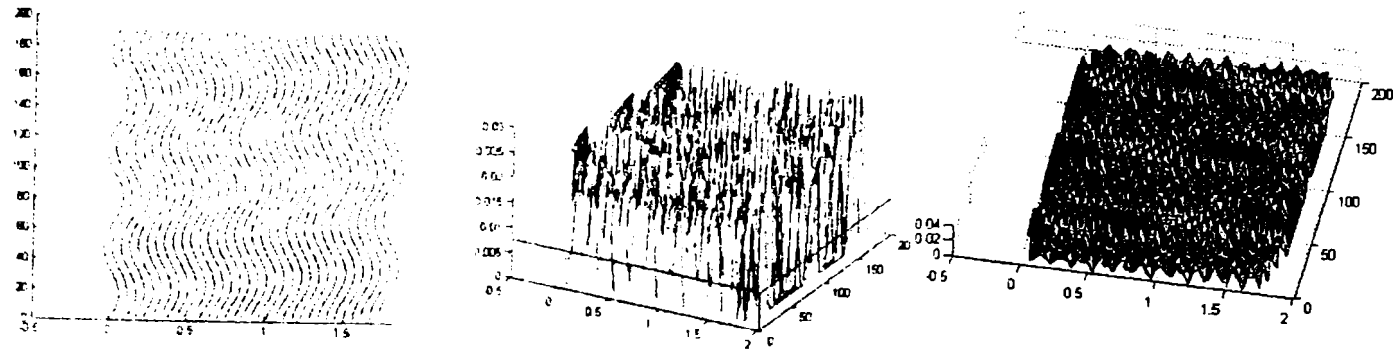
a) **b)** **c)**

Fig.A.5.1.3. Regimul mişcării vibratorii: f=28,875 Hz; A_x=0.028 mm; k_λ =2; i=3/4



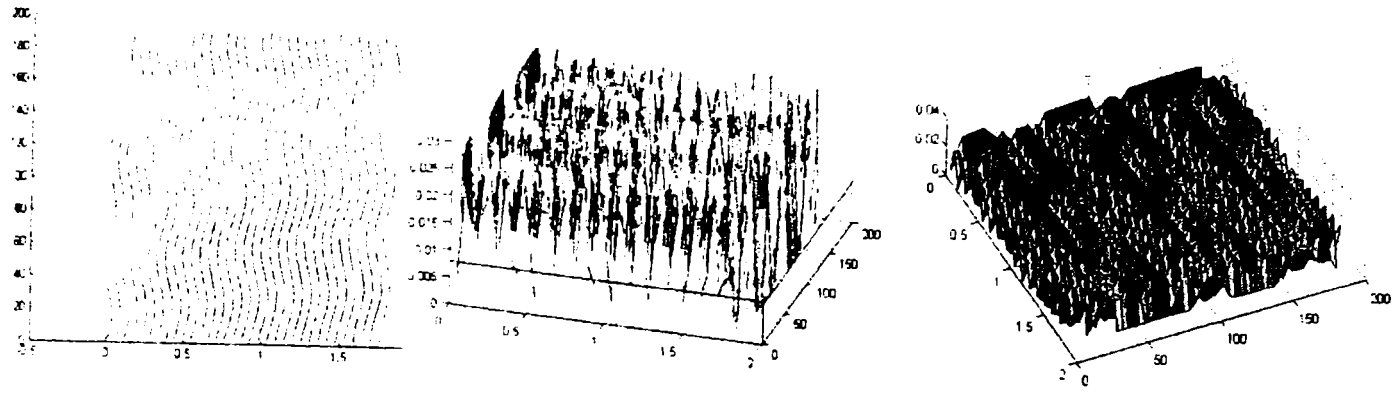
a) **b)** **c)**

Fig.A.5.2.1. Regimul mişcării vibratorii: f=31,5 Hz; A_x=0.24 mm; k_λ =2; i=1



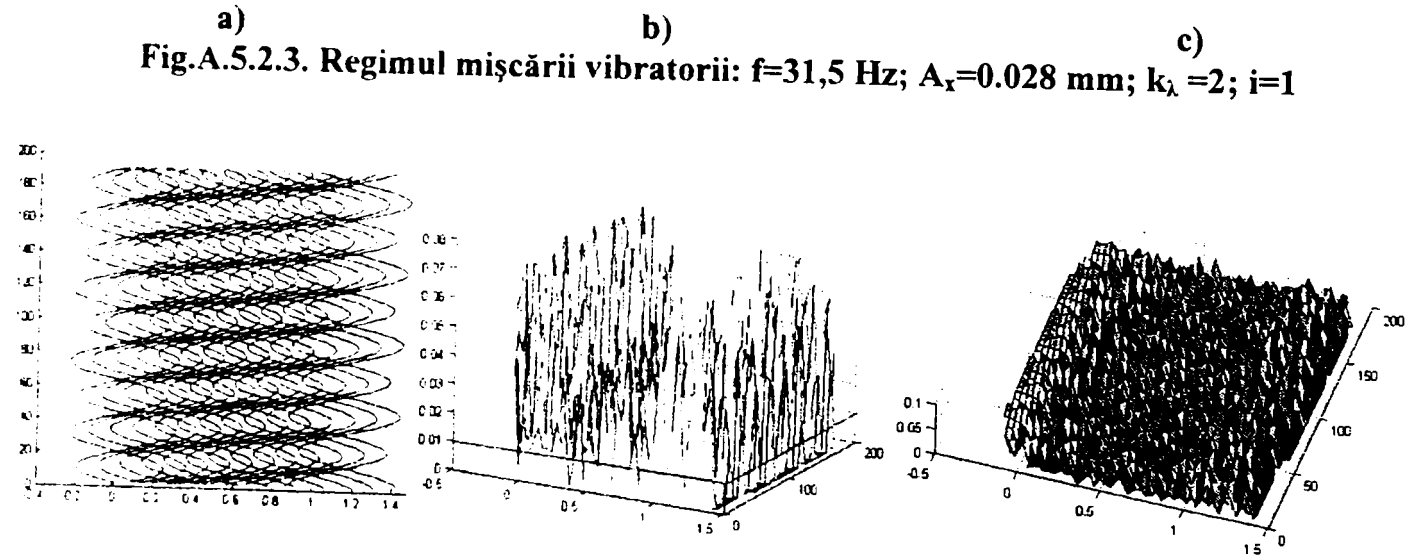
a) b) c)

Fig.A.5.2.2. Regimul mișcării vibratorii: f=31,5 Hz; A_x=0.056 mm; k_λ =2; i=1



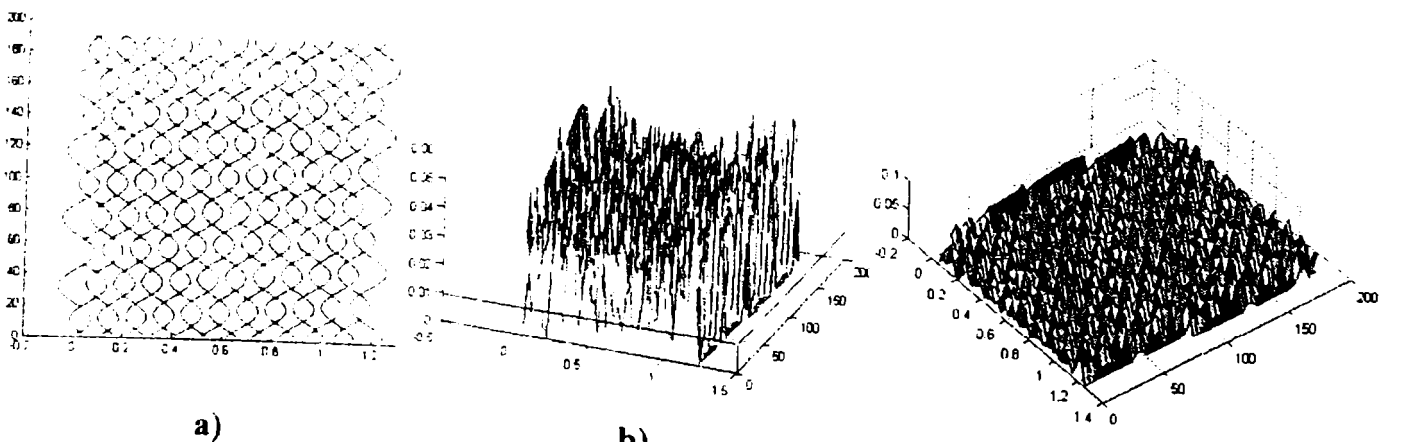
a) b) c)

Fig.A.5.2.3. Regimul mișcării vibratorii: f=31,5 Hz; A_x=0.028 mm; k_λ =2; i=1



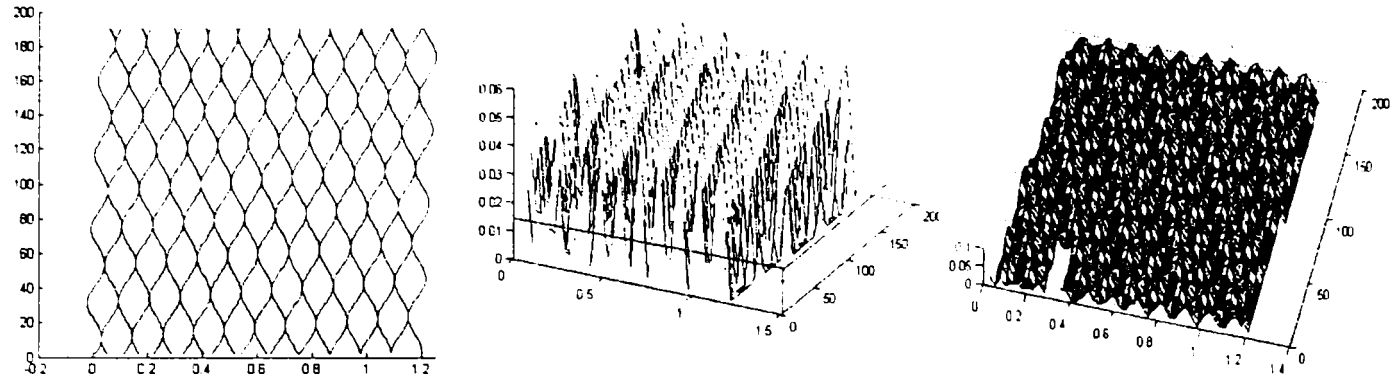
a) b) c)

Fig.A.5.3.1. Regimul mișcării vibratorii: f=47,25 Hz; A_x=0.25 mm; k_λ =4; 1/2



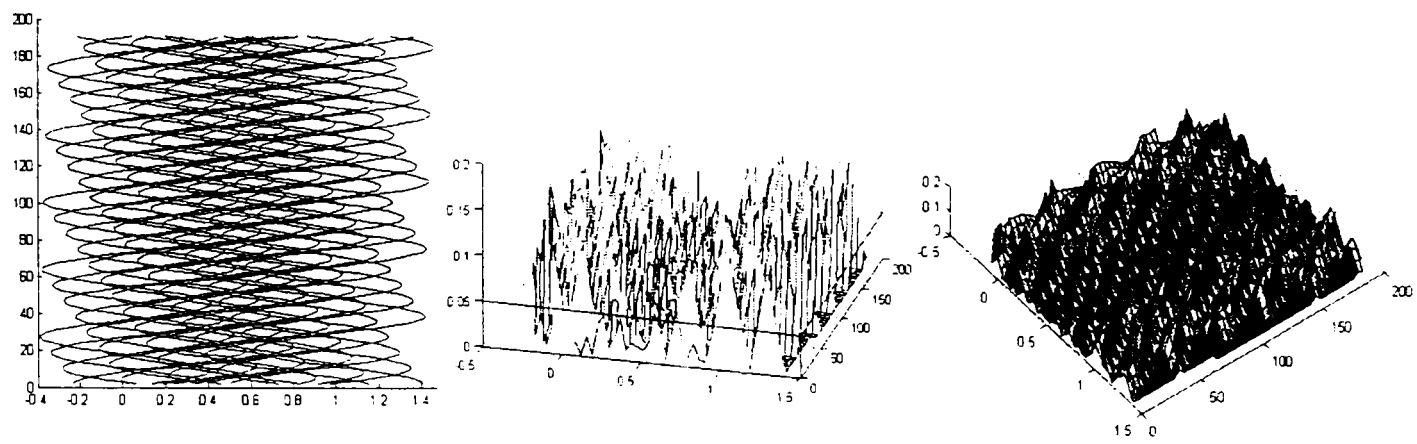
a) b) c)

Fig.A.5.3.2. Regimul mișcării vibratorii: f=47,25 Hz; A_x=0.07 mm; k_λ =4; i=1/2



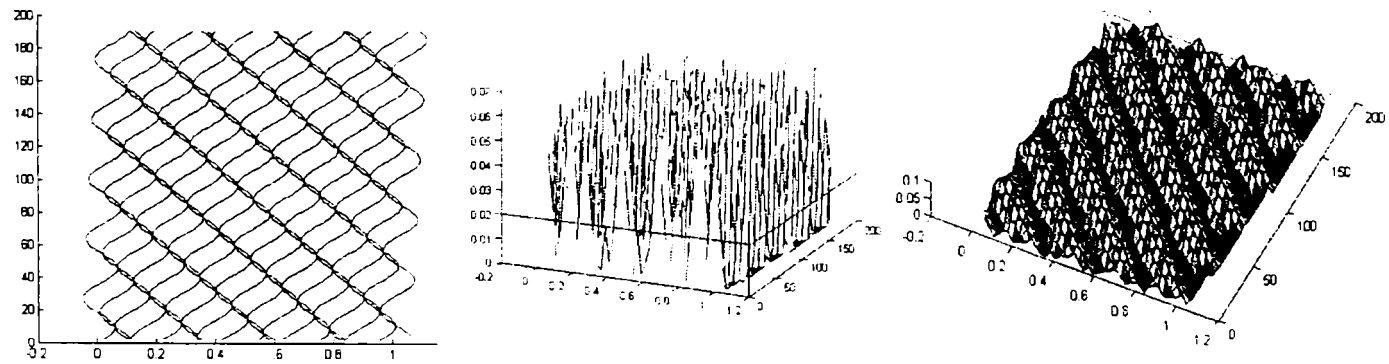
a) b) c)

Fig.A.5.3.3. Regimul mișcării vibratorii: f=47,25 Hz; A_x=0.028 mm; k_λ =4; 1/2



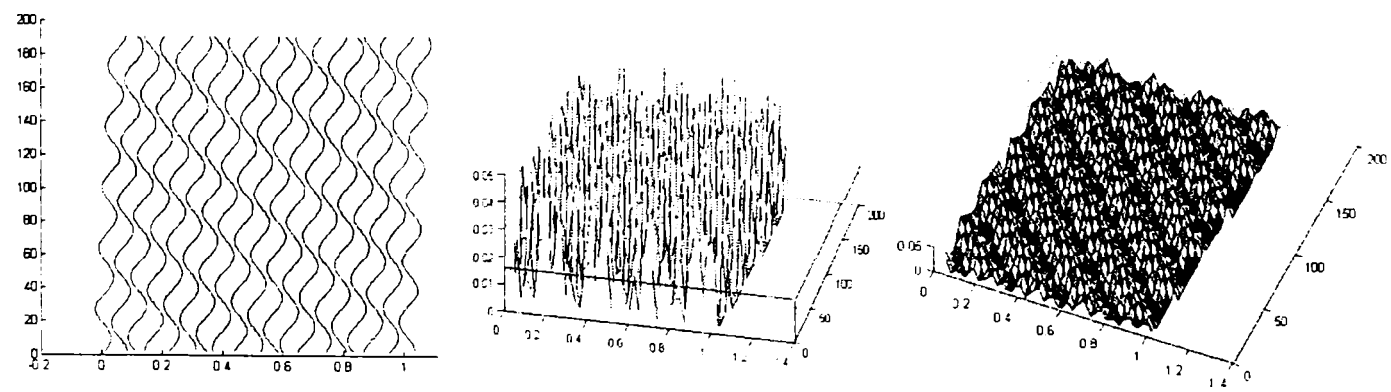
a) b) c)

Fig.A.5.4.1. Regimul mișcării vibratorii: f=55,125 Hz; A_x=0.4 mm ; k_λ =5; i=1/4



a) b) c)

Fig.A.5.4.2. Regimul mișcării vibratorii: f=55,125 Hz; A_x=0.056 mm; k_λ =5; i=1/4



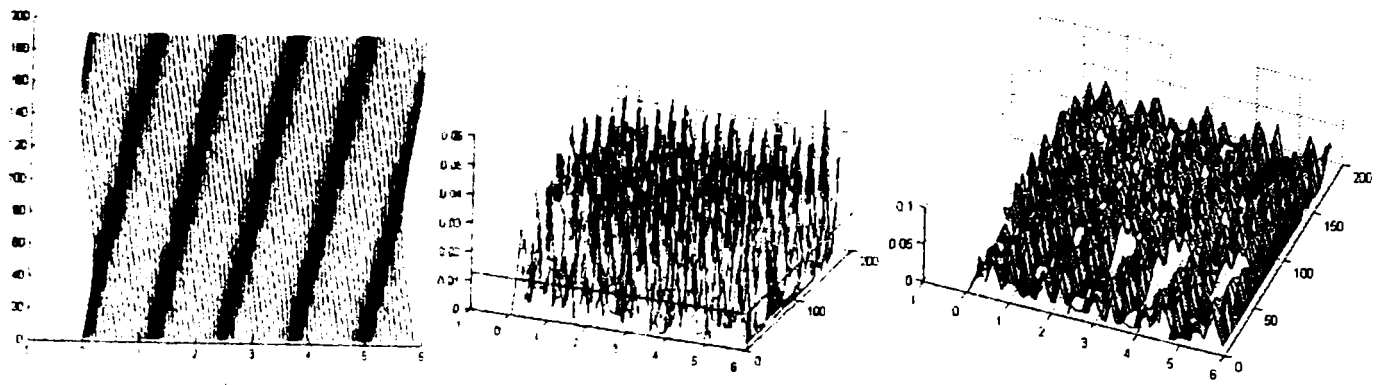
a) b) c)

Fig.A.5.4.3. Regimul mișcării vibratorii: f=55,125 Hz; A_x=0,028 mm; k_λ =5; i=1/4

Anexa 6.

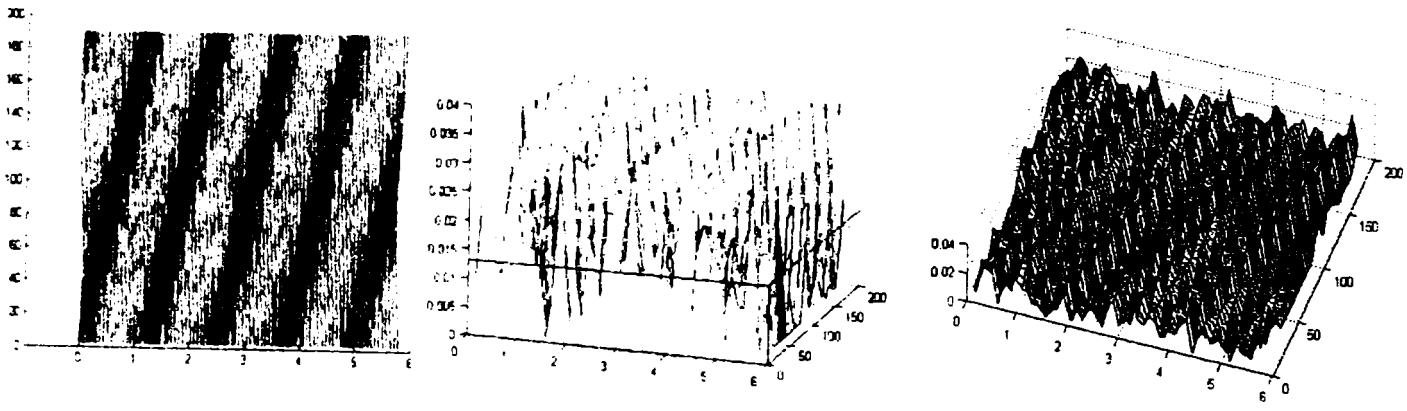
Simularea așezării unui număr întreg de lungimi de undă λ pe circumferința piesei.

$\pi \cdot d = \lambda(k_\lambda + i)$; $k_\lambda+i=60*f/n$; $i=0$; $\pi*d/n=60*f/n$; $f=\pi*d*n/60*\lambda$



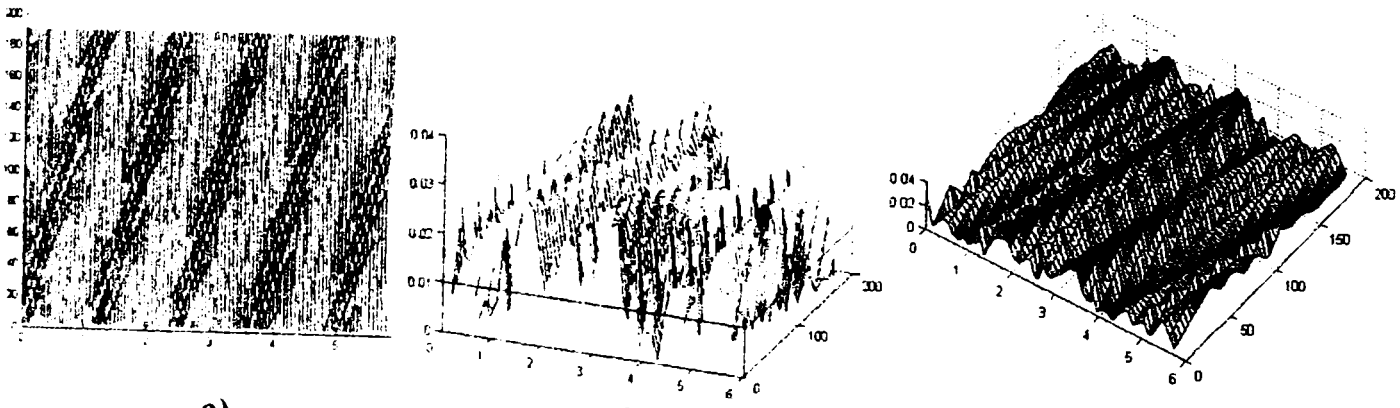
a) **b)** **c)**

Fig. A.6.1.1. Regimul mișcării vibratorii: f=10,06095 Hz; A_x=0,16mm; λ=200.



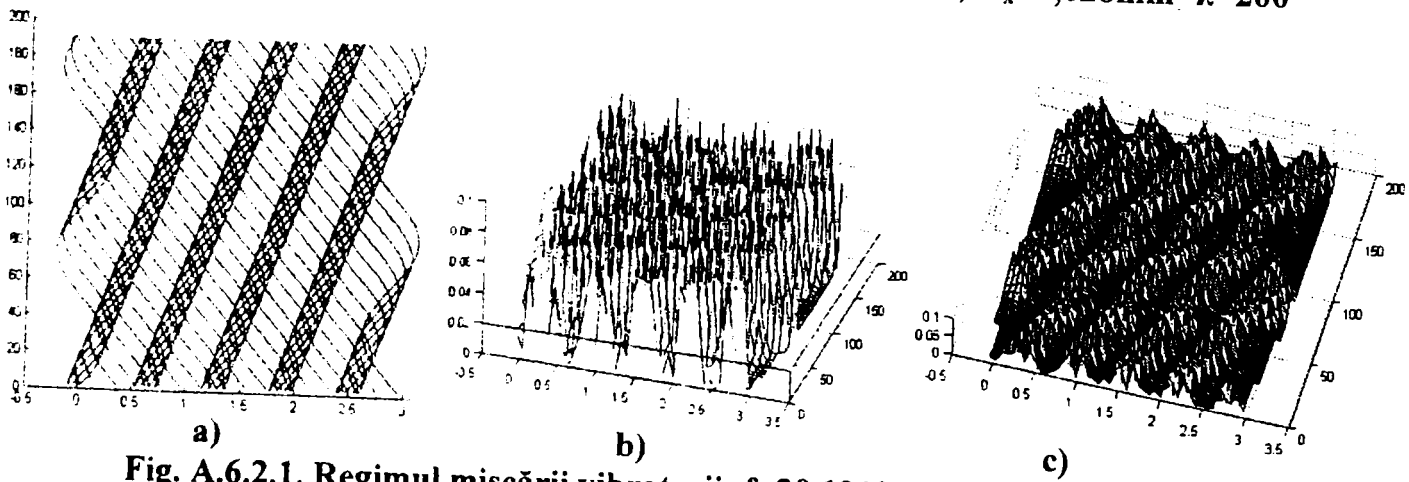
a) **b)** **c)**

Fig. A.6.1.2. Regimul mișcării vibratorii: f=10,06095 Hz; A_x=0,056mm; λ=200



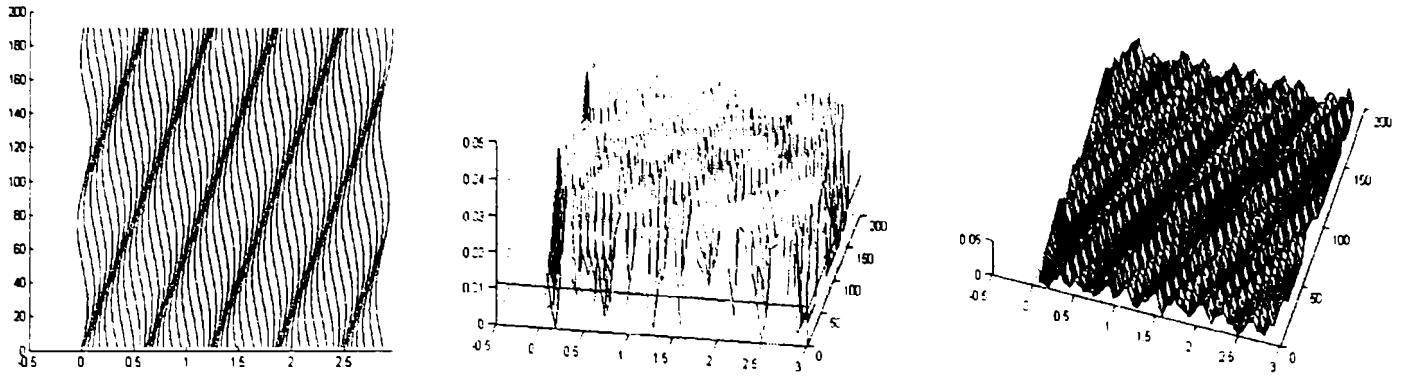
a) **b)** **c)**

Fig. A.6.1.3. Regimul mișcării vibratorii: f=10,06095 Hz; A_x=0,028mm λ=200



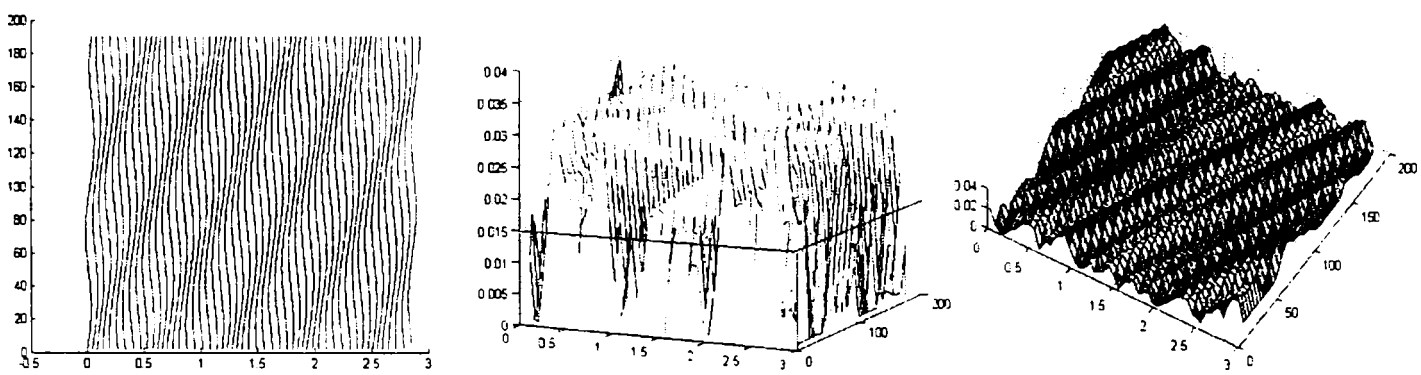
a) **b)** **c)**

Fig. A.6.2.1. Regimul mișcării vibratorii: f=20,1219 Hz; A_x=0,24mm; λ=100



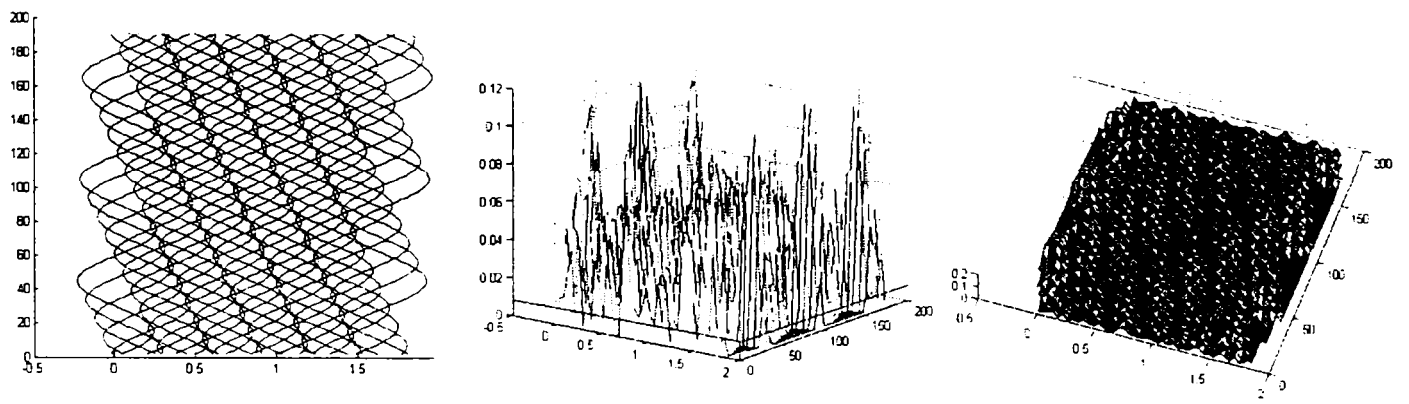
a) b) c)

Fig. A.6.2.2. Regimul mişcării vibratorii: f=20,1219 Hz; A_x=0,056mm; λ=100



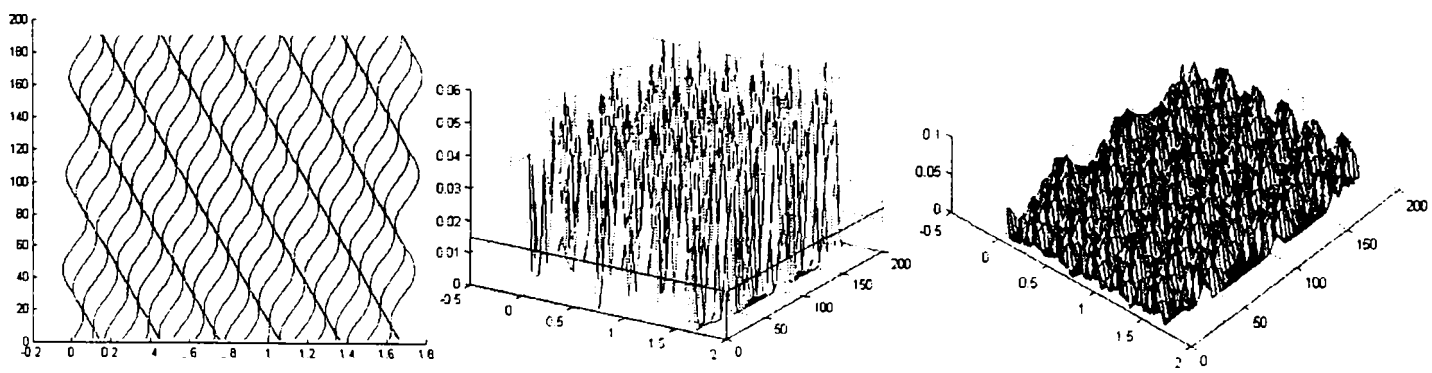
a) b) c)

Fig. A.6.2.3. Regimul mişcării vibratorii: f=20,1219 Hz; A_x=0,028mm; λ=100



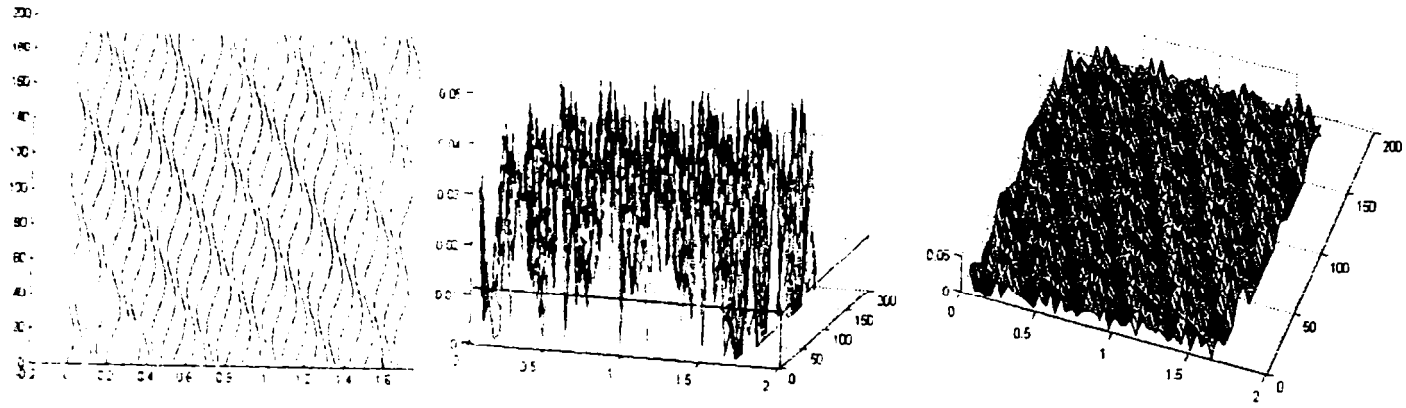
a) b) c)

Fig. A.6.3.1. Regimul mişcării vibratorii: f=33,536501 Hz; A_x=0,24mm; λ=60



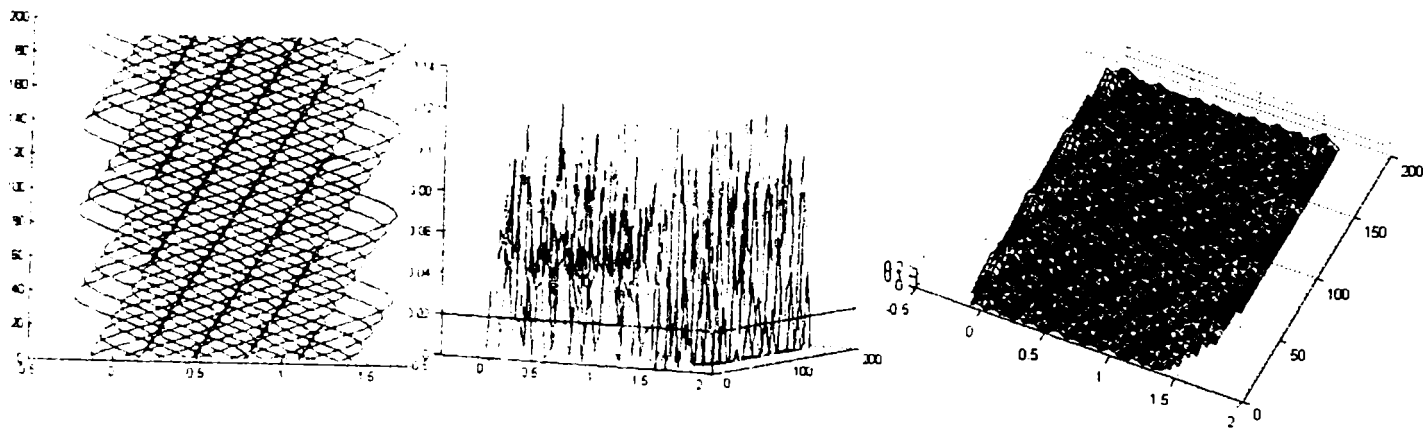
a) b) c)

Fig. A.6.3.2. Regimul mişcării vibratorii: f=33,536501 Hz; A_x=0,056mm; λ=60



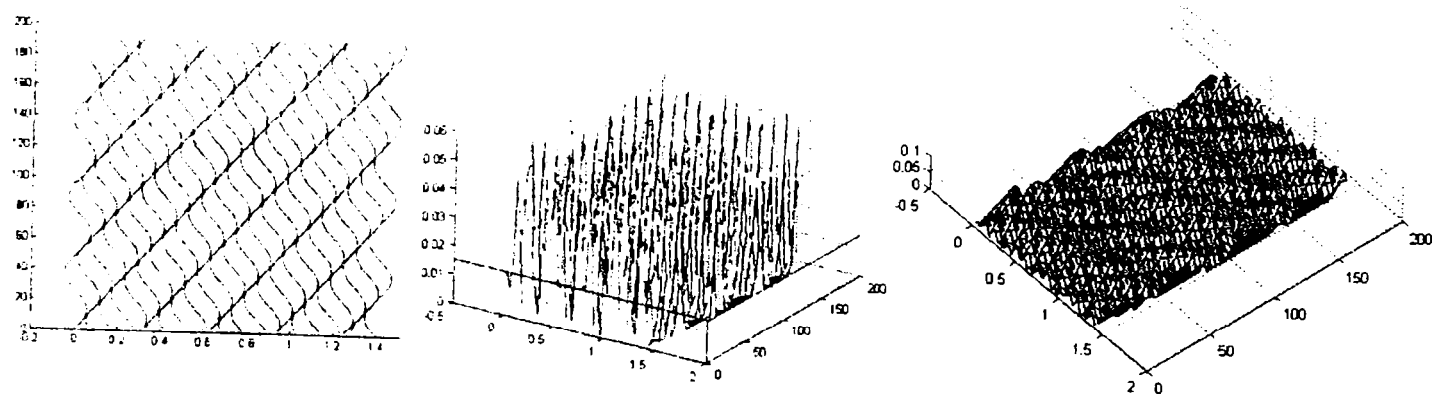
a) b) c)

Fig. A.6.3.3. Regimul mișcării vibratorii: f=33,536501 Hz; A_x=0,028mm; λ=60



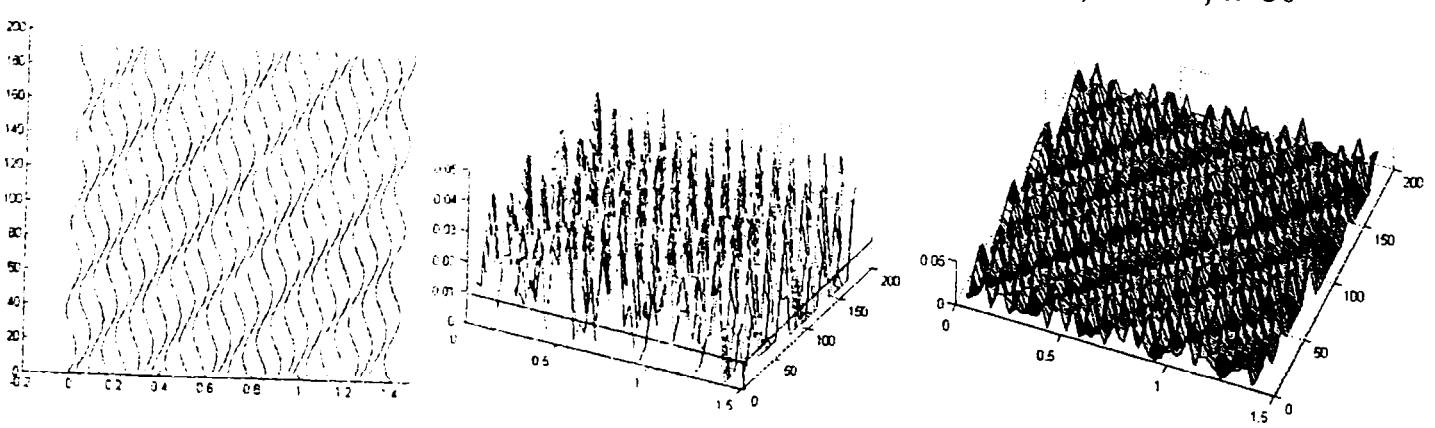
a) b) c)

Fig. A.6.4.1. Regimul mișcării vibratorii: f=40,243801 Hz; A_x=0,24mm; λ=50



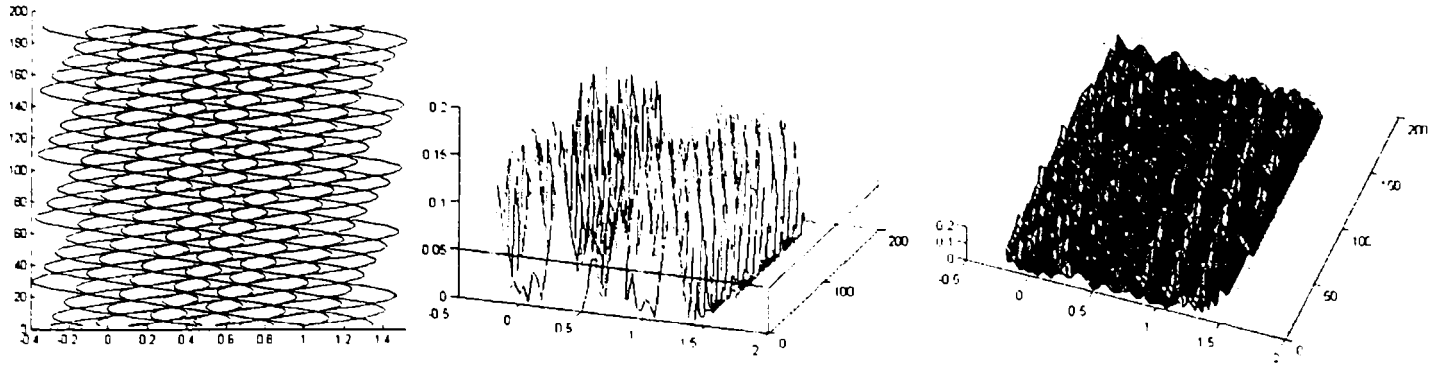
a) b) c)

Fig. A.6.4.2. Regimul mișcării vibratorii: f=40,243801 Hz; A_x=0,056mm; λ=50



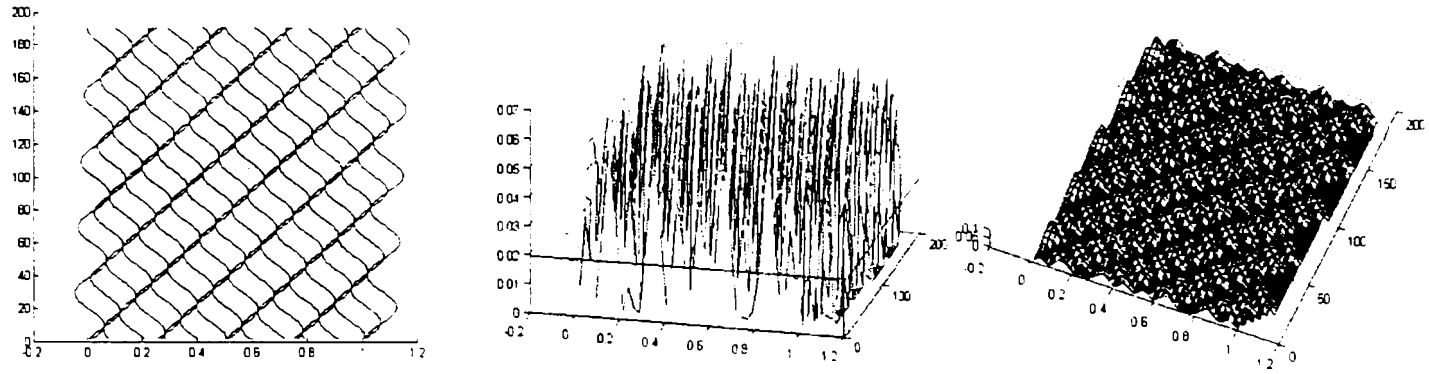
a) b) c)

Fig. A.6.4.3. Regimul mișcării vibratorii: f=40,243801 Hz; A_x=0,028mm; λ=50



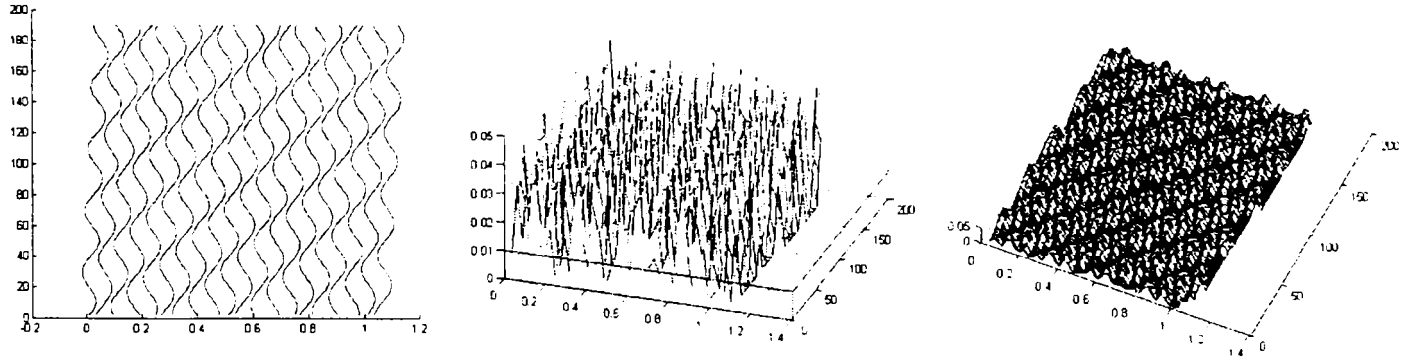
a) b) c)

Fig. A.6.5.1. Regimul mişcării vibratorii: f=50,304752 Hz; A_x=0,4mm; λ=40



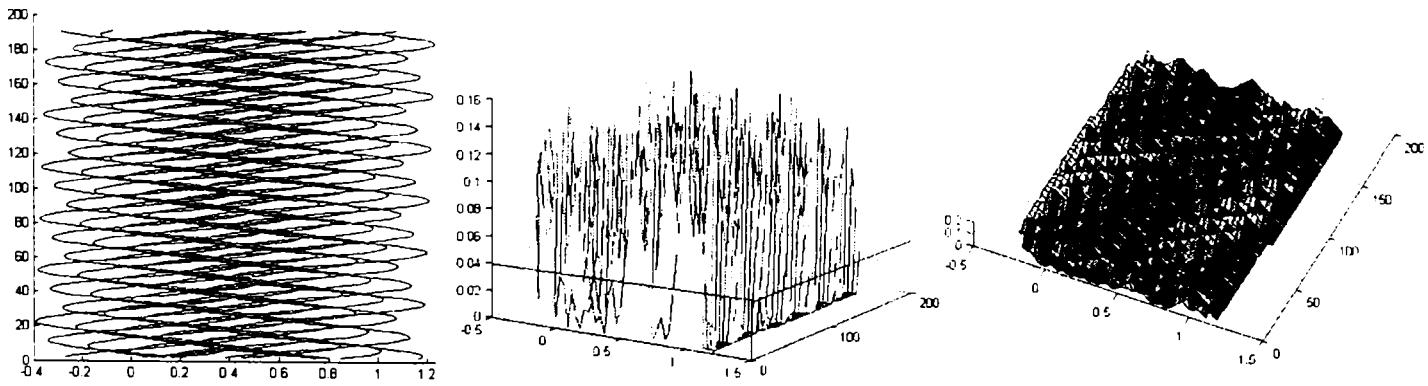
a) b) c)

Fig. A.6.5.2. Regimul mişcării vibratorii: f=50,304752 Hz; A_x=0,056mm; λ=40



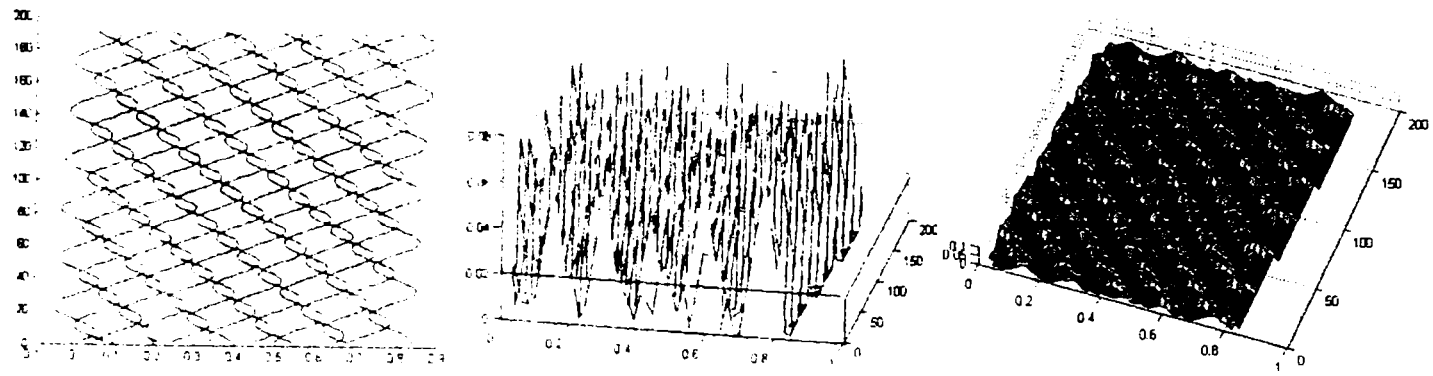
a) b) c)

Fig. A.6.5.3. Regimul mişcării vibratorii: f=50,304752 Hz; A_x=0,028mm; λ=40



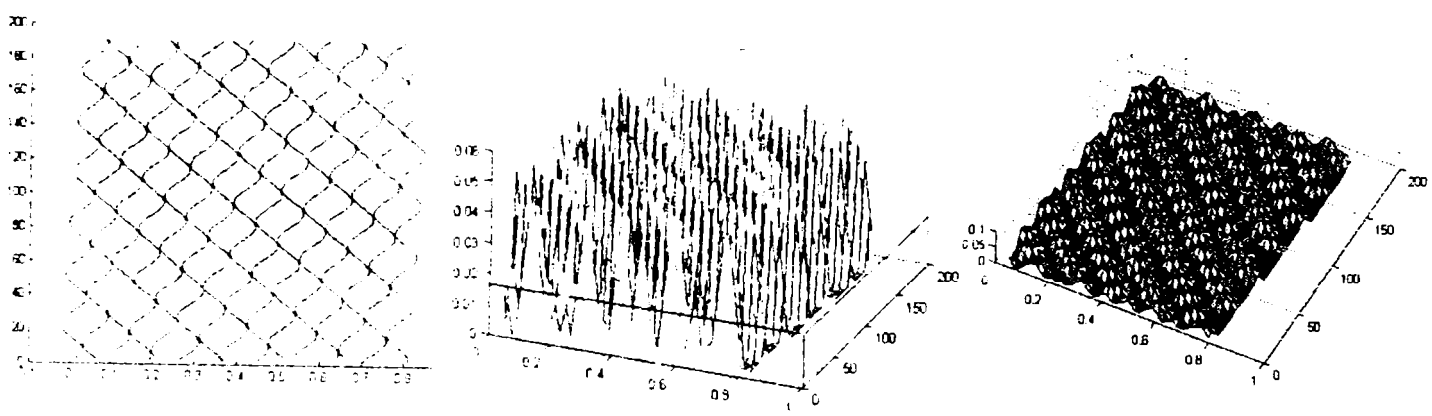
a) b) c)

Fig. A.6.6.1. Regimul mişcării vibratorii: f=67,073003 Hz; A_x=0,4mm; λ=30



a) b) c)

Fig. A.6.6.2. Regimul mișcării vibratorii: f=67,073003 Hz; A_x=0,056mm; λ=30

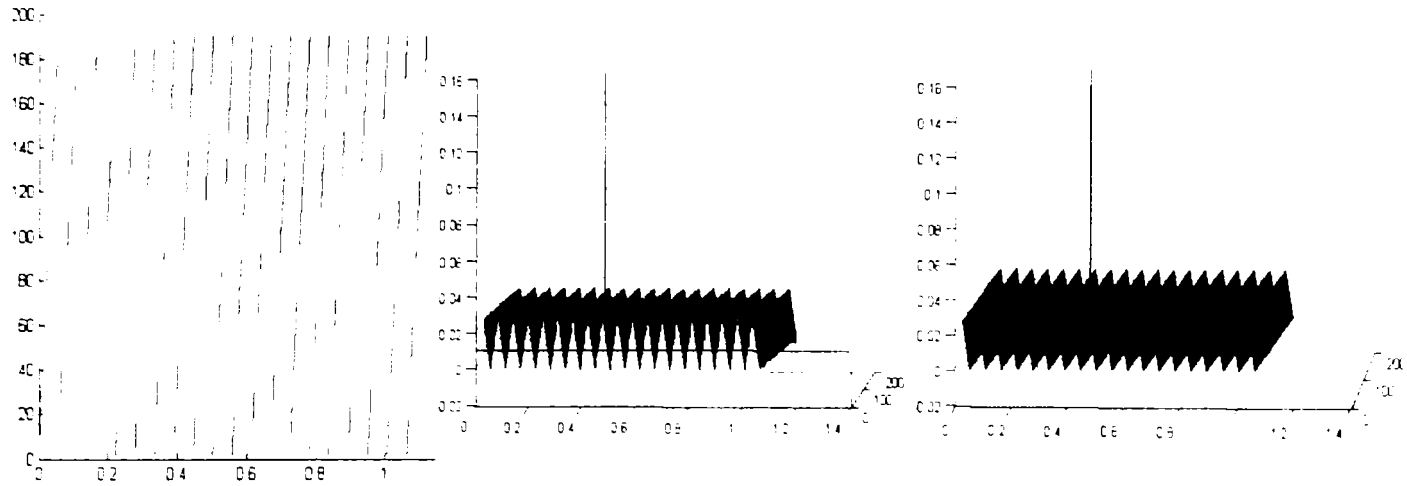


a) b) c)

Fig. A.6.6.3. Regimul mișcării vibratorii: f=67,073003 Hz; A_x=0,028mm; λ=30

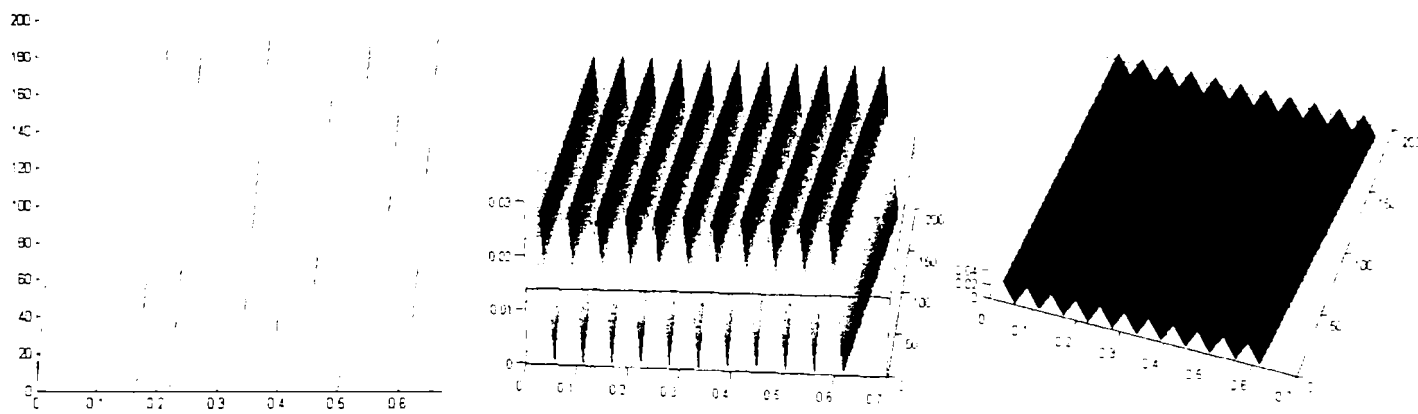
Anexa 7.

Simulare strunjire fără vibrații



a) b) c)

Fig. A.7.1. Regim de lucru: n=630 [rot/min], s_0=0,056 [mm/rot], nr.de perioade considerate =20

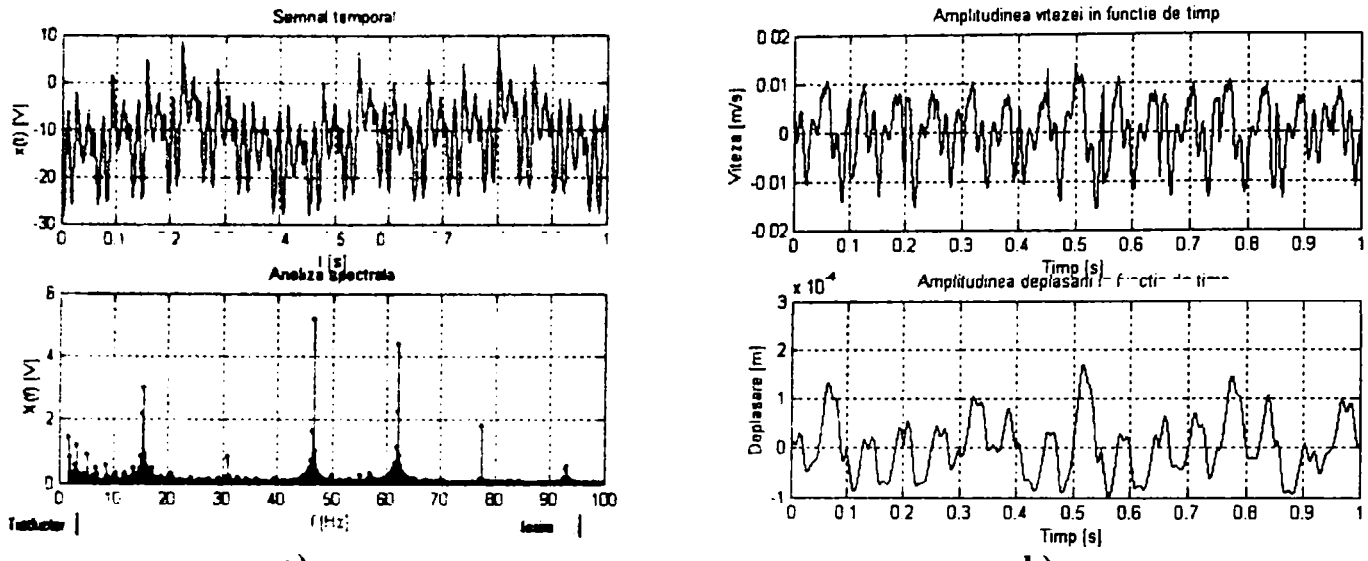


a) b) c)

Fig. A.7.2. Regim de lucru: n=630 [rot/min], s_0=0,056 [mm/rot], nr. de perioade considerate =12.

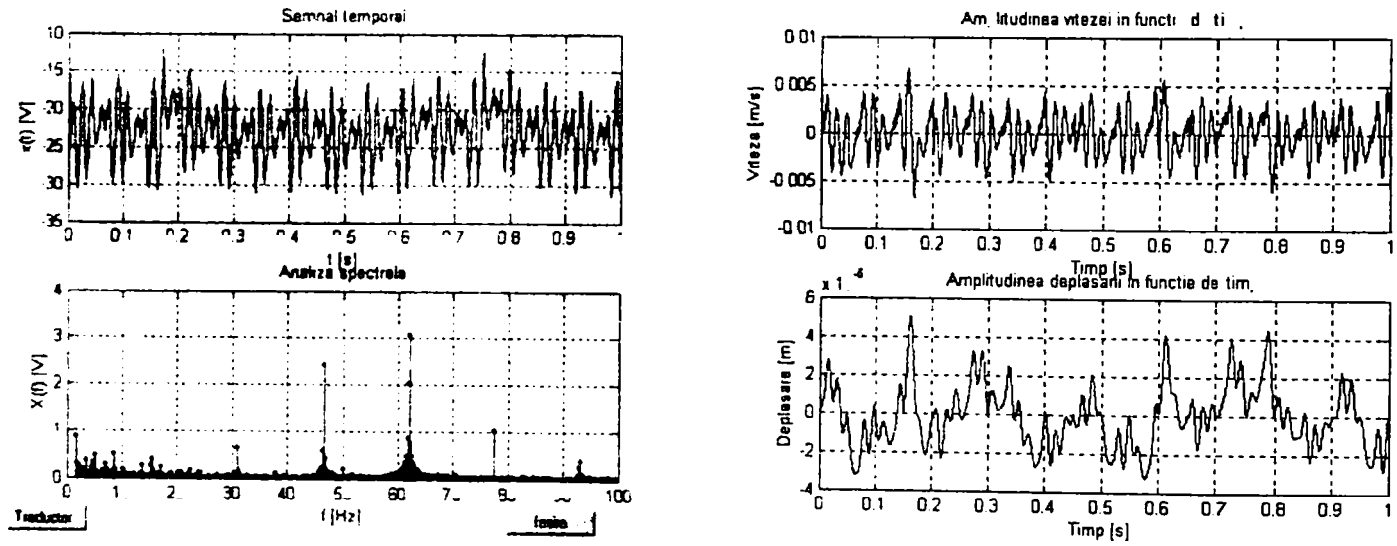
Anexa 8.

Frecvențe de lucru la mers în gol al dispozitivului electromecanic



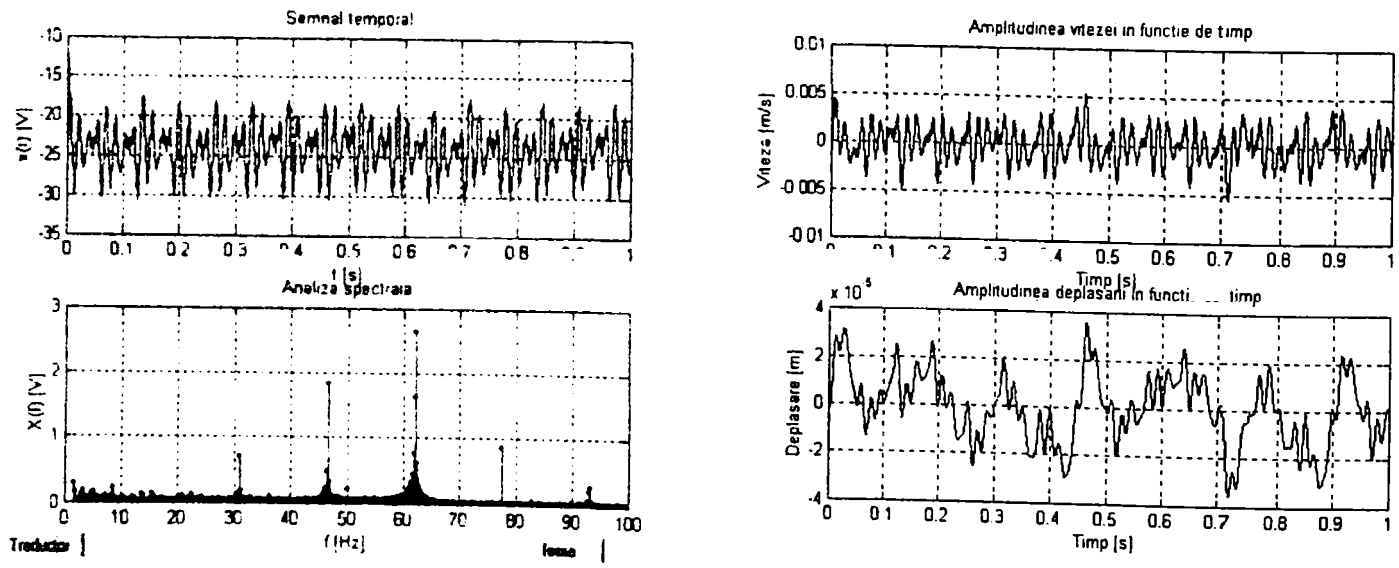
a) b)

Fig. A.8.1.1. Frecvența testată: f=15 Hz



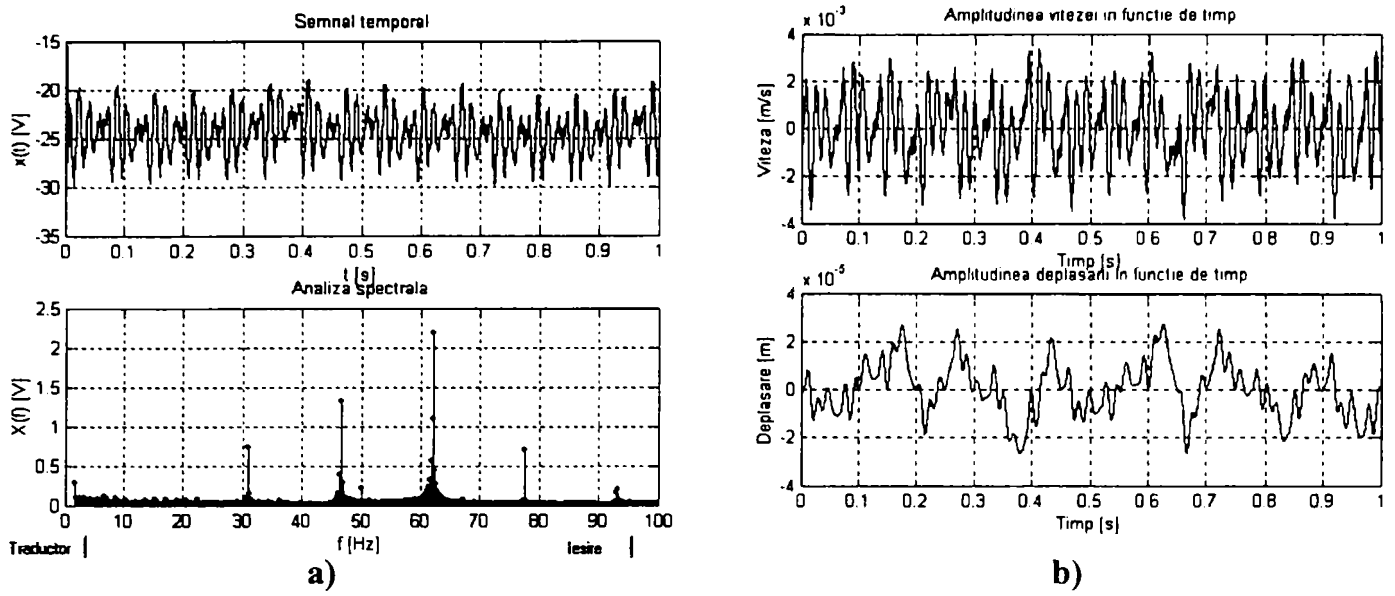
a) b)

Fig. A.8.1.2. Frecvența testată: f=15,1 Hz



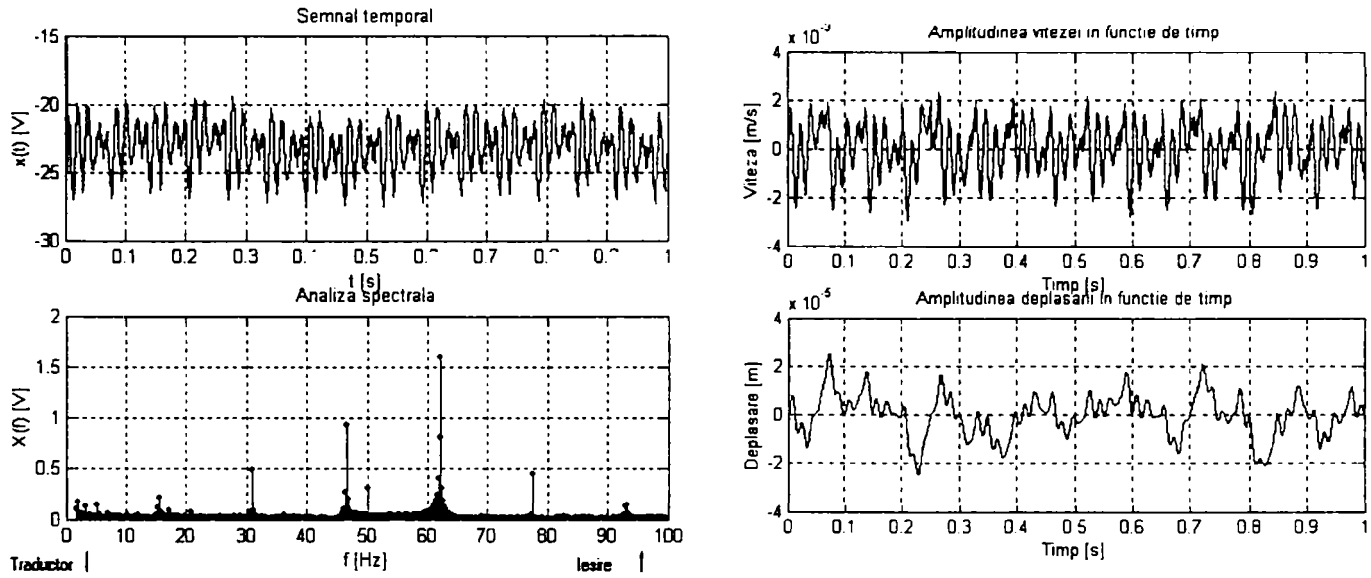
a) b)

Fig. A.8.1.3. Frecvența testată: f=15,2 Hz



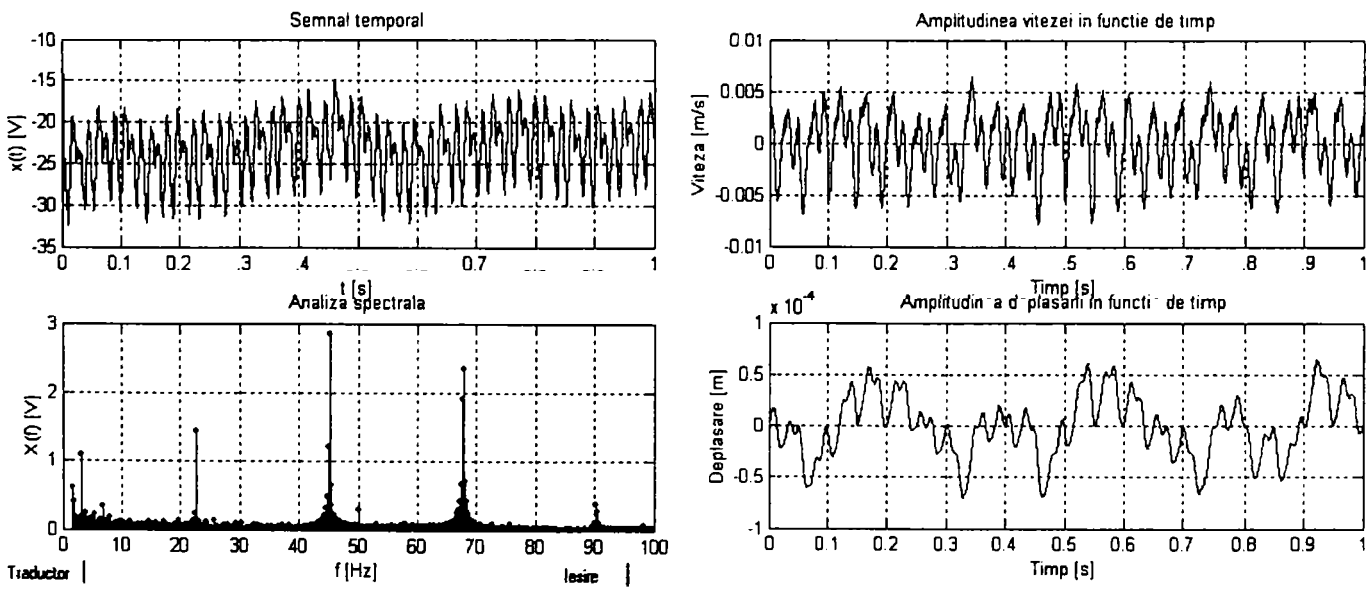
a) b)

Fig. A.8.1.4. Frecvența testată: f=15,3 Hz



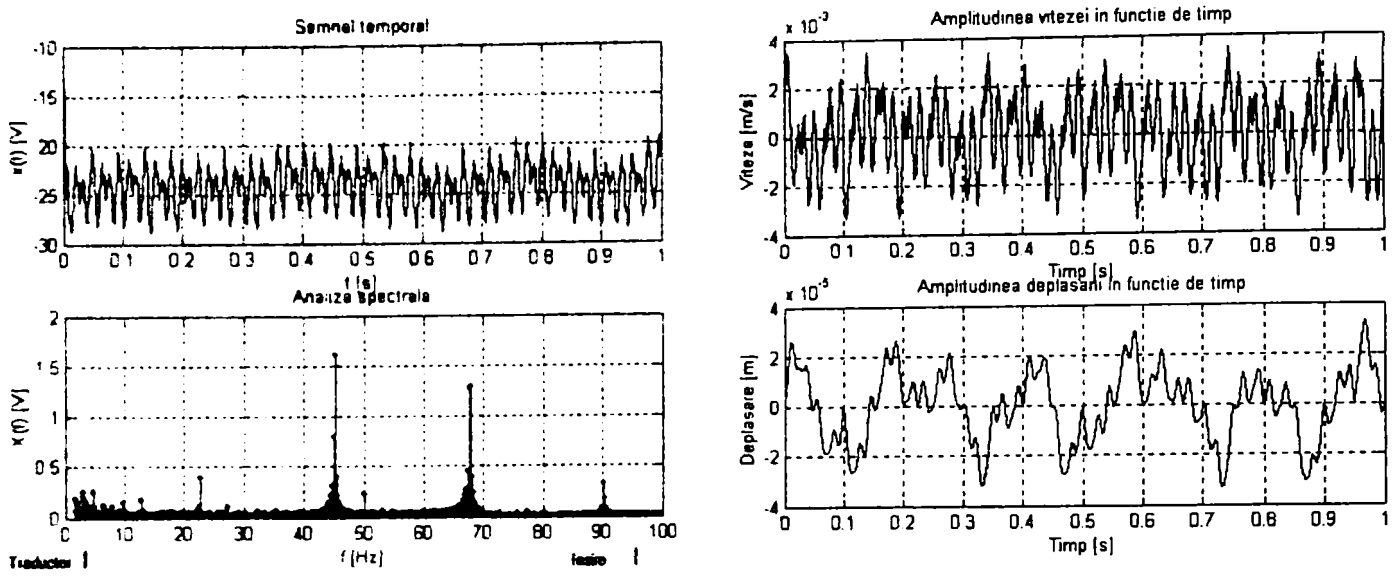
a) b)

Fig. A.8.1.5. Frecvența testată: f=15,4 Hz



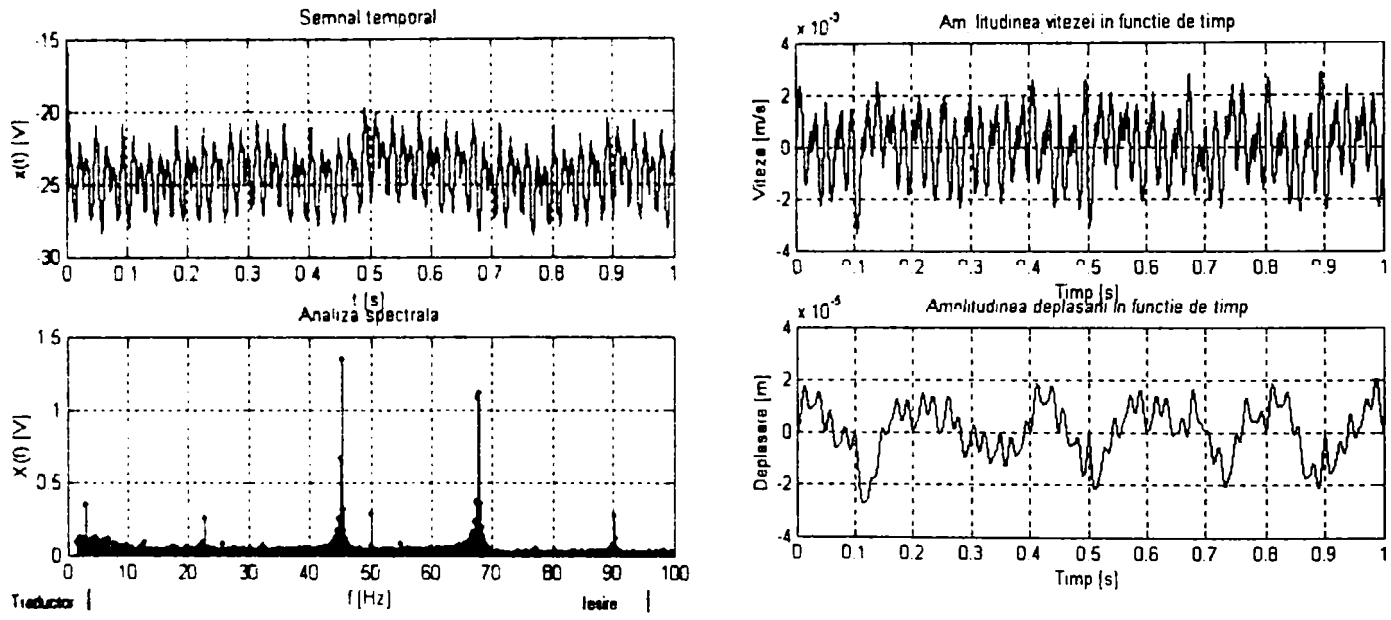
a) b)

Fig. A.8.2.1. Frecvența testată: f=23 Hz



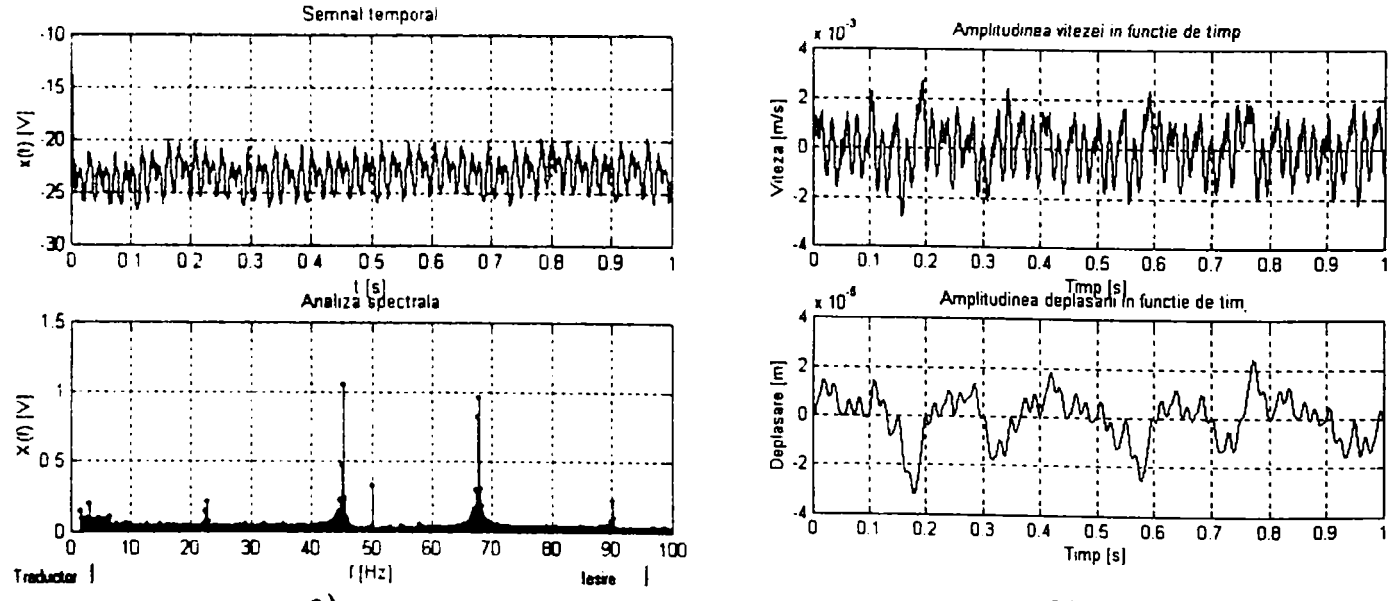
a) b)

Fig. A.8.2.2. Frecvența testată: f=23,1 Hz



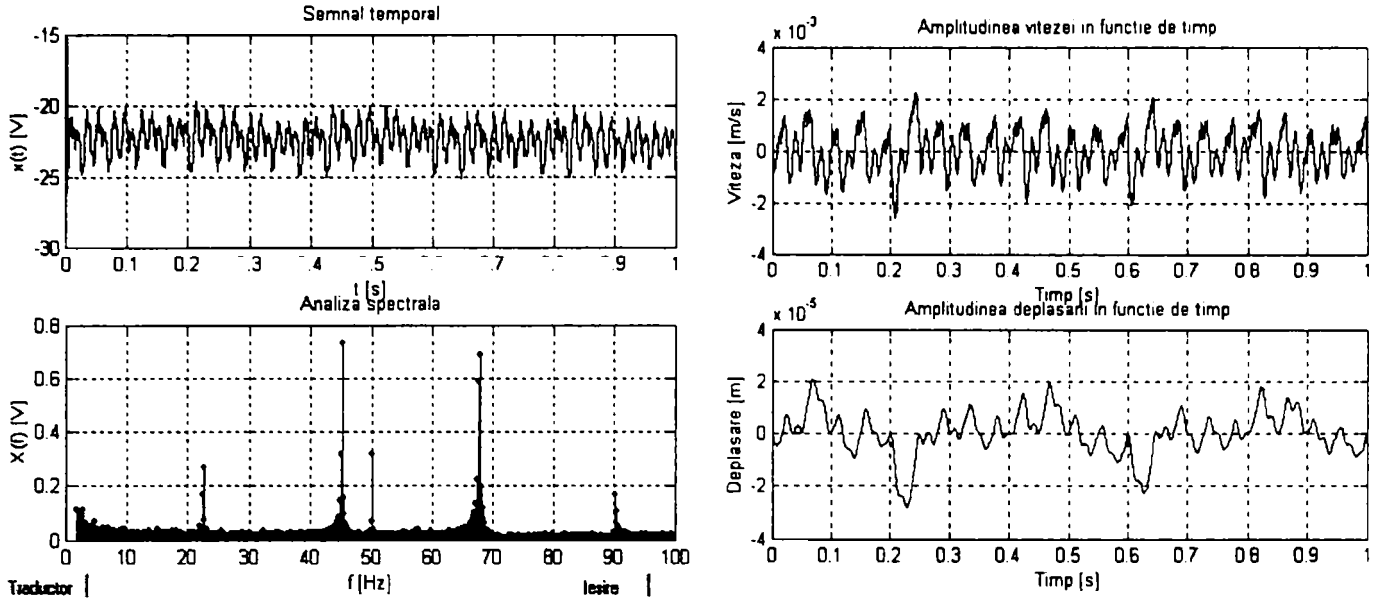
a) b)

Fig. A.8.2.3. Frecvența testată: f=23,2 Hz



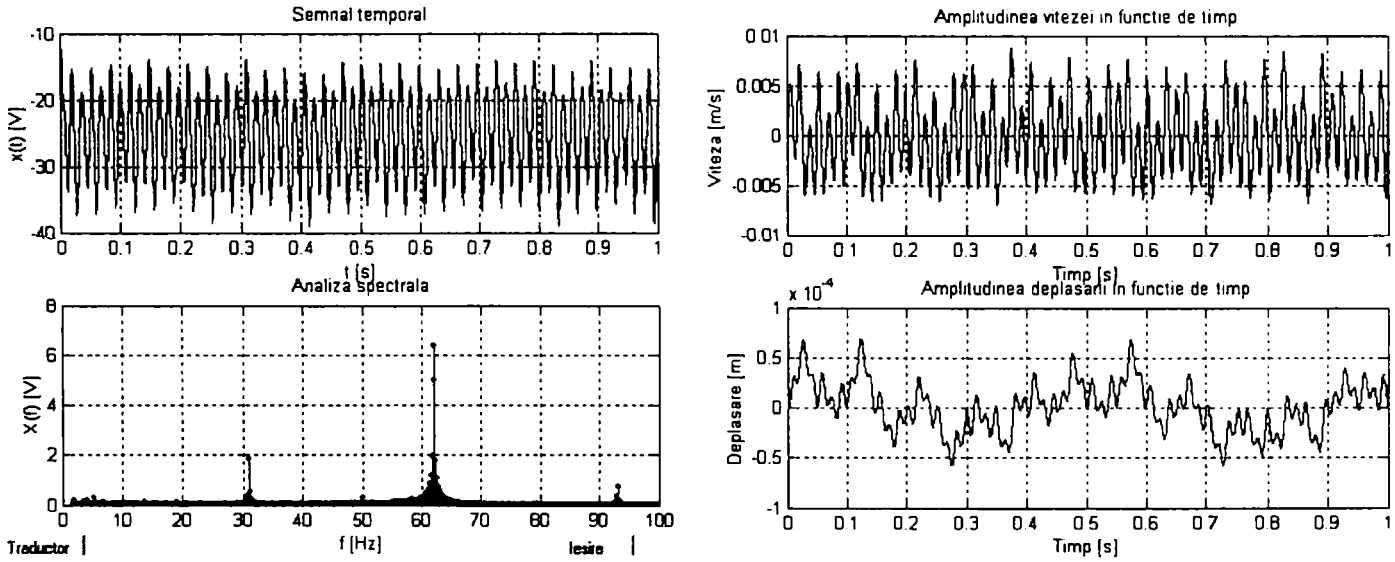
a) b)

Fig. A.8.2.4. Frecvența testată: f=23,3 Hz



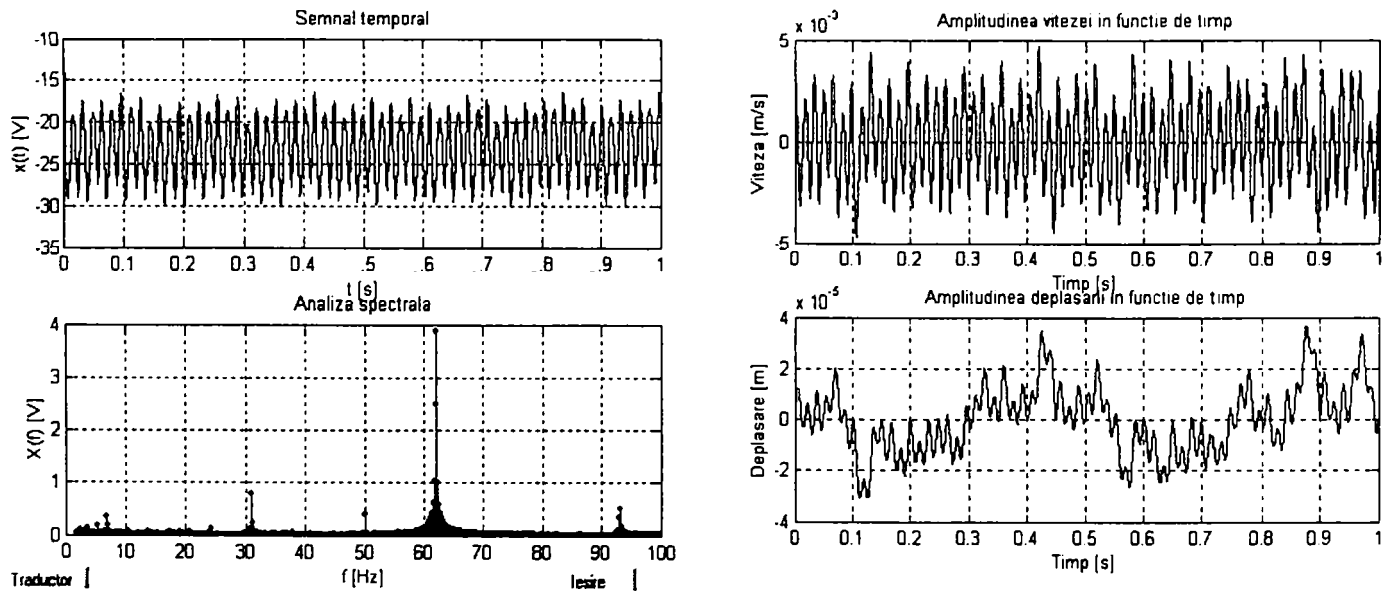
a) b)

Fig. A.8.2.5. Frecvența testată: f=23,4 Hz



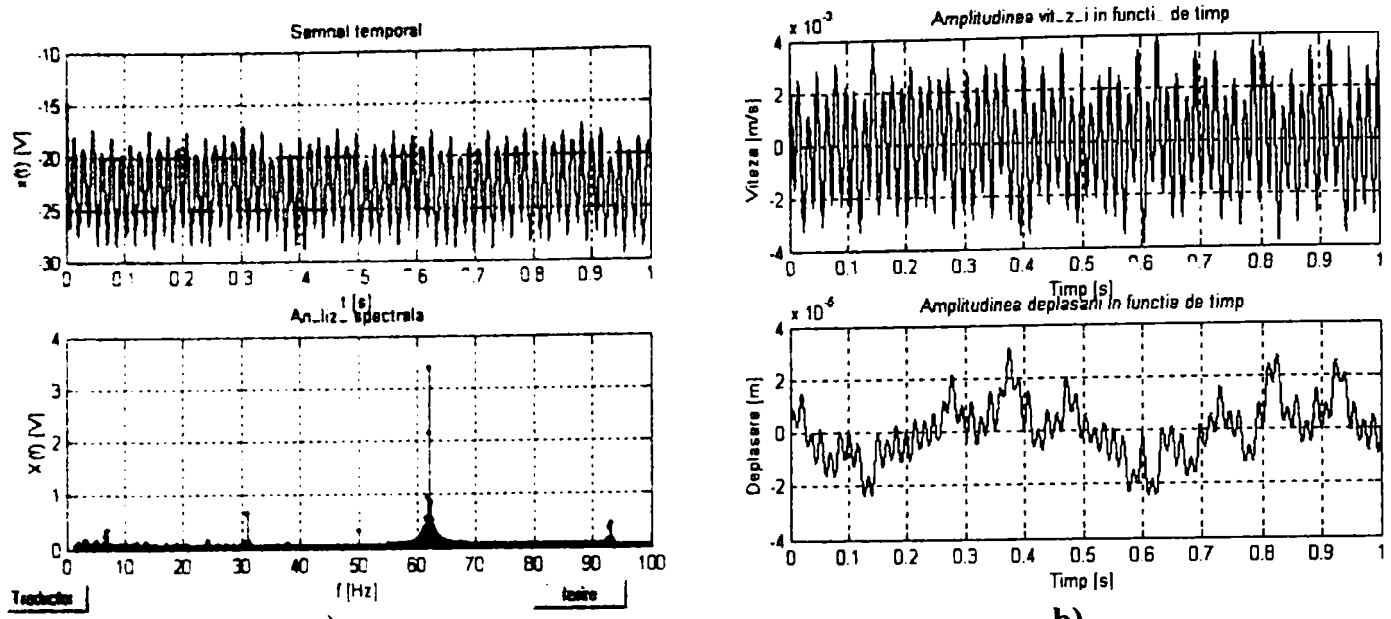
a) b)

Fig. A.8.3.1. Frecvența testată: f =31 Hz



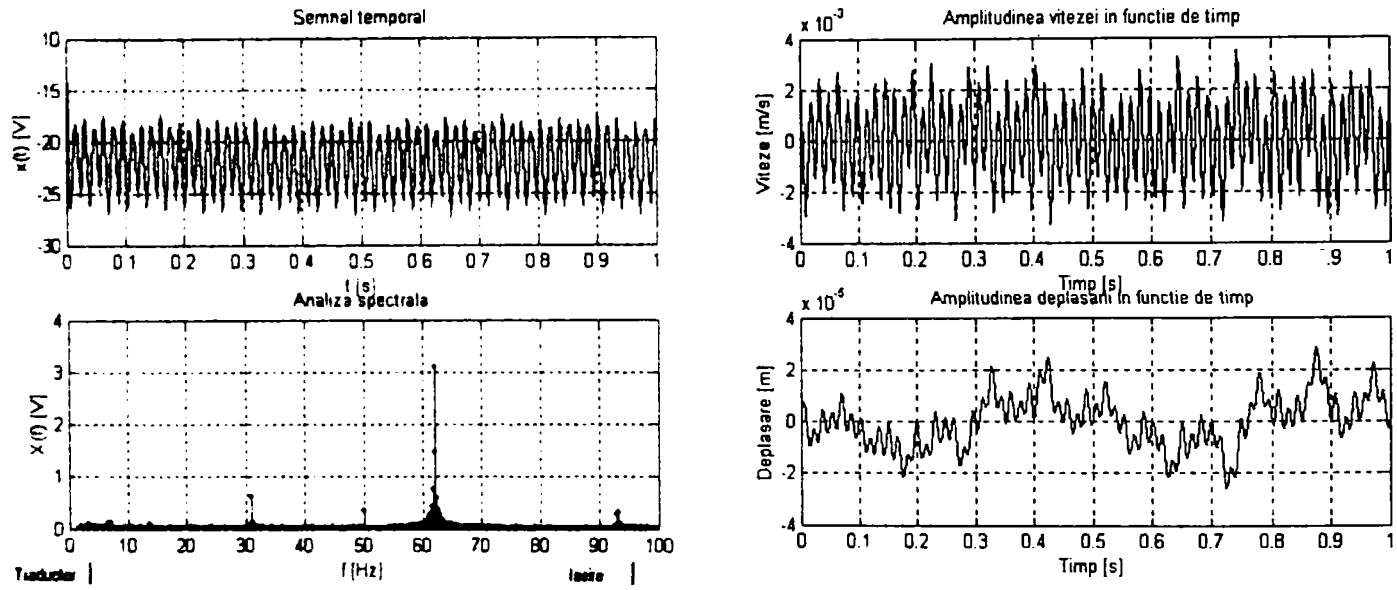
a) b)

Fig. A.8.3.2. Frecvența testată: f =31,1 Hz



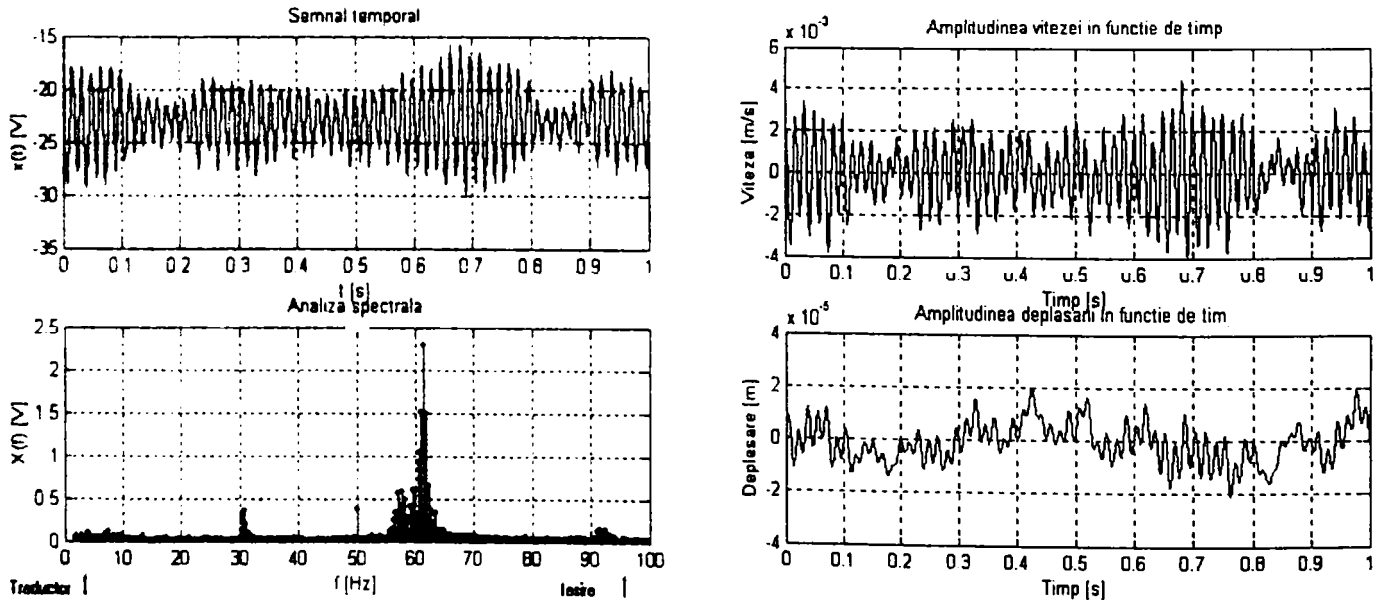
a) b)

Fig. A.8.3.3. Frecvența testată: f =31,2 Hz



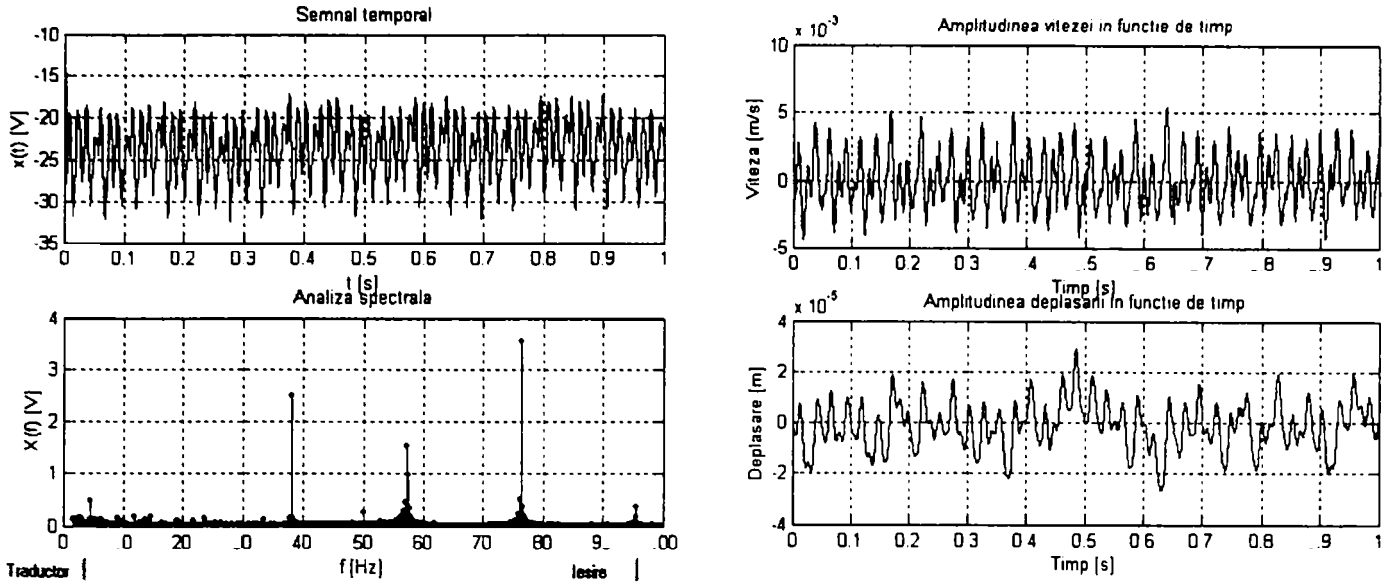
a) b)

Fig. A.8.3.4. Frecvența testată: f =31,3 Hz



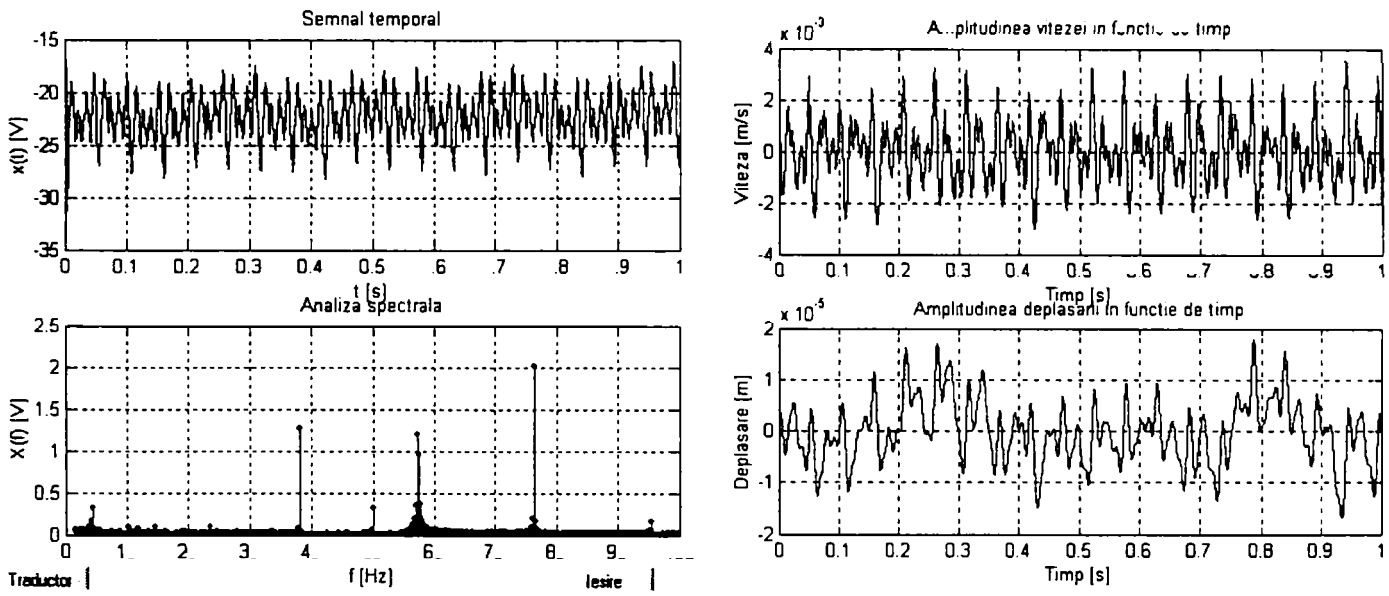
a) b)

Fig. A.8.3.5. Frecvența testată: f =31,4 Hz



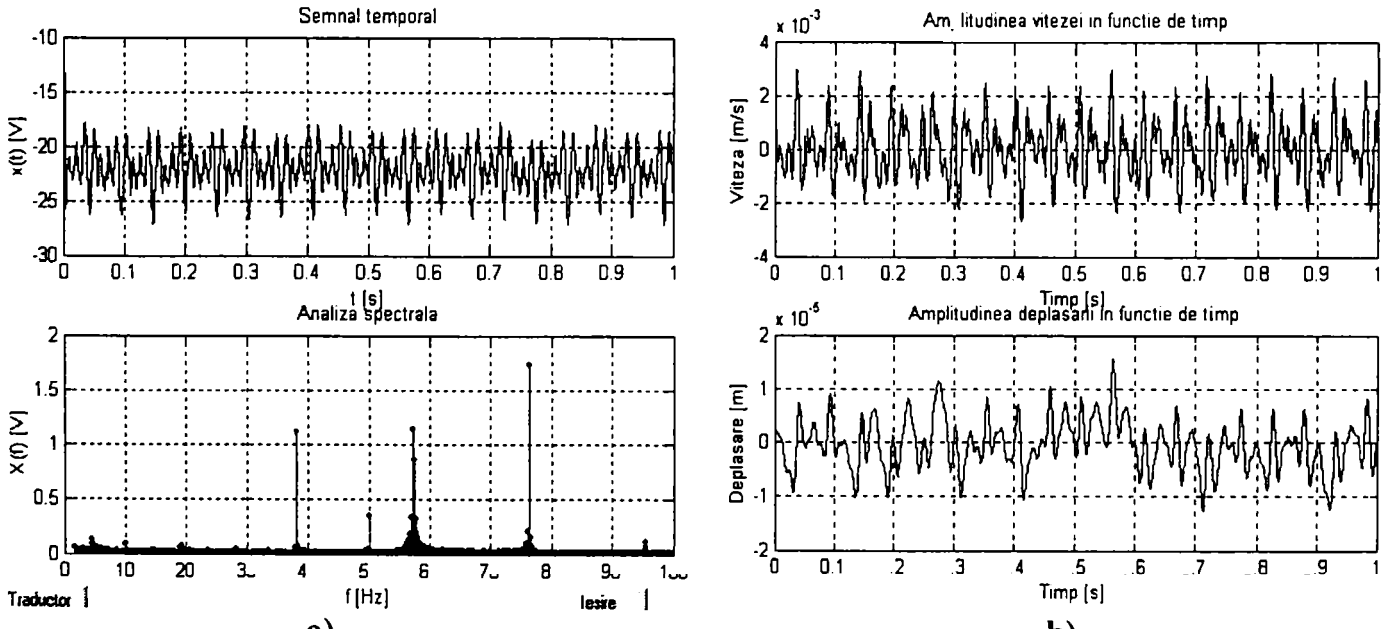
a) b)

Fig. A.8.4.1. Frecvența testată: f =39 Hz



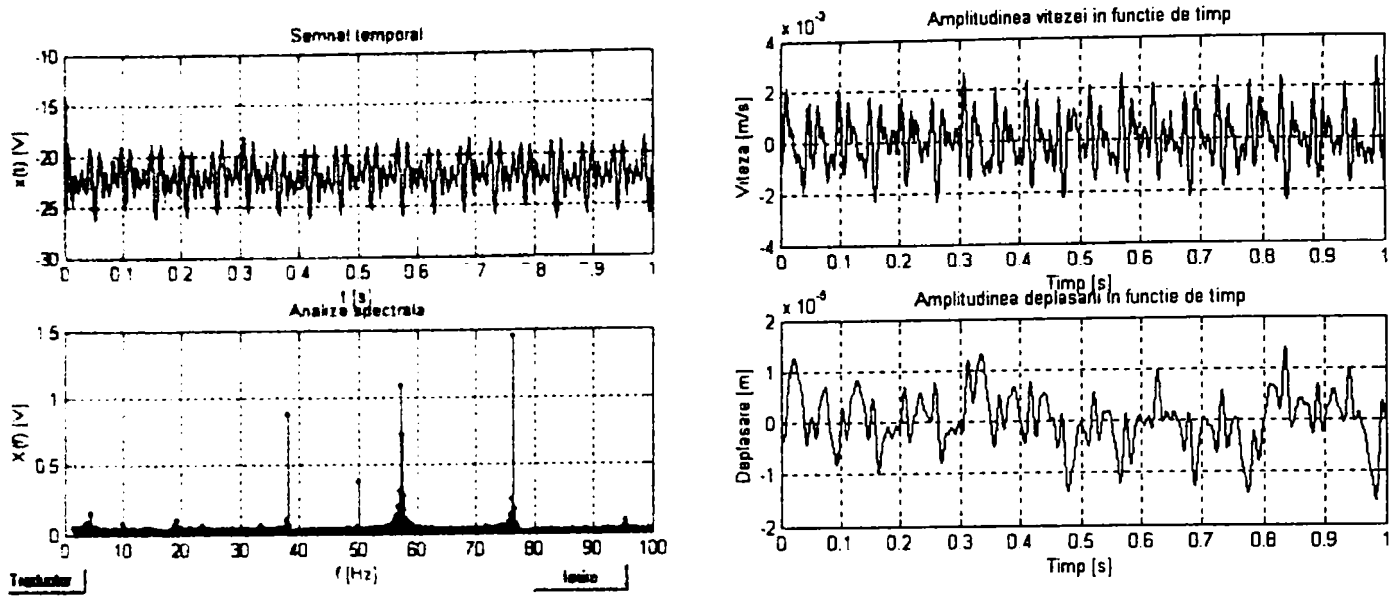
a) b)

Fig. A.8.4.2. Frecvența testată: f =39,1 Hz



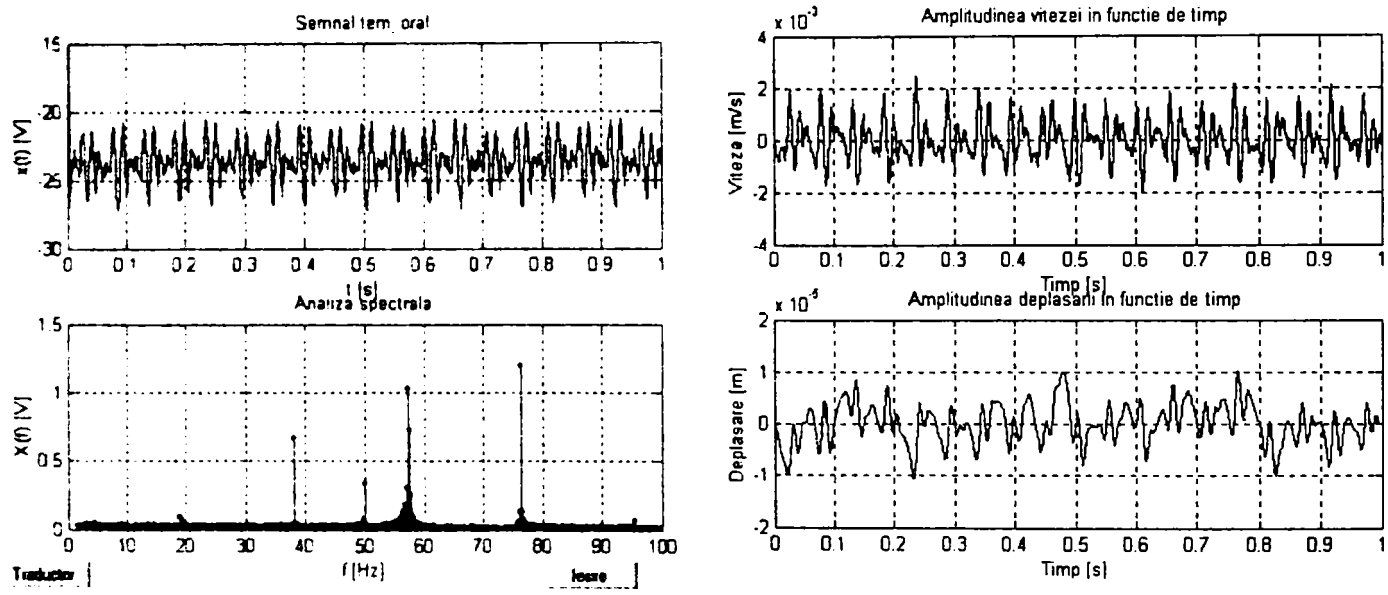
a) b)

Fig. A.8.4.3. Frecvența testată: f =39,2 Hz



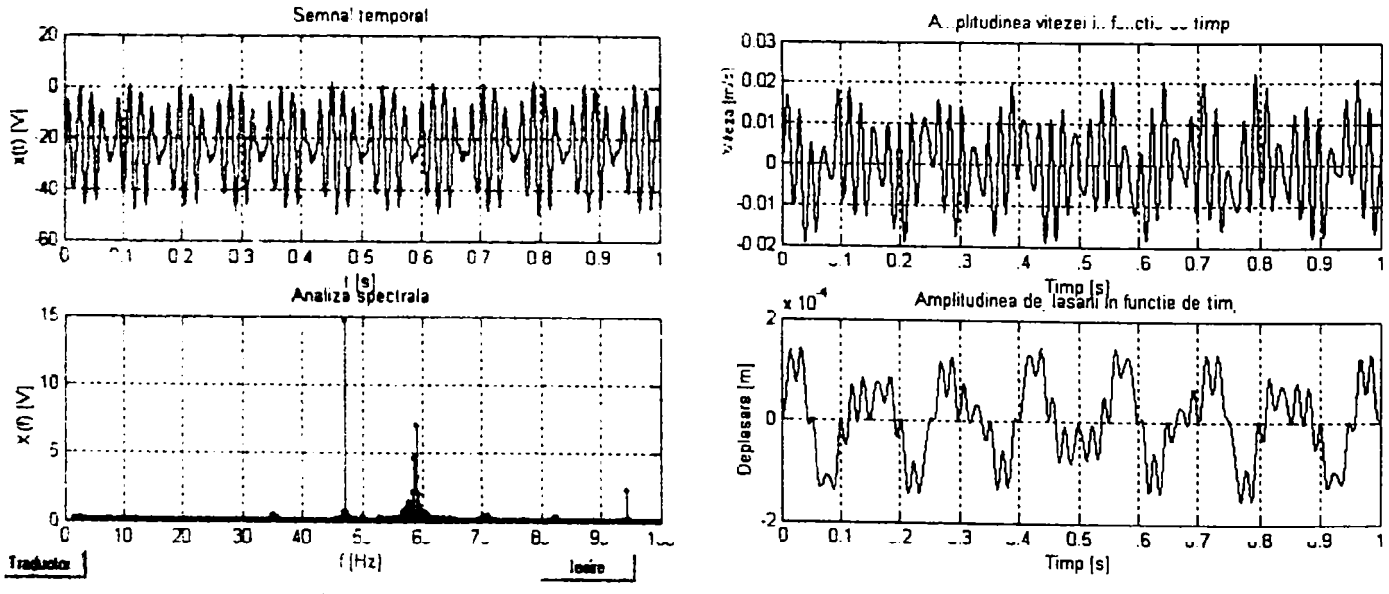
a) b)

Fig. A.8.4.4. Frecvența testată: f =39,3 Hz



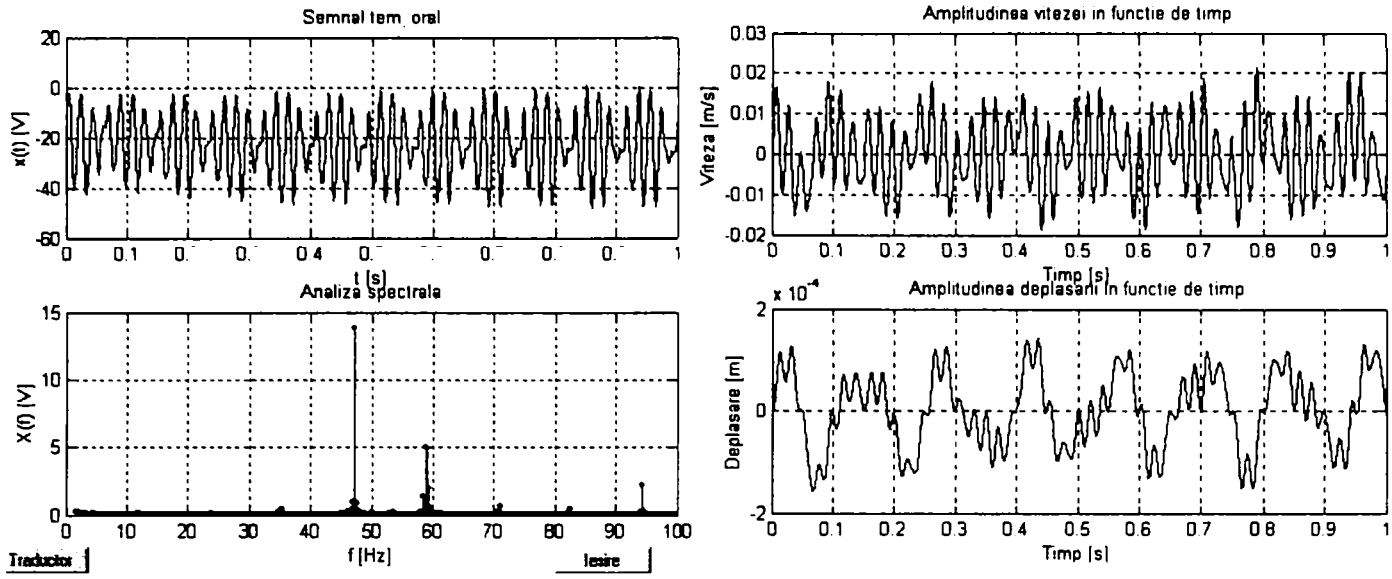
a) b)

Fig. A.8.4.5. Frecvența testată: f =39,4 Hz



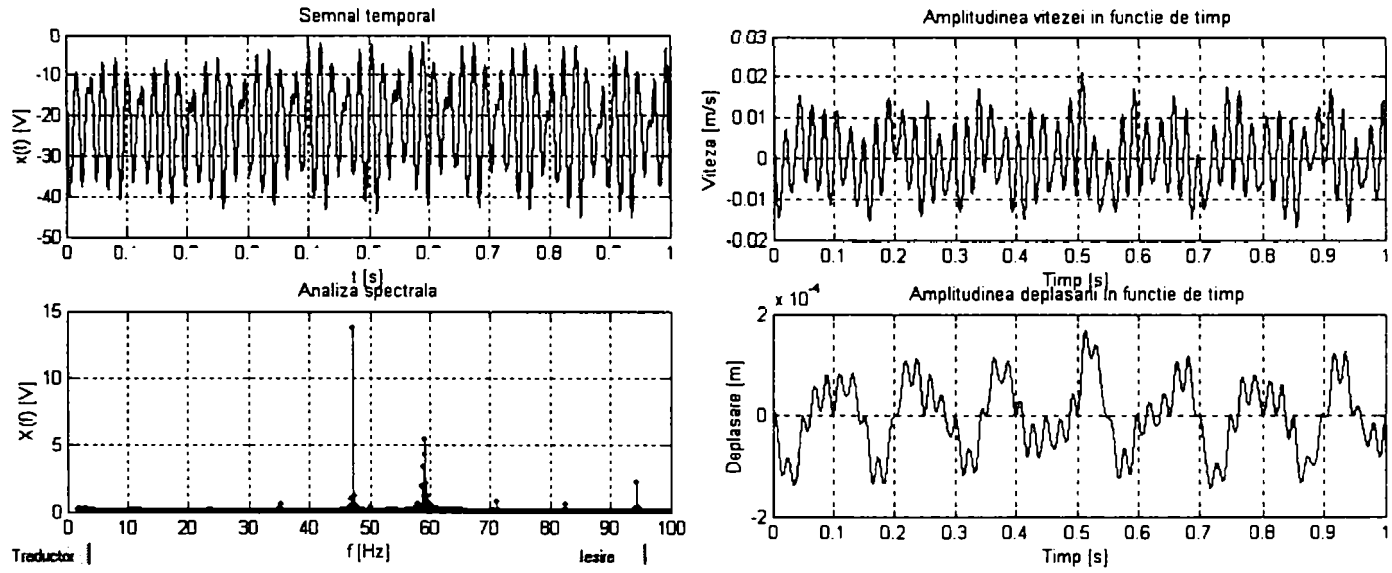
a) b)

Fig. A.8.5.1. Frecvența testată: f =47 Hz



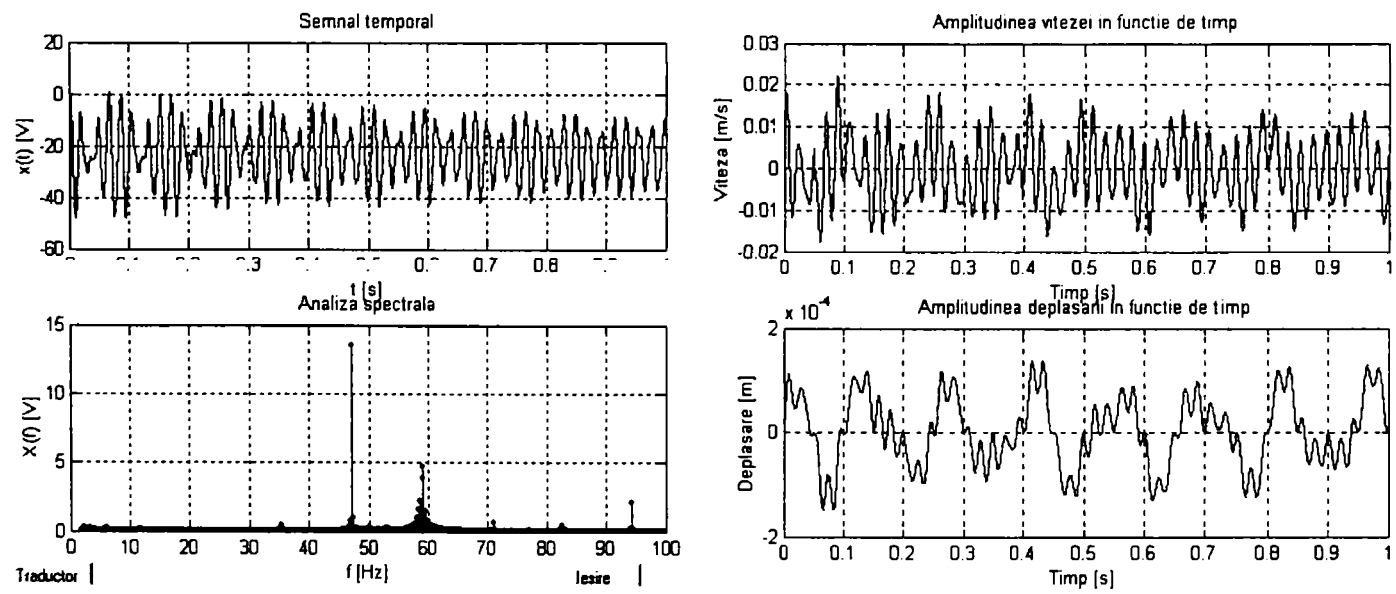
a) b)

Fig. A.8.5.2. Frecvența testată: f =47,1 Hz



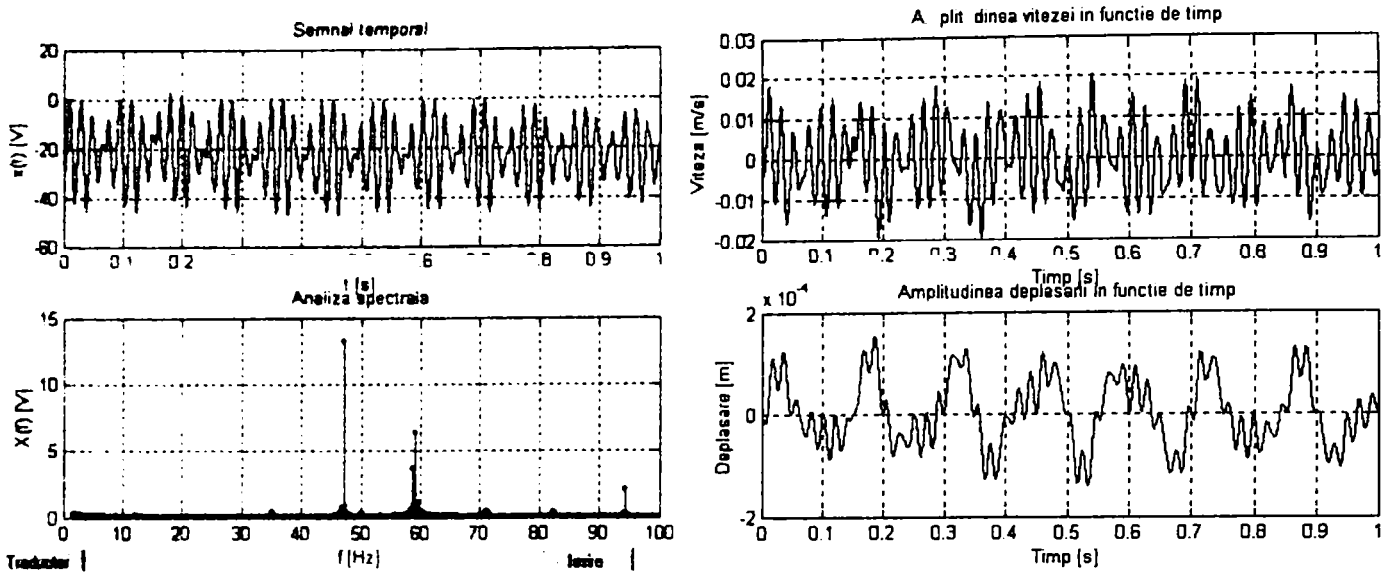
a) b)

Fig. A.8.5.3. Frecvența testată: f =47,2 Hz



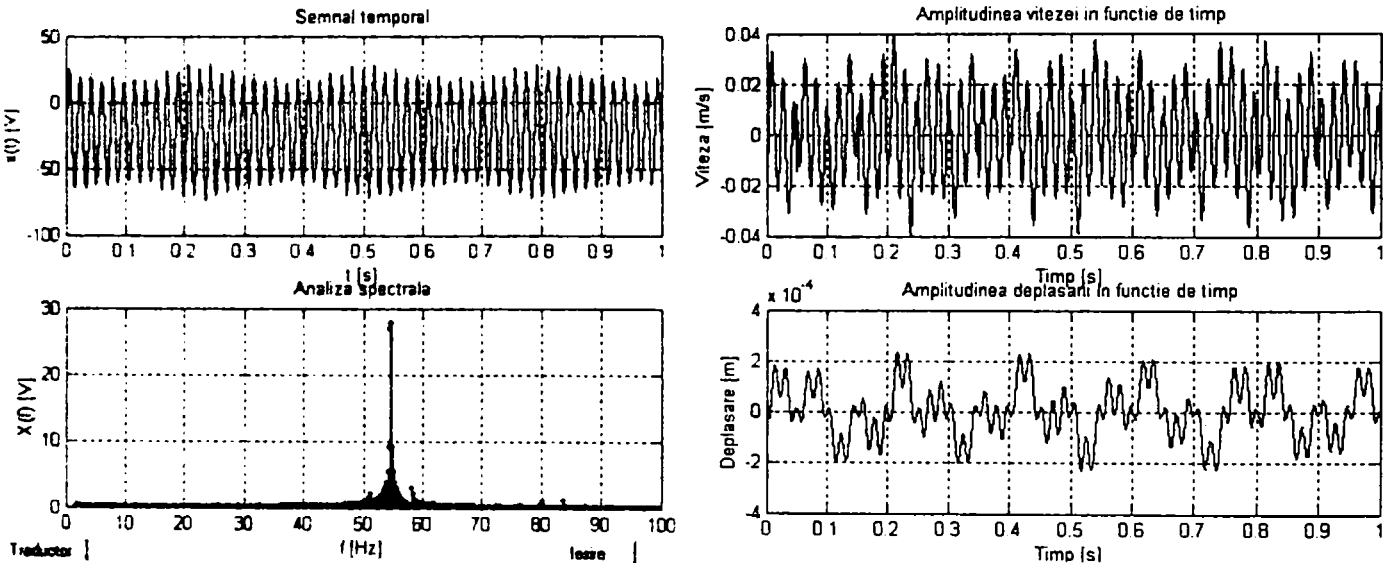
a) b)

Fig. A.8.5.4. Frecvența testată: f =47,3 Hz



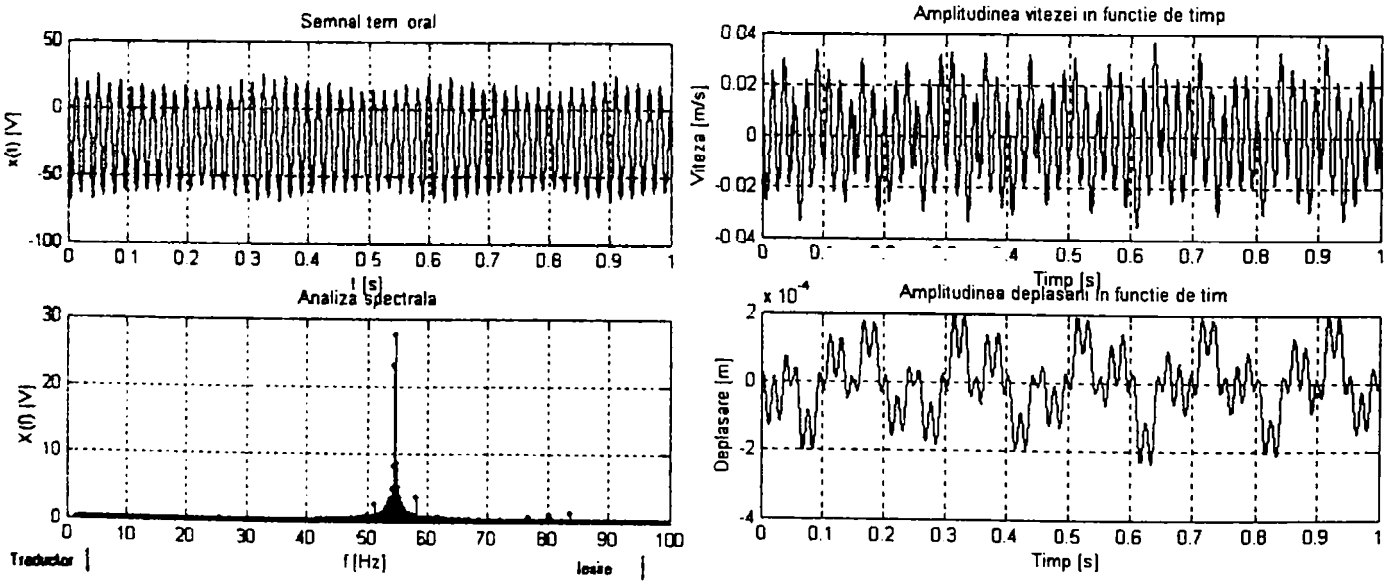
a) b)

Fig. A.8.5.5. Frecvența testată: f =47,4 Hz



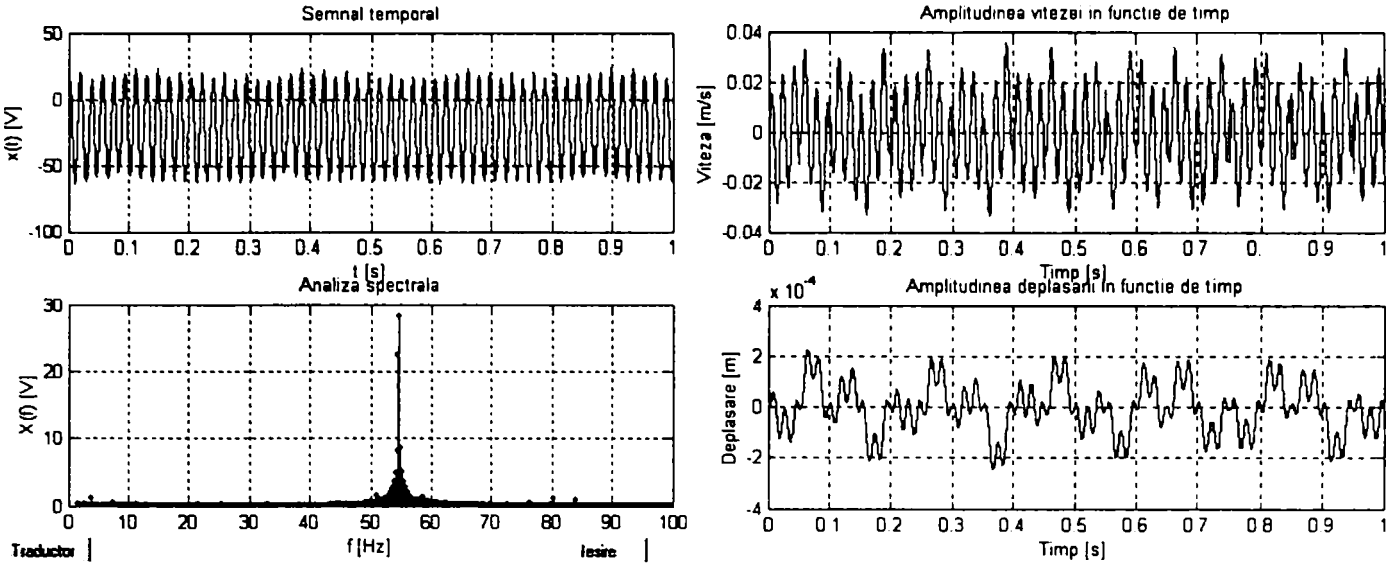
a) b)

Fig. A.8.6.1. Frecvența testată: f =55 Hz



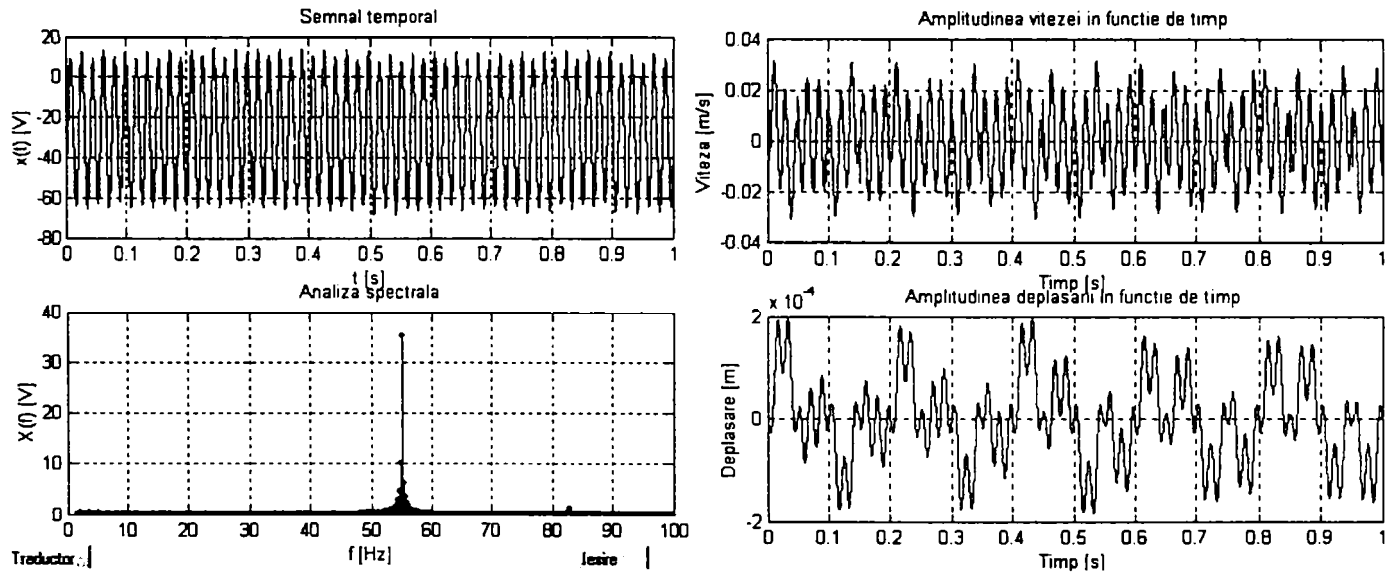
a) b)

Fig. A.8.6.2. Frecvența testată: f =55,1 Hz



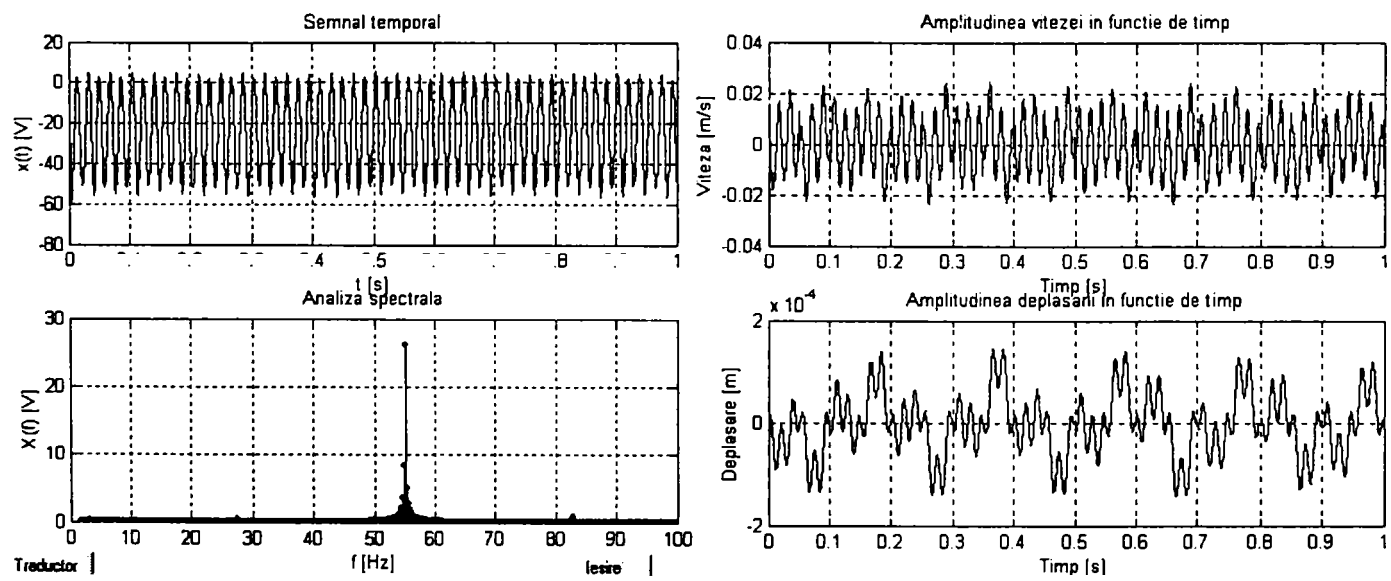
a) b)

Fig. A.8.6.3. Frecvența testată: f =55,2 Hz



a) b)

Fig. A.8.6.4. Frecvența testată: f =55,3 Hz

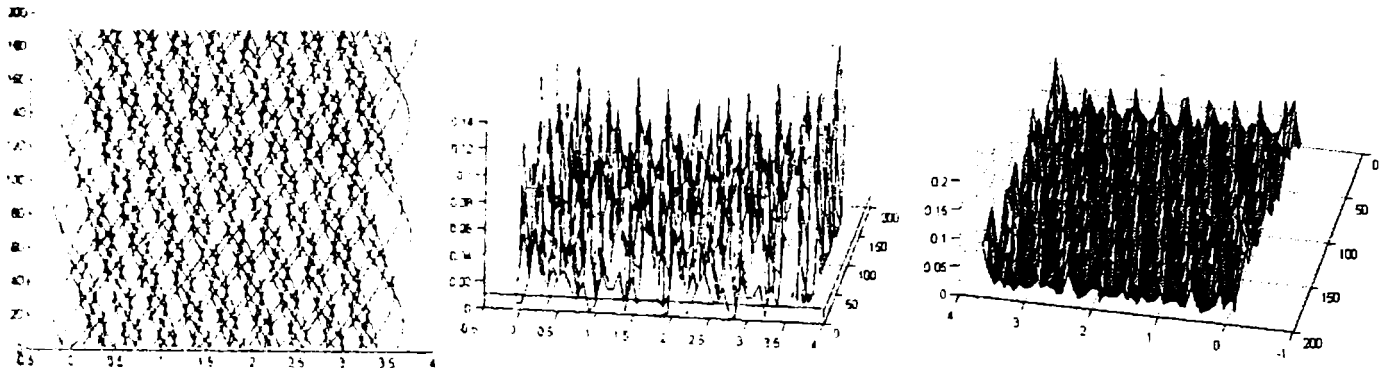


a) b)

Fig. A.8.6.5. Frecvența testată: f =55,4 Hz

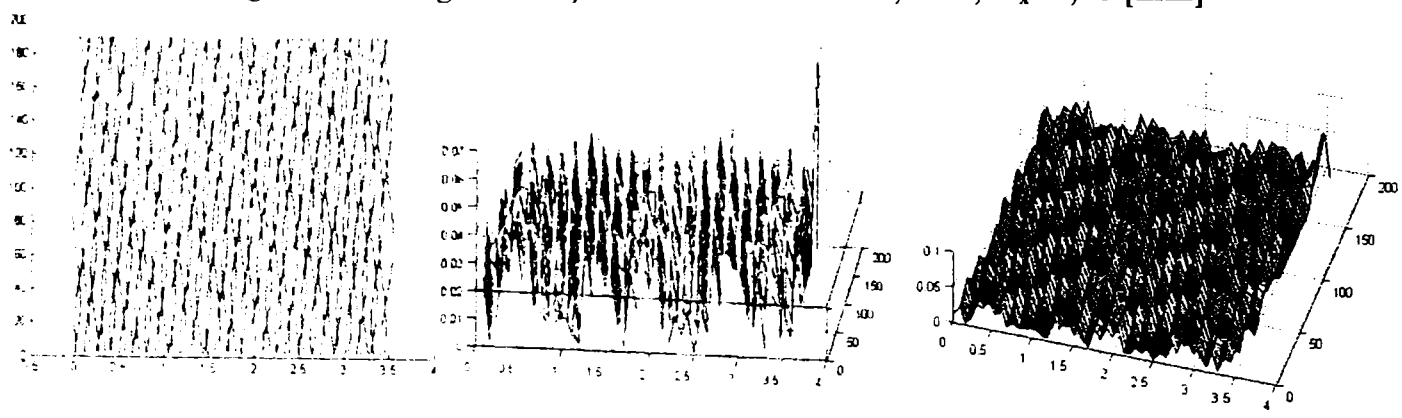
Anexa 9.

Traiectorii și rugozități simulate cu regimurile de așchiere:
s_o=0,056 [mm/rot]; n=630 [rot/min]; d=61 [mm]; $\alpha=\alpha_1=12^o$; $\gamma=8^o$; $\chi=45^o$; p=100.



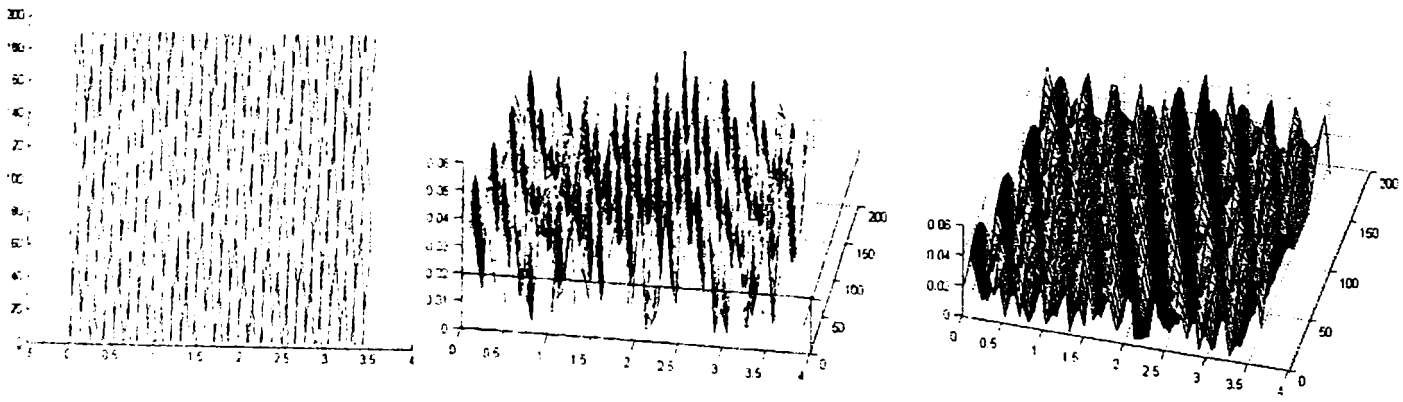
a) **b)** **c)**

Fig.A.9.1.1. Regimul mișcării vibratorii: f=16,6 Hz; A_x=0,26 [mm]



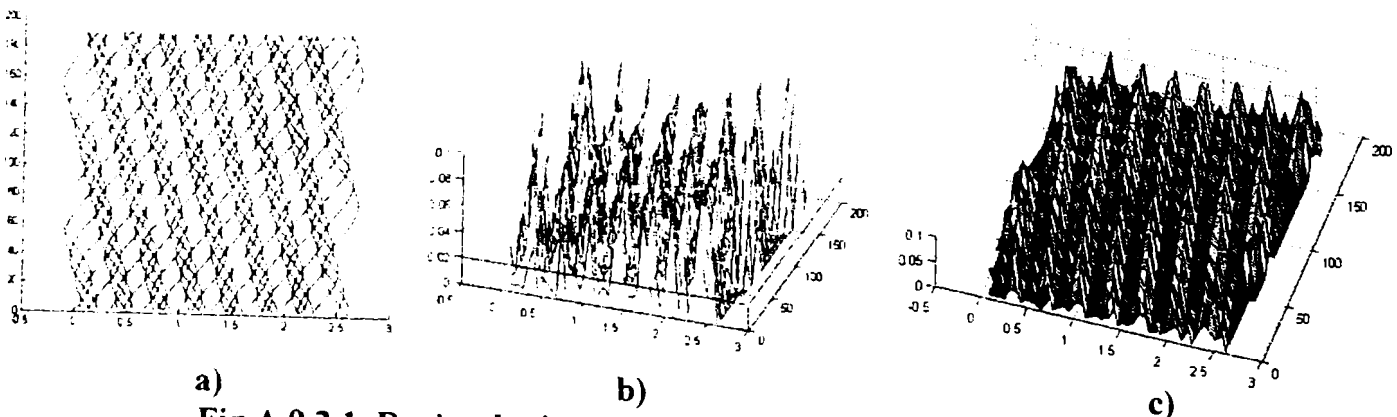
a) **b)** **c)**

Fig.A.9.1.2. Regimul mișcării vibratorii: f=16,6 Hz; A_x=0,056 [mm]



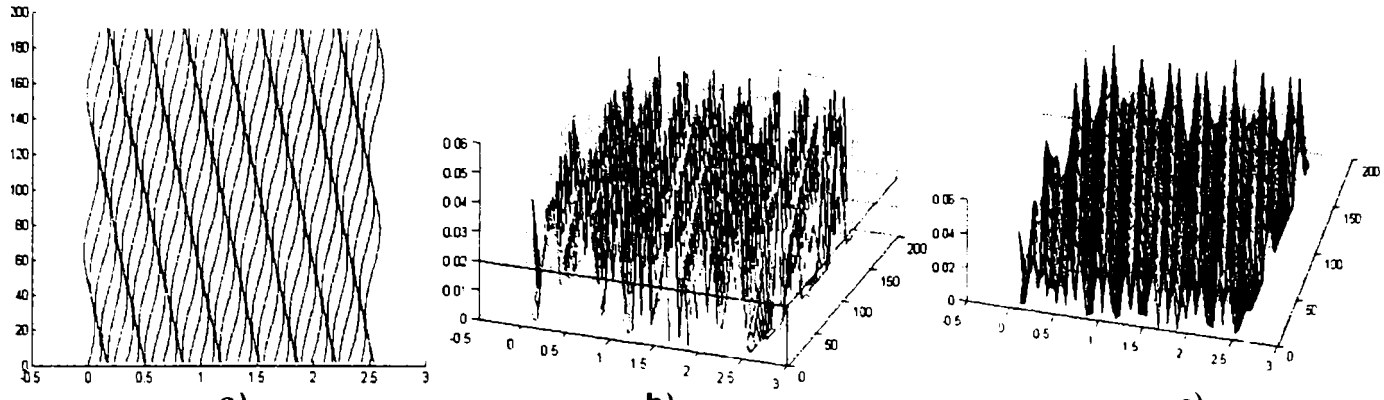
a) **b)** **c)**

Fig.A.9.1.3. Regimul mișcării vibratorii: f=16,6 Hz; A_x=0,028[mm]



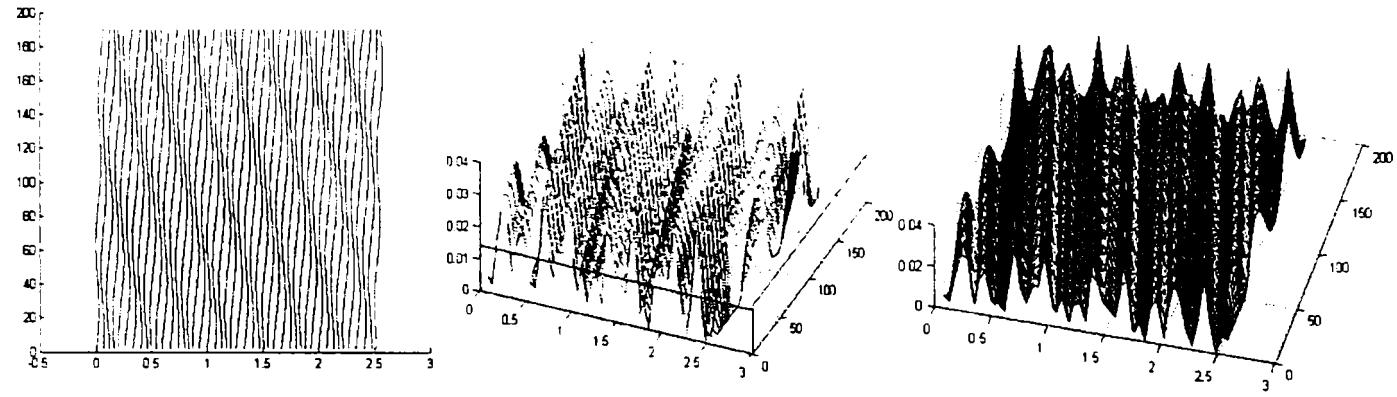
a) **b)** **c)**

Fig.A.9.2.1. Regimul mișcării vibratorii: f=22,8 Hz; A_x=0,16 [mm]



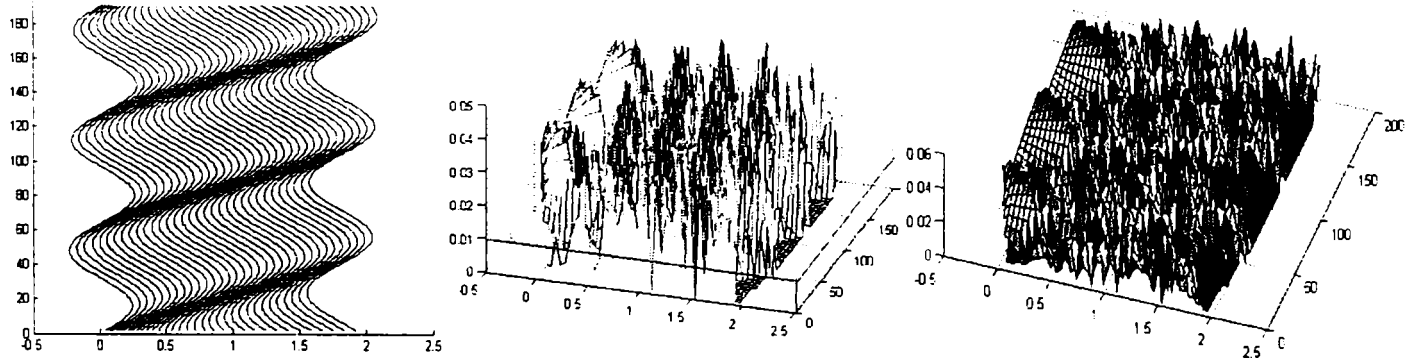
a) b) c)

Fig.A.9.2.2. Regimul mişcării vibratorii: f=22,8 Hz; A_x=0,056 [mm]



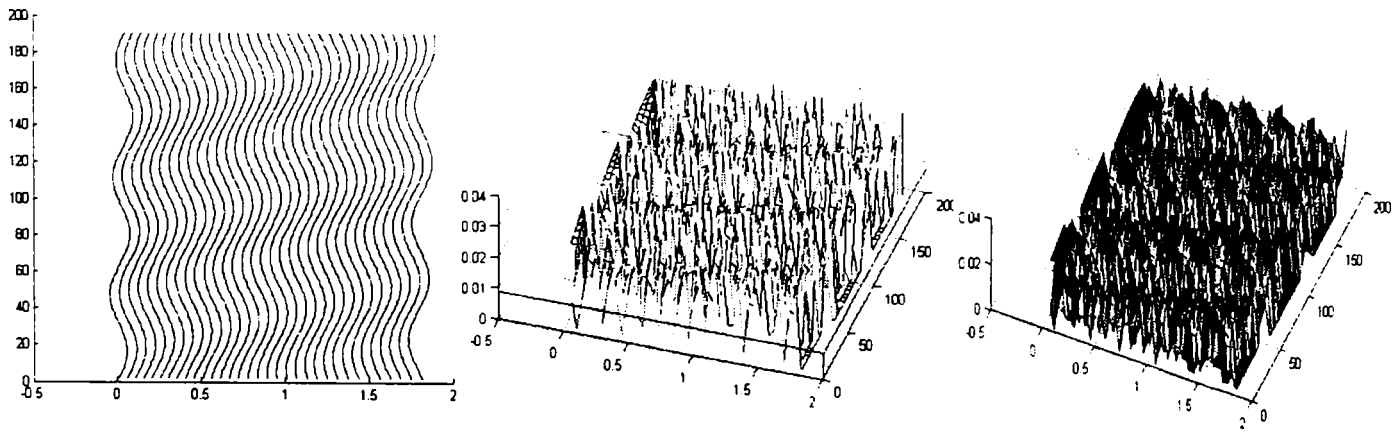
a) b) c)

Fig.A.9.2.3. Regimul mişcării vibratorii: f=22,8 Hz; A_x=0,028 [mm]



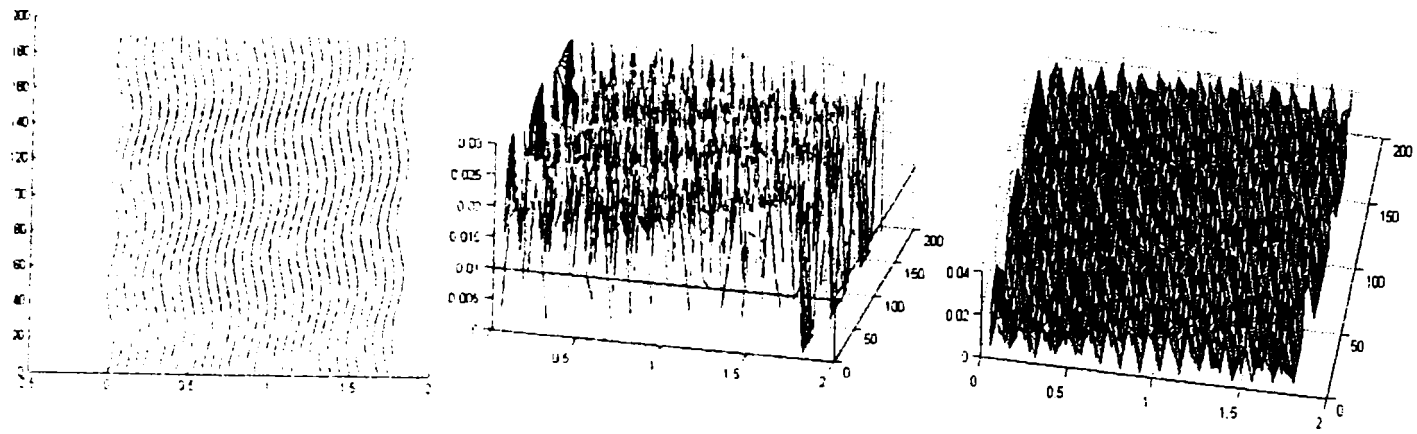
a) b) c)

Fig.A.9.3.1. Regimul mişcării vibratorii: f=31,4 Hz; A_x=0,24 [mm]



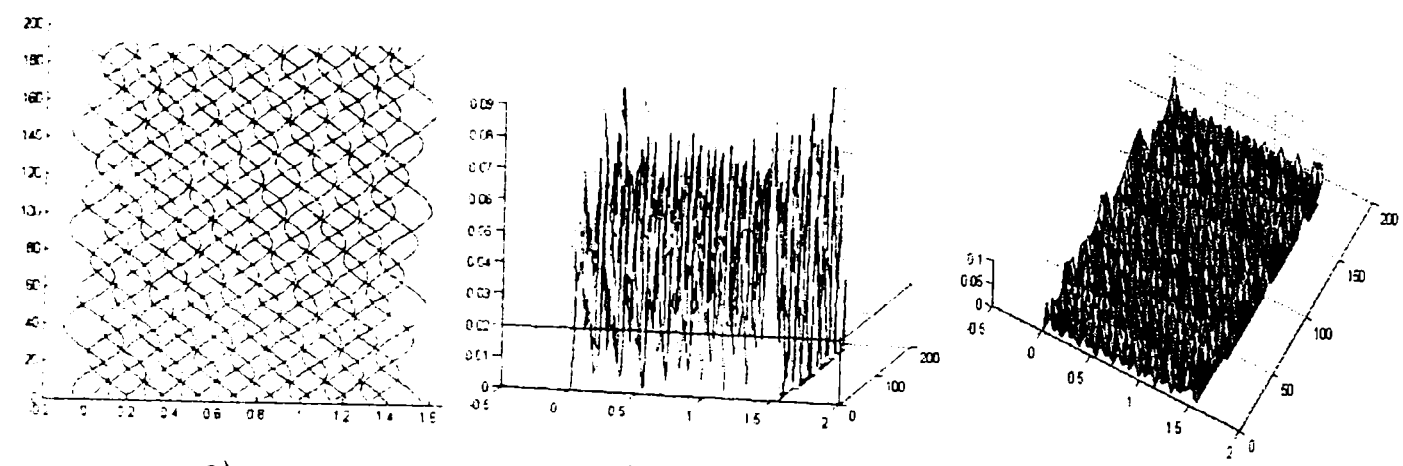
a) b) c)

Fig.A.9.3.2. Regimul mişcării vibratorii: f=31,4 Hz; A_x=0,056 [mm]



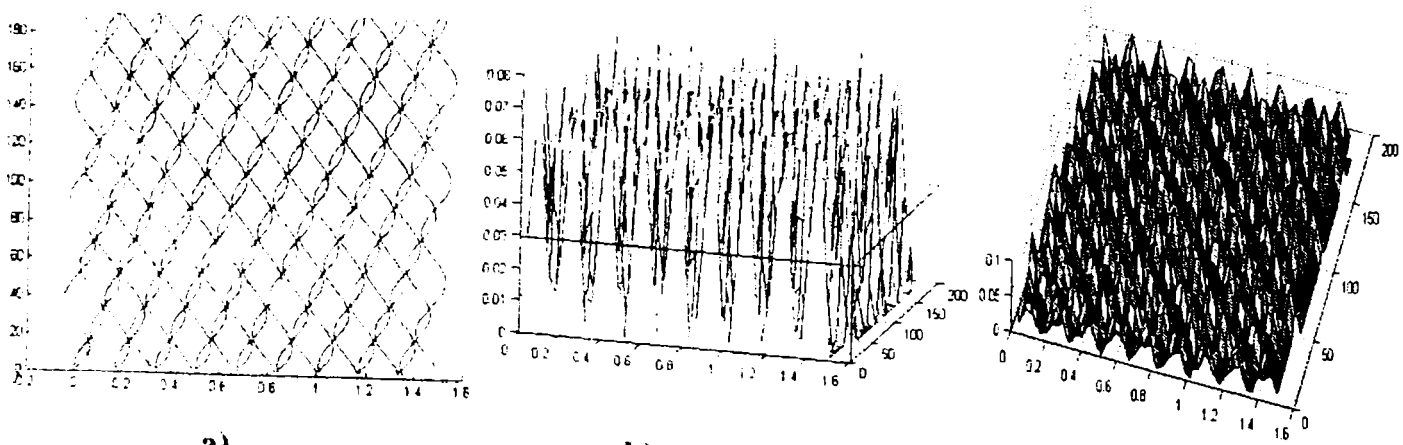
a) b) c)

Fig.A.9.3.3. Regimul mişcării vibratorii: f=31,4 Hz; A_x=0,028 [mm]



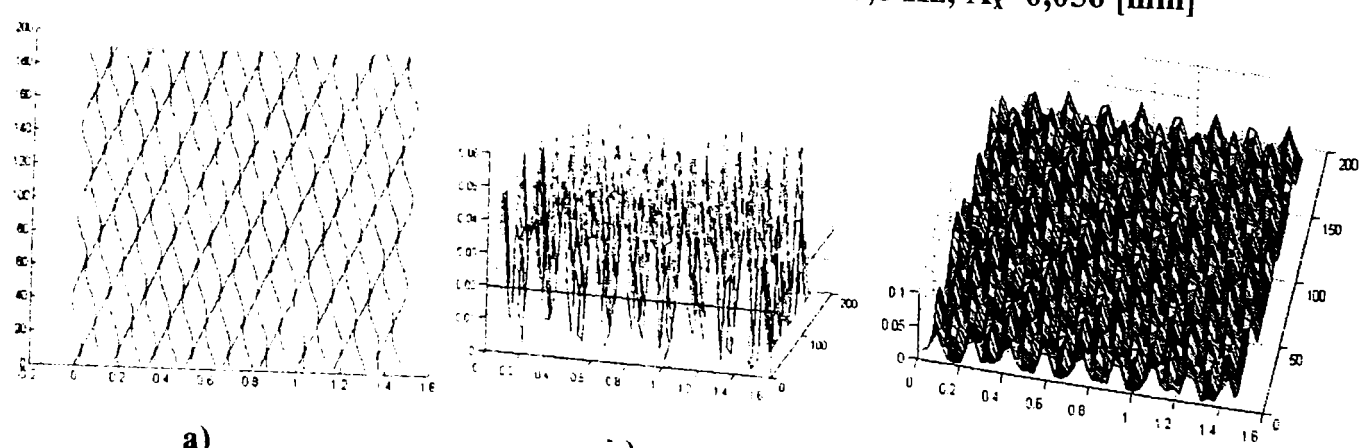
a) b) c)

Fig.A.9.4.1. Regimul mişcării vibratorii: f=38,6 Hz; A=0,14



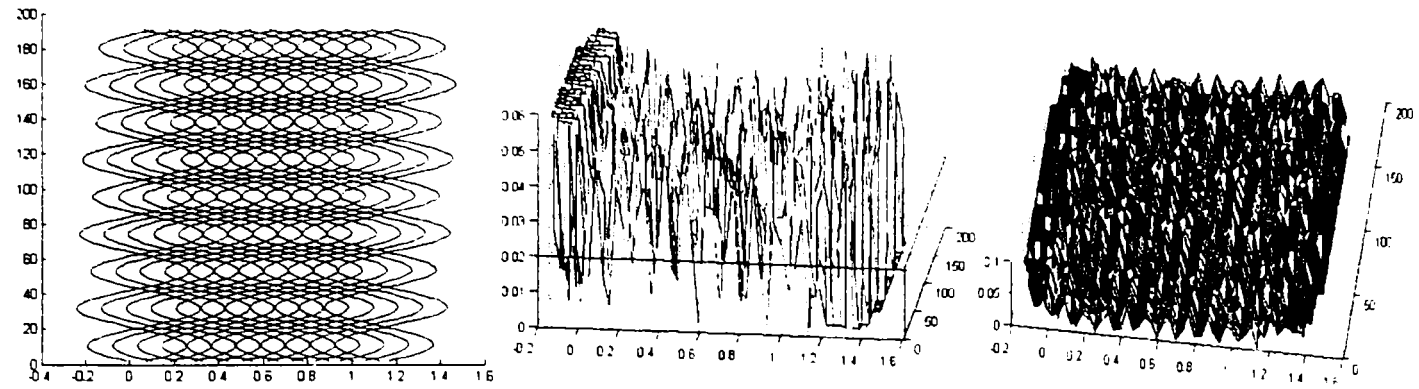
a) b) c)

Fig.A.9.4.2. Regimul mişcării vibratorii: f=38,6 Hz; A_x=0,056 [mm]



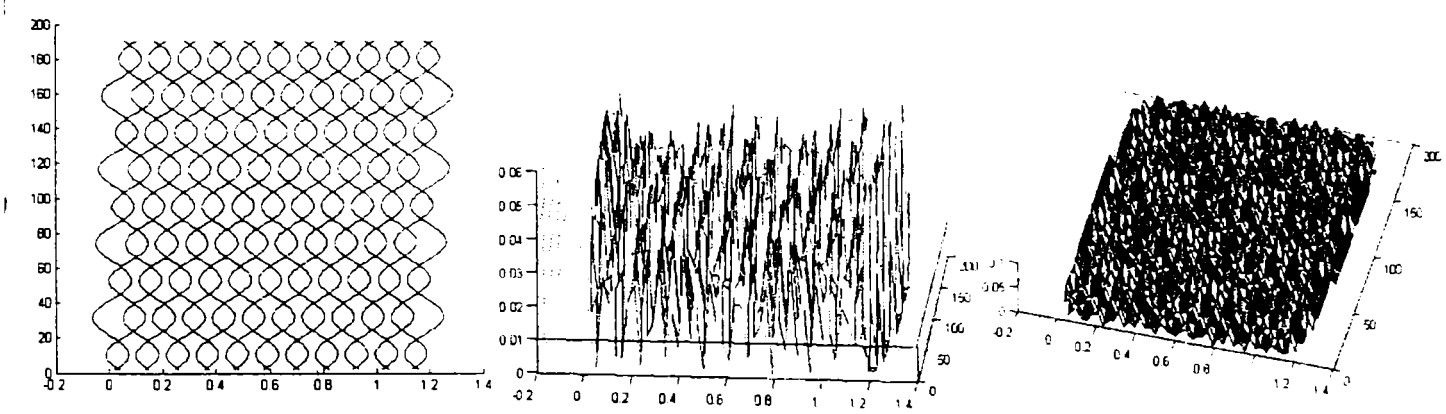
a)) c)

Fig.A.9.4.3. Regimul mişcării vibratorii: f=38,6 Hz; A_x=0,028 [mm]



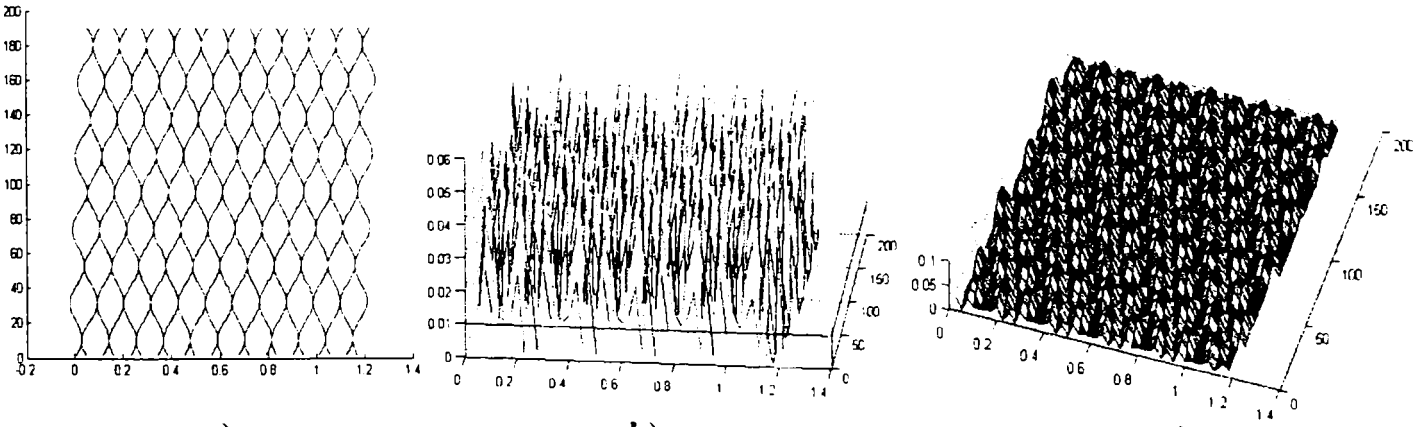
a) b) c)

Fig.A.9.5.1. Regimul mișcării vibratorii: f=47,4 Hz; A_x=0,25 [mm]



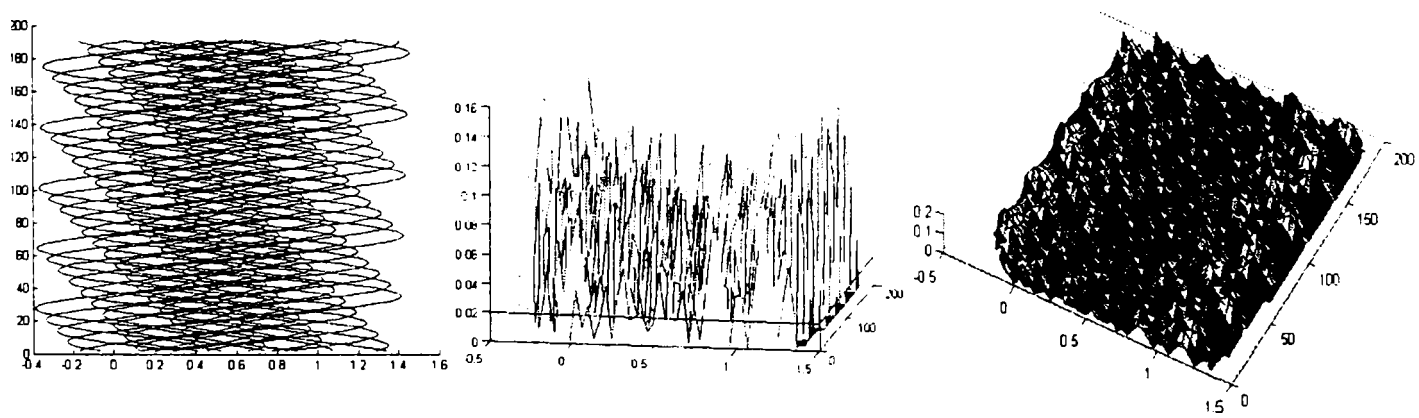
a) b) c)

Fig.A.9.5.2. Regimul mișcării vibratorii: f=47,4 Hz; A_x=0,07 [mm]



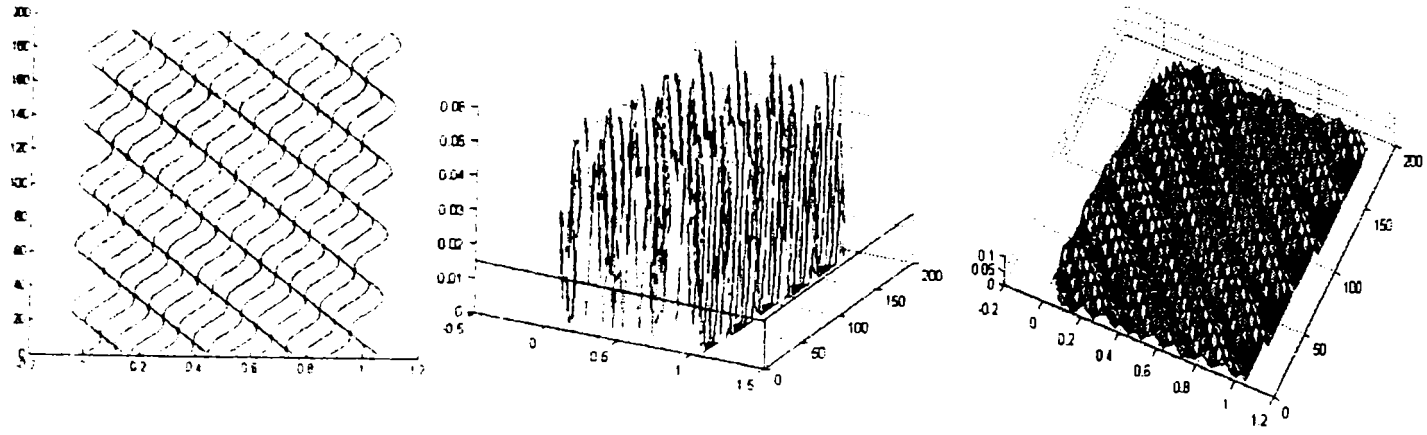
a) b) c)

Fig.A.9.5.3. Regimul mișcării vibratorii: f=47,4 Hz; A_x=0,028 [mm]



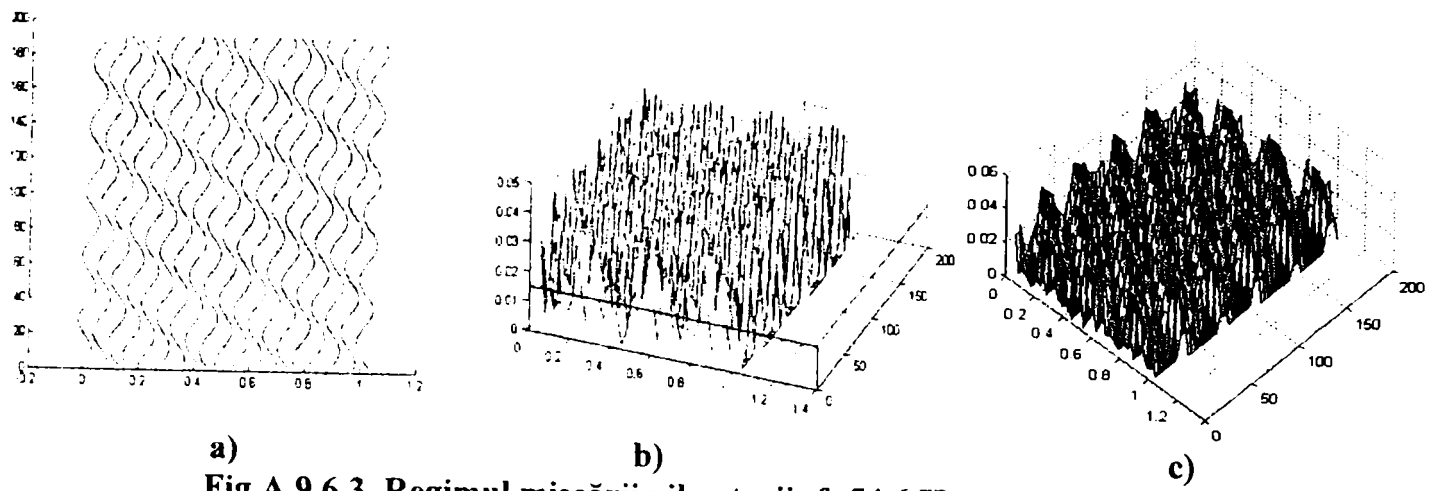
a) b) c)

Fig.A.9.6.1. Regimul mișcării vibratorii: f=54,6 Hz; A_x=0,4 [mm]



a) b) c)

Fig.A.9.6.2. Regimul mișcării vibratorii: f=54,6 Hz; A_x=0,056 [mm]



a) b) c)

Fig.A.9.6.3. Regimul mișcării vibratorii: f=54,6 Hz; A_x=0,028 [mm]

Notă:

a) – reprezintă traiectorii simulate ale vârfului cuțitului;
b) – reprezintă rugozitatea simulată cu funcția „mesh";
c) – reprezintă rugozitatea simulată cu funcția „surf".

Anexa 10.

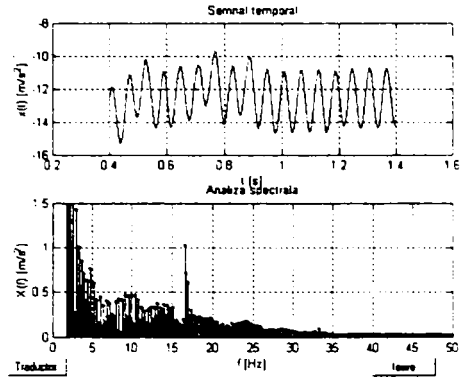
Programul „an_spec 1" pentru calculul spectrului de frecvențe

```
% Initializare variabile
% Definire suprafata grafica utilizator : "an_sp"

f40_1_1 = figure('Name','Analiza spectrala a semnalelor achizitionate',....
  'NumberTitle','off');
%'MenuBar','none',

% Grafic
f40_1_4  = uicontrol('Style', 'pushbutton', 'String', 'Traductor',...
  'Position', [20 5 70 20], 'Callback', 'an_spec3');

% Iesire

f40_1_6  = uicontrol('Style', 'pushbutton', 'String', 'Iesire',...
  'Position', [420 5 70 20], 'Callback', 'delete(f40_1_1)');

% analiza spectrala a unui semnal
% Traductor 2
[filename,pathname] = uigetfile('*.*','Selectati fisierul de date',300,100);
if filename ~=0
  fid = fopen(strcat(pathname,filename),'r');
  [x1,count1] = fread(fid,inf,'int16');
  fclose('all');
  an_spec5;
elseif filename == 0
  disp('Nu s-a selectat nici un fisier');
end

  ratach = 20000;
  limmin = 10;
  limmax = 100000;
  %   limmax = count1;
  limfmin = 10;
  limfmax = 500;
  x1 = x1'*0.1999;
  x2 = x1(limmin:limmax);
  lim_filt = 50;
  [B,A] = butter(5,lim_filt/20000);
  x = filter(B,A,x2);
  %x = x2; % Aceasta instructiune anuleaza filtraraea - daca dorim filtrare o anulam
  count = count1;
        fe = ratach; %frecventa de esantionare
  t = 0:1/fe:1/fe*count;  %momentele de esantionare
        Xt = fft(x); % transformata Fourier
  N = length(x); % lungimea secventei
        Xm = abs(Xt); % modulul semnalului
  X = Xm(1,1:N/2+1)/(N/2);
  f = (0:N/2)*fe/N; %frecventele pozitive
```

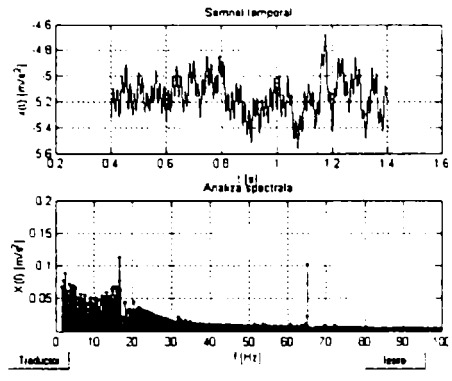
```
f_rep = f(limfmin:limfmax);
X_rep = X(limfmin:limfmax);
t_rep = t(8000:28000);
x_rep = x(8000:28000);
      subplot(211);
      plot(t_rep,x_rep,'k')
      grid
      xlabel('t [s]');
      ylabel('x(t) [m/s^2]');
      title('Semnal temporal');
      subplot(212);
      stem(f_rep,X_rep,'k.'); %reprezentare valori discrete
xlabel('f [Hz]');
      ylabel('X(f) [m/s^2]');
      grid
      title('Analiza spectrala');
```

Anexa 11.

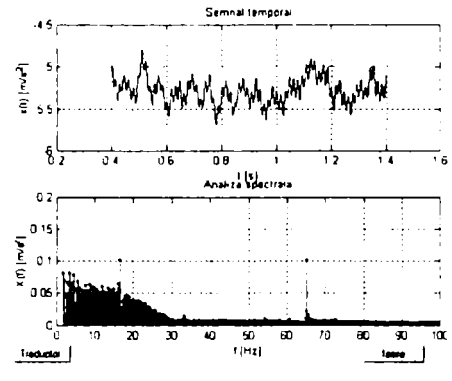
Grafice acceleraţii, spectre de frecvenţe, viteze şi deplasări înregistrate la strunjirea cu vibraţii a oţelului Armco.



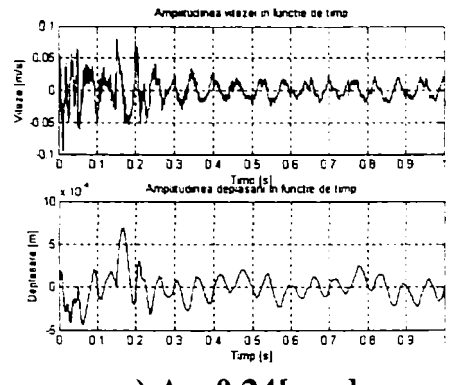
a) A_x=0,24[mm]



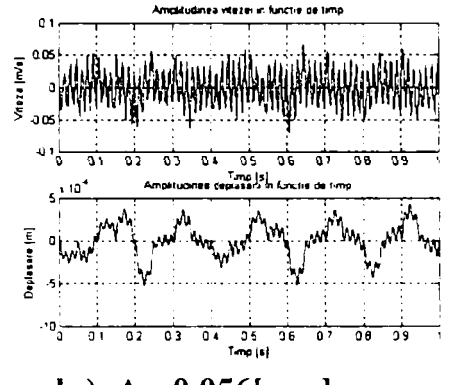
b). A_x=0,056[mm]



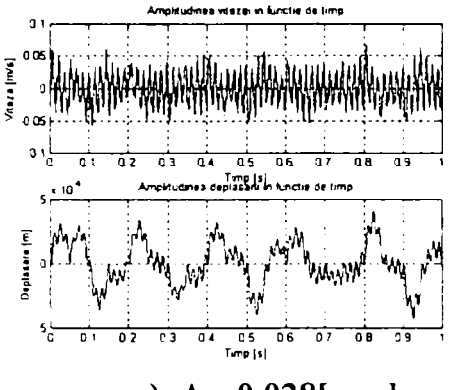
c)A_x=0,028[mm]



a_1).A_x=0,24[mm]

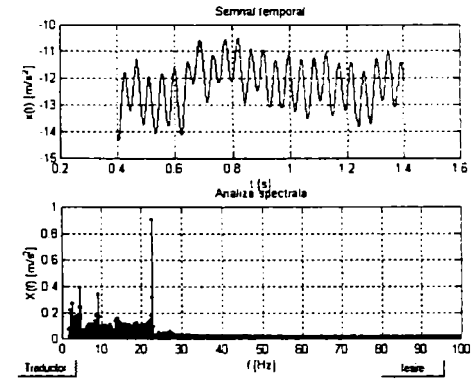


b_1). A_x=0,056[mm]

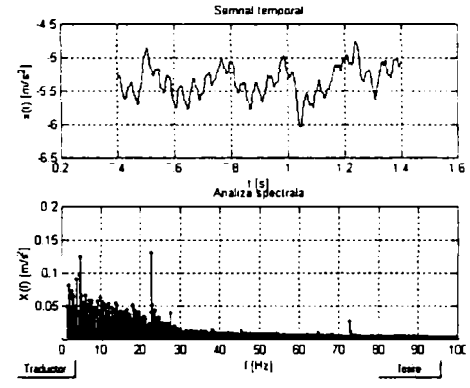


c_1). A_x=0,028[mm]

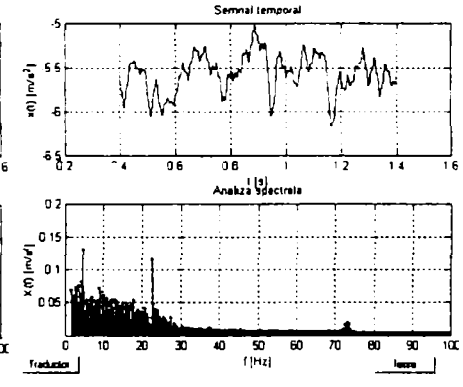
Fig.A.11.1. Acceleraţii, viteze şi deplasări măsurate la amplitudinile A_x şi frecvenţa f=16,6 Hz.



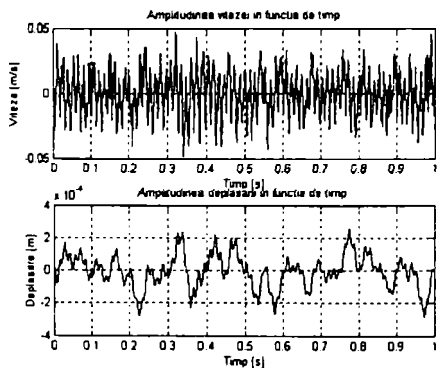
a) A_x=0,16 [mm]



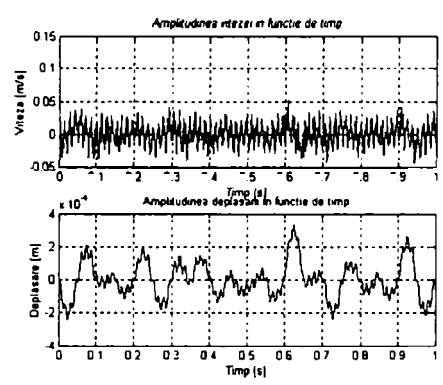
b) A_x=0,056 [mm]



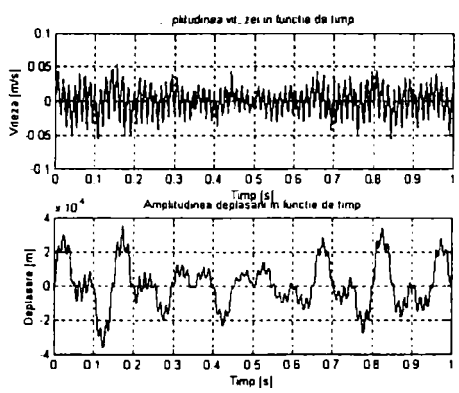
c) A_x=0,028 [mm]



a_1) A_x=0,16 [mm]

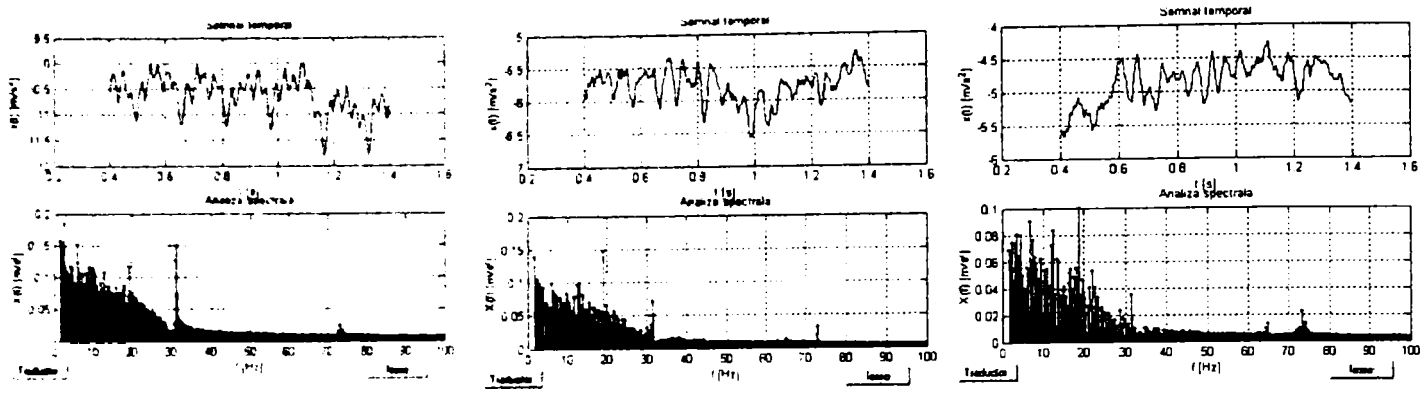


b_1) A_x=0,056 [mm]

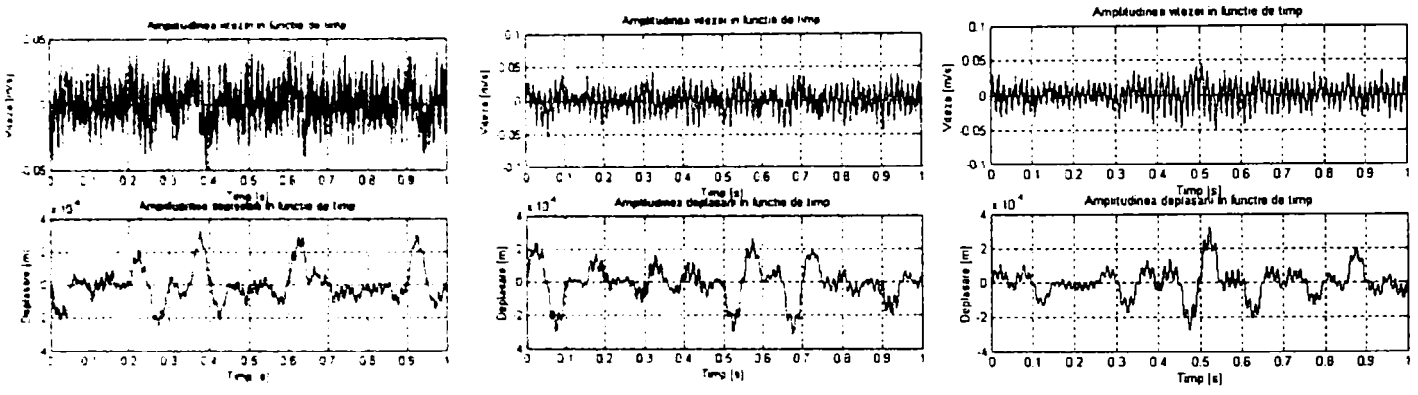


c_1) A_x=0,028 [mm]

Fig.A.11.2. Acceleraţii, viteze şi deplasări măsurate la amplitudinile A_x şi frecvenţa f=22,8 Hz.

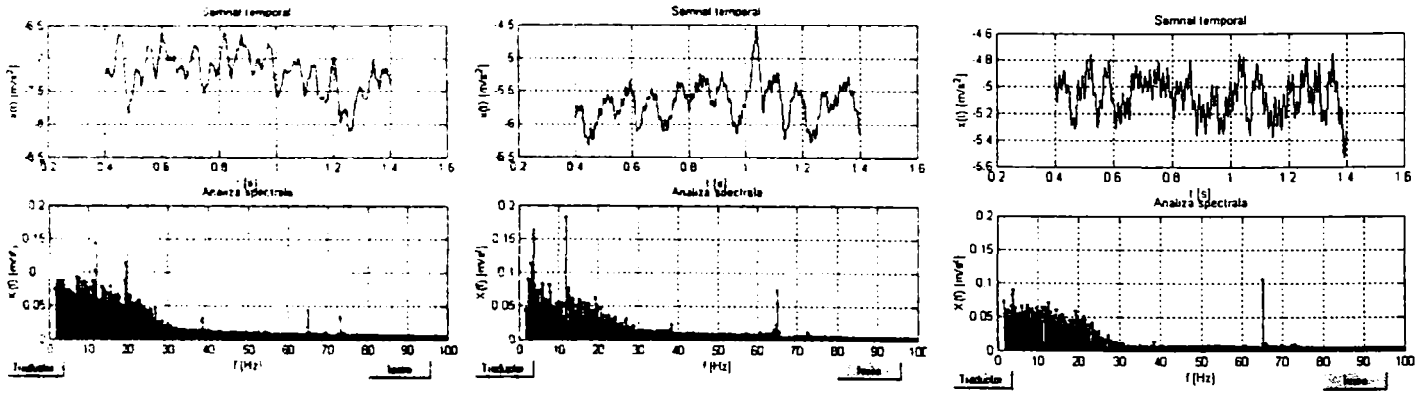


a) A_x=0,24 [mm] **b) A_x=0,056 [mm]** **c) A_x=0,028 [mm]**

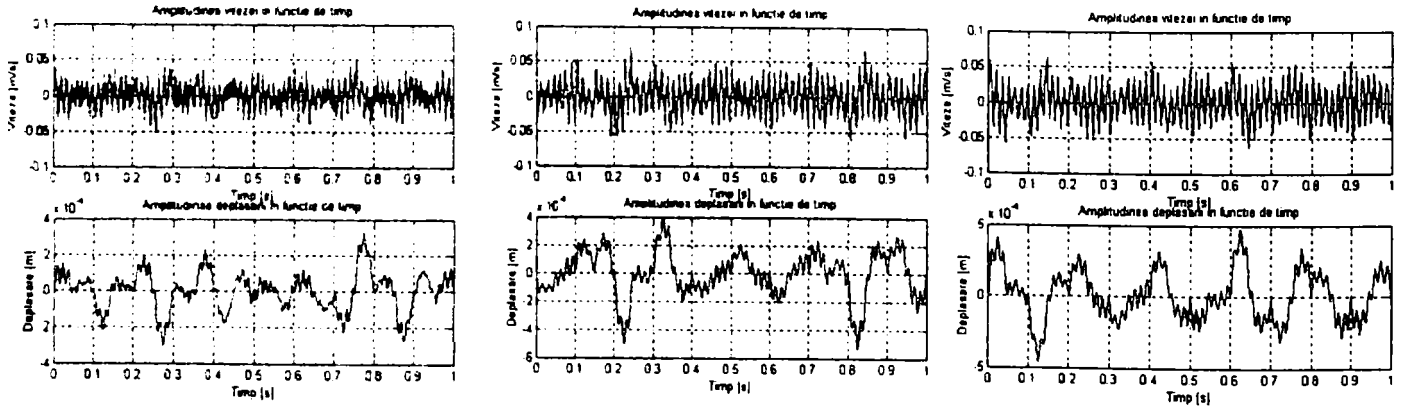


a_1) A_x=0,24 [mm] **b_1) A_x=0,056 [mm]** **c_1) A_x=0,028 [mm]**

Fig.A.11.3. Accelerații, viteze și deplasări măsurate la amplitudinile A_x și frecvența f=31,4 Hz

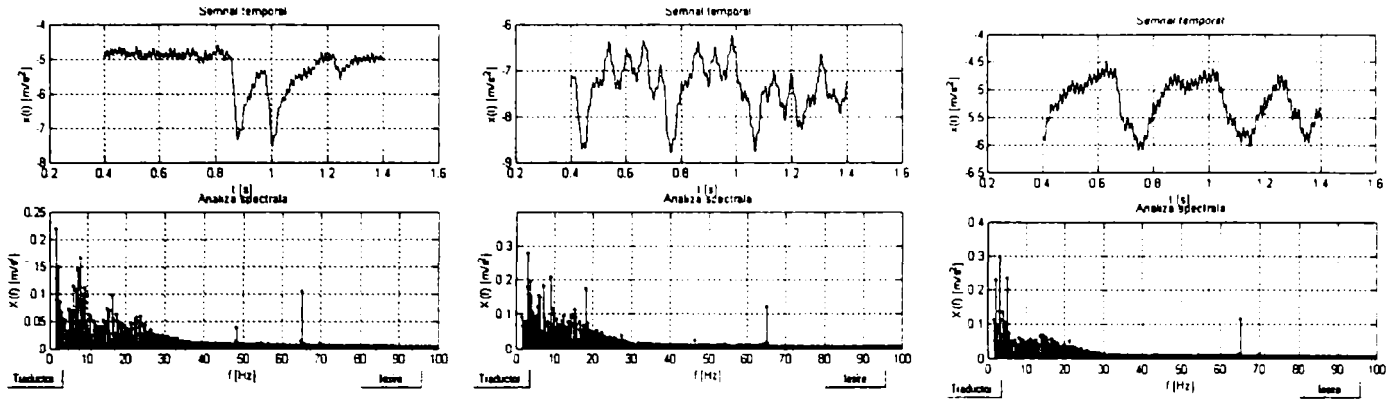


a) A_x=0,14 [mm] **b) A_x=0,056 [mm]** **c) A_x=0,028 [mm]**

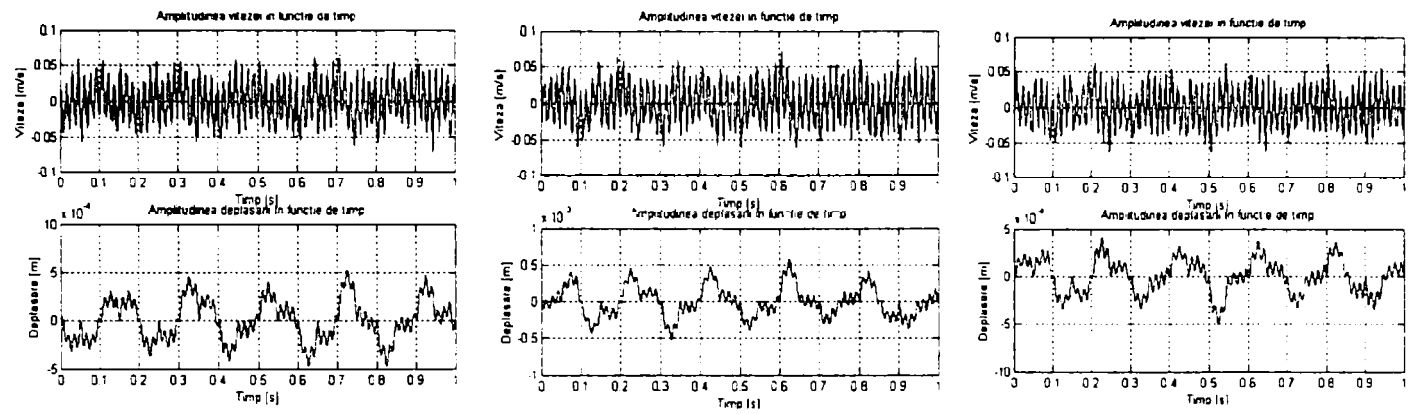


a_1) A_x=0,14 [mm] **b_1) A_x=0,056 [mm]** **c_1) A_x=0,028 [mm]**

Fig.A.11.4. Accelerații, viteze și deplasări măsurate la amplitudinile A_x și frecvența f=38,6 Hz

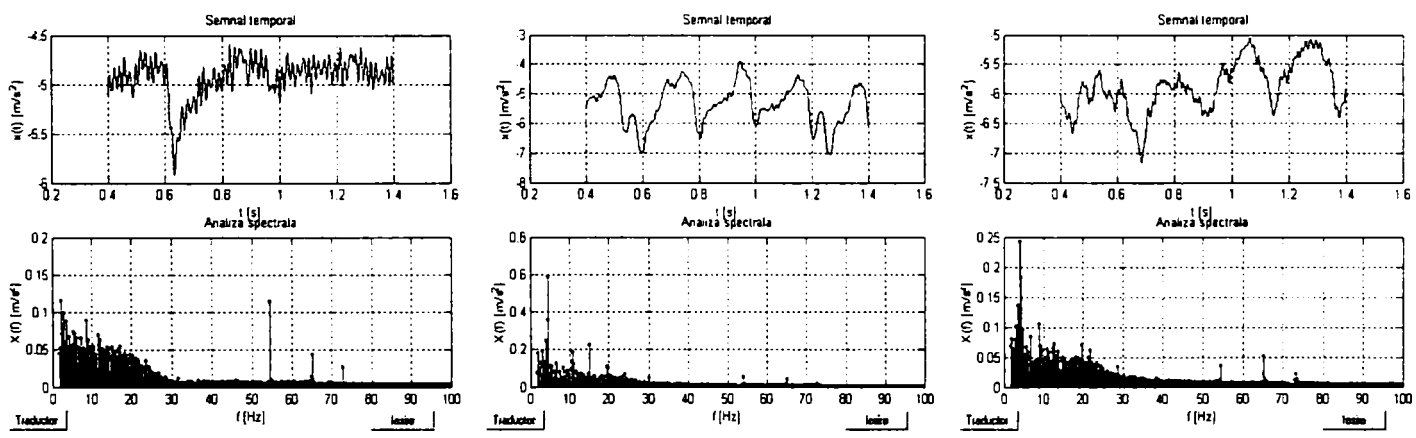


a) A_x=0,25 [mm] **b) A_x=0,07 [mm]** **c) A_x=0,028 [mm]**

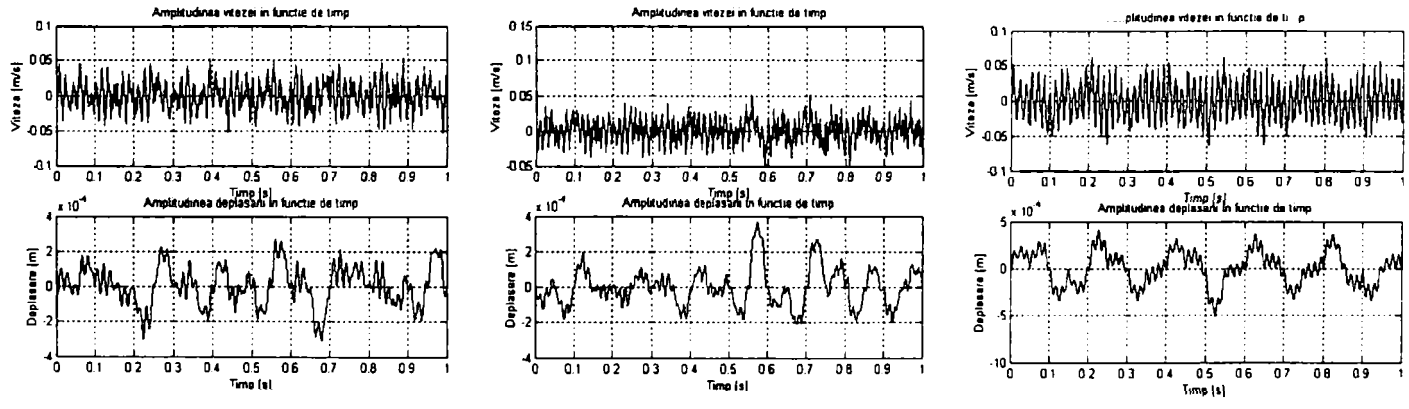


a_1) A_x=0,25 [mm] **b_1) A_x=0,07 [mm]** **c_1) A_x=0,028 [mm]**

Fig.A.11.5. Accelerații, viteze şi deplasări măsurate la amplitudinile A_x şi frecvența f=47,4 Hz

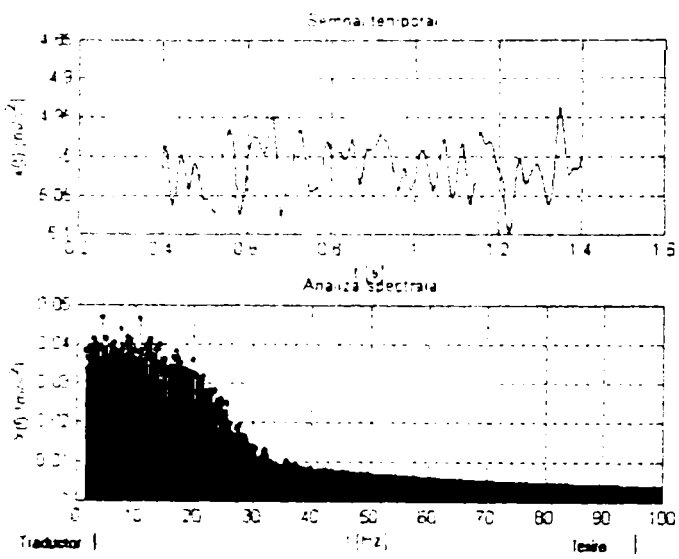


a) A_x=0,4 [mm] **b) A_x=0,056 [mm]** **c) A_x=0,028 [mm]**

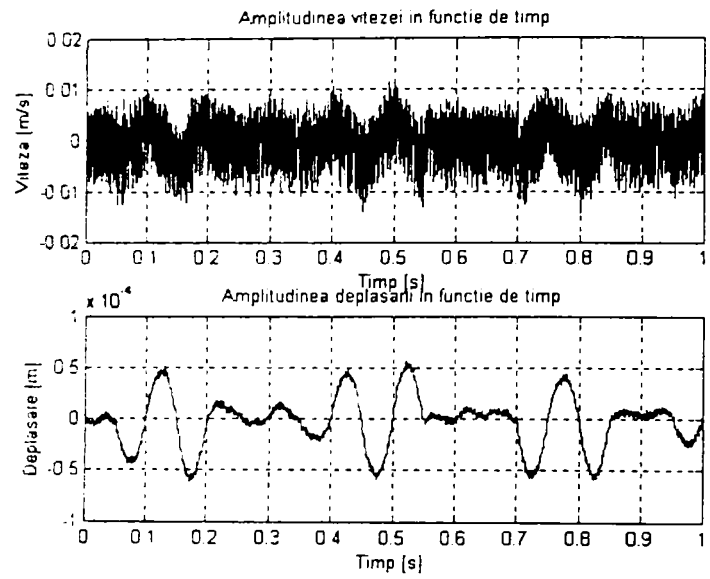


a_1) A_x=0,4 [mm] **b_1) A_x=0,056 [mm]** **c_1) A_x=0,028 [mm]**

Fig.A.11.6. Accelerații, viteze şi deplasări măsurate la amplitudinile A_x şi frecvența f=54,6 Hz



d)



d_1)

Fig.A.11.7. Accelerații, viteze și deplasări măsurate la strunjire oțel Armco cu dispozitiv blocat (fără vibrații forțate)

Anexa 12.

Apectul microscopic al suprafețelor prelucrate cu vibrații al oțelului Armco. Mărire:25x

Dispozitiv blocat
Fig.**A.12.1**

f=16,6 Hz; A_x=0,24 mm
Fig.**A.12.2.1.**

f=16,6 Hz; A_x=0,058 mm
Fig.**A.12.2.2.**

f=16,6 Hz; A_x=0,028 mm
Fig.**A.12.2.3.**

f=22,8 Hz; A_x=0,16 mm
Fig.**A.12.3.1**

f=22,8 Hz; A_x=0,056 mm
Fig.**A.12.3.2.**

f=22,8 Hz; A_x=0,028 mm
Fig.**A.12.3.3.**

f=31,4 Hz; A_x=0,24 mm
Fig.**A.12.4.1.**

f=31,4 Hz; A_x=0,056 mm
Fig.**A.12.4.2.**

f=31,4 Hz; A_x=0,028 mm
Fig.**A.12.4.3.**

f=38,6 Hz; A_x=0,14 mm
Fig.**A.12.5.1.**

f=38,6 Hz; A_x=0,056 mm
Fig.**A.12.5.2.**

f=38,6 Hz; A_x=0,056 mm

Fig.**A.12.5.3.**

f=47,4 Hz; A_x=0,25 mm

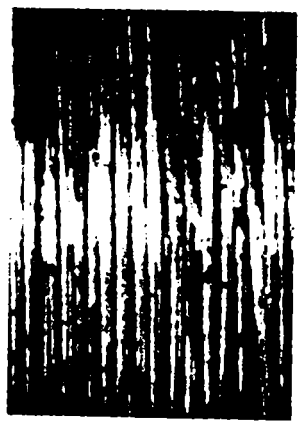
Fig.**A.12.6.1.**

f=47,4 Hz; A_x=0,07 mm

Fig.**A.12.6.2.**

f=47,4 Hz; A_x=0,028 mm

Fig.**A.12.6.3.**



f=54,6 Hz; A_x=0,4 mm

Fig.**A.12.7.1.**

f=54,6 Hz; A_x=0,056 mm

Fig.**A.12.7.2.**

f=0,028 Hz; A_x=0,028 mm

Fig.**A.12.7.3.**

Anexa 13.

Aşchii rezultate la prelucrarea obişnuită şi cu vibraţii a oţelului Armco

Fig. A. 13. 1.

f=

Fig. A. 13. 2. 1.

f=

Fig. A. 13. 2. 2.

.nm

Fig. A. 13. 2. 3.

f=22,8 Hz A_x=0,16 mm

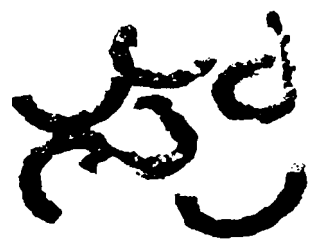
Fig. A. 13. 3. 1.

f=22,8 Hz A_x=0,056 mm

Fig. A. 13. 3.2.

f=22,8 Hz A_x=0,028 mm

Fig. A. 13. 3. 3.



f=31,4 Hz A_x=0,24 mm

Fig. A. 13. 4. 1.

f=31,4 Hz A_x=0,056 mm

Fig. A. 13. 4. 2.

f=31,4 Hz A_x=0,028 mm

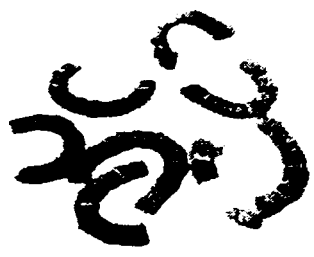
Fig. A. 13. 4. 3.

f=38,6 Hz A_x=0,14 mm

Fig. A. 13. 5. 1.

f=38,6 Hz A_x=0,056 mm

Fig. A. 13. 5. 2



f=38,6 Hz A_x=0,028 mm

Fig. A. 13. 5.3.

f=47,4 Hz A_x=0,25 mm

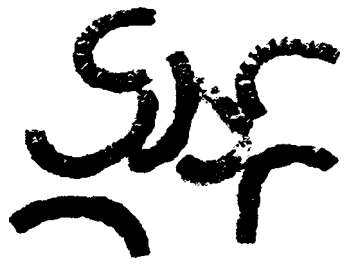
Fig. A. 13. 6. 1.

f=47,4 Hz A_x=0,07 mm

Fig. A. 13. 6. 2.

f=47,4 Hz A_x=0,028 mm

Fig. A. 13. 6. 3.



f=54,6 Hz A_x=0,4 mm

Fig. A. 13. 7. 1.

f=54,6 Hz A_x=0,056 mm

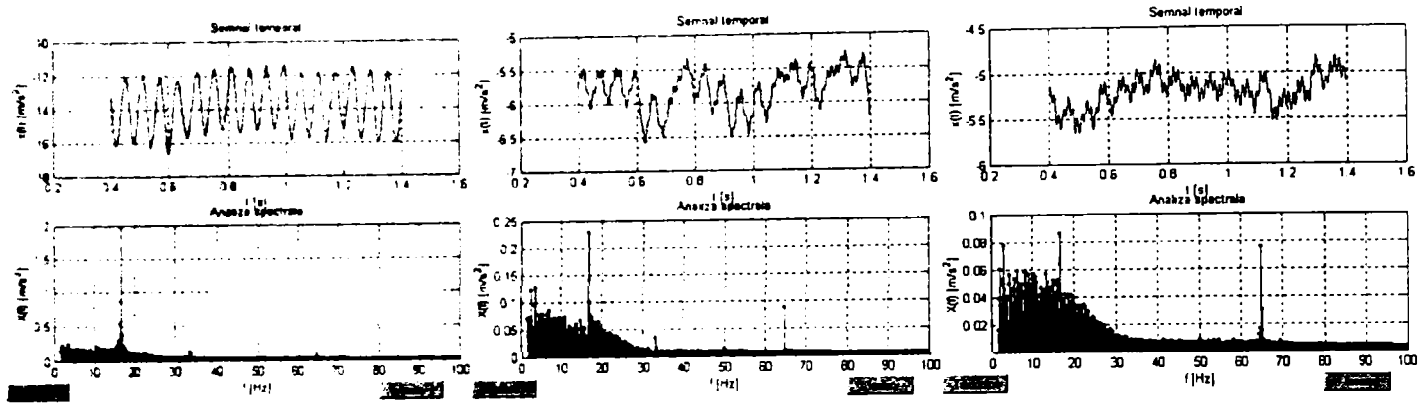
Fig. A. 13. 7. 2.

f=54,6 Hz A_x=0,028 mm

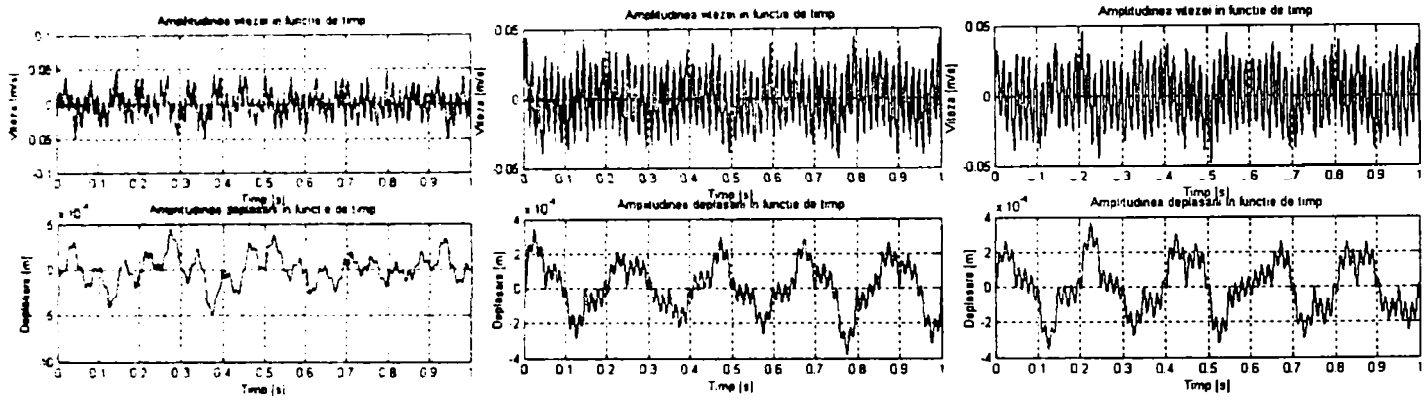
Fig. A. 13. 7. 3.

Anexa 14.

Grafice accelerații, spectre de frecvențe, viteze și deplasări înregistrate la strunjirea cu vibrații a cuprului OFHC

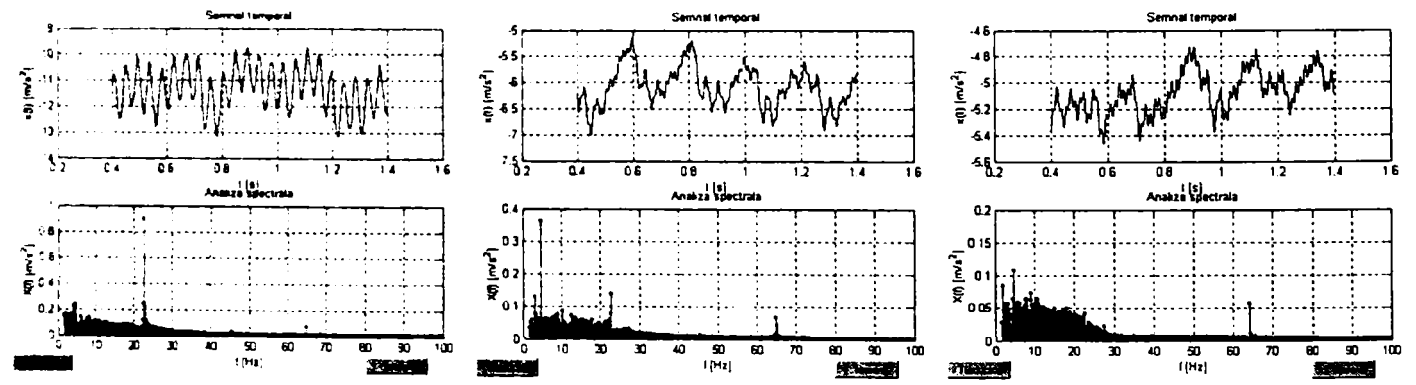


a) A_x=0,26 [mm] **b) A_x=0,056 [mm]** **c) A_x=0,028 [mm]**

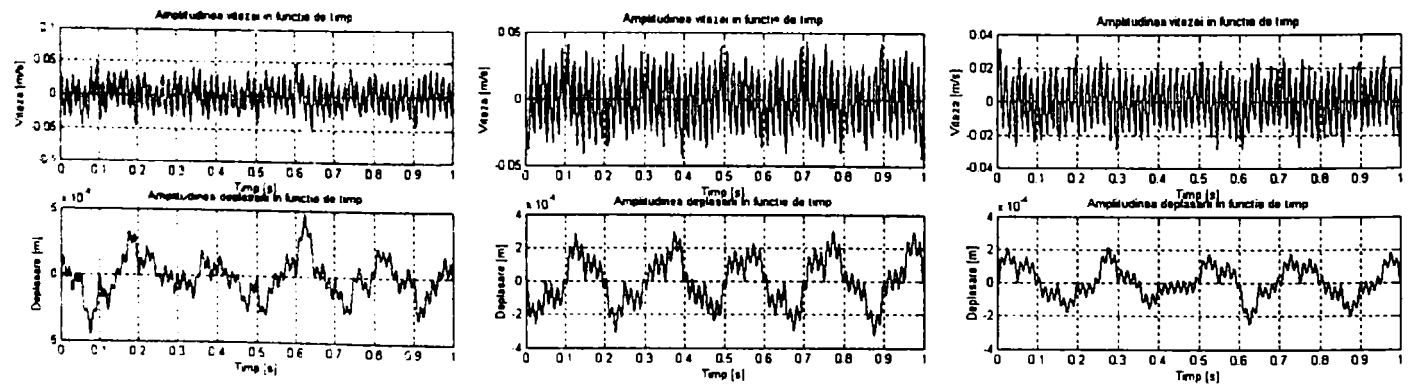


a_1) A_x=0,26 [mm] **b_1) A_x=0,056 [mm]** **c_1) A_x=0,028 [mm]**

Fig. A.14.1. Frecvența de lucru: f=16,6 Hz

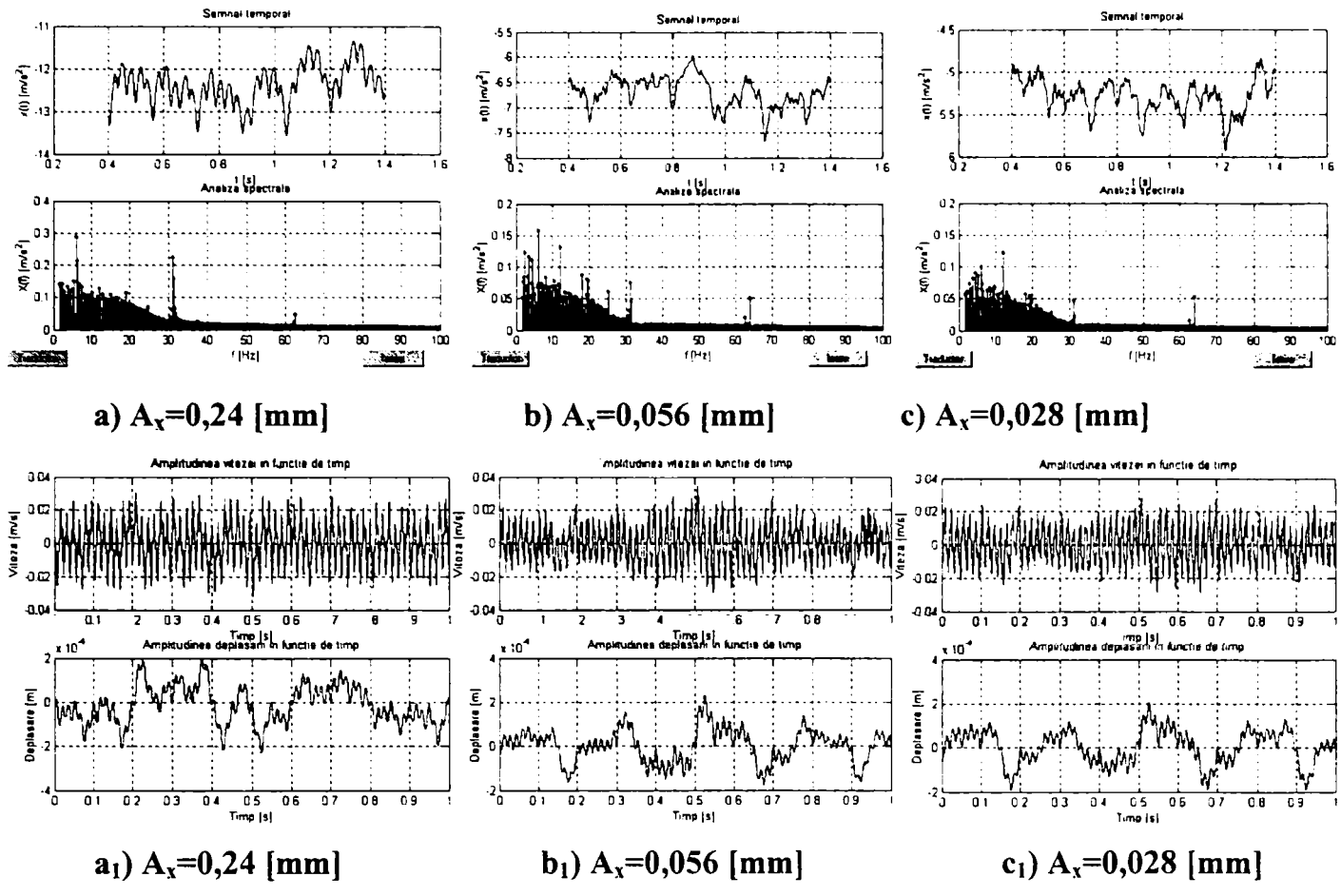


a) A_x=0,16 [mm] **b) A_x=0,056 [mm]** **c) A_x=0,028 [mm]**



a_1) A_x=0,16 [mm] **b_1) A_x=0,056 [mm]** **c_1) A_x=0,028 [mm]**

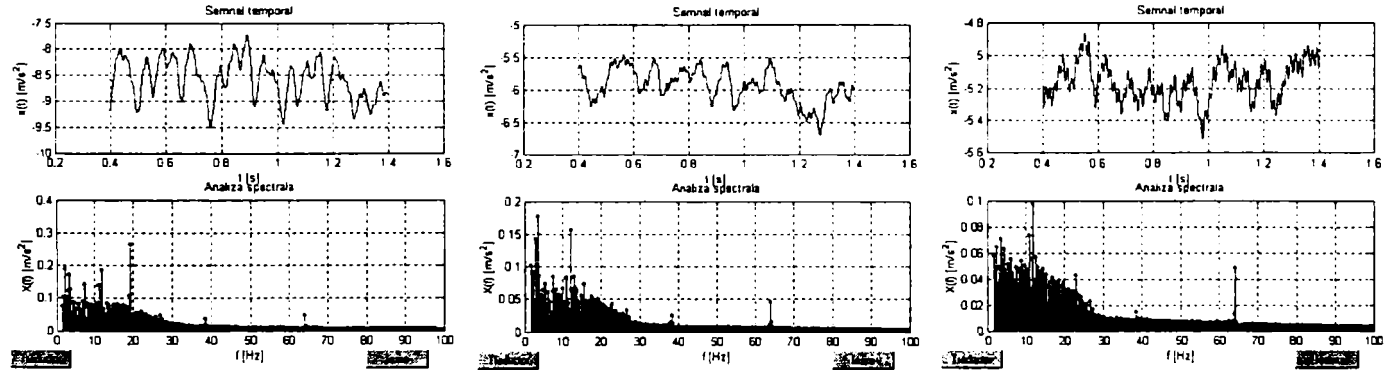
Fig. A.14.2. Frecvența de lucru: f=22,8 Hz.



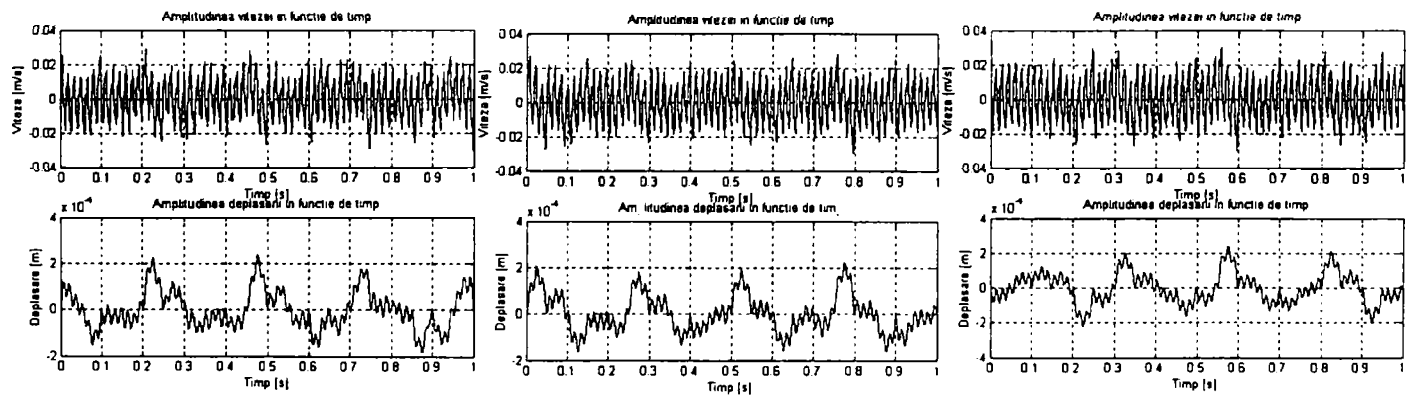
a) A_x=0,24 [mm] **b) A_x=0,056 [mm]** **c) A_x=0,028 [mm]**

a_1) A_x=0,24 [mm] **b_1) A_x=0,056 [mm]** **c_1) A_x=0,028 [mm]**

Fig. A.14.3. Frecvența de lucru: f=31,4 Hz

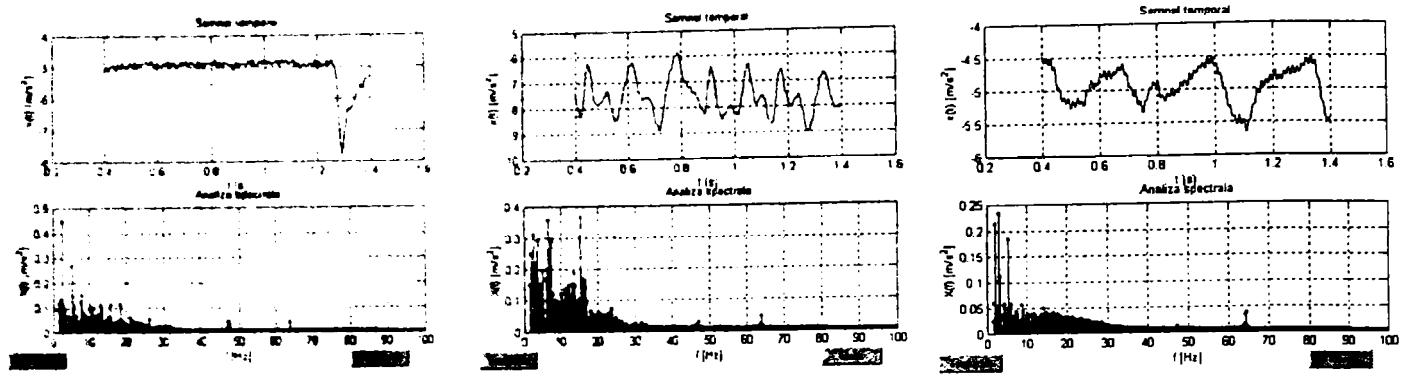


a) A_x=0,14 [mm] **b) A_x=0,056 [mm]** **c) A_x=0,028 [mm]**

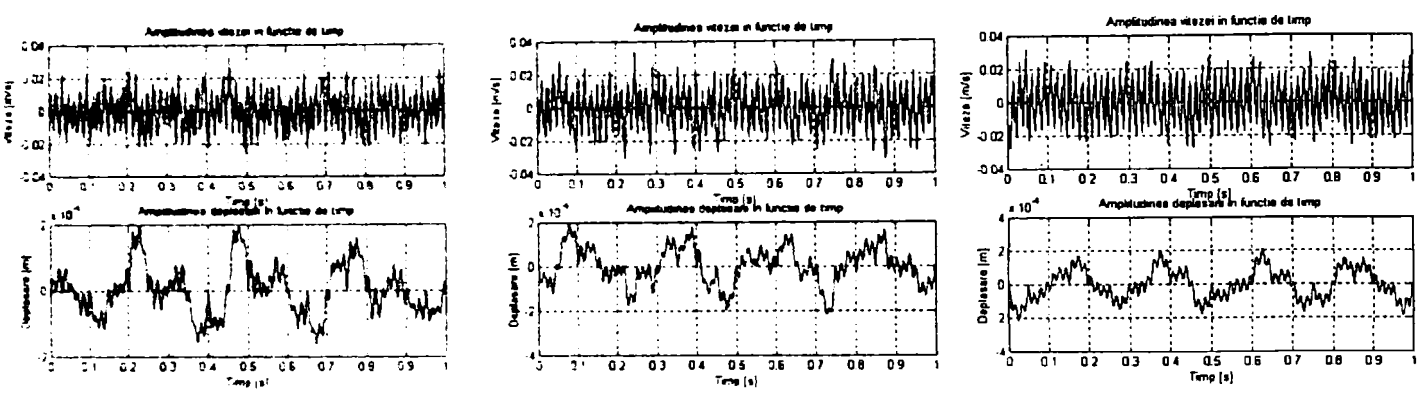


a_1) A_x=0,14 [mm] **b_1) A_x=0,056 [mm]** **c_1) A_x=0,028 [mm]**

Fig. A.14.4. Frecvența de lucru f=38,6 Hz

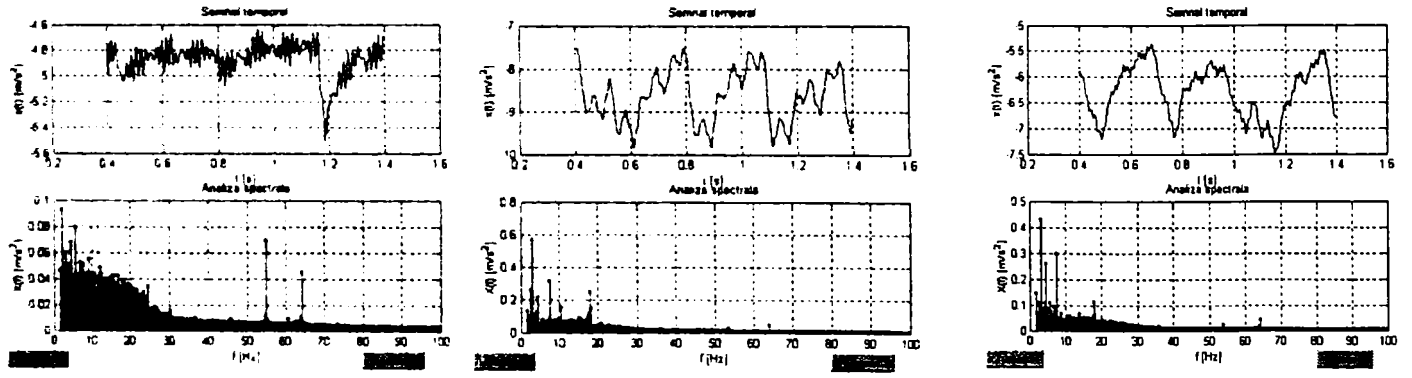


a) A_x=0,25 [mm] b) A_x=0,07 [mm] c) A_x=0,028 [mm]

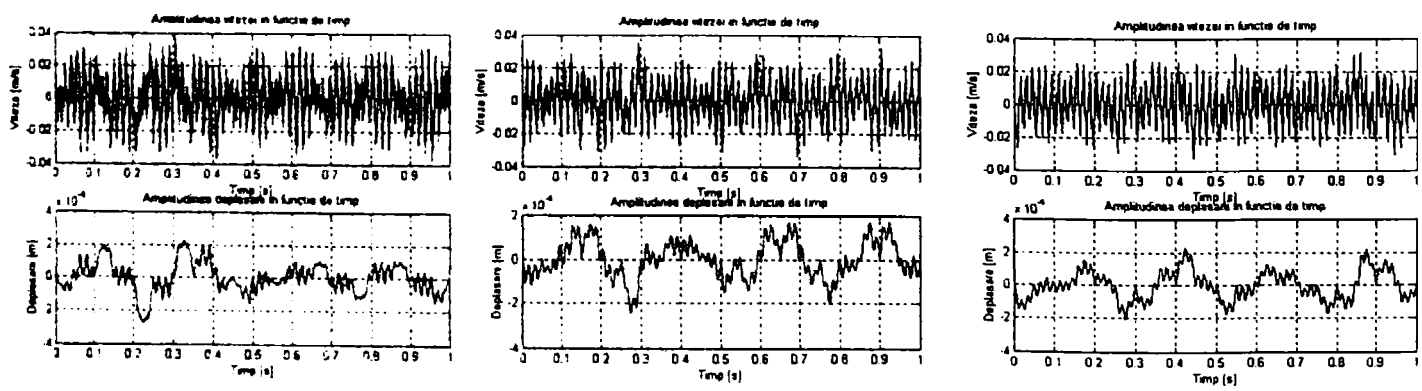


a_1) A_x=0,25 [mm] b_1) A_x=0,07 [mm] c_1) A_x=0,028 [mm]

Fig. A.14.5. Frecvența de lucru: f=47,4 Hz

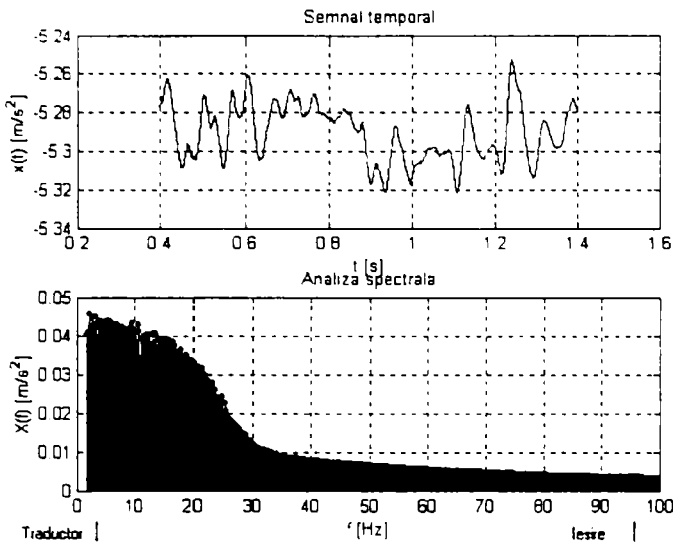


a) A_x=0,4 [mm] b) A_x=0,056 [mm] c) A_x=0,028 [mm]

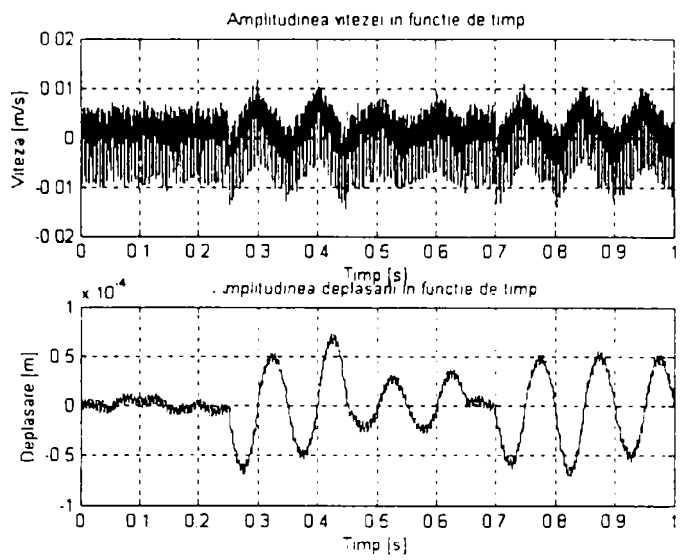


a_1) A_x=0,4 [mm] b_1) A_x=0,056 [mm] c_1) A_x=0,028 [mm]

Fig. A.14.6. Frecvența de lucru: f=54,6 Hz



d)



d_1)

Fig. A.14.7. Strunjire a cuprului OFHC cu dispozitiv blocat (fără vibrații).

Anexa 15.

Aspectul microscopic al suprafețelor prelucrate cu vibrații al cuprului OFHC

Dipozitiv blocat

Fig. A.15.1. x25.

f=16.6Hz;Ax=0,24mm

Fig. A.15.2.1. x25.

f=16.6Hz;Ax=0,056mm

Fig. A.15.2.2. x25.

f=16.6Hz;Ax=0,028mm

Fig. A.15.2.3. x25.

f=22,8Hz;Ax=0,16mm

Fig. A.15.3.1. x25.

f=22.8Hz;Ax=0,056mm

Fig. A.15.3.2. x25.

f=22,8Hz;Ax=0,028mm

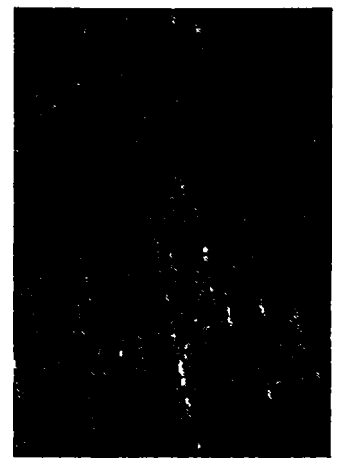
Fig. A.15.3.3. x25.

f=31,4Hz;Ax=0,24mm

Fig. A.15.4.1. x25.

f=31,4Hz;Ax=0,056mm

Fig. A.15.4.2. x25.

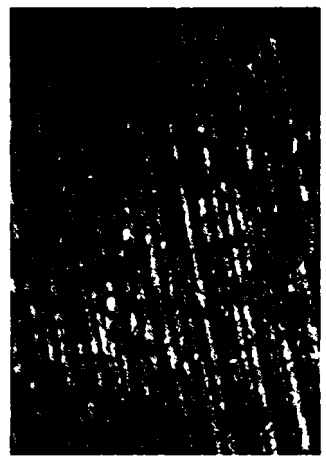


f=31,4Hz;Ax=0,028mm

Fig. A.15.4.3. x25.

f=38,6Hz;Ax=0,14mm

Fig. A.15.5.1. x25.



f=38,6Hz;Ax=0,056mm

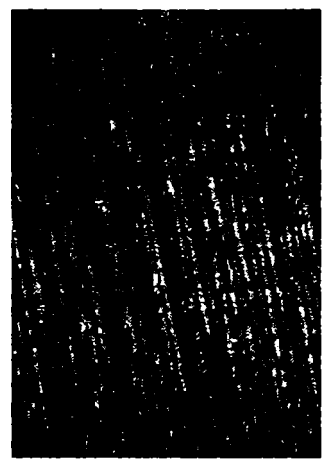
Fig. A.15.5.2. x25.

f=38,6Hz;Ax=0,028mm

Fig. A.15.5.3. x25.

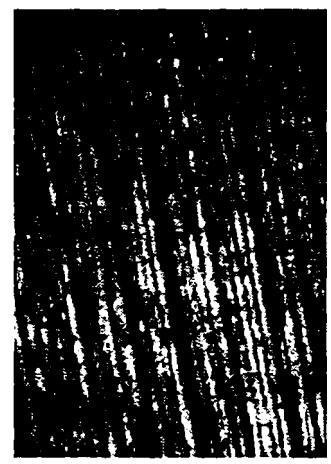
f=47,4Hz;Ax=0,25mm

Fig. A.15.6.1. x25.



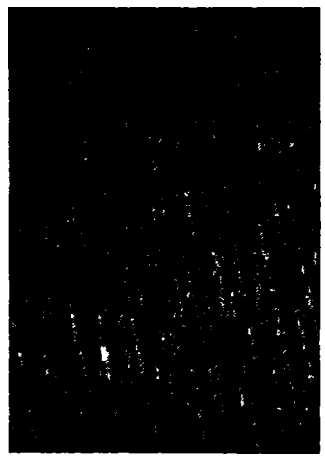
f=47,4Hz;Ax=0,07mm

Fig. A.15.6.2. x25.



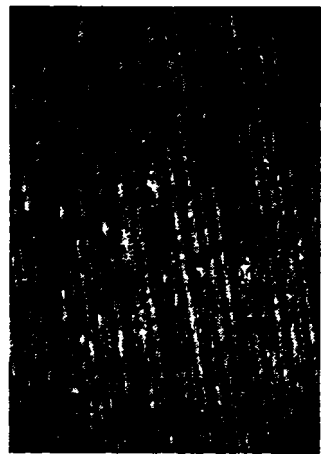
f=47,4Hz;Ax=0,028mm

Fig. A.15.6.3. x25.



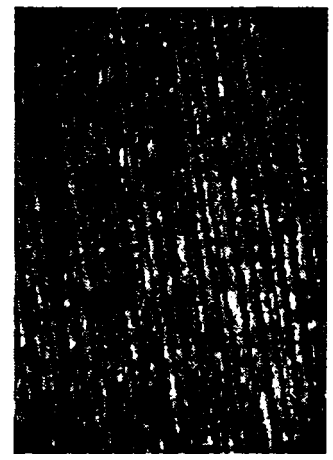
f=54,6Hz;Ax=0,4mm

Fig. A.15.7.1. x25.



f=54,6Hz;Ax=0,056mm

Fig. A.15.7.2. x25.



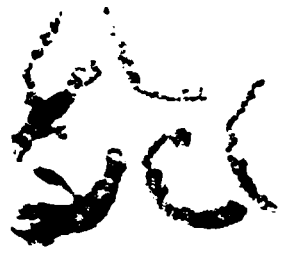
f=54,6Hz;Ax=0,028mm

Fig. A.15.7.3. x25.

Anexa 16.

Aşchii rezultate la prelucrarea cu vibraţii a cuprului OFHC

Dispozitiv blocat
Fig.A.16.1



f=16,6 Hz A_x=0,24 mm
Fig.A.16.2.1.

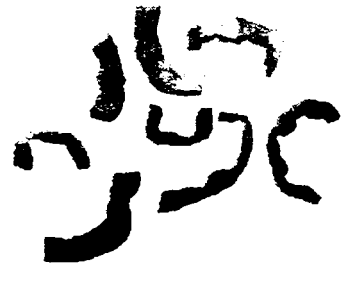
f=16,6 Hz A_x=0,056 mm
Fig.A.16.2.2.

f=16,6 Hz A_x=0,028 mm
Fig.A.16.2.3.

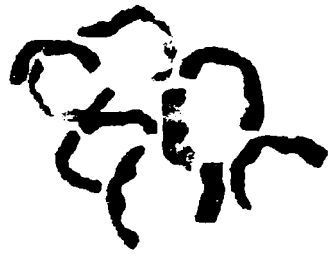
f=22,8 Hz A_x=0,16 mm
Fig.A.16.3.1.

f=22,8 Hz A_x=0,056 mm
Fig.A.16.3.2.

f=22,8 Hz A_x=0,028 mm
Fig.A.16.3.3.



f=31,4 Hz A_x=0,24 mm
Fig.A.16.4.1.



f=31,4 Hz A_x=0,056 mm
Fig.A.16.4.2.

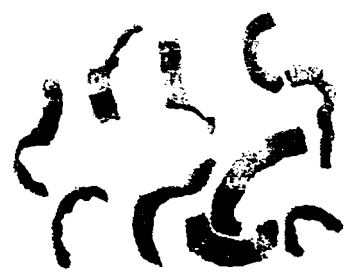
f=31,4 Hz A_x=0,028 mm
Fig.A.16.4.3.

f=38,6 Hz A_x=0,14 mm
Fig.A.16.5.1.

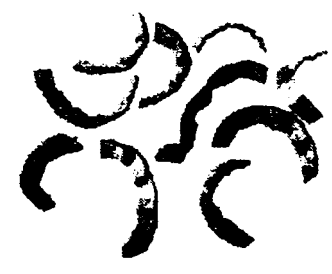
f=38,6 Hz A_x=0,056 mm
Fig.A.16.5.2.

f=38,6 Hz A_x=0,028 mm
Fig.A.16.5.3..

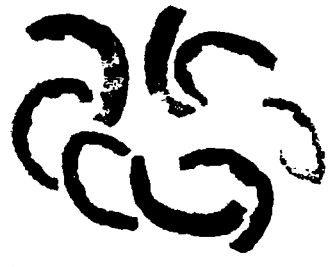
f=47,4 Hz A_x=0,25 mm
Fig.A.16.6.1.



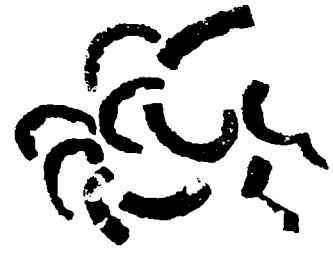
f=47,4 Hz A_x=0,07 mm
Fig.A.16.6.2.



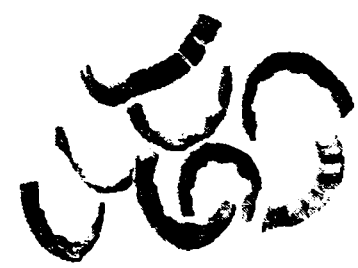
f=47,4 Hz A_x=0,028 mm
Fig.A.16.6.3.



f=54,6 Hz A_x=0,4 mm
Fig.A.16.7.1.



f=54,6 Hz A_x=0,056 mm
Fig.A.16.7.2.



f=54,6 Hz A_x=0,028 mm
Fig.A.16.7.3.

Anexa 17.

/* Program „C++_asc4" pentru stabilirea parametrilor de așchiere cu salvarea datelor în fișier

Module :

- masurarea vitezei de rotatie cu TIRO :
  - semnal A : GATEB1 (pin44)
    : CLKB2 (pin48)
  - semnal B : GATEB2 (pin47)
    CLKB1 (pin45)
  - alimentare TIRO : (pin49)
  - masa TIRO :DGND (pin50)

Functionarea programului.

Programul masoara viteza unghiulara data de catre TIRO.

/* Biblioteci*/

```
#include "nidaqex.h"
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <io.h>
#include <stdlib.h>
#include <time.h>
#include <sys/timeb.h>
#include <sys\stat.h>
#include <fcntl.h>
#include <ctype.h>
#include <string.h>
#include <sys/types.h>

unsigned short buffer;


                    /*Program principal*/

void main(void)
{
        /*variabile contorizare */

   i16 iStatus1 = 0;
   i16 iRetVal1 = 0;
   i16 iStatus0 = 0;
   i16 iRetVal0 = 0;
   i16 iDevice = 1;
   i16 iCtr0 = 0;
   i16 iCtr1 = 1;
   i16 iMode = 0;
   u16 uCount0 = 0;
   u16 uCount1 = 0;
```

```
i16 iBinBCD = 1;
i16 iIgnoreWarnings = 0;
i16 iYieldON = 1;

    /*variabila timp*/

    static u16 timp_c = 0;
    static u16 timp_c1[40000];

    /*variabila pozitie si viteza*/

    static u16 xc = 0;
    static u16 xc1[40000];
    static u16 n = 39999;
    static u16 i = 0;

    /*parametri*/

    char *strFilename1 = "m";
    unsigned char ch;
    ch = 'a';

    /* Specificarea numelui de fisier in care se inscriu datele*/
    printf("\n Numele fisierului : ");
    scanf("%s",strFilename1);
    printf("\n");
    FILE *m_asc;
    /*Deschidere fisier*/
    _fmode = _O_TEXT;
    if( (m_asc = fopen( strFilename1, "w+t" )) == NULL )
    {
            printf( "Fisierul 'm_asc' nu a fost deschis\n" );
    }
    /* Initializare contoare*/
iStatus0 = ICTR_Setup(iDevice, iCtr0, iMode, uCount0, iBinBCD);
iRetVal0 = NIDAQErrorHandler(iStatus0, "ICTR_Setup",iIgnoreWarnings);
iStatus1 = ICTR_Setup(iDevice, iCtr1, iMode, uCount1, iBinBCD);
iRetVal1 = NIDAQErrorHandler(iStatus1, "ICTR_Setup",iIgnoreWarnings);
    printf("Start achzitie!\n");
    /*Bucla de citire*/
    for (i = 0; i <= n ;i++)
    {
            /* Contor tiro*/
            iStatus1 = ICTR_Read(iDevice, iCtr1, &xc1[i]);
            /*xc1[i] = xc;*/
            /*Contor timp*/
            iStatus0 = ICTR_Read(iDevice, iCtr0, &timp_c1[i]);
            /*timp_c1[i] = timp_c;*/
    }
    printf( "Terminat citire\n" );
```

```
/*Scriere in fisier*/

for (i = 0; i <= n ;i++)
{
        fprintf(m_asc,"%d\n%d\n",xc1[i],timp_c1[i]);

}

/*Resetare contoare*/
iRetVal1 = NIDAQErrorHandler(iStatus1, "ICTR_Read", iIgnoreWarnings);
iStatus1 = ICTR_Reset(iDevice, iCtr1, 0);
iRetVal0 = NIDAQErrorHandler(iStatus0, "ICTR_Read", iIgnoreWarnings);
iStatus0 = ICTR_Reset(iDevice, iCtr0, 0);

/*Inchidere fisiere*/
if( fclose(m_asc) )
printf( "Fisierul 'm_asc' nu a fost inchis\n" );
```

% Program „Prel10.m" pentru prelucrarea datelor experimentale

```
clear all
% Preprocesare date
% A. Citire date din fisier
  [filenameg,pathname] = uigetfile('*.*','Selectati fisierul tip .mat',300,100);
  if filenameg ~=0
     load(filenameg);
     timp1 = timp_cont;
     tiro1 = tiro_cont;
     t_gol  = cumsum(timp_cont)/2000000;   % secunde
     d_gol  = 2*pi/1000*cumsum(tiro_cont); % radiani
     [filenamea,pathname] = uigetfile('*.*','Selectati fisierul tip .mat',300,100);
     if filenamea ~=0
        load(filenamea);
        t_asc  = cumsum(timp_cont)/2000000;   % secunde
        d_asc  = 2*pi/1000*cumsum(tiro_cont); % radiani
% A.1. Eliminarea revenirii pendulului dupa aschiere
        if filenamea == 'asc1b.mat'
           % Maxim
           d_asc(12926:length(d_asc)) = d_asc(12926);
           d_gol(12070:length(d_gol)) = d_gol(12070);
           % Minim
           d_asc(1:9986) = d_asc(9986);
           d_gol(1:7371) = d_gol(7371);
        end
        if filenamea == 'asc2b.mat'
           % Maxim
           d_asc(10068:length(d_asc)) = d_asc(10068);
           d_gol(13981:length(d_gol)) = d_gol(13981);
           % Minim
           d_asc(1:7204) = d_asc(7204);
```

```
    d_gol(1:9583) = d_gol(9583);
 end
 if filenamea == 'asc3b.mat'
    % Maxim
    d_asc(9076:length(d_asc)) = d_asc(9076);
    d_gol(15244:length(d_gol)) = d_gol(15244);
    % Minim
    d_asc(1:6389) = d_asc(6389);
    d_gol(1:10982) = d_gol(10982);
 end
 if filenamea == 'asc4b.mat'
    % Maxim
    d_asc(12218:length(d_asc)) = d_asc(12218);
    d_gol(12909:length(d_gol)) = d_gol(12909);
    % Minim
    d_asc(1:9416) = d_asc(9416);
    d_gol(1:8404) = d_gol(8404);
 end
 if filenamea == 'asc5b.mat'
    % Maxim
    d_asc(10631:length(d_asc)) = d_asc(10631);
    d_gol(12037:length(d_gol)) = d_gol(12037);
    % Minim
    d_asc(1:7862) = d_asc(7862);
    d_gol(1:7515) = d_gol(7515);
 end
 if filenamea == 'asc6b.mat'
    % Maxim
    d_asc(10170:length(d_asc)) = d_asc(10170);
    d_gol(13943:length(d_gol)) = d_gol(13943);
    % Minim
    d_asc(1:7372) = d_asc(7372);
    d_gol(1:9607) = d_gol(9607);
 end
 if filenamea == 'asc7b.mat'
    % Maxim
    d_asc(8402:length(d_asc)) = d_asc(8402);
    d_gol(13817:length(d_gol)) = d_gol(13817);
    % Minim
    d_asc(1:5801) = d_asc(5801);
    d_gol(1:9516) = d_gol(9516);
 end
% A.2. Eliminarea diferentelor de start
 [gol_max i_gol_max] = max(d_gol);
 [gol_min i_gol_min] = min(d_gol);
 med_gol = (gol_max - gol_min)/2;
 [asc_min i_asc_min] = min(d_asc);
 d_asc = d_asc - asc_min - med_gol;
 d_gol = d_gol - gol_min - med_gol;
 d_gol1 = d_gol>0;
 d_gol1a  = d_gol1.*d_gol;
 d_asc1 = d_asc>0;
```

```
d_asc1a  = d_asc1.*d_asc;
d_gol2 = find(d_gol1a == 0);
i_d_gol2 = d_gol2(length(d_gol2));
d_asc2 = find(d_asc1a == 0);
i_d_asc2 = d_asc2(length(d_asc2));
t_gol = t_gol - t_gol(i_d_gol2);
t_asc = t_asc - t_asc(i_d_asc2);
% Pentru conformitatea cu continuarea
d_gol4 = d_gol;
t_gol4 = t_gol;
d_asc4 = d_asc;
t_asc4 = t_asc;
```

% B. Interpolare date : se elimina caracterul de salturi in trepte al diagramei deplasarilor

% B.1. Interpolare date pentru deplasarea pendulului in gol

```
t1 = 0;
t2 = 0;
j  = 0;
for i = 2:length(d_gol4)
   if d_gol4(i-1) ~= d_gol4(i)
      j=j+1;
      t2 = t_gol4(i-1);
      tm = (t1 + t2)/2;
      t1 = t2;
      t_gol_m(j) = tm;
      d_gol_m(j) = d_gol4(i-1);
   end
end
```

% % B.2. Interpolare date pentru deplasarea pendulului in aschiere

```
t1 = 0;
t2 = 0;
j  = 0;
for i = 2:length(d_asc4)
   if d_asc4(i-1) ~= d_asc4(i)
      j=j+1;
      t2 = t_asc4(i-1);
      tm = (t1 + t2)/2;
      t1 = t2;
      t_asc_m(j) = tm;
      d_asc_m(j) = d_asc4(i-1);
   end
end

t_asc_m = [t_asc_m t_gol_m(length(t_gol_m))];
d_asc_m = [d_asc_m d_asc_m(length(d_asc_m))];
```

% C. Ajustarea lungimii seturilor de date.

% Interpolarea datelor pentru deplasare, pentru a avea intervale egale in timp

```
j = 0;
for i = 1:length(t_asc_m)
   if t_asc_m(i) < t_gol_m(1,1)
      j = i;
```

```
    end
end
if j ~= 0
    t_asc_m = t_asc_m(j:length(t_asc_m));
    d_asc_m = d_asc_m(j:length(d_asc_m));
end
t_gol_m = t_gol_m(2:length(t_gol_m));
d_gol_m = d_gol_m(2:length(d_gol_m));
t_sup = min(t_gol_m(length(t_gol_m)),t_asc_m(length(t_asc_m)));
t_inf = max(t_gol_m(1),t_asc_m(1));
n_int = 20000;

delta_t = (t_sup - t_inf)/n_int;
ti = [t_inf:delta_t:t_sup];
d_gol_mi = interp1(t_gol_m,d_gol_m,ti);
d_asc_mi = interp1(t_asc_m,d_asc_m,ti);
```

% D. Filtrarea de netezire - pentru eliminarea erorilor accidentale

```
% Filtrarea se realizeaza cu un filtru de convolutie pe 50 de puncte
n_conv = 50;
f_conv = zeros(1,n_conv)+1/n_conv;
d_gol_c = conv(d_gol_mi,f_conv);
d_asc_c = conv(d_asc_mi,f_conv);
d_gol_c = d_gol_c(1:length(ti));
d_asc_c = d_asc_c(1:length(ti));
```

% E. Calculul vitezelor pentru pendulul in gol si in aschiere

% E.1. Derivarea numerica a deplasarii

```
v_gol_c = diff(d_gol_c)./diff(ti);
v_asc_c = diff(d_asc_c)./diff(ti);
```

% E.2. Ajustare lungime vectori dupa derivare.

```
d_gol_c = d_gol_c(1:length(ti)-1);
d_asc_c = d_asc_c(1:length(ti)-1);
ti      = ti(1:length(ti)-1);
```

% F. Filtrarea de netezire a vitezelor - pentru eliminarea erorilor de derivare

```
% Filtrarea se realizeaza cu un filtru de convolutie pe 500 de puncte
n_conv = 1000;
f_conv = zeros(1,n_conv)+1/n_conv;
v_gol_cc = conv(v_gol_c,f_conv);
v_asc_cc = conv(v_asc_c,f_conv);
v_gol_cc = v_gol_cc(1:length(ti));
v_asc_cc = v_asc_cc(1:length(ti));
```

% G. Ajustarea punctului de inceput si de sfarsit pentru calculul lucrului mecanic

```
if filenamea == 'asc1b.mat'
    s = 6874;
    f = 11760;
end
if filenamea == 'asc2b.mat'
    s = 6726;
    f = 11190;
end
if filenamea == 'asc3b.mat'
    s = 6392;
```

```
    f = 10619;
end
if filenamea == 'asc4b.mat'
    s = 6789;
    f = 11015;
end
if filenamea == 'asc5b.mat'
    s = 7365;
    f = 11365;
end
if filenamea == 'asc6b.mat'
    s = 7044;
    f = 11012;
end
if filenamea == 'asc7b.mat'
    s = 6640;
    f = 10834;
end
```

% G. Calculul Lucrului mecanic de aschiere

% G.1. Definire parametrii pentru calculul lucrului mecanic

```
g        = 9.81; % acceleratia gravitationala
m_pen    = 7.618; % kg
l_pen    = 0.2578; % m
r_sc     = 0.04; % m
ro       = 8900; % kg/m^3
lat_asc  = 0.0053; % m
adinc_asc = 0.0001; % m
suprafata = lat_asc*adinc_asc;
alfa     = 0.5;
c_asc    = 1 - alfa;
c1_asc   = c_asc*ro*suprafata*r_sc^3;
```

% Calculul fi_rot

```
[v_imp i_v_imp] = max(v_gol_cc);
fi_rot_c = (v_gol_cc.^2 - v_asc_cc.^2)./((v_imp^2 - v_asc_cc.^2)./...
        d_asc_c - 7.758*g*sin(d_asc_c));
fi_rot  = fi_rot_c;
ti_s        = ti(s:length(ti));
d_gol_c_s   = d_gol_c(s:length(ti));
d_asc_c_s   = d_asc_c(s:length(ti));
v_gol_cc_s  = v_gol_cc(s:length(ti));
v_asc_cc_s  = v_asc_cc(s:length(ti));
fi_rot_s    = fi_rot(s:length(ti));
[max_fi_rot f] = max(fi_rot_s);
ti_f        = ti_s(1:f);
d_gol_c_f   = d_gol_c_s(1:f);
d_asc_c_f   = d_asc_c_s(1:f);
v_gol_cc_f  = v_gol_cc_s(1:f);
v_asc_cc_f  = v_asc_cc_s(1:f);
fi_rot_f    = fi_rot_s(1:f);
disp('fi_rot')
disp(max_fi_rot)
disp(max_fi_rot*180/pi)
```

```
            % Diagrame
            f_1 = figure('Name',",'NumberTitle','off');
            plot(ti_f,d_asc_c_f,'k');
            hold on
            plot(ti_f,d_gol_c_f,'k');
            title(strcat('Deplasarea unghiulara 0p - rosu; 0ap - negru'));
            xlabel('Timp [s]');ylabel('[rad]');
            f_2 = figure('Name',",'NumberTitle','off');
            plot(ti_f,v_asc_cc_f,'k');
            hold on
            plot(ti_f,v_gol_cc_f,'k');
            title(strcat('Viteza unghiulara Vp - rosu; Vap - negru'));
            xlabel('Timp [s]');ylabel('[rad/s]');
            hold off
            f_4 = figure('Name',",'NumberTitle','off');
            plot(ti_f,fi_rot_f(1:length(ti_f)),'k');
            title(strcat('Deplasarea unghiulara 0a'));
            xlabel('Timp [s]');ylabel('[rad]');
        end
end
```

Anexa 18.

% Program „tiro_fis_a.m" de conversie a fişierelor ASCII în fişiere MATLAB

```
global viteza_rot timp_sec x1 viteza_conv_final timp_conv
global timp deplasare viteza
global t_i d_i v_i
% Citire din fisier
[filename,pathname] = uigetfile('*.*','Selectati fisierul de date',300,100);
if filename ~=0
   fid = fopen(strcat(pathname,filename));
   j = 1;
   h_bara = waitbar(0,'Citire valori din fisier...');
   while 1
      line = fgetl(fid);
      if ~isstr(line),break,end
      tiro(j) = str2num(line);
      line = fgetl(fid);
      if ~isstr(line),break,end
      timp(j) = str2num(line);
      j = j + 1;
      waitbar(j/40000)
   end
   fclose(fid);
   close(h_bara);
   timp_cont = zeros(1,length(timp));
   tiro_cont = zeros(1,length(tiro));
   % Calcul deltat si deltax
   h_bara1 = waitbar(0,'Conversie date...');
   for i = 1:length(timp)-1
      if timp(i+1) <= timp(i)
         timp_cont(i)    = abs(timp(i) - timp(i+1));
      else
         timp_cont(i)    = abs(timp(i) + (65535 - timp(i+1)));
      end
      if tiro(i+1) <= tiro(i)
         tiro_cont(i)    = abs(tiro(i) - tiro(i+1));
      else
         tiro_cont(i)    = abs(tiro(i) + (65535 - tiro(i+1)));
      end
      if tiro_cont(i) >= 2
         tiro_cont(i) = 1;
      end
      if timp_cont(i) >= 3000
         timp_cont(i) = 275;
      end
      waitbar(i/(length(timp)-1));
   end
   % Salvare mat
   save(filename,'tiro_cont','timp_cont');
   close(h_bara1);
end
```

% Subprogram „sub fi_rot.m" pentru calculul energiei specifice corespunzătoare unghiului fi_rot

```
clear all

% Calculul fi_rot

        [v_imp i_v_imp] = max(v_gol_cc);
        fi_rot_c = (v_gol_cc.^2 - v_asc_cc.^2)./((v_imp^2 - v_asc_cc.^2)./...
                d_asc_c - 7.758*g*sin(d_asc_c));

        fi_rot_c1 = (v_gol_cc.^2)./((v_imp^2)./d_asc_c - 7.758*g*sin(d_asc_c));


        fi_rot  = fi_rot_c;

        ti_s        = ti(s:length(ti));
        d_gol_c_s   = d_gol_c(s:length(ti));
        d_asc_c_s   = d_asc_c(s:length(ti));
        v_gol_cc_s  = v_gol_cc(s:length(ti));
        v_asc_cc_s  = v_asc_cc(s:length(ti));
        fi_rot_s    = fi_rot(s:length(ti));

        [max_fi_rot f] = max(fi_rot_s);
        ti_f        = ti_s(1:f);
        d_gol_c_f   = d_gol_c_s(1:f);
        d_asc_c_f   = d_asc_c_s(1:f);
        v_gol_cc_f  = v_gol_cc_s(1:f);
        v_asc_cc_f  = v_asc_cc_s(1:f);
        fi_rot_f    = fi_rot_s(1:f);
        fi_rot  = fi_rot_c1;
        fi_rot_s_c1         = fi_rot(s:length(ti));
        [max_fi_rot_c1 f_c1] = max(fi_rot_s_c1);
        teta_ap = d_asc_c_f(1,f);

%           ws = m_pen*(v_imp^2 - 2*m_pen*g*l_pen*teta_ap*sin(teta_ap))./...
%            (2*0.53*teta_ap*1000*l_pen);
        delta_fa = m_pen*(l_pen*v_imp^2/(2*teta_ap) - g*sin(teta_ap));
        ws = (delta_fa*(l_pen/r_sc))/(2*0.53*teta_ap);

        disp('fi_rot_c')
        disp(max_fi_rot)
        disp(max_fi_rot*180/pi)
        disp('fi_rot_c1')
        disp(max_fi_rot_c1)
        disp(max_fi_rot_c1*180/pi)
        disp('ws')
        disp(ws)
    end
end
```

% Program „calcul_ws.m" pentru prelucrarea datelor experimentale aschiabilitate

```
clear all
% Preprocesare date
% A.1. Citire date din fisier
[filenameg,pathname] = uigetfile('*.*','Selectati fisierul tip .mat',300,100);
if filenameg ~=0
   load(filenameg);
   timp1 = timp_cont;
   tiro1 = tiro_cont;
   t_gol  = cumsum(timp_cont)/2000000;   % secunde
   d_gol  = 2*pi/1000*cumsum(tiro_cont); % radiani
   [filenamea,pathname] = uigetfile('*.*','Selectati fisierul tip .mat',300,100);
   if filenamea ~=0
      load(filenamea);
      t_asc  = cumsum(timp_cont)/2000000;   % secunde
      d_asc  = 2*pi/1000*cumsum(tiro_cont); % radiani

% A.2. Eliminarea revenirii pendului dupa aschiere
% i_asc_min = [9740 7047  6277 9286 7862 7371 5762]
% i_gol_min = [7340 9452 10772 8309 7463 9607 9471]
      if filenamea == 'asc1b.mat'
         % Maxim
         d_asc(12926:length(d_asc)) = d_asc(12926);
         d_gol(12070:length(d_gol)) = d_gol(12070);
         % Minim
         d_asc(1:9986) = d_asc(9986);
         d_gol(1:7371) = d_gol(7371);
      end
      if filenamea == 'asc2b.mat'
         % Maxim
         d_asc(10068:length(d_asc)) = d_asc(10068);
         d_gol(13981:length(d_gol)) = d_gol(13981);
         % Minim
         d_asc(1:7204) = d_asc(7204);
         d_gol(1:9583) = d_gol(9583);
      end
      if filenamea == 'asc3b.mat'
         % Maxim
         d_asc(9076:length(d_asc)) = d_asc(9076);
         d_gol(15244:length(d_gol)) = d_gol(15244);
         % Minim
         d_asc(1:6389) = d_asc(6389);
         d_gol(1:10982) = d_gol(10982);
      end
      if filenamea == 'asc4b.mat'
         % Maxim
         d_asc(12218:length(d_asc)) = d_asc(12218);
         d_gol(12909:length(d_gol)) = d_gol(12909);
         % Minim
         d_asc(1:9416) = d_asc(9416);
```

```
    d_gol(1:8404) = d_gol(8404);
 end
 if filenamea == 'asc5b.mat'
    % Maxim
    d_asc(10631:length(d_asc)) = d_asc(10631);
    d_gol(12037:length(d_gol)) = d_gol(12037);
    % Minim
    d_asc(1:7862) = d_asc(7862);
    d_gol(1:7515) = d_gol(7515);
 end
 if filenamea == 'asc6b.mat'
    % Maxim
    d_asc(10170:length(d_asc)) = d_asc(10170);
    d_gol(13943:length(d_gol)) = d_gol(13943);
    % Minim
    d_asc(1:7372) = d_asc(7372);
    d_gol(1:9607) = d_gol(9607);
 end
 if filenamea == 'asc7b.mat'
    % Maxim
    d_asc(8402:length(d_asc)) = d_asc(8402);
    d_gol(13817:length(d_gol)) = d_gol(13817);
    % Minim
    d_asc(1:5801) = d_asc(5801);
    d_gol(1:9516) = d_gol(9516);
 end
```

% A.3. Eliminarea diferentelor de start

```
[gol_max i_gol_max] = max(d_gol);
[gol_min i_gol_min] = min(d_gol);
med_gol = (gol_max - gol_min)/2;
[asc_min i_asc_min] = min(d_asc);
d_asc = d_asc - asc_min - med_gol;
d_gol = d_gol - gol_min - med_gol;
d_gol1 = d_gol>0;
d_gol1a  = d_gol1.*d_gol;
d_asc1 = d_asc>0;
d_asc1a  = d_asc1.*d_asc;
d_gol2 = find(d_gol1a == 0);
i_d_gol2 = d_gol2(length(d_gol2));
d_asc2 = find(d_asc1a == 0);
i_d_asc2 = d_asc2(length(d_asc2));
t_gol = t_gol - t_gol(i_d_gol2);
t_asc = t_asc - t_asc(i_d_asc2);
```

% Pentru conformitatea cu continuarea

```
d_gol4 = d_gol;
t_gol4 = t_gol;
d_asc4 = d_asc;
t_asc4 = t_asc;
```

% A.4. Taierea valorilor nesemnificative

```
%       d_gol1 = d_gol>0;
%       d_gol  = d_gol1.*d_gol;
%       d_asc1 = d_asc>0;
%       d_asc  = d_asc1.*d_asc;
%       d_gol2 = find(d_gol == 0);
%       i_d_gol2 = d_gol2(length(d_gol2));
%       d_asc2 = find(d_asc == 0);
%       i_d_asc2 = d_asc2(length(d_asc2));
%       t_gol = t_gol - t_gol(i_d_gol2);
%       t_asc = t_asc - t_asc(i_d_asc2);
%
%       d_gol3 = d_gol(i_d_gol2:length(d_gol));
%       t_gol3 = t_gol(i_d_gol2:length(d_gol));
%       d_asc3 = d_asc(i_d_asc2:length(d_asc));
%       t_asc3 = t_asc(i_d_asc2:length(d_asc));
%       [gol_max3 i_gol_max3] = max(d_gol3);
%
%       d_gol4 = d_gol3(1:i_gol_max3+400);
%       t_gol4 = t_gol3(1:i_gol_max3+400);
%       d_asc4 = d_asc3(1:i_gol_max3+400);
%       t_asc4 = t_asc3(1:i_gol_max3+400);
```

% B. Interpolare date : se elimina caracterul de salturi in trepte al diagramei deplasarilor

% B.1. Interpolare date pentru deplasarea pendulului in gol

```
t1 = 0;
t2 = 0;
j  = 0;
for i = 2:length(d_gol4)
   if d_gol4(i-1) ~= d_gol4(i)
      j=j+1;
      t2 = t_gol4(i-1);
      tm = (t1 + t2)/2;
      t1 = t2;
      t_gol_m(j) = tm;
      d_gol_m(j) = d_gol4(i-1);
   end
end
```

% % B.2. Interpolare date pentru deplasarea pendulului in aschiere

```
t1 = 0;
t2 = 0;
j  = 0;
for i = 2:length(d_asc4)
   if d_asc4(i-1) ~= d_asc4(i)
      j=j+1;
      t2 = t_asc4(i-1);
```

```
            tm = (t1 + t2)/2;
            t1 = t2;
            t_asc_m(j) = tm;
            d_asc_m(j) = d_asc4(i-1);
        end
    end

    t_asc_m = [t_asc_m t_gol_m(length(t_gol_m))];
    d_asc_m = [d_asc_m d_asc_m(length(d_asc_m))];
```

% C. Ajustarea lungimii seturilor de date.

% Interpolarea datelor pentru deplasare, pentru a avea intervale egale in timp

```
    j = 0;
    for i = 1:length(t_asc_m)
        if t_asc_m(i) < t_gol_m(1,1)
            j = i;
        end
    end
    if j ~= 0
        t_asc_m = t_asc_m(j:length(t_asc_m));
        d_asc_m = d_asc_m(j:length(d_asc_m));
    end
    t_gol_m = t_gol_m(2:length(t_gol_m));
    d_gol_m = d_gol_m(2:length(d_gol_m));

%       figure
%       plot(t_asc_m,d_asc_m,'k.')
%       hold on
%       plot(t_gol_m,d_gol_m,'r.')

    t_sup = min(t_gol_m(length(t_gol_m)),t_asc_m(length(t_asc_m)));
    t_inf = max(t_gol_m(1),t_asc_m(1));
    n_int = 20000;

    delta_t = (t_sup - t_inf)/n_int;
    ti = [t_inf:delta_t:t_sup];
    d_gol_mi = interp1(t_gol_m,d_gol_m,ti);
    d_asc_mi = interp1(t_asc_m,d_asc_m,ti);

%       plot(ti,d_asc_mi,'b.')
%       plot(ti,d_gol_mi,'m.')
```

% D. Filtrarea de netezire - pentru eliminarea erorilor accidentale

% Filtrarea se realizeaza cu un filtru de convolutie pe 50 de puncte

```
    n_conv = 50;
    f_conv = zeros(1,n_conv)+1/n_conv;
    d_gol_c = conv(d_gol_mi,f_conv);
    d_asc_c = conv(d_asc_mi,f_conv);
    d_gol_c = d_gol_c(1:length(ti));
    d_asc_c = d_asc_c(1:length(ti));
```

```
% E. Calculul vitezelor pentru pendulul in gol si in aschiere
% E.1. Derivarea numerica a deplasarii

    v_gol_c = diff(d_gol_c)./diff(ti);
    v_asc_c = diff(d_asc_c)./diff(ti);

% E.2. Ajustare lungime vectori dupa derivare.

    d_gol_c = d_gol_c(1:length(ti)-1);
    d_asc_c = d_asc_c(1:length(ti)-1);
    ti      = ti(1:length(ti)-1);

% F. Filtrarea de netezire a vitezelor - pentru eliminarea erorilor de derivare
% Filtrarea se realizeaza cu un filtru de convolutie pe 500 de puncte

    n_conv = 1000;
    f_conv = zeros(1,n_conv)+1/n_conv;
    v_gol_cc = conv(v_gol_c,f_conv);
    v_asc_cc = conv(v_asc_c,f_conv);
    v_gol_cc = v_gol_cc(1:length(ti));
    v_asc_cc = v_asc_cc(1:length(ti));

% G. Ajustarea punctului de inceput si de sfarsit pentru calculul lucrului mecanic
%       1    2    3    4    5    6    7
% s = [ 6874  6726  6392  6789  7365  7044  6640]
% f = [11760 11190 10619 11015 11365 11012 10834]
if filenamea == 'asc1b.mat'
   s = 6874;
   f = 11760;
end
if filenamea == 'asc2b.mat'
   s = 6726;
   f = 11190;
end
if filenamea == 'asc3b.mat'
   s = 6392;
   f = 10619;
end
if filenamea == 'asc4b.mat'
   s = 6789;
   f = 11015;
end
if filenamea == 'asc5b.mat'
   s = 7365;
   f = 11365;
end
if filenamea == 'asc6b.mat'
   s = 7044;
   f = 11012;
end
if filenamea == 'asc7b.mat'
```

```
    s = 6640;
    f = 10834;
end

% G. Calculul Lucrului mecanic de aschiere
% G.1. Definire parametrii pentru calculul lucrului mecanic
    g        = 9.81; % acceleratia gravitationala
    m_pen    = 7.618; % kg
    l_pen    = 0.2578; % m
    r_sc     = 0.04; % m
    ro       = 8900; % kg/m^3
    lat_asc  = 0.0053; % m
    adinc_asc = 0.0001; % m
    suprafata = lat_asc*adinc_asc;
    alfa     = 0.5;
    c_asc    = 1 - alfa;
    c1_asc   = c_asc*ro*suprafata*r_sc^3;

% Calculul fi_rot

    [v_imp i_v_imp] = max(v_gol_cc);
    fi_rot_c = (v_gol_cc.^2 - v_asc_cc.^2)./((v_imp^2 - v_asc_cc.^2)./...
            d_asc_c - 7.758*g*sin(d_asc_c));

    fi_rot_c1 = (v_gol_cc.^2)./((v_imp^2)./d_asc_c - 7.758*g*sin(d_asc_c));

    fi_rot  = fi_rot_c;

    ti_s        = ti(s:length(ti));
    d_gol_c_s   = d_gol_c(s:length(ti));
    d_asc_c_s   = d_asc_c(s:length(ti));
    v_gol_cc_s  = v_gol_cc(s:length(ti));
    v_asc_cc_s  = v_asc_cc(s:length(ti));
    fi_rot_s    = fi_rot(s:length(ti));

    [max_fi_rot f] = max(fi_rot_s);
    ti_f        = ti_s(1:f);
    d_gol_c_f   = d_gol_c_s(1:f);
    d_asc_c_f   = d_asc_c_s(1:f);
    v_gol_cc_f  = v_gol_cc_s(1:f);
    v_asc_cc_f  = v_asc_cc_s(1:f);
    fi_rot_f    = fi_rot_s(1:f);

    fi_rot  = fi_rot_c1;
    fi_rot_s_c1       = fi_rot(s:length(ti));
    [max_fi_rot_c1 f_c1] = max(fi_rot_s_c1);
    teta_ap = d_asc_c_f(1,f);

%       ws = m_pen*(v_imp^2 - 2*m_pen*g*l_pen*teta_ap*sin(teta_ap))./...
%        (2*0.53*teta_ap*1000*l_pen);
    delta_fa = m_pen*(l_pen*v_imp^2/(2*teta_ap) - g*sin(teta_ap));
    ws = (delta_fa*(l_pen/r_sc))/(2*0.53*teta_ap);
```

```
        disp('fi_rot_c')
        disp(max_fi_rot)
        disp(max_fi_rot*180/pi)
        disp('fi_rot_c1')
        disp(max_fi_rot_c1)
        disp(max_fi_rot_c1*180/pi)
        disp('ws')
        disp(ws)

        % Diagrame
%           f_1 = figure('Name','','NumberTitle','off');
%           plot(ti_f,d_asc_c_f,'k');
%           hold on
%           plot(ti_f,d_gol_c_f,'k');
%           title(strcat('Deplasarea unghiulara 0p - rosu: 0ap - negru'));
%           xlabel('Timp [s]');ylabel('[rad]');
%
%           f_2 = figure('Name','','NumberTitle','off');
%           plot(ti_f,v_asc_cc_f,'k');
%           hold on
%           plot(ti_f,v_gol_cc_f,'k');
%           title(strcat('Viteza unghiulara Vp - rosu; Vap - negru'));
%           xlabel('Timp [s]'); ylabel('[rad/s]');
%           hold off
%
%           f_4 = figure('Name','','NumberTitle','off');
%           plot(ti_f,fi_rot_f(1:length(ti_f)),'k');
%           title(strcat('Deplasarea unghiulara 0a'));
%           xlabel('Timp [s]');ylabel('[rad]');
      end
   end
```

% Subrutina „ws.m" pentru trasarea diagramelor fi_rot functie de viteze si energie specifica functie de viteze

```
v = [11.664 13.219 14.774 16.329 17.885 19.440 20.995];

fi_rot = [0.1740 0.1893 0.3065 0.2656 0.2841 0.3407 0.4936];
plot(v,fi_rot,'k')
hold on
plot(v,fi_rot,'ko')

ws = [3.669 4.003 2.085 2.921 3.237 2.565 1.437];
figure
plot(v,ws,'k')
hold on
plot(v,ws,'kd')
```

Anexa 19.

Diagrammele deplasărilor θ_a , θ_{ap} , θ_p şi vitezelor unghiulare ω_{ap} ω_p corespunzătoare vitezelor din tabelul 7.2

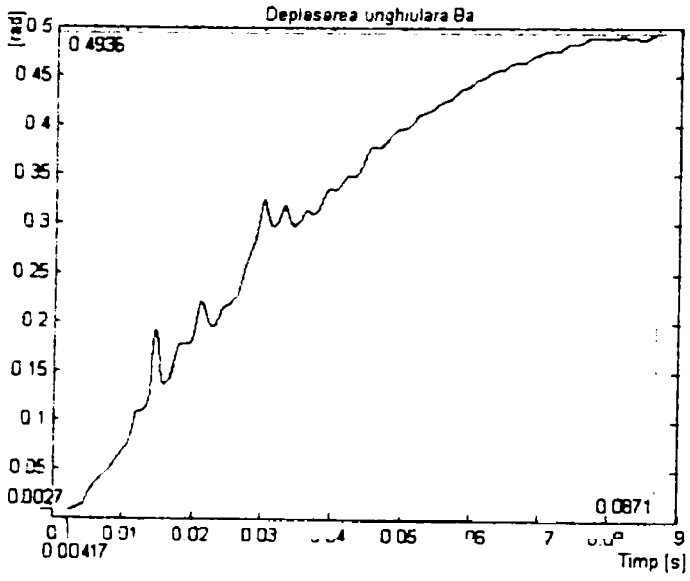


Fig. A.19.1.a

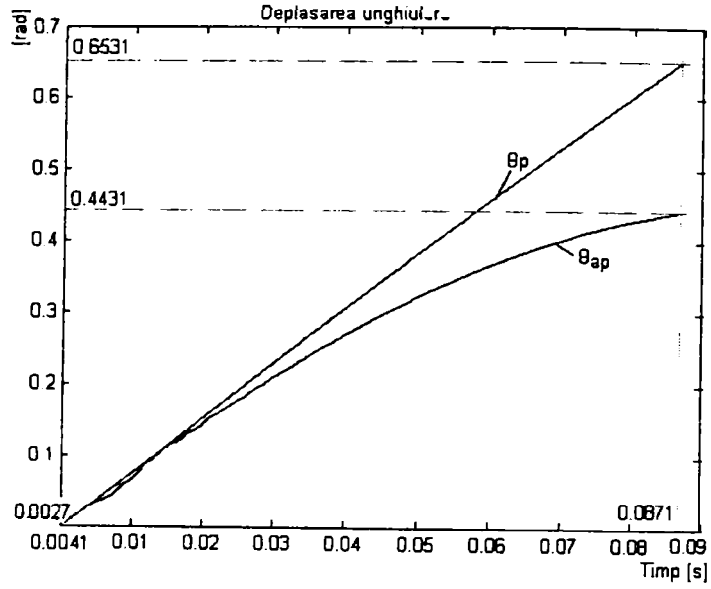


Fig. A.19.1.b

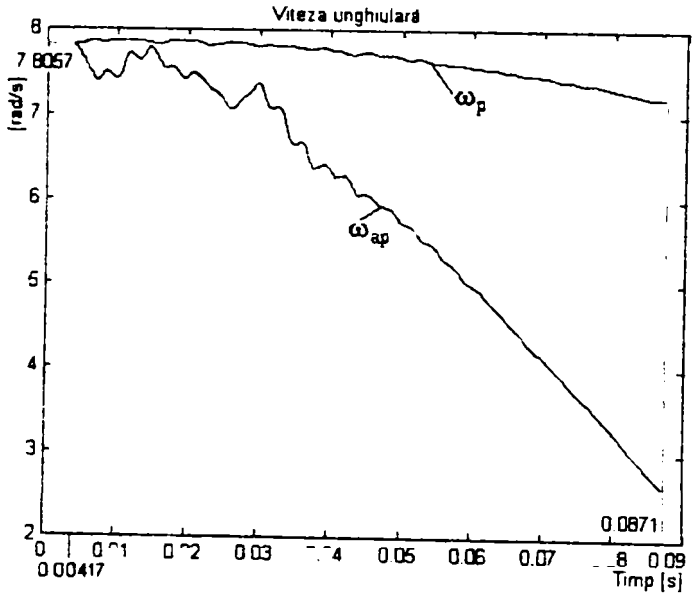


Fig. A.19.1.c.

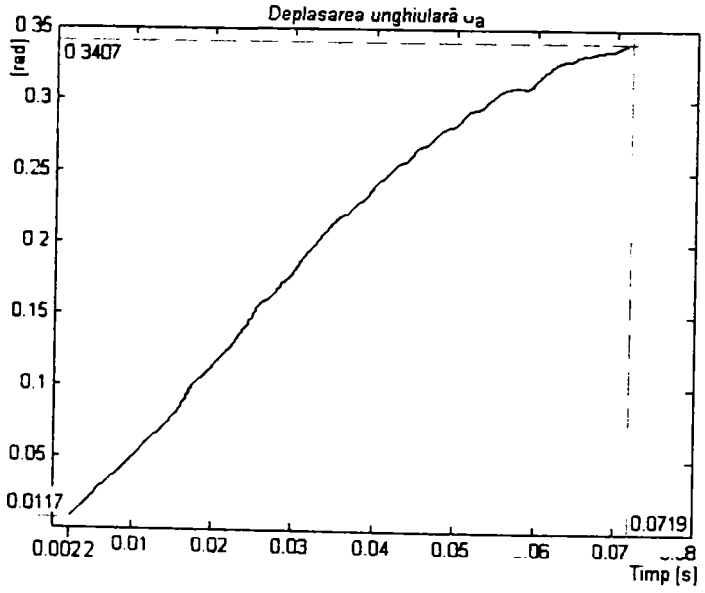


Fig. A.19.2.a.

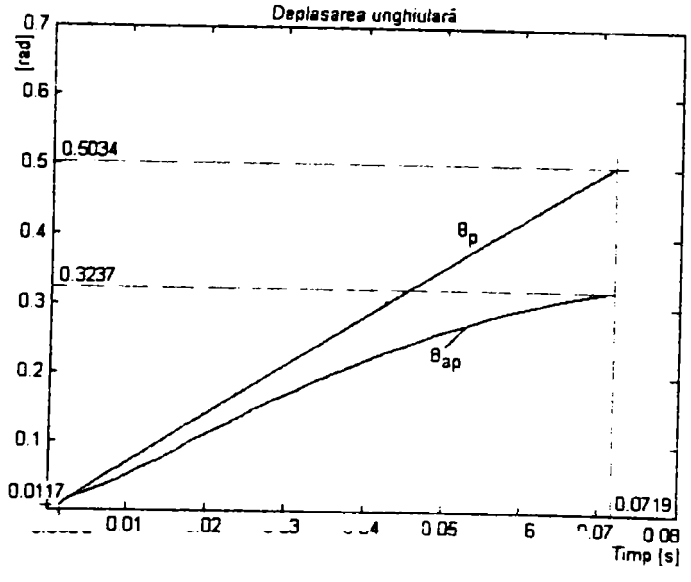


Fig. A.19.2.b.

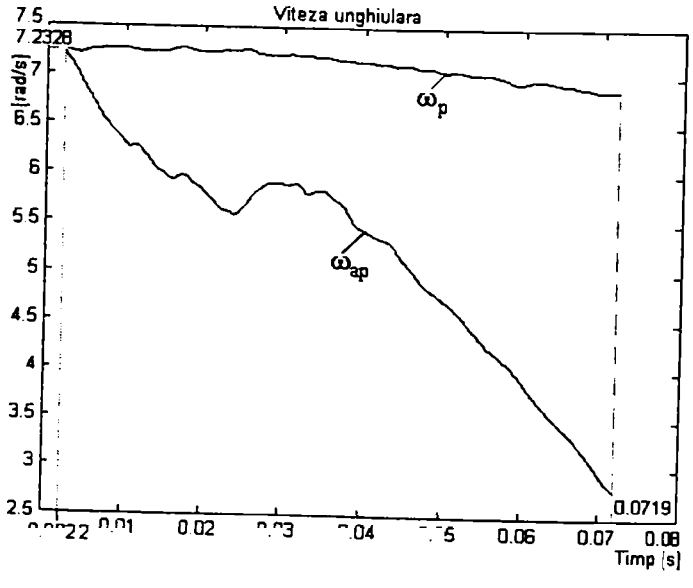


Fig. A.19.2.c.

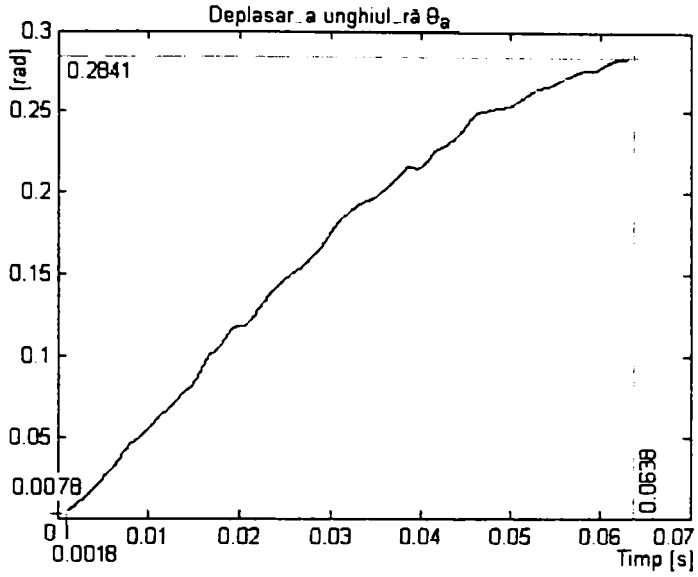


Fig. A.19.3.a.

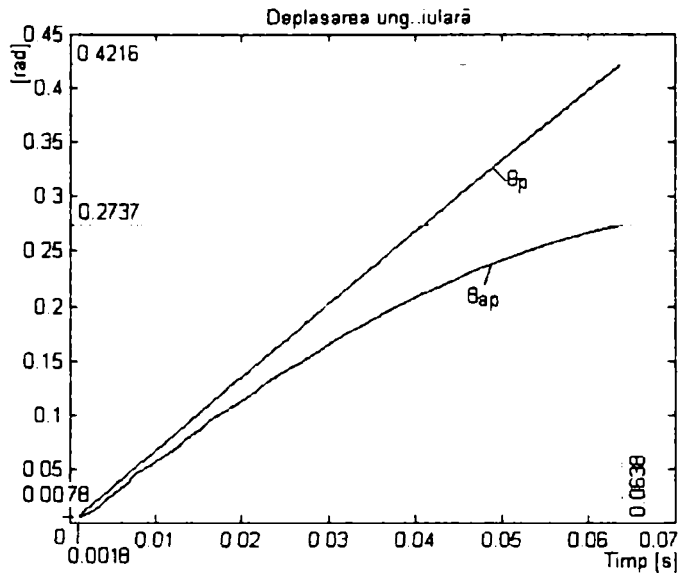


Fig. A.19.3.b.

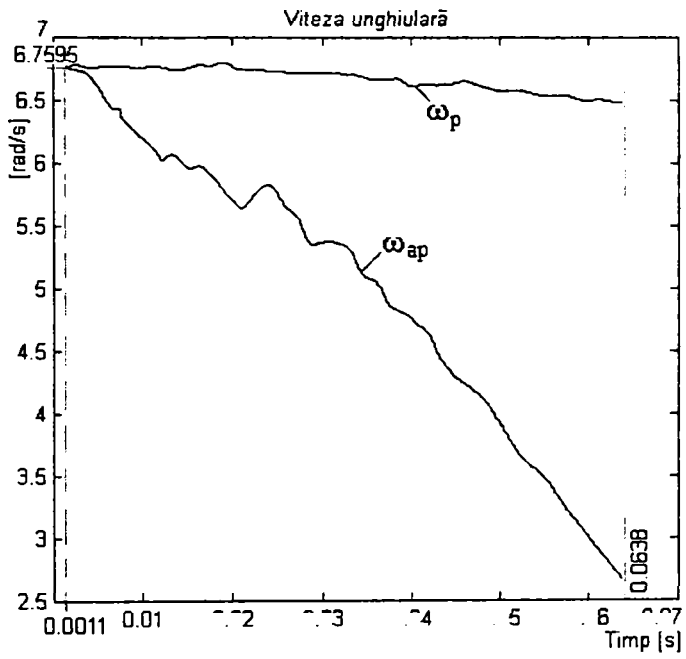


Fig. A.19.3.c.

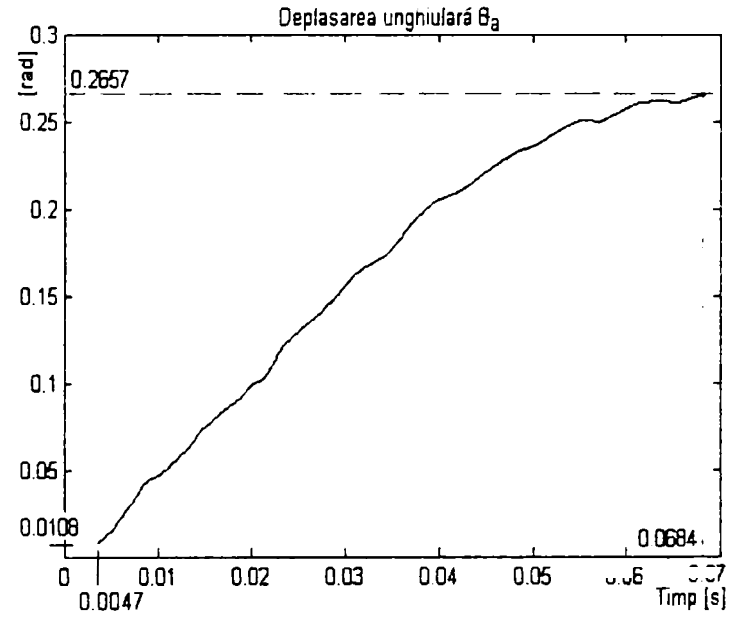


Fig. A.19.4.a.

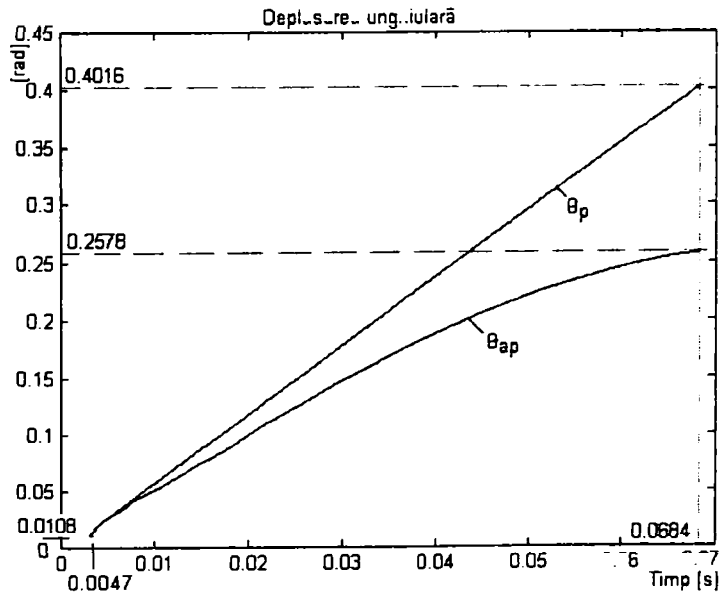


Fig. A.19.4.b.

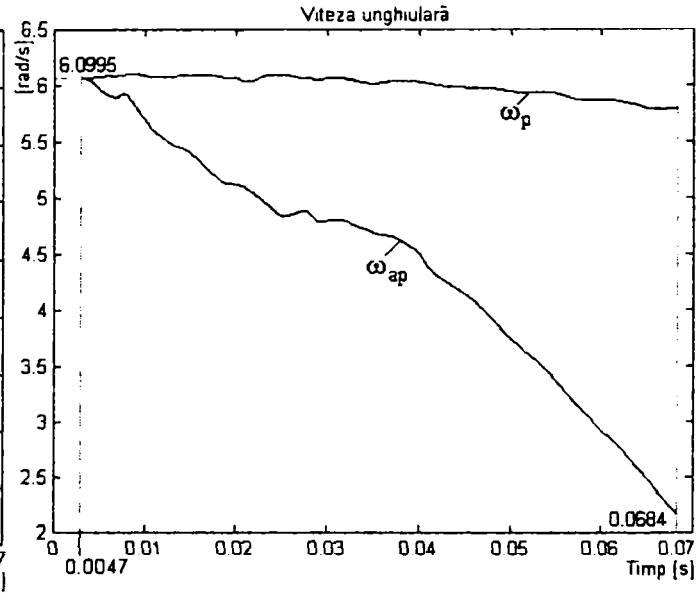


Fig. A.19. 4.c.

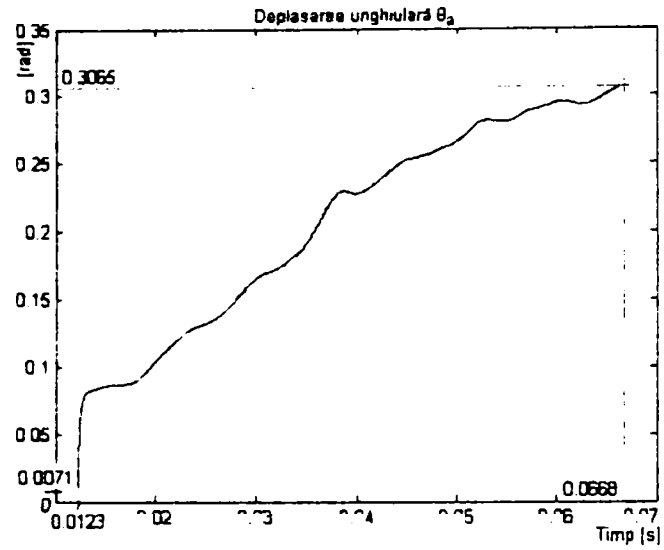


Fig. A.19.5.a.

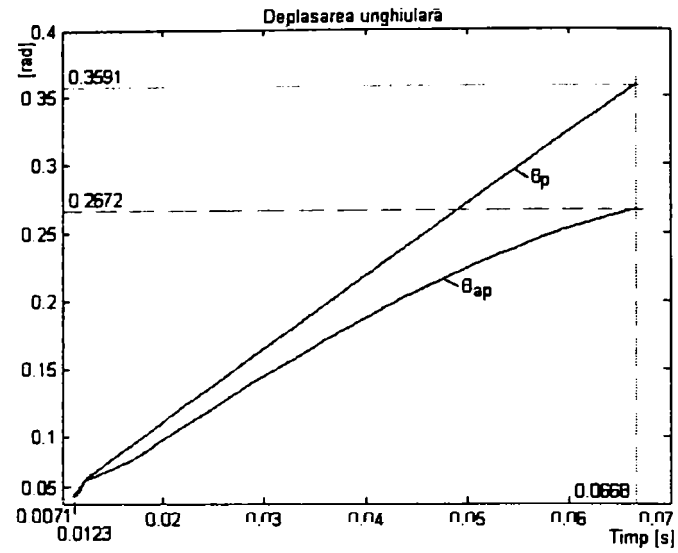


Fig. A.19.5.b.

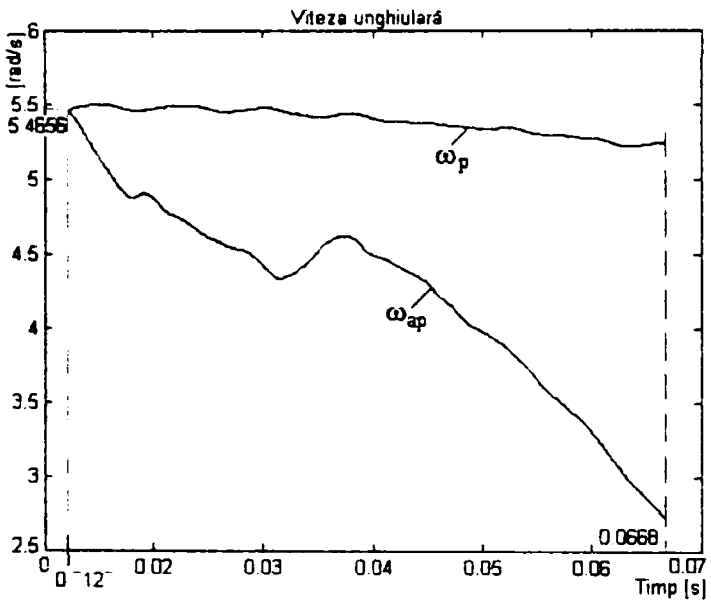


Fig. A.19.5.c.

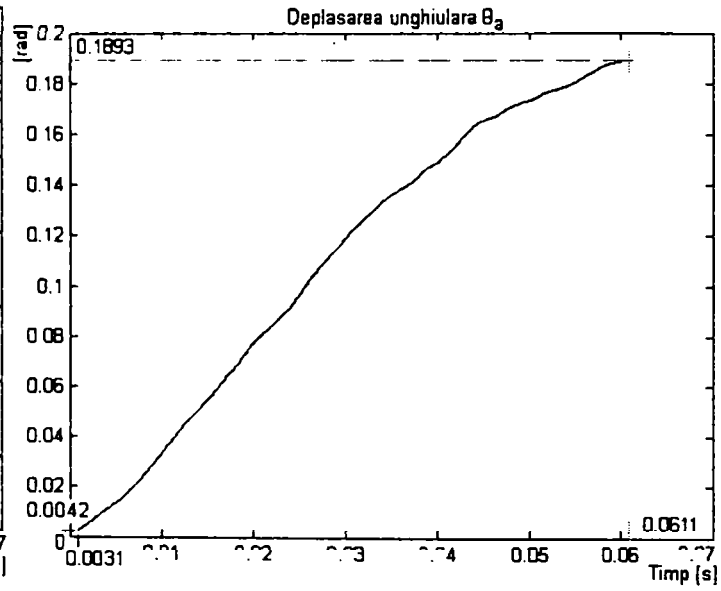


Fig. A.19.6.a.

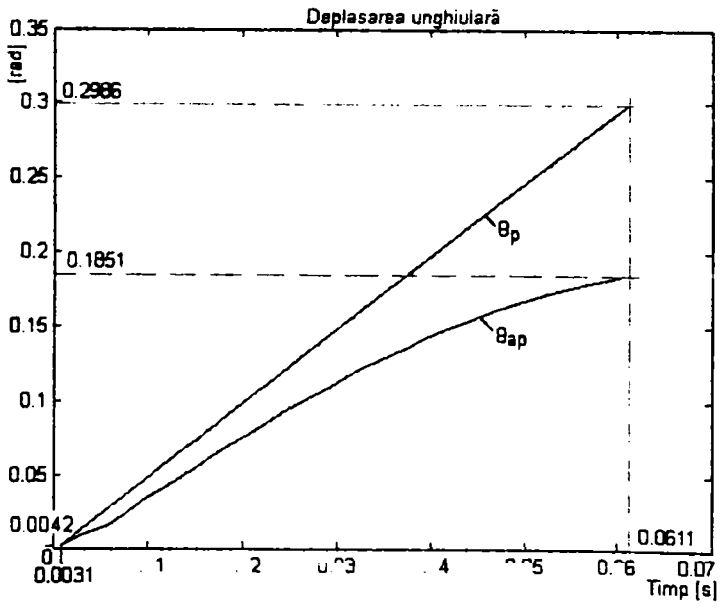


Fig. A.19.6.b.

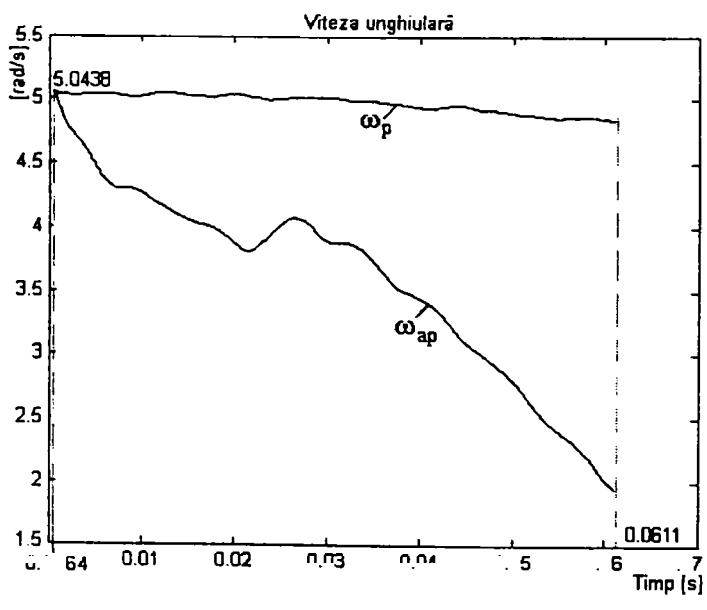


Fig. A.19.6.c.

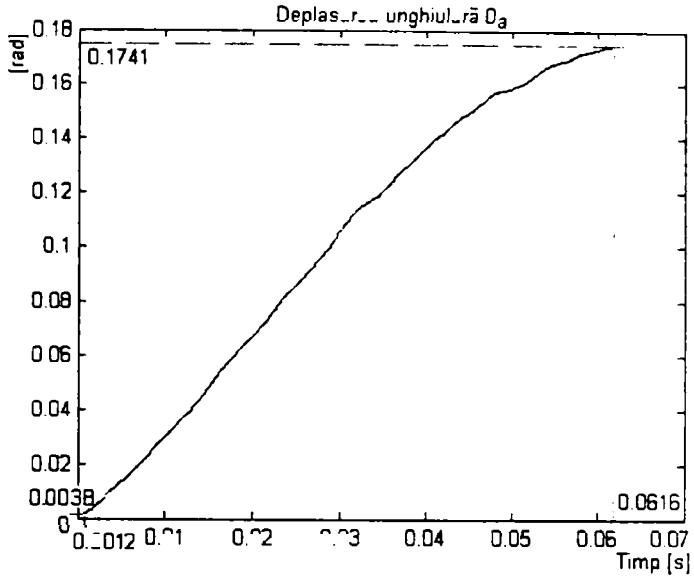


Fig. A.19.7.a.

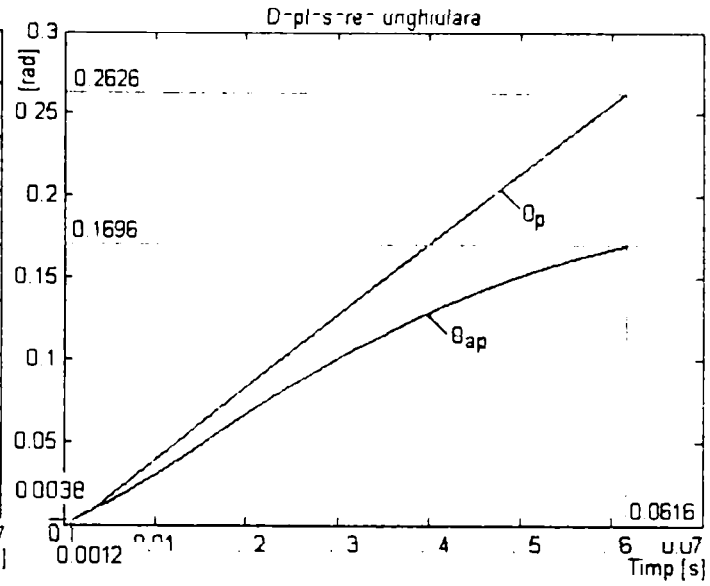


Fig. A.19.7.b.

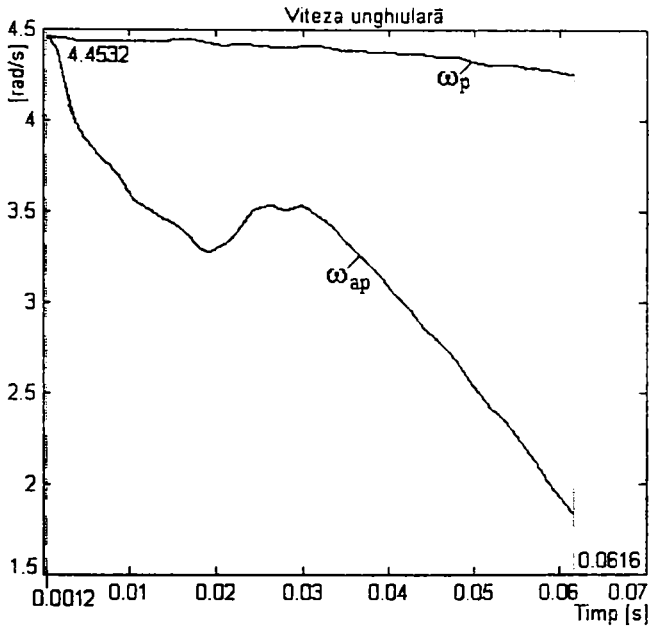


Fig. A.19.7.c.

Tabelul 7.2.

v_{A1} [m/min]	135	125	115	105	95	85	75
v_{asp} [m/min]	20,9953	19,4401	17,8849	16,3297	14,7744	13,2192	11,6640

Notă: Numerotarea figurilor corespunde astfel:

- fig.A.19.1 pentru viteza cea mai mare respectiv 135 [m/min] (20,9953), iar fig.7 pentru viteza cea mai mică respectiv 75[m/min] (11,6640) din tabelul 7.2.;

- cu litera a sunt reprezentate diagramele deplasării unghiulare θ_a a cuțitului numai datorită forțelor de aşchiere;

- cu litera b sunt reprezentate pe aceeaşi diagramă deplasările unghiulare θ_{ap} ale cuțitului în material datorită forțelor de aşchiere şi greutății pendulului (G=m·g) şi deplasările unghiulare θ_p datorate numai greutății pendulului (respectiv mersul în gol al pendulului);

- cu litera c sunt reprezentate vitezele unghiulare ω_{ap} în timpul mişcării de aşchiere, iar ω_p vitezele unghiulare la mers în gol al pendulului pentru vitezele impuse corespunzătoare (tab.7.2).

Anexa 20.

Structuri ale rădăcinii aşchiei la aşchierea ortogonlă a cuprului OFHC prin şoc


Fig. A.20.1as.x200.


Fig. A.20.1s.x200


Fig. A.20.1s.x250

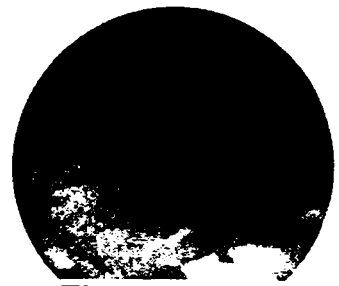


Fig. A.20.1s.x400


Fig. A.20.1s.x500


Fig. A.20.2s.x200


Fig. A.20.2s.x250

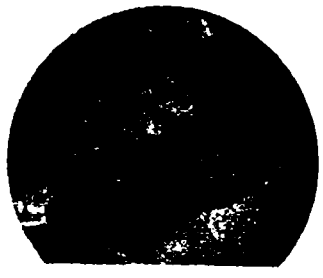

Fig. A.20.2s.x500


Fig. A.20.3s.x200


Fig. A.20.3s.x250


Fig. A.20.3s.x500


Fig. A.20.4s.x200


g. A. .4s.x250

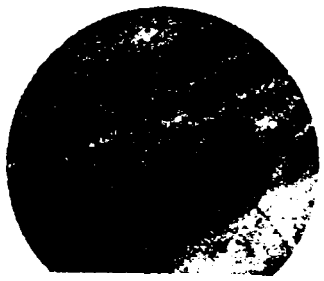


Fig. A.20.4s.x500

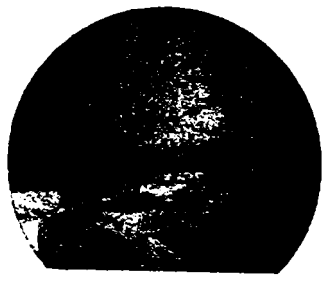


Fig. A.20.5s.x200


Fig. A.20.5s.x250


Fig. A.20.5s.x500


Fig. A.20.6s.x200


Fig. A.20.6s.x250

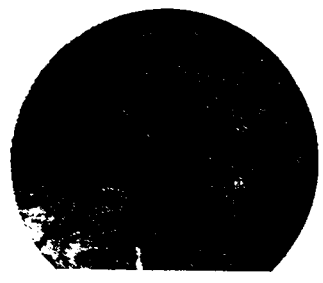


Fig. A.20.6s.x500

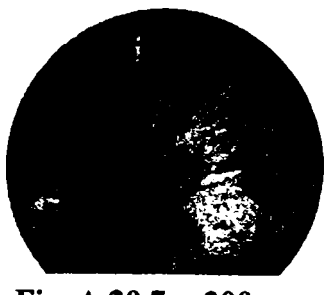


Fig. A.20.7s.x200


Fig. A.20.7s.x250


Fig. A.20.7s.x500

Notă:Numerotarea figurilor corespunde astfel: fig.1 pentru viteza cea mai mare respectiv 135 [m/min] (20,9953), iar fig.7 pentru viteza cea mai mică respectiv 75[m/min] (11,6640) din tab. 7.2.

Anexa 21.

%*Program „model_disp3.m" de modelare a dispozitivului de gaurire cu vibropercuții *****

```
clear all
global percutor_pol arbore_pol delta_s delta_sa delta_test_imp n_pauza

global int_arb1 int_per1 x_arb y_arb x_arba y_arba x_per y_per x_pera y_pera xc3a yc3a

global semn_omega3_per omega3_per omega0_arb omega3_per_imp omega0_arb_imp
omega3_per_imp

global timp_amortizare_arb timp_amortizare_per
pack;
n_pauza = 0;
% Initializare elemente
% Pasul de calcul al desimii punctelor pe profil
delta_s = 0.01;                 % delta-s≡ Δni
delta_sa = 0.2;                 % delta-sa≡Δmk
% Pasul de calcul al timpului
delta_t = 0.001;
% Cerc exterior
x_cerc_ext  = -47.5;
y_cerc_ext  = -47.5;
w_cerc_ext  = 95;
h_cerc_ext  = 95;
x1_cerc_ext = 1;
y1_cerc_ext = 1;
axis equal;
% Calculul coordonatelor pentru pozitia initiala
coord_piesa;
ttt1 = 0;
ttt2 = 0;
ttt3 = 0;
ttt4 = 0;
xmin = -70;
xmax =  70;
ymin = -70;
ymax =  70;
nr_pas = 1000;
omega0_per_real =  10;
omega3_per_real = 25;
semn_omega0_per = -1;
omega0_per = 10;
semn_omega3_per = -1;
omega3_per = omega3_per_real*(omega0_per/omega0_per_real);
omega3_per_imp = omega3_per;
timp_amortizare_per = 40;
teta_cum = 0;
delta_test_imp = 1;
semn_omega0_arb = -1;
omega0_arb_imp = 10;
omega0_arb = 0;
```

```
timp_amortizare_arb = 30;
gama_cum = 0;
timp = 0;
r_rot0 = xc3;
nr_pct1_per = size(percutor_pol);
nr_pct_per  = nr_pct1_per(1,1);
nr_pct1_arb = size(arbore_pol);
nr_pct_arb  = nr_pct1_arb(1,1);
test_start = 0;
% Bucla principala
for i = 1:nr_pas
   alfa_i = semn_omega0_per*omega0_per*i*delta_t;
   teta_j = semn_omega3_per*omega3_per*delta_t;
   for q = 1:nr_pct_per
      r_p     = percutor_pol(q,2);
      teta_p  = percutor_pol(q,1);
      x_per(q) = r_rot0*cos(alfa_i) + r_p*cos(teta_p + teta_cum + teta_j);
      y_per(q) = r_rot0*sin(alfa_i) + r_p*sin(teta_p + teta_cum + teta_j);
      xc3a(q)  = r_rot0*cos(alfa_i);
      yc3a(q)  = r_rot0*sin(alfa_i);
   end
   teta_cum = teta_cum + teta_j;
   gama_k = semn_omega0_arb*omega0_arb*delta_t;        %gama_k≡fi_rot arbore
   for q = 1:nr_pct_arb
      r_a     = arbore_pol(q,2);
      gama_a  = arbore_pol(q,1);
      x_arb(q) = r_a*cos(gama_a + gama_cum + gama_k);
      y_arb(q) = r_a*sin(gama_a + gama_cum + gama_k);
   end
   gama_cum = gama_cum + gama_k;
   if i == 1
      h_fig = figure;
      hold on
      h_arb = plot(x_arb,y_arb,'k','EraseMode','none');
      h_arb1 = plot(0,0,'k.','EraseMode','none');
      set(h_fig,'DoubleBuffer','on','Render','painters');
      axis equal;
      axis([xmin xmax ymin ymax]);
      %hold on
      h_per = plot(x_per,y_per,'r','EraseMode','none');
      h_per1 = plot(xc3a,yc3a,'r.','EraseMode','none');
      h_ext = rectangle('Position',[x_cerc_ext,y_cerc_ext,w_cerc_ext,h_cerc_ext]...
         ,'Curvature',[x1_cerc_ext,y1_cerc_ext]);

      %drawnow;
      F(i) = getframe;
   else
      set(h_arb,'xData',x_arb,'yData',y_arb);
      set(h_arb1,'xData',0,'yData',0);
      set(h_per,'xData',x_per,'yData',y_per);
      set(h_per1,'xData',xc3a,'yData',yc3a);
      drawnow;
```

```
      F(i) = getframe;
    end
    if test_start == 0;
      x_pera = x_per;
      y_pera = y_per;
      x_arba = x_arb;
      y_arba = y_arb;
    end
    test_impact1;
    test_impact2;
    test_impact3;
    test_impact4;
    vit_impact1;
    if test_start == 1;
      x_pera = x_per;
      y_pera = y_per;
      x_arba = x_arb;
      y_arba = y_arb;
    end
    test_start = 1;
end
plot(6,60,'r.');
% movie(F,3,200)
```

Anexa 22.

% *********Subprogram „coord_piesa.m" pentru definire coordonate inițiale **********

%************* Definire percutor *******************

```
% delta_s = 0.01;
```

%* Definire percutor in sistemul cartezian c0 ***

```
% Constante
xc1 = -10;
xc3 = -30;
xc2 = -33.38;
xp1 = -20.05;
yc2 = 9.05;
yp2 = 9.4;
yp6 = 24;
r0  = 19.05;
r1  = 35;
r2  = 25.85;
                % r2a=24.35;
r3  = 5.025;
xc0 = 0;
yc0 = 0;
yc1 = 0;
yc3 = 0;
yp1 = 0;
yp8 = 0;
```

```
% Relatii
xp2  = xp1;
xp6  = xp1;
xp7  = xp1;
yp3  = yp2;
yp5  = yp6;
xp8  = (xc1-r1);
yp7  = sqrt(r1^2-(xp1-xc1)^2);
xp3  = -sqrt(r0^2-yp3^2);
xp5  = xc2+sqrt(r2^2-(yp6-yc2)^2);
a_p4 = (xc2^2+yc2^2+r0^2-r2^2)/2;
xp4  = (2*a_p4*xc2 + sqrt(4*a_p4^2*xc2^2-4*(a_p4^2-
r0^2*yc2^2)*(xc2^2+yc2^2)))/(2*(xc2^2+yc2^2));
yp4  = sqrt(r0^2-xp4^2);

% Coordonate
xp9  = xp7;
yp9  = -yp7;
xp10 = xp6;
yp10 = -yp6;
xp11 = xp5;
yp11 = -yp5;
xp12 = xp4;
yp12 = -yp4;
```

```
xp13 = xp3;
yp13 = -yp3;
xp14 = xp2;
yp14 = -yp2;

% Definire matrice de coordonate percutor
percutor1a = gen_p1_p2(xp1,yp1,xp2,yp2);

percutor1 = [xp1,yp1
        %percutor1a
        xp2,yp2];

percutor2a = gen_p2_p3(xp2,yp2,xp3,yp3);

percutor2 = [xp2,yp2
         percutor2a
         xp3,yp3];

percutor3a = gen_p3_p4(xp3,yp3,r0,xc0,yc0,xp4,yp4);

percutor3 = [xp3,yp3
         percutor3a
         xp4,yp4];

percutor4a = gen_p4_p5((xc2-xp4),(yp4-yc2),r2,xc2,yc2,(xc2-xp5),(yp5-yc2));

percutor4 = [xp4,yp4
         percutor4a
         xp5,yp5];

percutor5 = [xp5 ,yp5
         xp6 ,yp6
         xp7 ,yp7];

percutor6a = gen_p7_p8((xp7-xc1),yp7,r1,xc1,yc1,(xp8-xc1),yp8,delta_sa);

percutor6 = [xp7,yp7
        percutor6a
        xp8,yp8];

percutor7_y = -percutor6(:,2);
percutor7_x =  percutor6(:,1);
percutor7a = flipud([percutor7_x,percutor7_y]);

percutor7 = [xp8,yp8
        percutor7a
        xp9,yp9];

percutor8 = [xp9,yp9
        xp10,yp10
        xp11,yp11];
```

```
percutor9_y = -percutor4(:,2);
percutor9_x =  percutor4(:,1);
percutor9a = flipud([percutor9_x,percutor9_y]);

percutor9 = [xp11,yp11
        percutor9a
        xp12,yp12];

percutor10_y = -percutor3(:,2);
percutor10_x =  percutor3(:,1);
percutor10a = flipud([percutor10_x,percutor10_y]);

percutor10 = [xp12,yp12
         percutor10a
         xp13,yp13];

percutor11_y = -percutor2(:,2);
percutor11_x =  percutor2(:,1);
percutor11a = flipud([percutor11_x,percutor11_y]);

percutor11 = [xp13,yp13
        percutor11a
        xp14,yp14];

percutor12_y = -percutor1(:,2);
percutor12_x =  percutor1(:,1);
percutor12a = flipud([percutor12_x,percutor12_y]);

percutor12 = [xp14,yp14
        %percutor12a
        xp1,yp1];

% Definirea intervalelor pentru elementele profilului percutor
int_per1 = size(percutor1);
int_per2 = size(percutor2);
int_per3 = size(percutor3);
int_per4 = size(percutor4);
int_per5 = size(percutor5);
int_per6 = size(percutor6);
int_per7 = size(percutor7);
int_per8 = size(percutor8);
int_per9 = size(percutor9);
int_per10 = size(percutor10);
int_per11 = size(percutor11);
int_per12 = size(percutor12);

int_per =  [int_per1(1,1)
        int_per2(1,1)
        int_per3(1,1)
        int_per4(1,1)
        int_per5(1,1)
        int_per6(1,1)
```

```
        int_per7(1,1)
        int_per8(1,1)
        int_per9(1,1)
        int_per10(1,1)
        int_per11(1,1)
        int_per12(1,1)];
int_per1 = zeros(12,2);
int_per1(1,1) = 1;
int_per1(1,2) = int_per(1,1);
for i = 2:12
   int_per1(i,1) = sum(int_per(1:i-1))+1;
   int_per1(i,2) = sum(int_per(1:i));
end
```

% Matricea coordonatelor punctelor de pe profilul percutorului

```
percutor_c0 = [percutor1
        percutor2
        percutor3
        percutor4
        percutor5
        percutor6
        percutor7
        percutor8
        percutor9
        percutor10
        percutor11
        percutor12];
```

%* Definire percutor in sistemul cartezian c3 (coordonate u0 si v0)*****

```
percutor_c3(:,1) = percutor_c0(:,1) - xc3;
percutor_c3(:,2) = percutor_c0(:,2);

%*** Definire percutor in sistemul polar c3 ***

[teta_p,r_p] = cart2pol(percutor_c3(:,1),percutor_c3(:,2));
percutor_pol = [teta_p,r_p];
```

%******** Definire arbore **********

```
% Constante

xp15 = 5.82;
xp16 = 7.5;
r4   = 12.45;
r5   = 19;

% Relatii

yp15 = sqrt(r4^2-xp15^2);
yp16 = sqrt(r5^2-xp16^2);

% Coordonate
xp17 = -xp16;
```

```
yp17 =  yp16;
xp18 = -xp15;
yp18 =  yp15;
xp19 =  xp18;
yp19 = -yp18;
xp20 =  xp17;
yp20 = -yp17;
xp21 =  xp16;
yp21 = -yp16;
xp22 =  xp15;
yp22 = -yp15;


arbore1a = gen_p22_p15(xp22,yp22,r4,xc0,yc0,xp15,yp15);

arbore1 = [xp22,yp22
        arbore1a
        xp15,yp15];

arbore2a = gen_p15_p16(xp15,yp15,xp16,yp16);

arbore2 = [xp15,yp15
        arbore2a
        xp16,yp16];

arbore3a = gen_p16_p17(xp16,yp16,r5,xc0,yc0,xp17,yp17);

arbore3 = [xp16,yp16
        arbore3a
        xp17,yp17];

arbore4a = gen_p17_p18(xp17,yp17,xp18,yp18);

arbore4 = [xp17,yp17
        arbore4a
        xp18,yp18];

arbore5a = gen_p18_p19(xp18,yp18,r4,xc0,yc0,xp19,yp19);

arbore5 = [xp18,yp18
        arbore5a
        xp19,yp19];

arbore6a = flipud(gen_p19_p20(xp20,yp20,xp19,yp19));

arbore6 = [xp19,yp19
        arbore6a
        xp20,yp20];

arbore7a = gen_p20_p21(xp20,yp20,r5,xc0,yc0,xp21,yp21);

arbore7 = [xp20,yp20
```

```
        arbore7a
        xp21,yp21];

arbore8a = flipud(gen_p21_p22(xp22,yp22,xp21,yp21));

arbore8 = [xp21,yp21
        arbore8a
        xp22,yp22];

% Definirea intervalelor pentru elementele profilului arbore
int_arb1 = size(arbore1);
int_arb2 = size(arbore2);
int_arb3 = size(arbore3);
int_arb4 = size(arbore4);
int_arb5 = size(arbore5);
int_arb6 = size(arbore6);
int_arb7 = size(arbore7);
int_arb8 = size(arbore8);
int_arb = [int_arb1(1,1)
        int_arb2(1,1)
        int_arb3(1,1)
        int_arb4(1,1)
        int_arb5(1,1)
        int_arb6(1,1)
        int_arb7(1,1)
        int_arb8(1,1)];
int_arb1 = zeros(8,2);
int_arb1(1,1) = 1;
int_arb1(1,2) = int_arb(1,1);
for i = 2:8
   int_arb1(i,1) = sum(int_arb(1:i-1))+1;
   int_arb1(i,2) = sum(int_arb(1:i));
end

%*** Definire arbore in sistem cartezian ***
arbore = [arbore1
        arbore2
        arbore3
        arbore4
        arbore5
        arbore6
        arbore7
        arbore8];

%*** Definire arbore in sistem polar ***
[gama_a,r_a] = cart2pol(arbore(:,1),arbore(:,2));
arbore_pol = [gama_a,r_a];

%plot(percutor_c0(:,1),percutor_c0(:,2),'r.');
%hold on
%plot(percutor_c3(:,1),percutor_c3(:,2),'r.');
%plot(arbore(:,1),arbore(:,2),'k');
```

```
%axis equal;
```

% Programul de generare subrutinele „gen_p.._p..”

```
function percutor_n = gen_p1_p2(x_start,y_start,x_stop,y_stop)

global delta_s

nr_iter   = floor(abs((y_start - y_stop)/delta_s));
percutor_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   percutor_n(j1,1) = x_start;
   percutor_n(j1,2) = y_start + j1*delta_s;
end


function arbore_n = gen_p15_p16(x_start,y_start,x_stop,y_stop)

global delta_s

m_p_p = ((y_start - y_stop)/(x_start - x_stop));
nr_iter   = floor(abs((x_start - x_stop)/delta_s));
arbore_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   arbore_n(j1,1) = x_start + j1*delta_s;
   arbore_n(j1,2) = m_p_p*(arbore_n(j1,1) - x_start) + y_start;
end
```

```
function arbore_n = gen_p16_p17(x_start,y_start,raza,x_c,y_c,x_stop,y_stop)

global delta_s

csi_start = atan(abs(y_start/x_start));
csi_stop  = pi - atan(abs(y_stop/x_stop));
nr_iter   = floor(abs((csi_stop-csi_start)/delta_s));
arbore_n = zeros(nr_iter,2);
j1 = 1;
for j1 = 1:nr_iter
   arbore_n(j1,1) = raza*cos(csi_start + j1*delta_s) + x_c;
   arbore_n(j1,2) = raza*sin(csi_start + j1*delta_s) + y_c;
end

function arbore_n = gen_p17_p18(x_start,y_start,x_stop,y_stop)

global delta_s

m_p_p = ((y_start - y_stop)/(x_start - x_stop));
nr_iter   = floor(abs((x_start - x_stop)/delta_s));
arbore_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   arbore_n(j1,1) = x_start + j1*delta_s;
   arbore_n(j1,2) = m_p_p*(arbore_n(j1,1) - x_start) + y_start;
```

```
end
function arbore_n = gen_p18_p19(x_start,y_start,raza,x_c,y_c,x_stop,y_stop)

global delta_s

csi_start = pi - atan(abs(y_start/x_start));
csi_stop  = pi + atan(abs(y_stop/x_stop));
nr_iter   = floor(abs((csi_stop-csi_start)/delta_s));
arbore_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   arbore_n(j1,1) = raza*cos(csi_start + j1*delta_s) + x_c;
   arbore_n(j1,2) = raza*sin(csi_start + j1*delta_s) + y_c;
end


function arbore_n = gen_p19_p20(x_start,y_start,x_stop,y_stop)

global delta_s

m_p_p = ((y_start - y_stop)/(x_start - x_stop));
nr_iter   = floor(abs((x_start - x_stop)/delta_s));
arbore_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   arbore_n(j1,1) = x_start + j1*delta_s;
   arbore_n(j1,2) = m_p_p*(arbore_n(j1,1) - x_start) + y_start;
end
```

```
function percutor_n = gen_p2_p3(x_start,y_start,x_stop,y_stop)

global delta_s

nr_iter   = floor(abs((x_start - x_stop)/delta_s));
percutor_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   percutor_n(j1,1) = x_start + j1*delta_s;
   percutor_n(j1,2) = y_start;
end


function arbore_n = gen_p20_p21(x_start,y_start,raza,x_c,y_c,x_stop,y_stop)

global delta_s

csi_start = pi + atan(abs(y_start/x_start));
csi_stop  = 2*pi - atan(abs(y_stop/x_stop));
nr_iter   = floor(abs((csi_stop-csi_start)/delta_s));
arbore_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   arbore_n(j1,1) = raza*cos(csi_start + j1*delta_s) + x_c;
   arbore_n(j1,2) = raza*sin(csi_start + j1*delta_s) + y_c;
end
```

```
function arbore_n = gen_p22_p15(x_start,y_start,raza,x_c,y_c,x_stop,y_stop)

global delta_s

csi_start = -atan(abs(y_start/x_start));
csi_stop  = atan(abs(y_stop/x_stop));
nr_iter   = floor(abs((csi_stop-csi_start)/delta_s));
arbore_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   arbore_n(j1,1) = raza*cos(csi_start + j1*delta_s) + x_c;
   arbore_n(j1,2) = raza*sin(csi_start + j1*delta_s) + y_c;
end
```

```
function percutor_n = gen_p3_p4(x_start,y_start,raza,x_c,y_c,x_stop,y_stop)

global delta_s

csi_start = pi - atan(abs(y_start/x_start));
csi_stop  = pi - atan(abs(y_stop/x_stop));
nr_iter   = floor(abs((csi_stop-csi_start)/delta_s));
percutor_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   percutor_n(j1,1) = raza*cos(csi_start-j1*delta_s) + x_c;
   percutor_n(j1,2) = raza*sin(csi_start-j1*delta_s) + y_c;
end
```

```
function percutor_n = gen_p4_p5(x_start,y_start,raza,x_c,y_c,x_stop,y_stop)

global delta_s

csi_start = atan(abs(y_start/x_start));
csi_stop  = atan(abs(y_stop/x_stop));
nr_iter   = floor(abs((csi_stop-csi_start)/delta_s));
percutor_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   percutor_n(j1,1) = raza*cos(csi_start + j1*delta_s) + x_c;
   percutor_n(j1,2) = raza*sin(csi_start + j1*delta_s) + y_c;
end
```

```
function percutor_n = gen_p7_p8(x_start,y_start,raza,x_c,y_c,x_stop,y_stop,delta_s)
csi_start = pi - atan(abs(y_start/x_start));
csi_stop  = pi - atan(abs(y_stop/x_stop));
nr_iter   = floor(abs((csi_stop-csi_start)/delta_s));
percutor_n = zeros(nr_iter,2);
for j1 = 1:nr_iter
   percutor_n(j1,1) = raza*cos(csi_start + j1*delta_s) + x_c;
   percutor_n(j1,2) = raza*sin(csi_start + j1*delta_s) + y_c;
end
```

Anexa 23.

```
                % Testarea impactului între percutor si arbore („test_impact1")
        % int_arb1 int_per1 x_arb y_arb x_per y_per
global delta_test_imp n_pauza

global semn_omega3_per omega3_per omega0_arb omega3_per_imp omega0_arb_imp

                        % ********* Caz I  - Impact Percutor - Arbore ***********
          % ******* Testare contact suprafata P4P5 cu suprafata P17P18 ********
    plot(60,60,'b.');
    plot(60,60,'bo');
    drawnow;
    for iti = 1:int_arb1(4,2) - int_arb1(4,1)
      t_p4p5_p17p18(iti)  = min(sqrt((x_per(int_per1(4,1):int_per1(4,2)) -...
                                x_arb(iti + int_arb1(4,1))).^2 + ...
                               (y_per(int_per1(4,1):int_per1(4,2))-...
                                y_arb(iti + int_arb1(4,1))).^2));
    end
    t_p4p5_p17p18b = min(t_p4p5_p17p18);
    delta_test_imp = sqrt(((x_pera(int_per1(4,1)) -...
                x_per(int_per1(4,1))).^2 + (y_pera(int_per1(4,1)) -...
                y_per(int_per1(4,1))).^2));
    if t_p4p5_p17p18b < delta_test_imp
       semn_omega3_per = 1;
       omega0_arb = omega0_arb_imp;
       omega3_per = omega3_per_imp;
       plot(60,60,'g.');
       drawnow;
       pause(n_pauza);
    end
          % ********* Testare contact suprafata P4P5 cu suprafata P22P21 *********
    for iti = 1:int_arb1(8,2) - int_arb1(8,1)
      t_p4p5_p22p21(iti)  = min(sqrt((x_per(int_per1(4,1):int_per1(4,2)) -...
                                x_arb(iti + int_arb1(8,1))).^2 + ...
                               (y_per(int_per1(4,1):int_per1(4,2))-...
                                y_arb(iti + int_arb1(8,1))).^2));
    end
    t_p4p5_p22p21b = min(t_p4p5_p22p21);
    if t_p4p5_p22p21b < delta_test_imp
       semn_omega3_per = 1;
       omega0_arb = omega0_arb_imp;
       omega3_per = omega3_per_imp;
       plot(60,60,'g.');
       drawnow;
       pause(n_pauza);
    end
          % ******* Testare contact suprafata P11P12 cu suprafata P19P20 *********
    for iti = 1:int_arb1(6,2) - int_arb1(6,1)
      t_p11p12_p19p20(iti)  = min(sqrt((x_per(int_per1(9,1):int_per1(9,2)) -...
                                x_arb(iti + int_arb1(6,1))).^2 + ...
                               (y_per(int_per1(9,1):int_per1(9,2))-...
                                y_arb(iti + int_arb1(6,1))).^2));
```

```
end
t_p11p12_p19p20b = min(t_p11p12_p19p20);
delta_test_imp = sqrt(((x_pera(int_per1(9,1)) -...
            x_per(int_per1(9,1))).^2 + (y_pera(int_per1(9,1)) -...
            y_per(int_per1(9,1))).^2));
if t_p11p12_p19p20b < delta_test_imp
   semn_omega3_per = -1;
   omega0_arb = 0;
   omega3_per = omega3_per_imp;
   plot(60,60,'go');
   drawnow;
   pause(n_pauza);
end
      % ******* Testare contact suprafata P11P12 cu suprafata P15P16 *********
for iti = 1:int_arb1(2,2) - int_arb1(2,1)
   t_p11p12_p15p16(iti)  = min(sqrt((x_per(int_per1(9,1):int_per1(9,2)) -...
                           x_arb(iti + int_arb1(2,1))).^2 + ...
                          (y_per(int_per1(9,1):int_per1(9,2))-...
                           y_arb(iti + int_arb1(2,1))).^2));
end
t_p11p12_p15p16b = min(t_p11p12_p15p16);
if t_p11p12_p15p16b < delta_test_imp
   semn_omega3_per = -1;
   omega0_arb = 0;
   omega3_per = omega3_per_imp;
   plot(60,60,'go');
   drawnow;
   pause(n_pauza);
end
```

```
                % Testarea impactului intre percutor si arbore („test_impact2")
% int_arb1 int_per1 x_arb y_arb x_per y_per n_pauza
global delta_test_imp n_pauza

global semn_omega3_per omega3_per omega0_arb omega3_per_imp

      % ********* Caz I  - Impact Percutor - Arbore **********
      % ********** Testare contact punct P4 cu suprafetele arborelui **********
      % ********* Testarea contact P4 cu suprafata P18P19 (arbore5) ********
   plot(60,60,'b.');
   plot(60,60,'bo');
   drawnow;
   t_p4_p18p19 = sqrt(min((x_arb(int_arb1(5,1):int_arb1(5,2)) -...
            x_per(int_per1(4,1))).^2 + (y_arb(int_arb1(5,1):int_arb1(5,2))-...
            y_per(int_per1(4,1))).^2));
   delta_test_imp = sqrt(((x_pera(int_per1(4,1)) -...
            x_per(int_per1(4,1))).^2 + (y_pera(int_per1(4,1)) -...
            y_per(int_per1(4,1))).^2));
   if t_p4_p18p19 < delta_test_imp
      semn_omega3_per = 1;
      omega3_per = omega3_per_imp;
```

```
   plot(60,60,'r.');
   drawnow;
   pause(n_pauza);
end
   % ******** Testarea contact P4 cu suprafata P17P16 (arbore3) ********
t_p4_p17p16 = sqrt(min((x_arb(int_arb1(3,1):int_arb1(3,2)) -...
            x_per(int_per1(4,1))).^2 + (y_arb(int_arb1(3,1):int_arb1(3,2))-...
            y_per(int_per1(4,1))).^2));
if t_p4_p17p16 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'r.');
   drawnow;
   pause(n_pauza);
end
   % ************ Testarea contact P4 cu suprafata P15P22 (arbore1) ************
t_p4_p15p22 = sqrt(min((x_arb(int_arb1(1,1):int_arb1(1,2)) -...
            x_per(int_per1(4,1))).^2 + (y_arb(int_arb1(1,1):int_arb1(1,2))-...
            y_per(int_per1(4,1))).^2));
if t_p4_p15p22 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'r.');
   drawnow;
   pause(n_pauza);
end
   % ************ Testarea contact P4 cu suprafata P20P21 (arbore7) ************
t_p4_p20p21 = sqrt(min((x_arb(int_arb1(7,1):int_arb1(7,2)) -...
            x_per(int_per1(4,1))).^2 + (y_arb(int_arb1(7,1):int_arb1(7,2))-...
            y_per(int_per1(4,1))).^2));
if t_p4_p20p21 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'r.');
   drawnow;
   pause(n_pauza);
end
   % ************ Testare contact punct P12 cu suprafetele arborelui **************
   % ****** Testarea contact P12 cu suprafata P18P19 (arbore5) ******
t_p12_p18p19 = sqrt(min((x_arb(int_arb1(5,1):int_arb1(5,2)) -...
            x_per(int_per1(10,1))).^2 + (y_arb(int_arb1(5,1):int_arb1(5,2))-...
            y_per(int_per1(10,1))).^2));
delta_test_imp = sqrt(((x_pera(int_per1(10,1)) -...
            x_per(int_per1(10,1))).^2 + (y_pera(int_per1(10,1)) -...
            y_per(int_per1(10,1))).^2));
if t_p12_p18p19 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'ro');
   drawnow;
   pause(n_pauza);
end
```

```
   % ************ Testarea contact P12 cu suprafata P17P16 (arbore3) ************
t_p12_p17p16 = sqrt(min((x_arb(int_arb1(3,1):int_arb1(3,2)) -...
            x_per(int_per1(10,1))).^2 + (y_arb(int_arb1(3,1):int_arb1(3,2))-...
            y_per(int_per1(10,1))).^2));
if t_p12_p17p16 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'ro');
   drawnow;
   pause(n_pauza);
end
   % ************ Testarea contact P12 cu suprafata P15P22 (arbore1) ************
t_p12_p15p22 = sqrt(min((x_arb(int_arb1(1,1):int_arb1(1,2)) -...
            x_per(int_per1(10,1))).^2 + (y_arb(int_arb1(1,1):int_arb1(1,2))-...
            y_per(int_per1(10,1))).^2));
if t_p12_p15p22 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'ro');
   drawnow;
   pause(n_pauza);
end
   % ************ Testarea contact P12 cu suprafata P20P21 (arbore7) ************
t_p12_p20p21 = sqrt(min((x_arb(int_arb1(7,1):int_arb1(7,2)) -...
            x_per(int_per1(10,1))).^2 + (y_arb(int_arb1(7,1):int_arb1(7,2))-...
            y_per(int_per1(10,1))).^2));
if t_p12_p20p21 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'ro');
   drawnow;
   pause(n_pauza);
end
```

% Testarea impactului intre percutor si arbore („test_impact3")

```
% int_arb1 int_per1 x_arb y_arb x_per y_per
global delta_test_imp n_pauza

global semn_omega3_per omega3_per omega0_arb omega3_per_imp

   % ********* Caz I - Impact Percutor - Arbore **********
    %************ Testare contact punct P3 cu suprafetele arborelui **************
% ************ Testarea contact P3 cu suprafata P17P16 (arbore3) ************
plot(60,60,'b.');
plot(60,60,'bo');
drawnow;
t_p3_p17p16 = sqrt(min((x_arb(int_arb1(3,1):int_arb1(3,2)) -...
            x_per(int_per1(3,1))).^2 + (y_arb(int_arb1(3,1):int_arb1(3,2))-...
            y_per(int_per1(3,1))).^2));
delta_test_imp = sqrt(((x_pera(int_per1(3,1)) -...
            x_per(int_per1(3,1))).^2 + (y_pera(int_per1(3,1)) -...
```

```
            y_per(int_per1(3,1))).^2));
if t_p3_p17p16 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'y.');
   drawnow;
   pause(n_pauza);
end
        % ******* Testarea contact P3 cu suprafata P16P15 (arbore2) ************
t_p3_p16p15 = sqrt(min((x_arb(int_arb1(2,1):int_arb1(2,2)) -...
            x_per(int_per1(3,1))).^2 + (y_arb(int_arb1(2,1):int_arb1(2,2))-...
            y_per(int_per1(3,1))).^2));
if t_p3_p16p15 < delta_test_imp
   semn_omega3_per = 1;
   %omega0_arb = 0;
   omega3_per = omega3_per_imp;
   plot(60,60,'y.');
   drawnow;
   pause(n_pauza);
end
        % ******* Testarea contact P3 cu suprafata P19P20 (arbore6) ************
t_p3_p19p20 = sqrt(min((x_arb(int_arb1(6,1):int_arb1(6,2)) -...
            x_per(int_per1(3,1))).^2 + (y_arb(int_arb1(6,1):int_arb1(6,2))-...
            y_per(int_per1(3,1))).^2));
if t_p3_p19p20 < delta_test_imp
   semn_omega3_per = 1;
   %omega0_arb = 0;
   omega3_per = omega3_per_imp;
   plot(60,60,'y.');
   drawnow;
   pause(n_pauza);
end
        % ******* Testarea contact P3 cu suprafata P20P21 (arbore7) ************
t_p3_p20p21 = sqrt(min((x_arb(int_arb1(7,1):int_arb1(7,2)) -...
            x_per(int_per1(3,1))).^2 + (y_arb(int_arb1(7,1):int_arb1(7,2))-...
            y_per(int_per1(3,1))).^2));
if t_p3_p20p21 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'y.');
   drawnow;
   pause(n_pauza);
end
          %****** Testare contact punct P13 cu suprafetele arborelui **************
        % ******* Testarea contact P13 cu suprafata P17P16 (arbore3) ************
t_p13_p17p16 = sqrt(min((x_arb(int_arb1(3,1):int_arb1(3,2)) -...
            x_per(int_per1(11,1))).^2 + (y_arb(int_arb1(3,1):int_arb1(3,2))-...
            y_per(int_per1(11,1))).^2));
delta_test_imp = sqrt(((x_pera(int_per1(11,1)) -...
            x_per(int_per1(11,1))).^2 + (y_pera(int_per1(11,1)) -...
            y_per(int_per1(11,1))).^2));
if t_p13_p17p16 < delta_test_imp
```

```
    semn_omega3_per = -1;
    omega3_per = omega3_per_imp;
    plot(60,60,'yo');
    drawnow;
    pause(n_pauza);
end
        % ****** Testarea contact P13 cu suprafata P17P18 (arbore2) ************
t_p13_p17p18 = sqrt(min((x_arb(int_arb1(4,1):int_arb1(4,2)) -...
            x_per(int_per1(11,1))).^2 + (y_arb(int_arb1(4,1):int_arb1(4,2))-...
            y_per(int_per1(11,1))).^2));
if t_p13_p17p18 < delta_test_imp
    semn_omega3_per = -1;
    %omega0_arb = 0;
    omega3_per = omega3_per_imp;
    plot(60,60,'yo');
    drawnow;
    pause(n_pauza);
end
        % ******* Testarea contact P13 cu suprafata P21P22 (arbore6) ************
t_p13_p21p22 = sqrt(min((x_arb(int_arb1(8,1):int_arb1(8,2)) -...
            x_per(int_per1(11,1))).^2 + (y_arb(int_arb1(8,1):int_arb1(8,2))-...
            y_per(int_per1(11,1))).^2));
if t_p13_p21p22 < delta_test_imp
    semn_omega3_per = -1;
    %omega0_arb = 0;
    omega3_per = omega3_per_imp;
    plot(60,60,'yo');
    drawnow;
end
        % ****** Testarea contact P13 cu suprafata P20P21 (arbore7) ************
t_p13_p20p21 = sqrt(min((x_arb(int_arb1(7,1):int_arb1(7,2)) -...
            x_per(int_per1(11,1))).^2 + (y_arb(int_arb1(7,1):int_arb1(7,2))-...
            y_per(int_per1(11,1))).^2));
if t_p13_p20p21 < delta_test_imp
    semn_omega3_per = -1;
    omega3_per = omega3_per_imp;
    plot(60,60,'yo');
    drawnow;
    pause(n_pauza);
end


            % Testarea impactului intre percutor si arbore („test_impact4")
% int_arb1 int_per1 x_arb y_arb x_per y_per
global delta_test_imp n_pauza

global semn_omega3_per omega3_per omega0_arb omega3_per_imp
            % ********* Caz II  - Impact Arbore - Percutor ***********
        %********* Testare contact punct P17 cu suprafetele percutorului **************
      % ****** Testarea contact P17 cu suprafata P13P14 (percutor11) ************
    plot(60,60,'b.');
    plot(60,60,'bo');
```

```
drawnow;
t_p17_p13p14 = sqrt(min((x_per(int_per1(11,1):int_per1(11,2)) -...
            x_arb(int_arb1(4,1))).^2 + (y_per(int_per1(11,1):int_per1(11,2))-...
            y_arb(int_arb1(4,1))).^2));
delta_test_imp = sqrt(((x_arba(int_arb1(4,1)) -...
            x_arb(int_arb1(4,1))).^2 + (y_arba(int_arb1(4,1)) -...
            y_arb(int_arb1(4,1))).^2));
if t_p17_p13p14 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'c.');
   drawnow;
   pause(n_pauza);
end
        % ******* Testarea contact P17 cu suprafata P12P13 (percutor10) ************
t_p17_p12p13 = sqrt(min((x_per(int_per1(10,1):int_per1(10,2)) -...
            x_arb(int_arb1(4,1))).^2 + (y_per(int_per1(10,1):int_per1(10,2))-...
            y_arb(int_arb1(4,1))).^2));
if t_p17_p12p13 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'c.');
   drawnow;
   pause(n_pauza);
end
      %******** Testare contact punct P21 cu suprafetele percutorului **********
   % ******* Testarea contact P21 cu suprafata P13P14 (percutor11) *********
t_p21_p13p14 = sqrt(min((x_per(int_per1(11,1):int_per1(11,2)) -...
            x_arb(int_arb1(8,1))).^2 + (y_per(int_per1(11,1):int_per1(11,2))-...
            y_arb(int_arb1(8,1))).^2));
delta_test_imp = sqrt(((x_arba(int_arb1(8,1)) -...
            x_arb(int_arb1(8,1))).^2 + (y_arba(int_arb1(8,1)) -...
            y_arb(int_arb1(8,1))).^2));
if t_p21_p13p14 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'c.');
   drawnow;
   pause(n_pauza);
end
   % ********* Testarea contact P21 cu suprafata P12P13 (percutor11) ************
t_p21_p12p13 = sqrt(min((x_per(int_per1(10,1):int_per1(10,2)) -...
            x_arb(int_arb1(8,1))).^2 + (y_per(int_per1(10,1):int_per1(10,2))-...
            y_arb(int_arb1(8,1))).^2));
if t_p21_p12p13 < delta_test_imp
   semn_omega3_per = -1;
   omega3_per = omega3_per_imp;
   plot(60,60,'c.');
   drawnow;
   pause(n_pauza);
end
        %********** Testare contact punct P20 cu suprafetele percutorului ********
```

```
   % ******** Testarea contact P20 cu suprafata P2P3 (percutor2) ************
t_p20_p2p3 = sqrt(min((x_per(int_per1(2,1):int_per1(2,2)) -...
            x_arb(int_arb1(7,1))).^2 + (y_per(int_per1(2,1):int_per1(2,2))-...
            y_arb(int_arb1(7,1))).^2));
delta_test_imp = sqrt(((x_arba(int_arb1(7,1)) -...
            x_arb(int_arb1(7,1))).^2 + (y_arba(int_arb1(7,1)) -...
            y_arb(int_arb1(7,1))).^2));
if t_p20_p2p3 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'co');
   drawnow;
   pause(n_pauza);
end
   % ************ Testarea contact P20 cu suprafata P3P4 (percutor3) ************
t_p20_p3p4 = sqrt(min((x_per(int_per1(3,1):int_per1(3,2)) -...
            x_arb(int_arb1(7,1))).^2 + (y_per(int_per1(3,1):int_per1(3,2))-...
            y_arb(int_arb1(7,1))).^2));
if t_p20_p3p4 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'co');
   drawnow;
   pause(n_pauza);
end
      %******* Testare contact punct P16 cu suprafetele percutorului *******
   % ************ Testarea contact P16 cu suprafata P2P3 (percutor2) ************
plot(60,60,'bo');
t_p16_p2p3 = sqrt(min((x_per(int_per1(2,1):int_per1(2,2)) -...
            x_arb(int_arb1(3,1))).^2 + (y_per(int_per1(2,1):int_per1(2,2))-...
            y_arb(int_arb1(3,1))).^2));
delta_test_imp = sqrt(((x_arba(int_arb1(3,1)) -...
            x_arb(int_arb1(3,1))).^2 + (y_arba(int_arb1(3,1)) -...
            y_arb(int_arb1(3,1))).^2));
if t_p16_p2p3 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'co');
   drawnow;
   pause(n_pauza);
end
   % ************ Testarea contact P20 cu suprafata P3P4 (percutor3) ************
t_p16_p3p4 = sqrt(min((x_per(int_per1(3,1):int_per1(3,2)) -...
            x_arb(int_arb1(3,1))).^2 + (y_per(int_per1(3,1):int_per1(3,2))-...
            y_arb(int_arb1(3,1))).^2));
if t_p16_p3p4 < delta_test_imp
   semn_omega3_per = 1;
   omega3_per = omega3_per_imp;
   plot(60,60,'co');
   drawnow;
   pause(n_pauza);
end
```

Anexa 24.

% Calculul vitezei unghiulare momentane a arborelui și a percutorului

```
%global percutor_pol arbore_pol delta_s delta_test_imp n_pauza

%global int_arb1 int_per1 x_arb y_arb x_arba y_arba x_per y_per x_pera y_pera

global semn_omega3_per omega3_per omega0_arb timp_amortizare_arb timp_amortizare_per

% Calcul viteza unghiulara percutor

if omega0_arb >= 0
   omega0_arb = omega0_arb - omega0_arb/timp_amortizare_arb;
end
if omega0_arb <= 0
   omega0_arb = 0;
end

% Calcul viteza unghiulara arbore

if omega3_per > 0
   omega3_per = omega3_per - omega3_per/timp_amortizare_per;
end
if omega3_per <= 0
   omega3_per = 0;
end
```

%gen_avi

```
aviobj = avifile('disp_pr_real_.avi')
for i=1:length(F)
   frame = F(i);
   aviobj = addframe(aviobj,frame);
end
aviobj = close(aviobj);
```

%Cerc_ext

```
x_cerc_ext  = -47.5;
y_cerc_ext  = -47.5;
w_cerc_ext  = 95;
h_cerc_ext  = 95;
x1_cerc_ext = 1;
y1_cerc_ext = 1;
axis equal;
rectangle('Position',[x,y,w,h],'Curvature',[x1,y1])
```

% Cerc_C3

```
x_cerc_c3  = -30;
y_cerc_c3  = -30;
w_cerc_c3  = 60;
h_cerc_c3  = 60;
x1_cerc_c3 = 1;
y1_cerc_c3 = 1;
axis equal;
```

Anexa 25

Succesiunea ciocnirilor suprafețelor și punctelor percutor-arbore în cazul dimensiunilor rezultate din proiectare

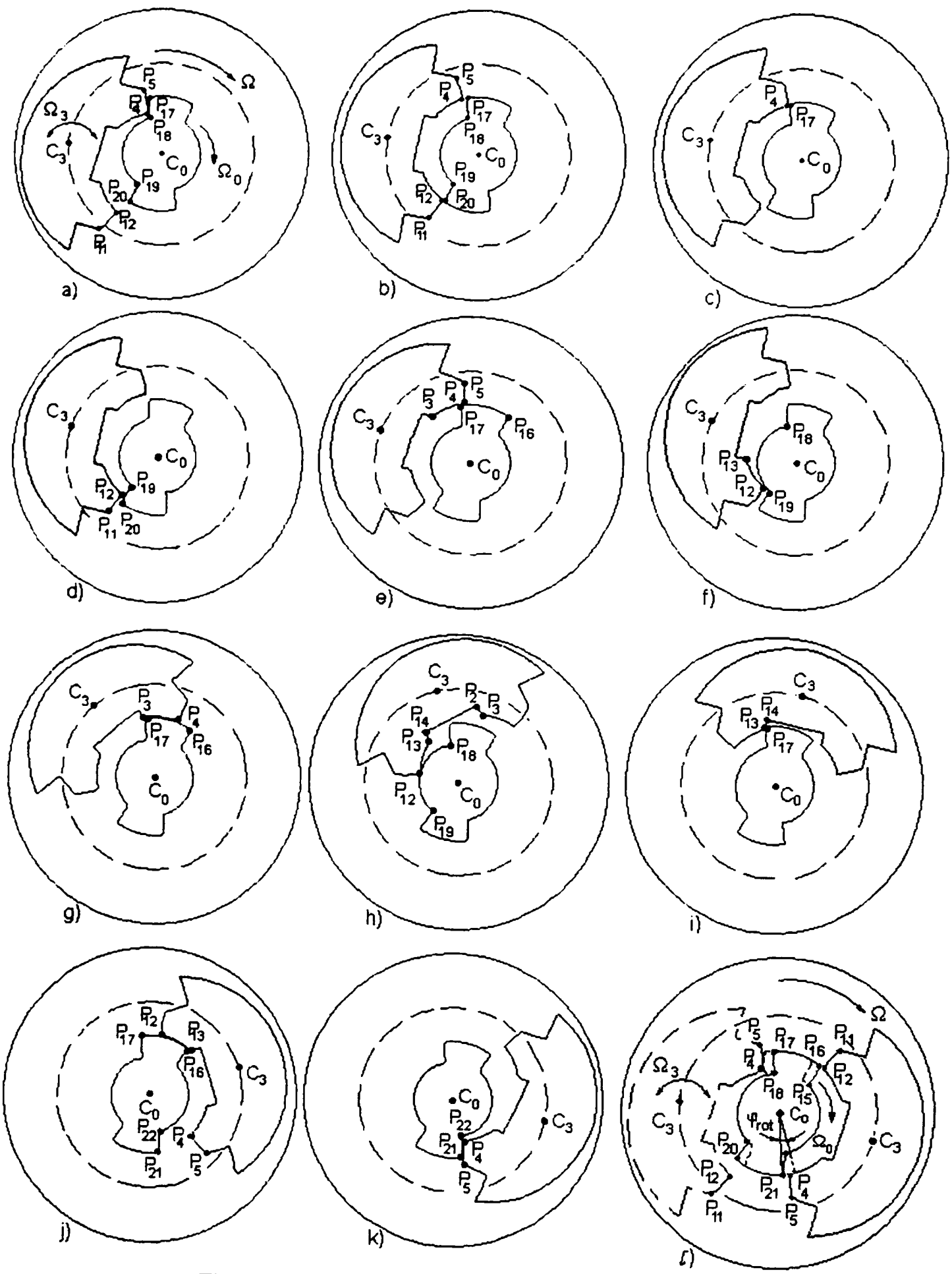


Fig. A.25. Succesiunea ciocnirilor percutor – arbore.

Anexa 26

Succesiunea ciocnirilor suprafețelor și punctelor percutor-arbore în cazul percutorului asimetric rezultat prin scurtarea coordonatelor punctului P_{12}

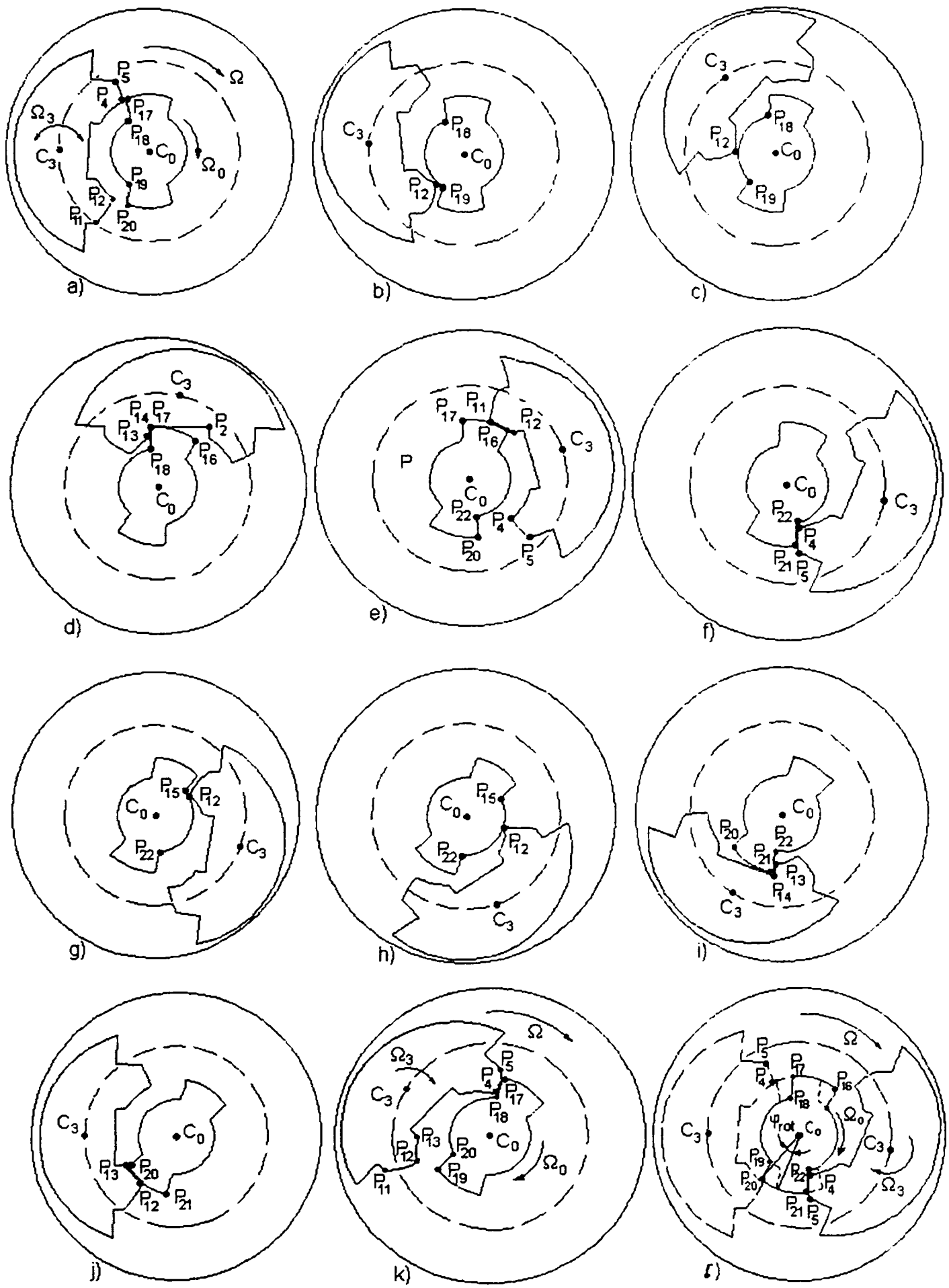


Fig. A. 26. Succesiunea ciocnirilor percutor – arbore cu percutor reproiectat

Anexa 27.

Succesiunea ciocnirilor suprafețelor și punctelor percutor-arbore în cazul percutorului asimetric nefuncțional rezultat prin scurtarea coordonatelor punctului P_{12} mai mult decât trebuie

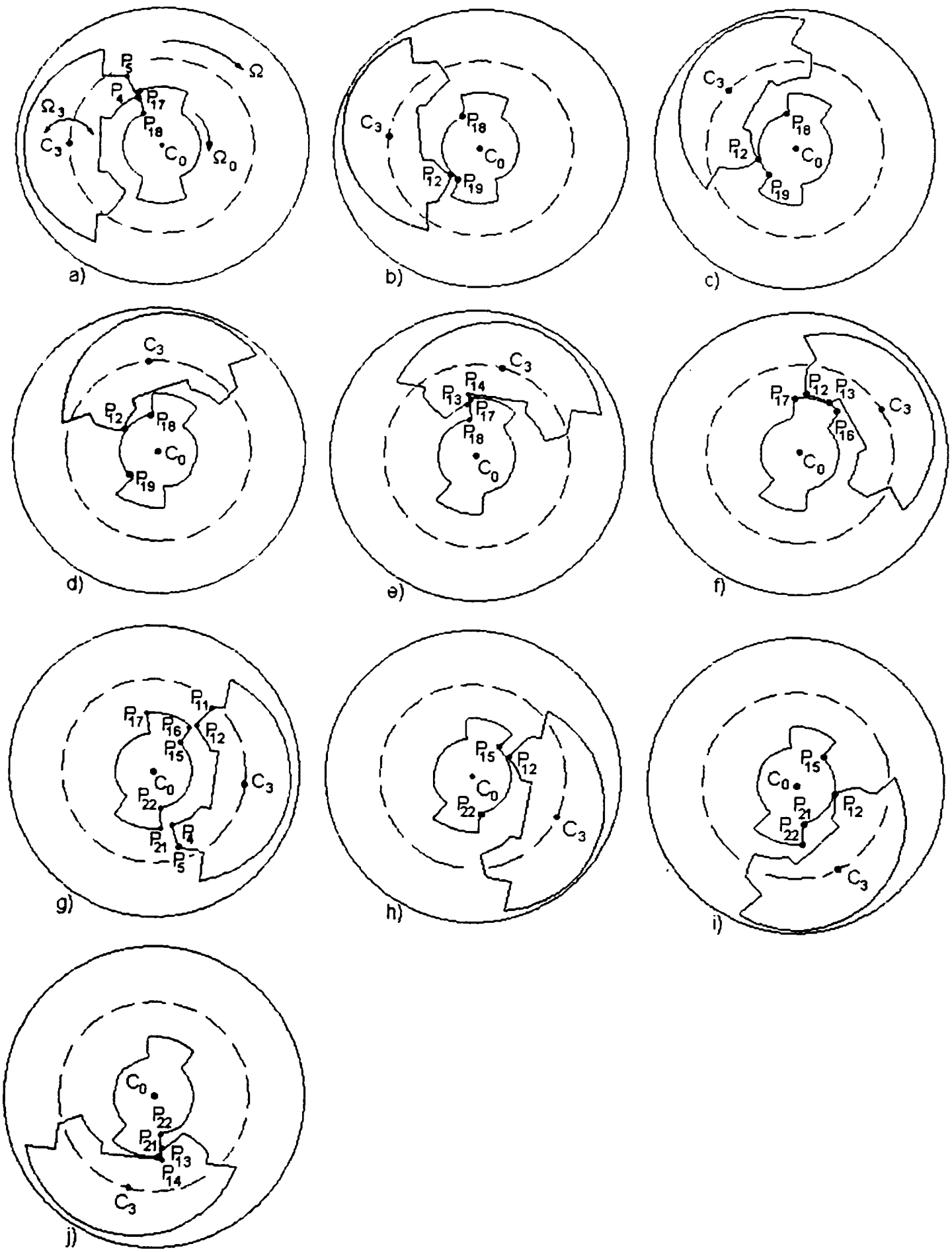


Fig. A.27. Succesiunea ciocnirilor percutor – arbore în cazul scurtării greșite a percutorului.

Anexa 28.

Diagrame accelerații înregistrate pe arborele dispozitivului în două situații: cu dispozitivul blocat (găurire normală) și cu dispozitivul neblocat (găurire vibropercutantă)

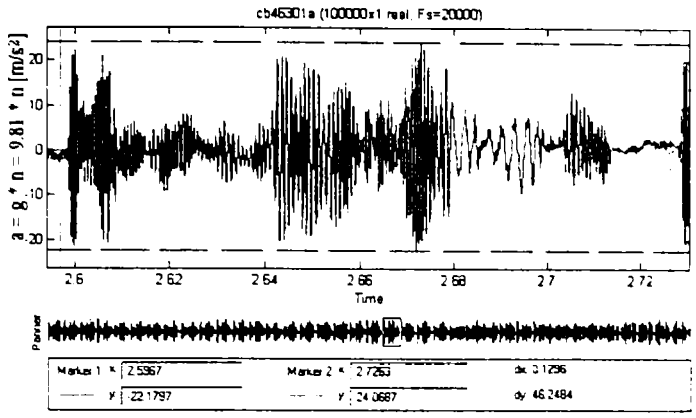


Fig. A. 28. 1 (cb. 463. 01a)

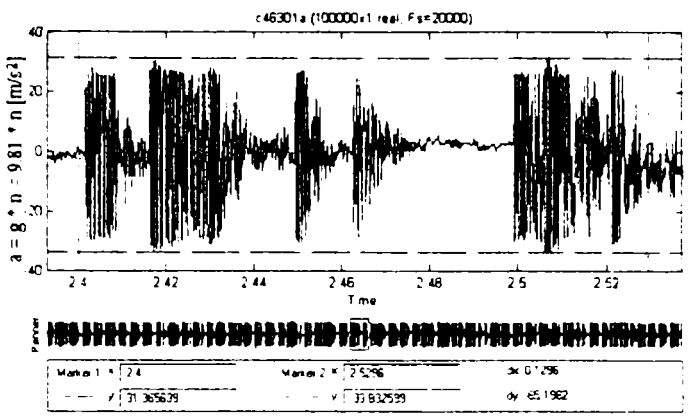


Fig. A. 28. 2. (c. 463. 01a)

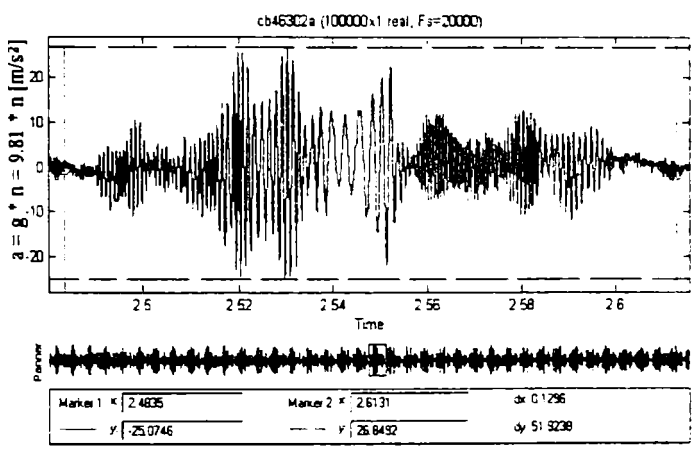


Fig. A. 28. 3. (cb. 463. 02a)

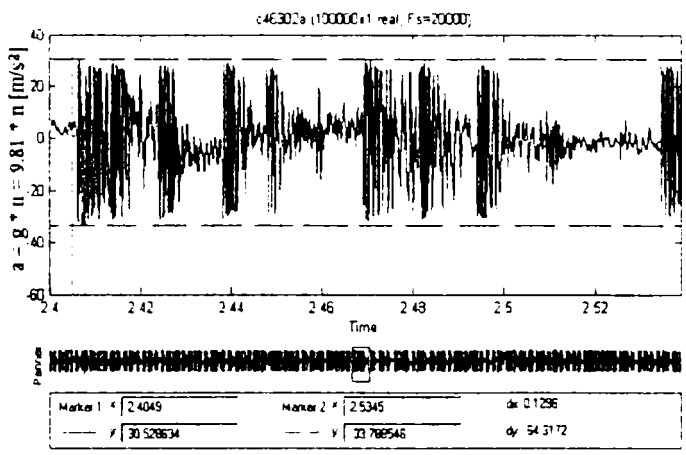


Fig. A. 28. 4 (c. 463. 02a)

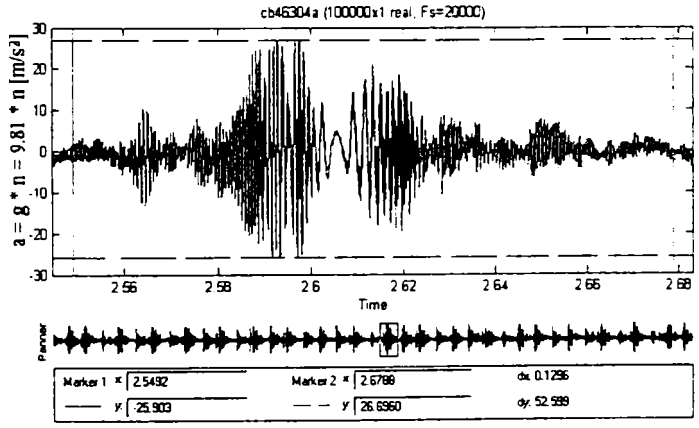


Fig. A. 28. 5 (cb. 463. 04a)

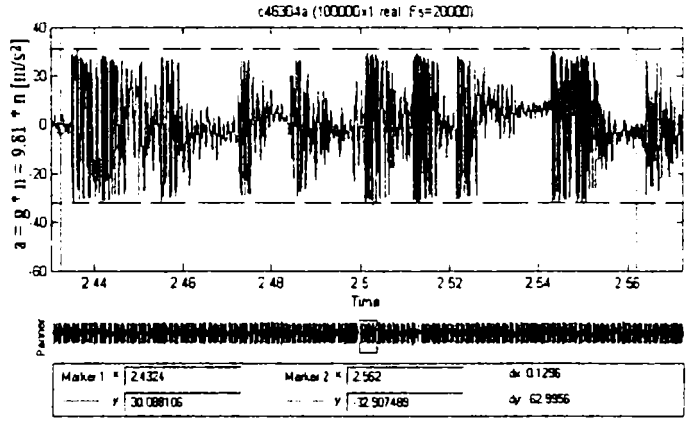


Fig. A. 28. 6. (c. 463. 04a)

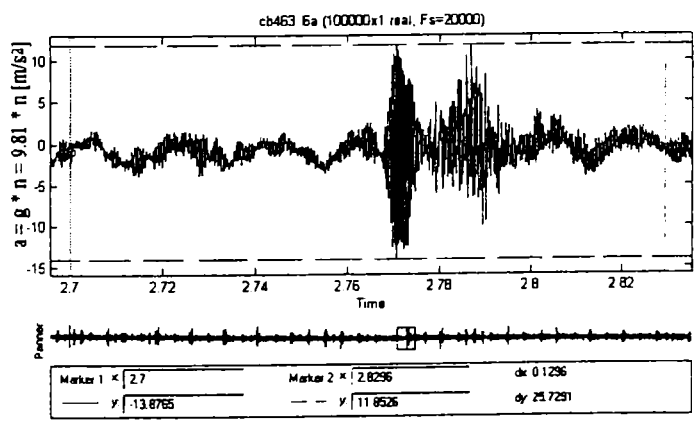


Fig. A. 28. 7. (cb. 463. 06a)

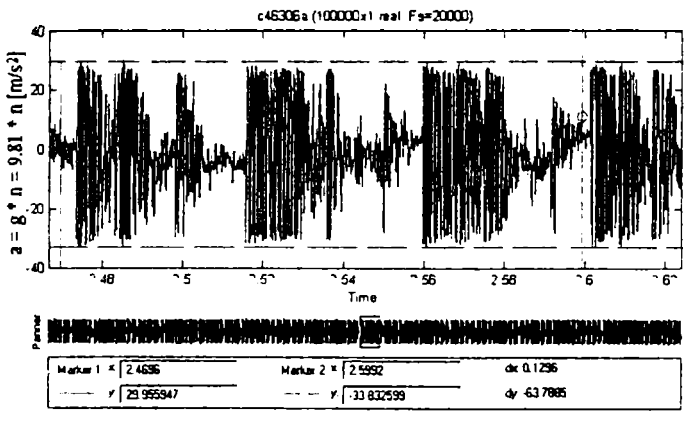


Fig. . A. 28. 8. (c. 463. 06a)

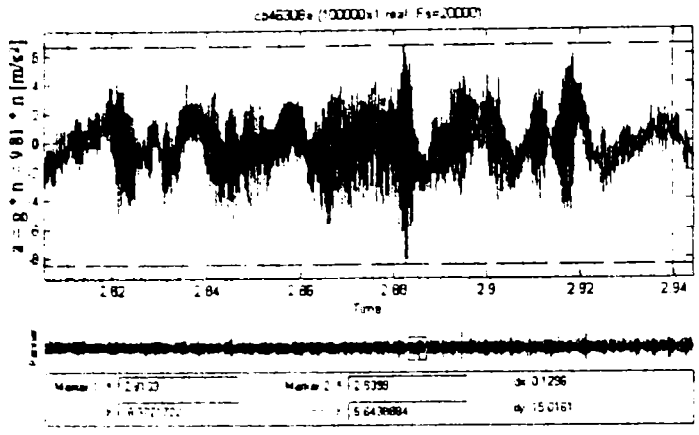


Fig. A.28. 9. (cb. 463. 08a)

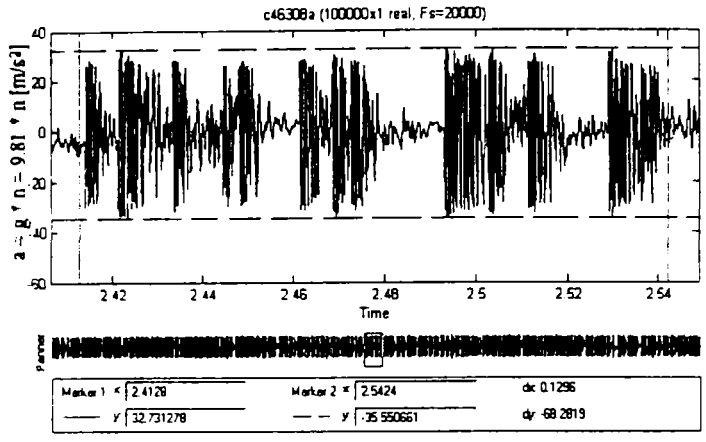


Fig. A. 28. 10. (c. 463. 08a)

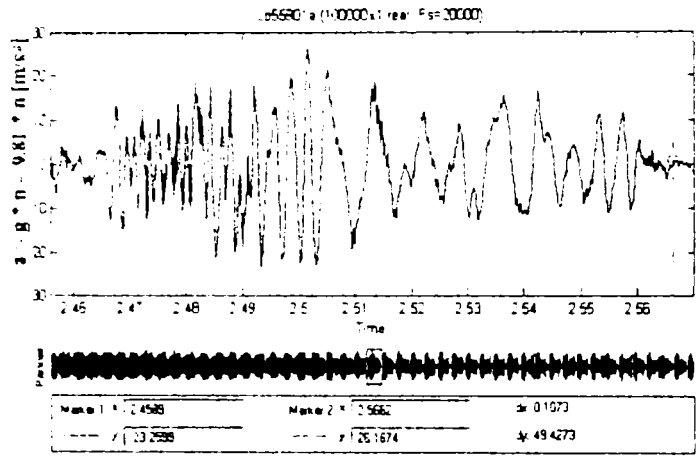


Fig. A.28. 11. (cb. 559. 01a)

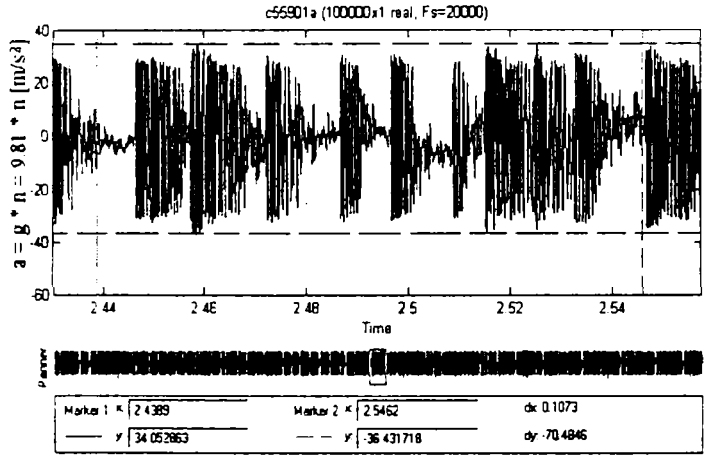


Fig. A.28. 12 (c. 559. 01a)

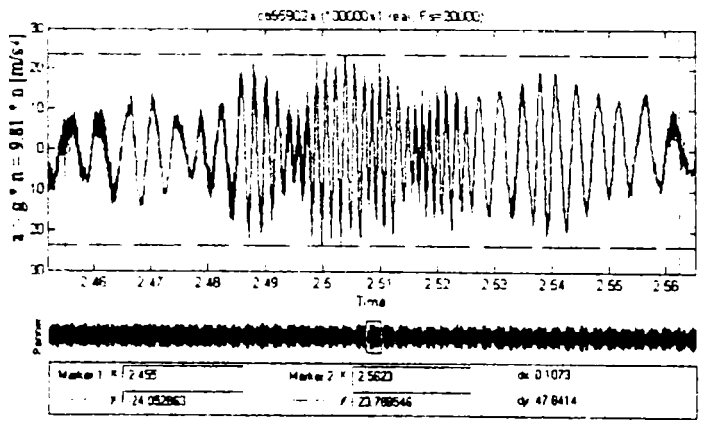


Fig. A.28. 13. (cb. 559. 02a)

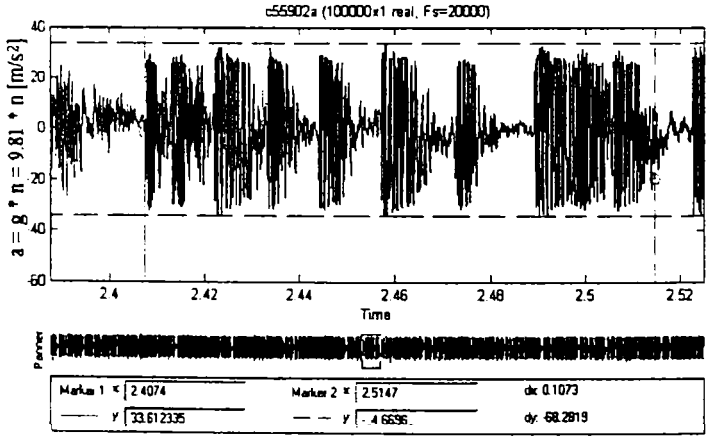


Fig. A.28. 14. (c. 559. 02a)

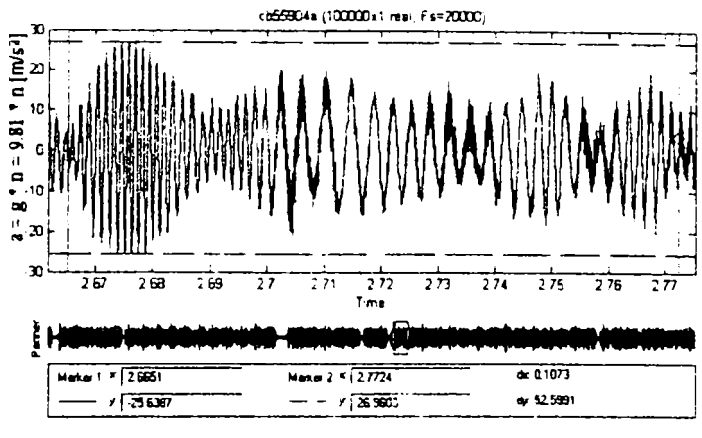


Fig. A.28. 15. (cb. 559. 04a)

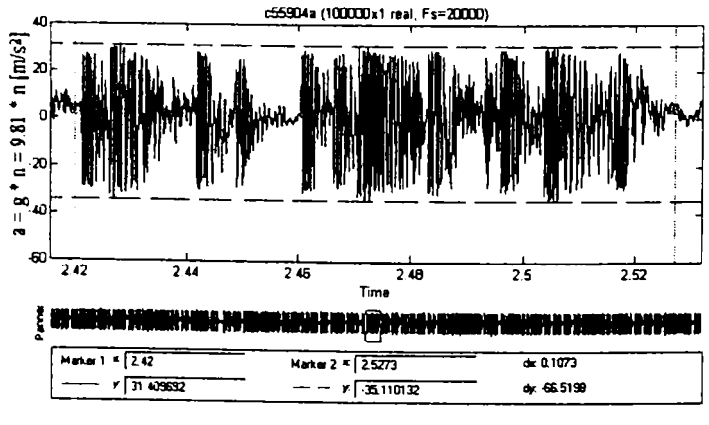


Fig. A.28. 16 (c. 559. 04a)

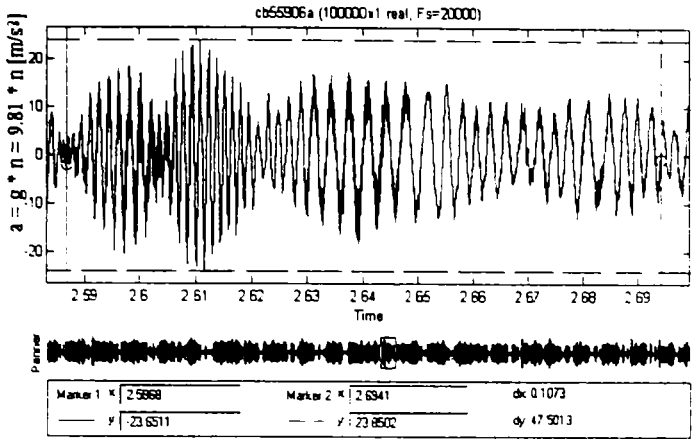


Fig. A.28. 17. (cb. 559. 06a)

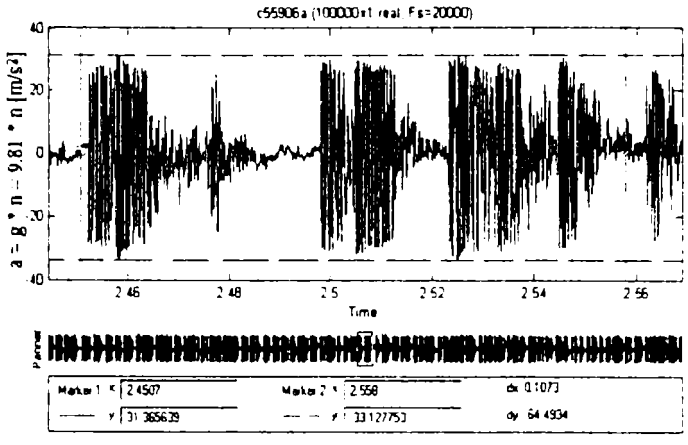


Fig. A.28. 18. (c. 559. 06a)

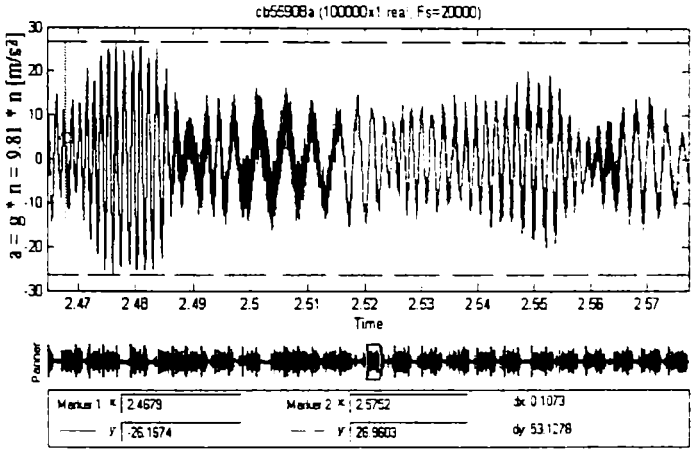


Fig. A.28. 19. (cb. 559. 08a)

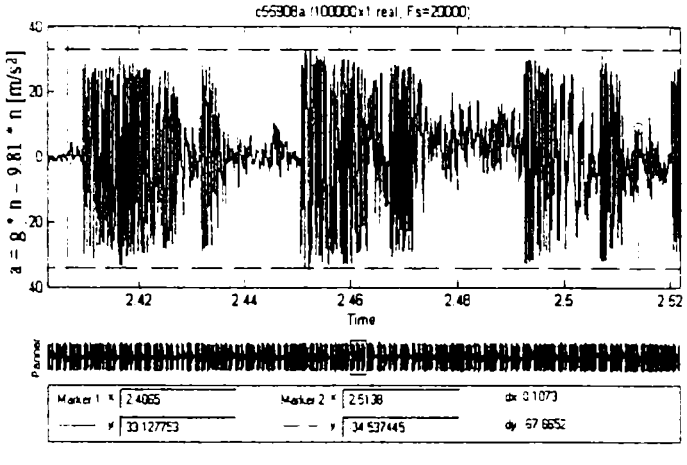


Fig. A.28. 20. (c. 559. 08a)

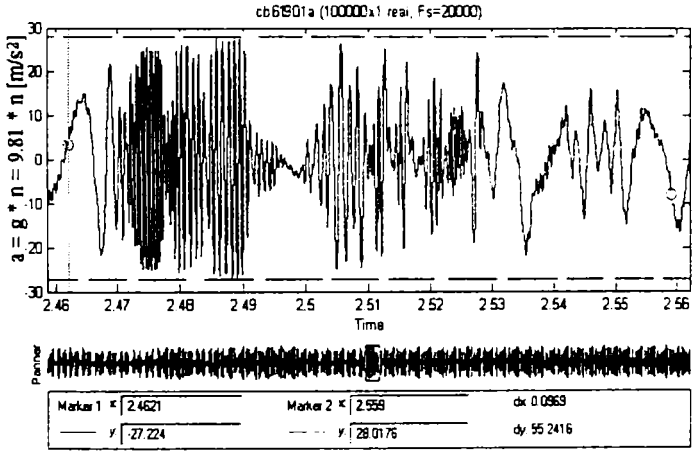


Fig. A.28. 21. (cb. 619. 01a)

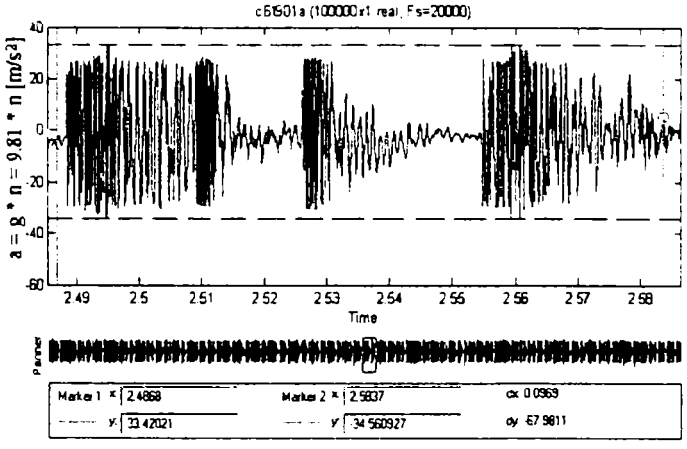


Fig. A.28. 22 (c.619. 01a)

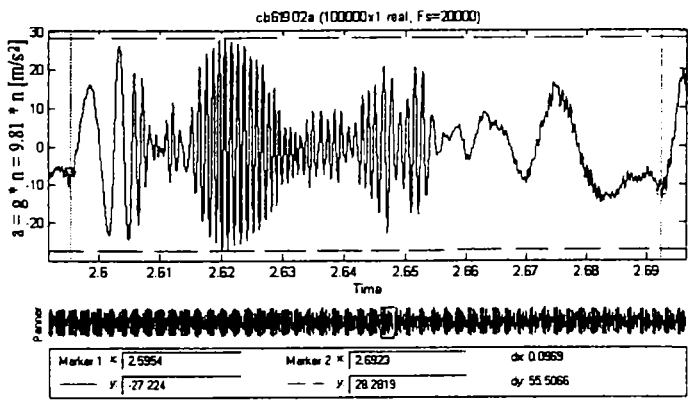


Fig. A.28. 23. (cb. 619. 02a)

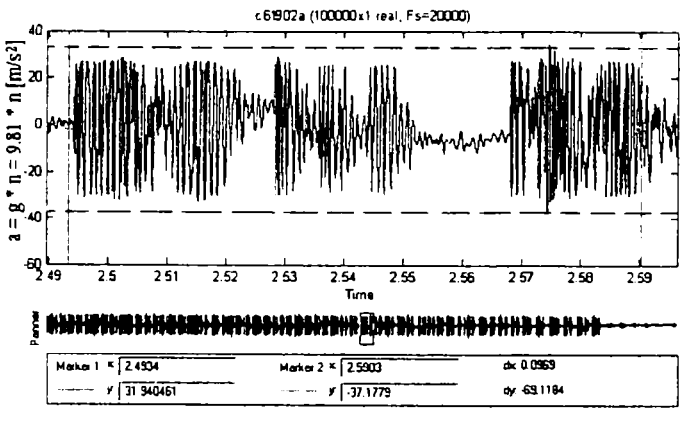


Fig. A.28. 24. (c.619. 02a)

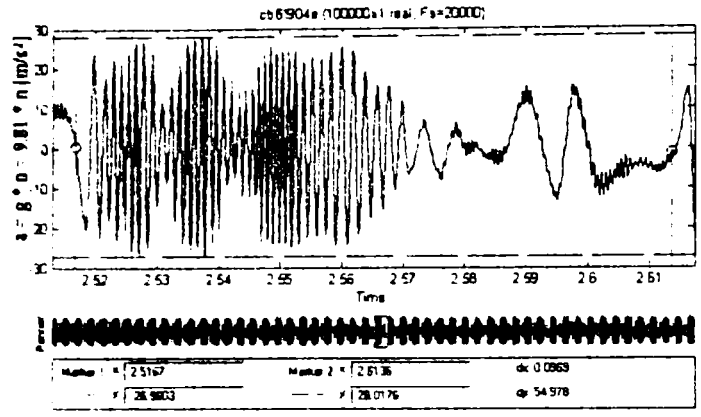


Fig. A.28. 25. (cb. 619. 04a)

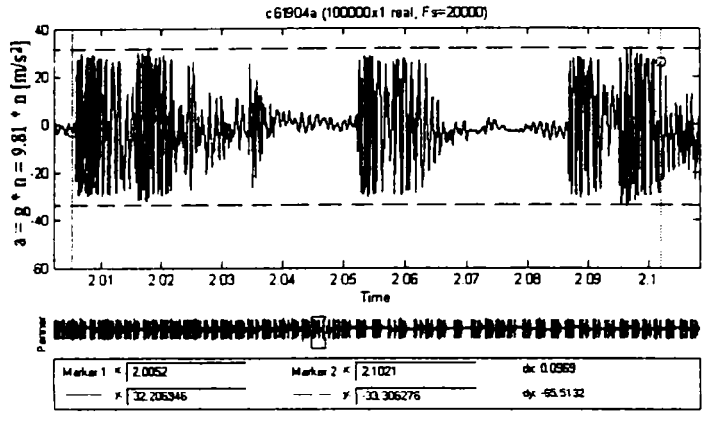


Fig. A.28. 26. (c.619. 04a)

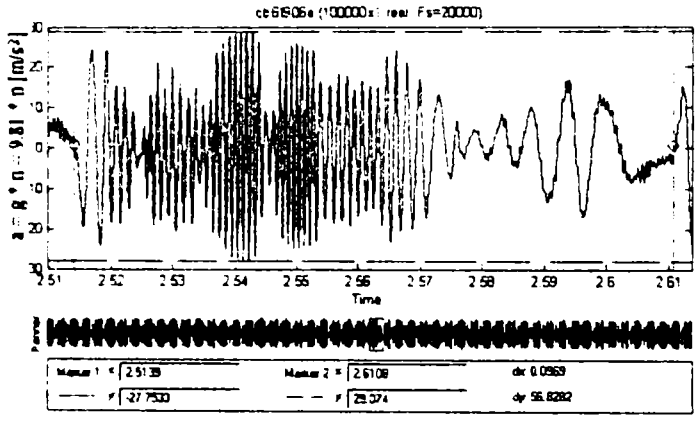


Fig. A.28. 27. (cb. 619. 06a)

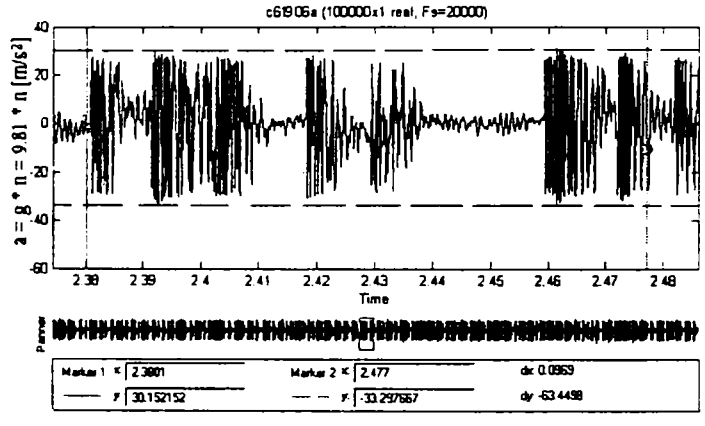


Fig. A.28. 28. (c.619. 06a)

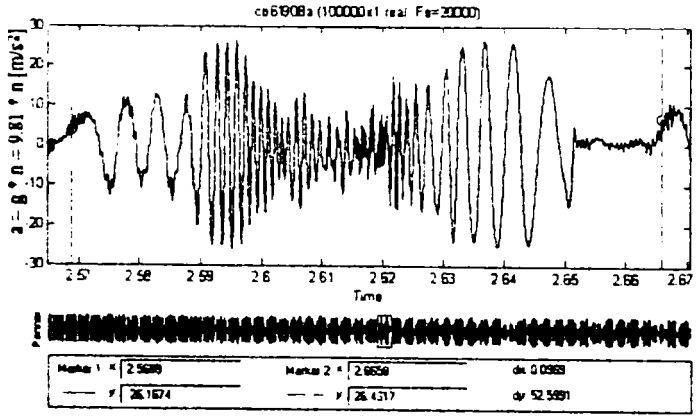


Fig. A.28. 29. (cb. 619. 08a)

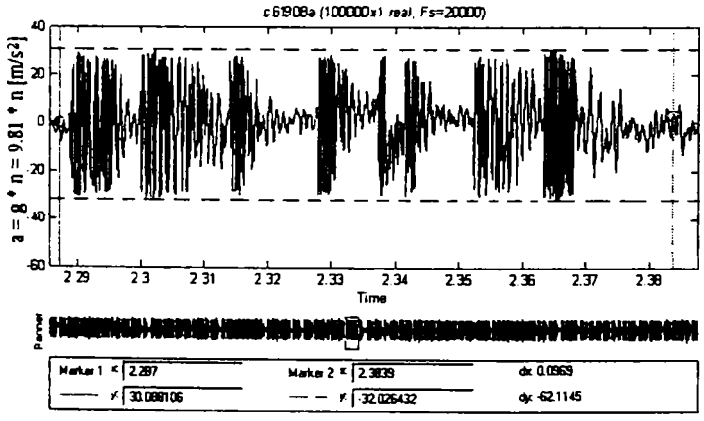


Fig. A.28. 30. (c.619. 08a)

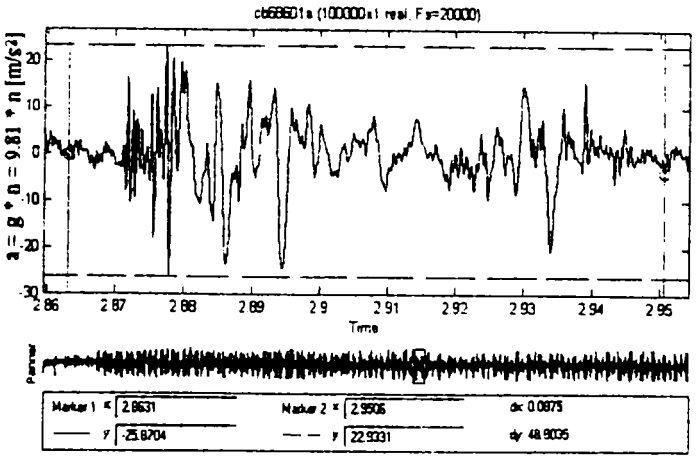


Fig. A.28. 31. (cb. 686. 01a)

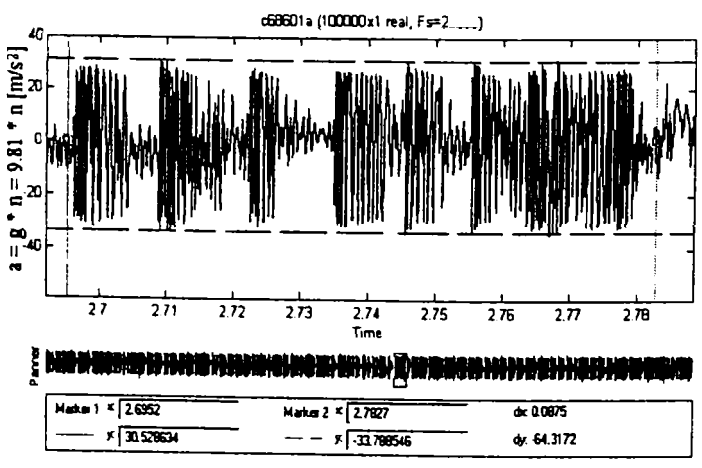


Fig. A.28. 32. (c. 686. 01a)

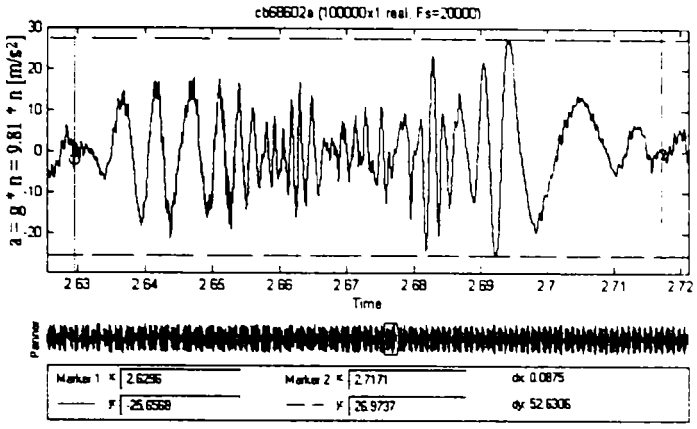


Fig. A.28. 33. (cb. 686. 02a)

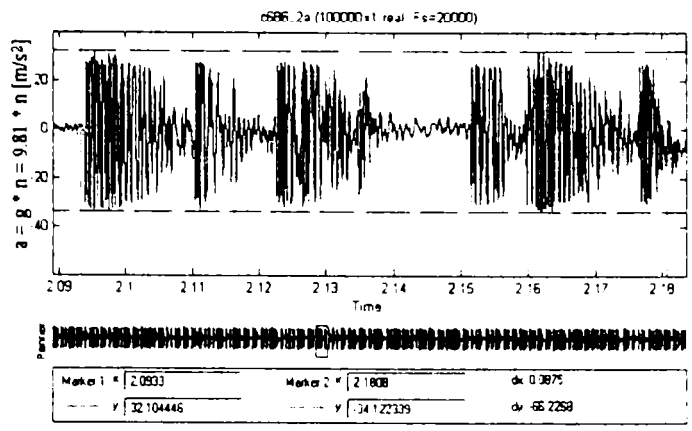


Fig. A.28. 34. (c. 686. 02a)

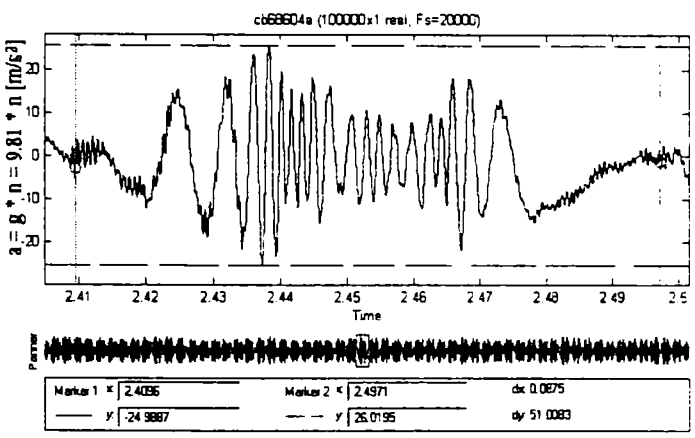


Fig. A.28. 35. (cb. 686. 04a)

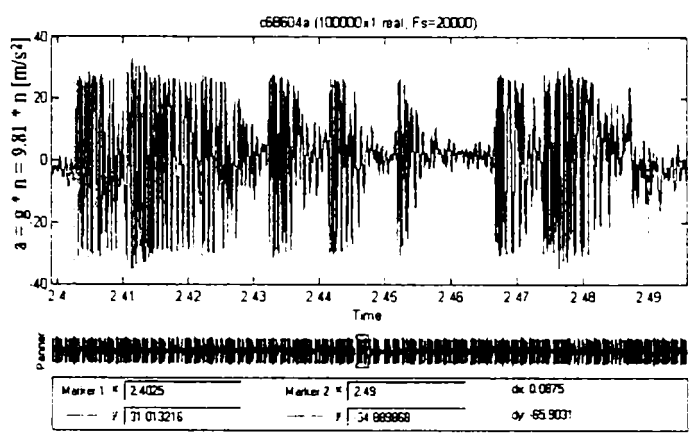


Fig. A.28. 36. (c. 686. 04a)

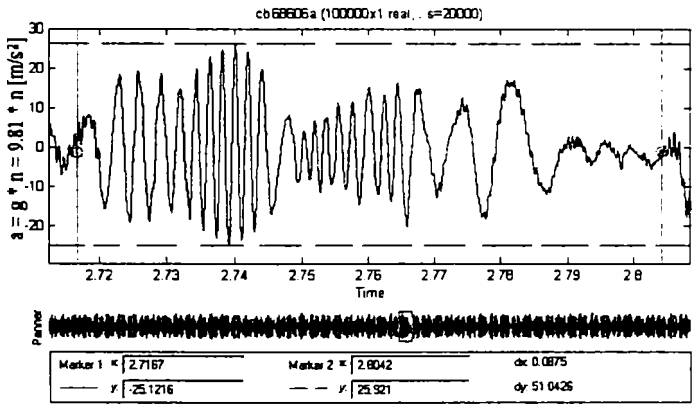


Fig. A.28. 37. (cb. 686. 06a)

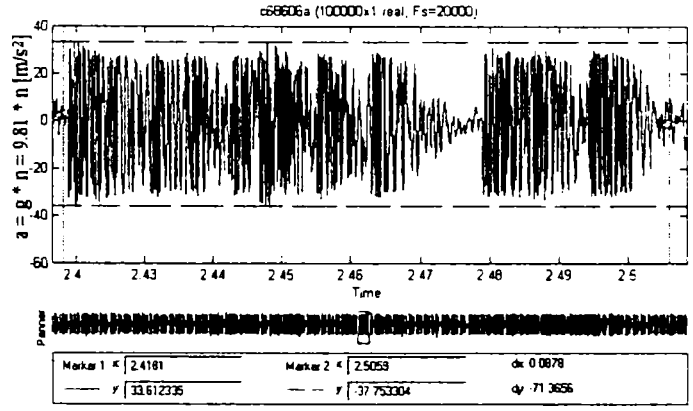


Fig. A.28. 38. (c. 686. 06a)

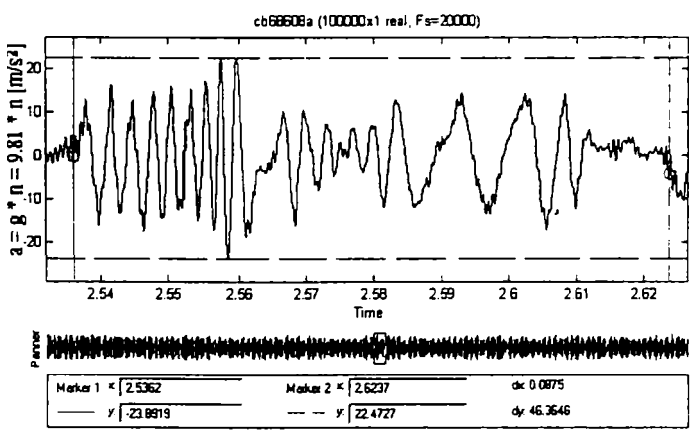


Fig. A.28. 39. (cb. 686. 08a)

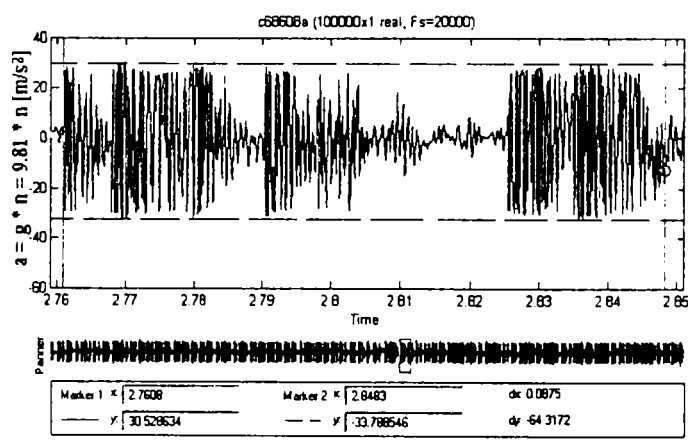


Fig. A.28. 40. (c. 686. 08a)

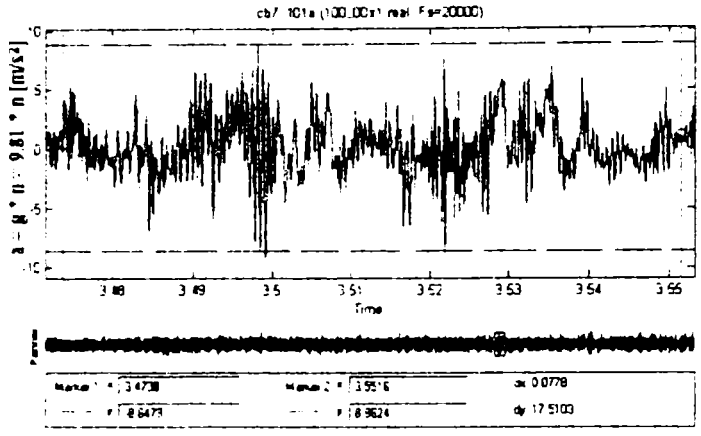


Fig. A.28. 41. (cb. 771. 01a)

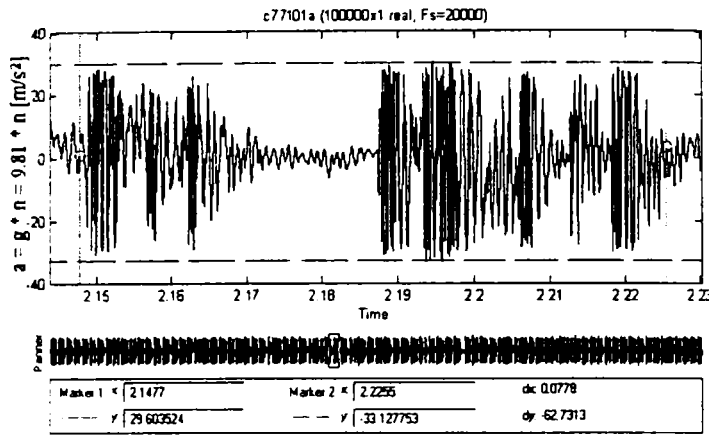


Fig. A.28. 42. (c. 771. 01a0

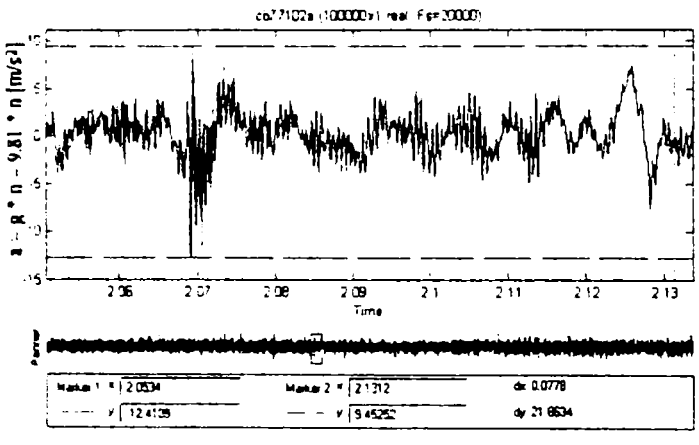


Fig. A.28. 43. (cb. 771. 02a)

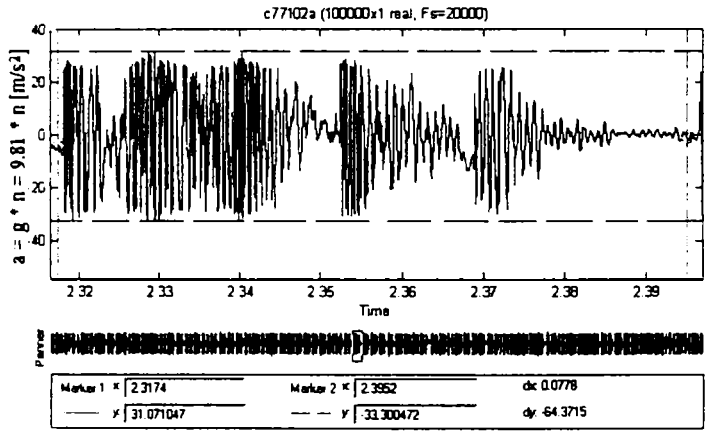


Fig. A.28. 44. (c. 771. 02a)

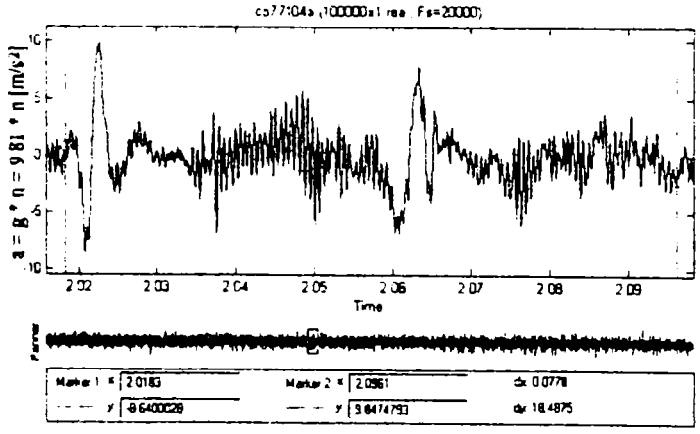


Fig. A.28. 45. (cb. 771. 04a)

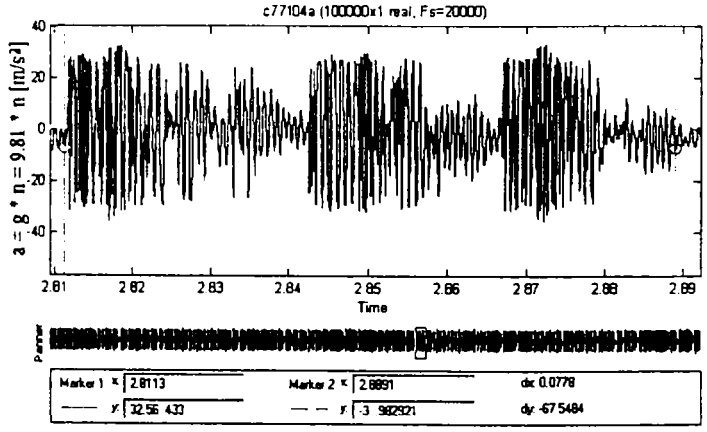


Fig. A.28. 46. (c. 771. 04a)

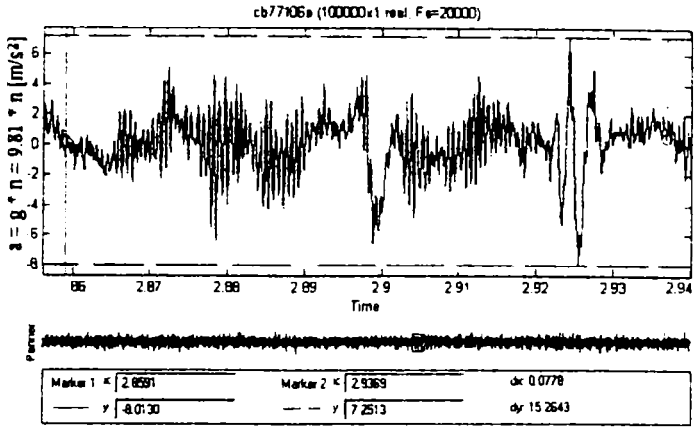


Fig. A.28. 47. (cb. 771. 06a)

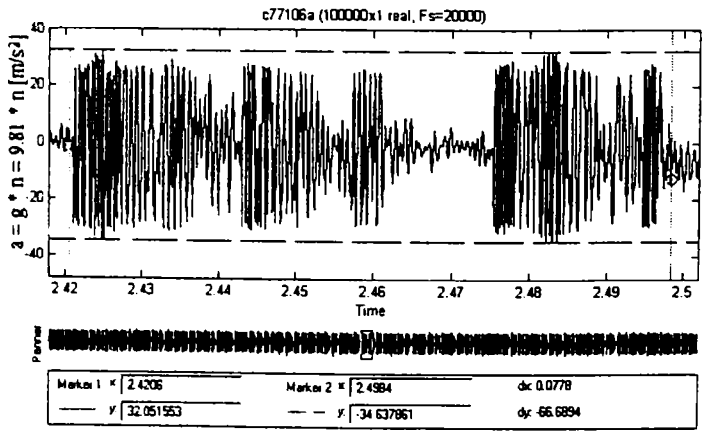


Fig. A.28. 48. (c. 771. 06a)

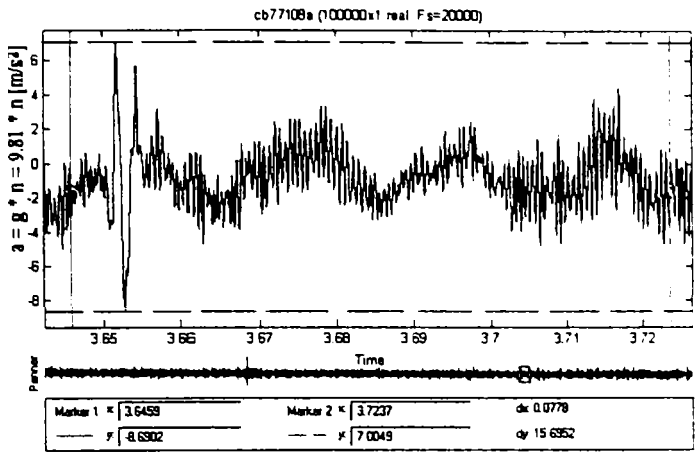


Fig. A.28. 49. (cb. 771. 08a)

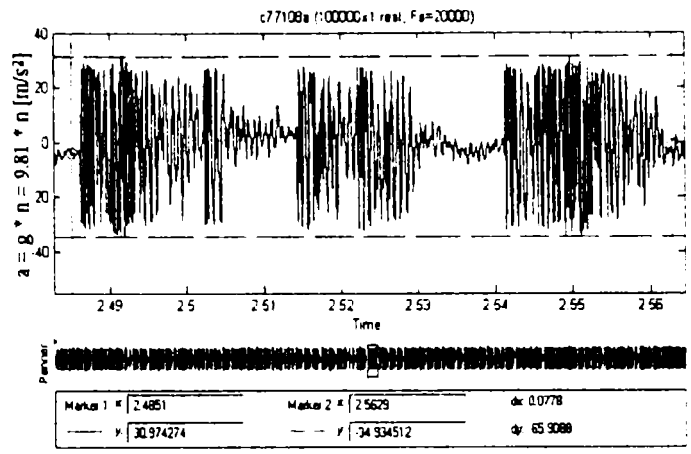


Fig. A.28. 50. (c. 771. 08a)

Notă. Semnificația notațiilor din paranteză:

- primele trei cifre reprezintă turațiile arborelui mașinii-unelte, [rot/min];
- următoarele două cifre reprezintă viteza de avans, [mm/min];
- cb - aşchiere obişnuită;
- c – aşchiere vibropercutantă;
- a – măsurare accelerații.

Anexa 29.

Diagrame accelerații măsurate pe universal cu dispozitivul blocat (cb) și neblocat (c)

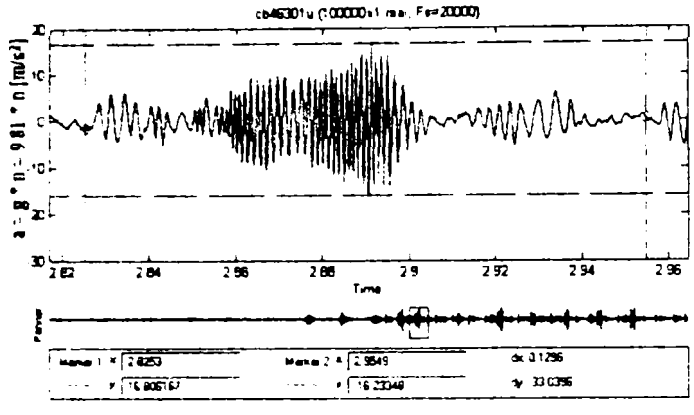


Fig. A.29. 1. (cb. 463. 01u)

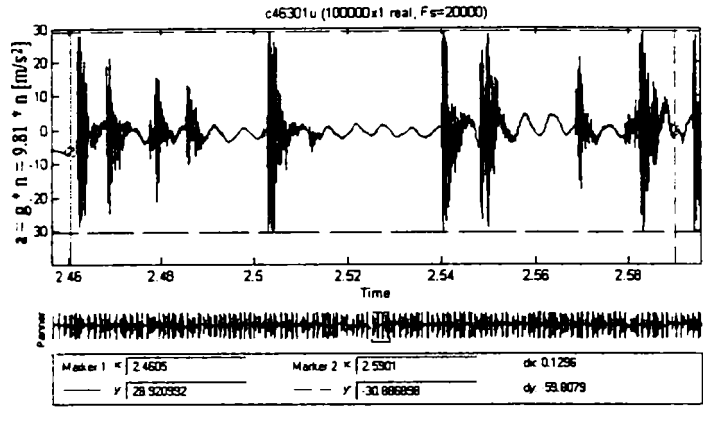


Fig. A.29. 2. (c. 463. 01u)

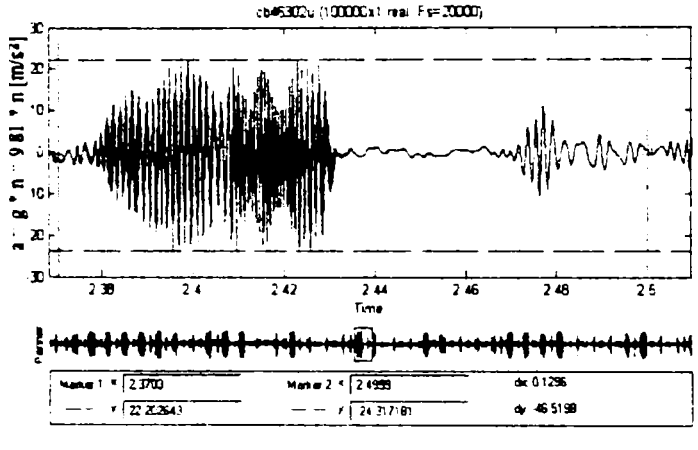


Fig. A.29. 3. (cb. 463. 02u)

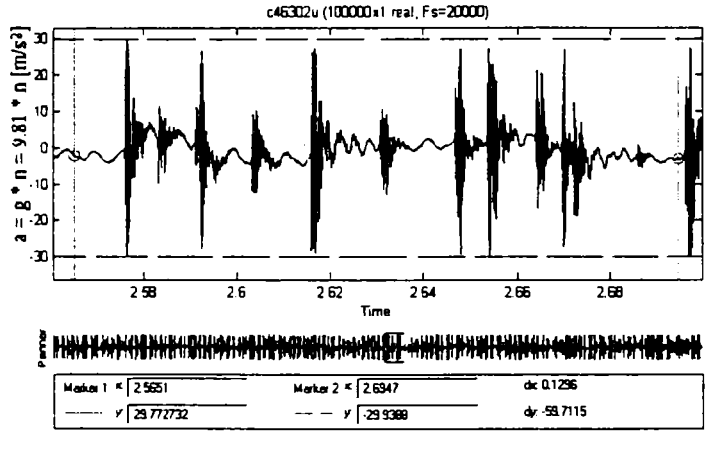


Fig. A.29. 4. (c. 463. 02u)

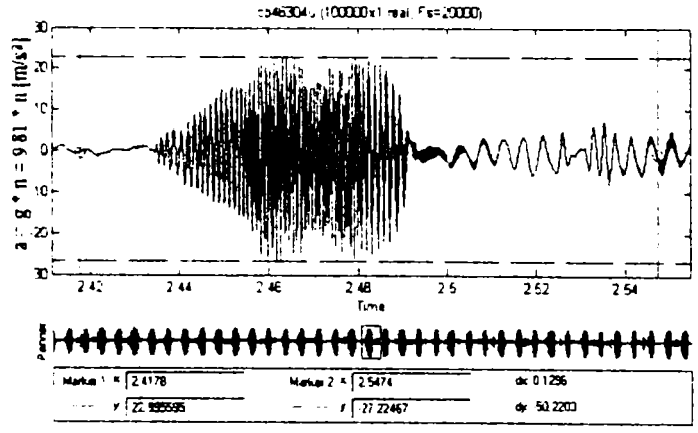


Fig. A.29. 5. (cb. 463. 04u)

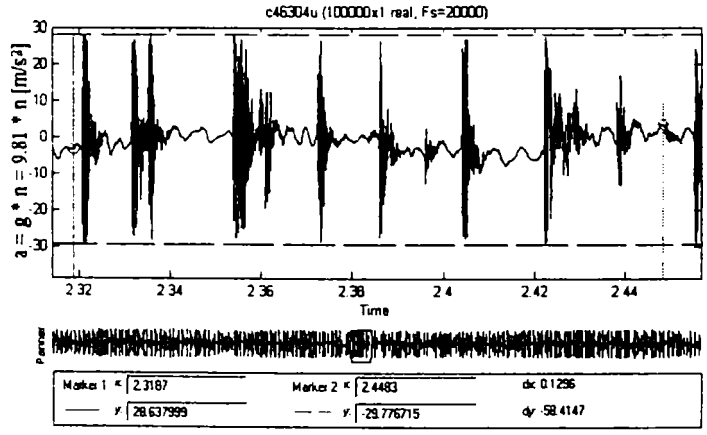


Fig. A.29. 6. (c. 463. 04u0

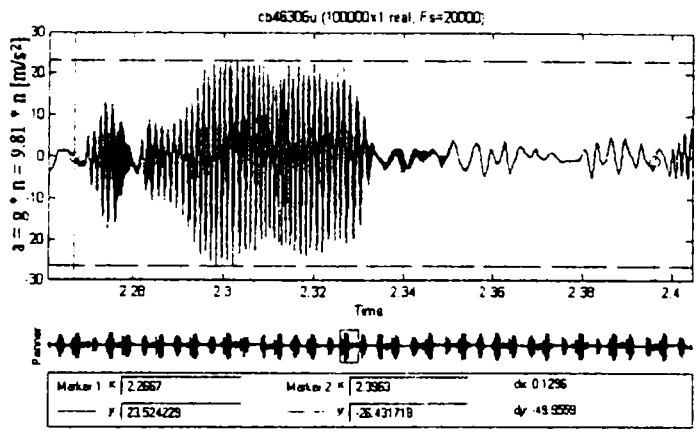


Fig. A.29. 7. (cb. 463. 06u)

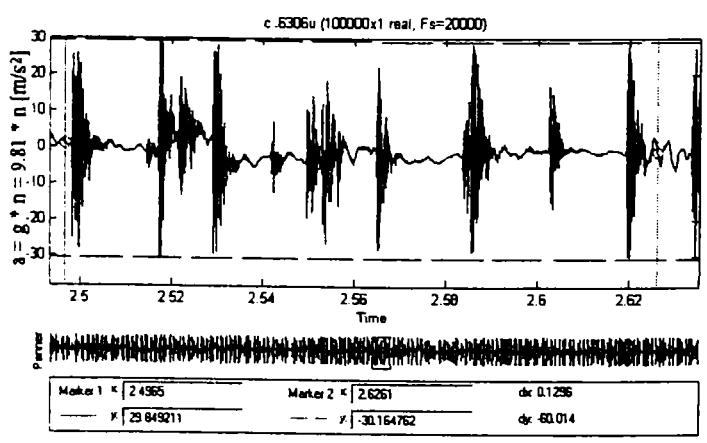


Fig. A.29. 8. (c. 463. 06u0

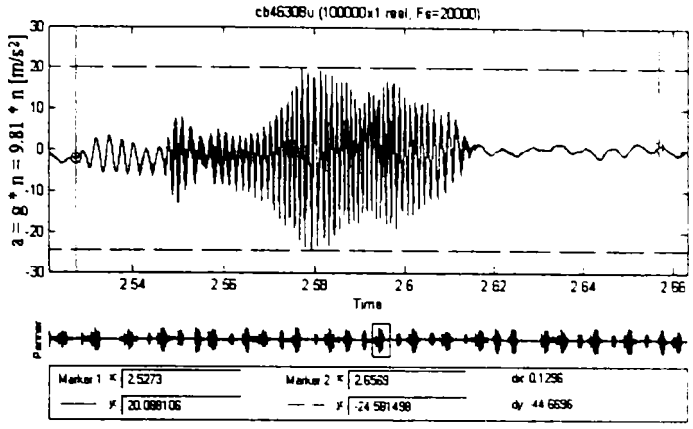


Fig. A.29. 9. (cb. 463. 08u)

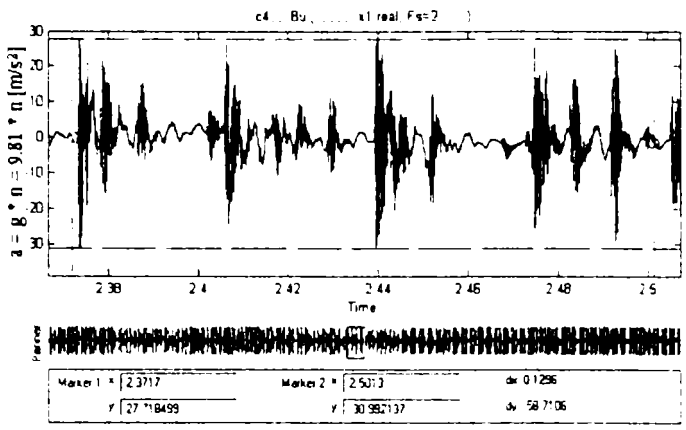


Fig. A.29. 10. (c. 463. 08u)

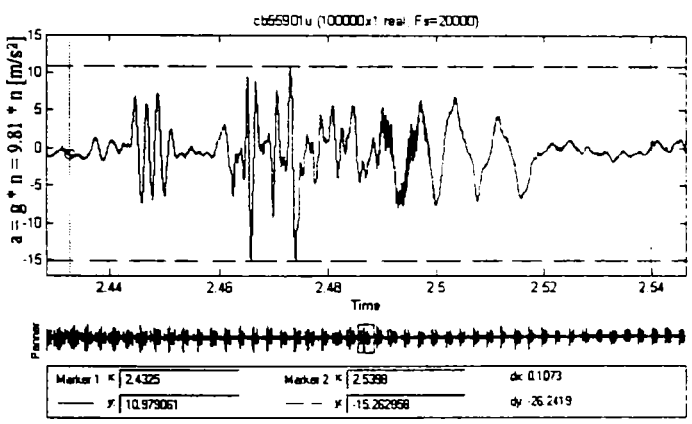


Fig. A.29. 11. (cb. 559. 01u)

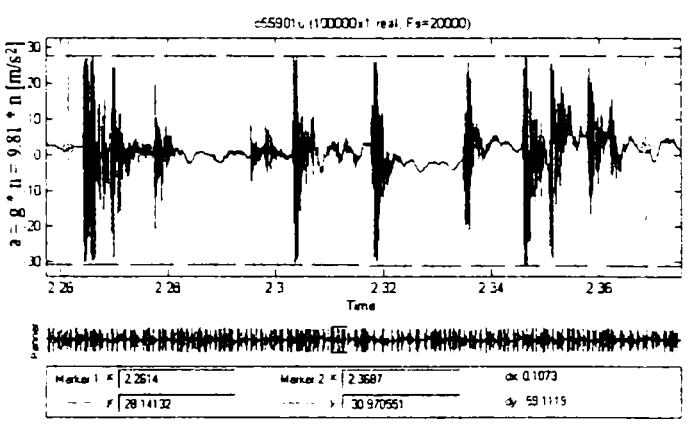


Fig. A.29. 12. (c.559. 01u)

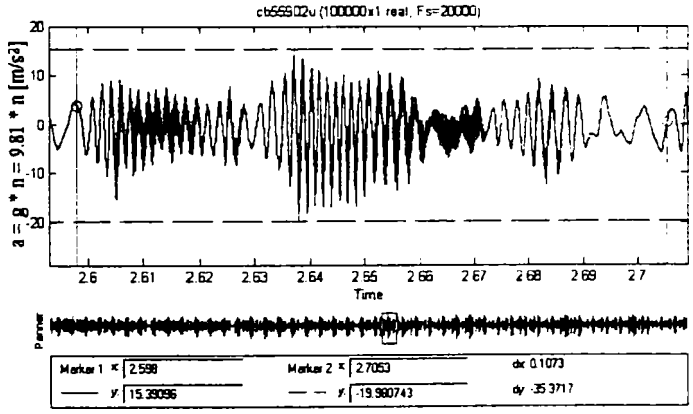


Fig. A.29. 13. (cb. 559. 02u)

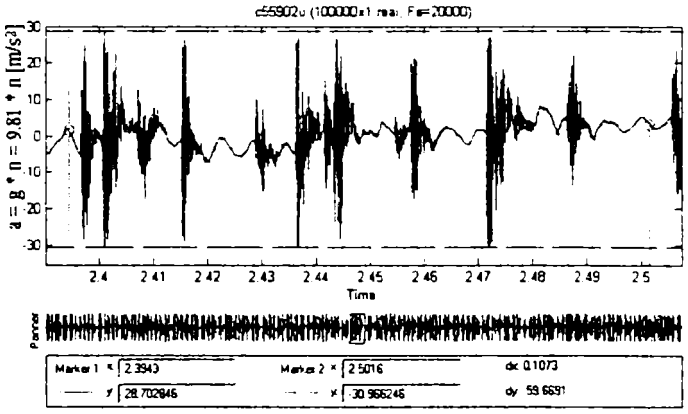


Fig. A.29. 14. (c.559. 02u)

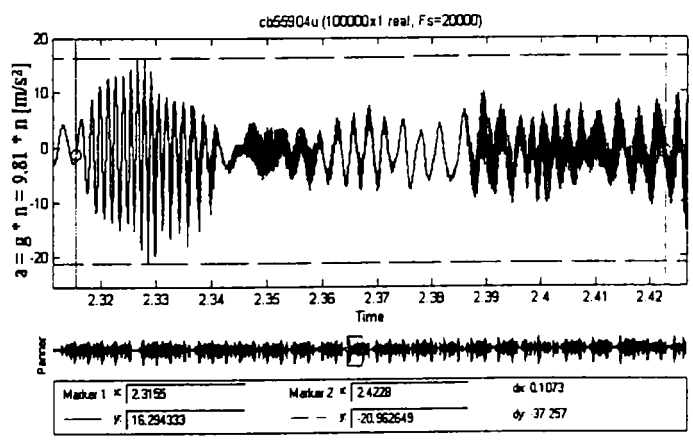


Fig. A.29. 15. (cb. 559. 04u)

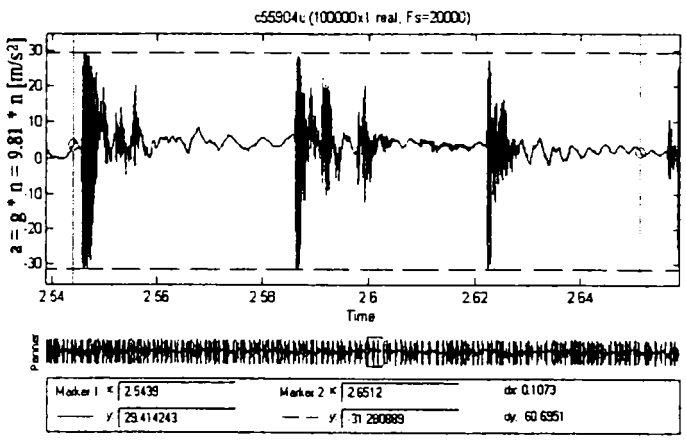


Fig. A.29.16. (c.559. 04u)

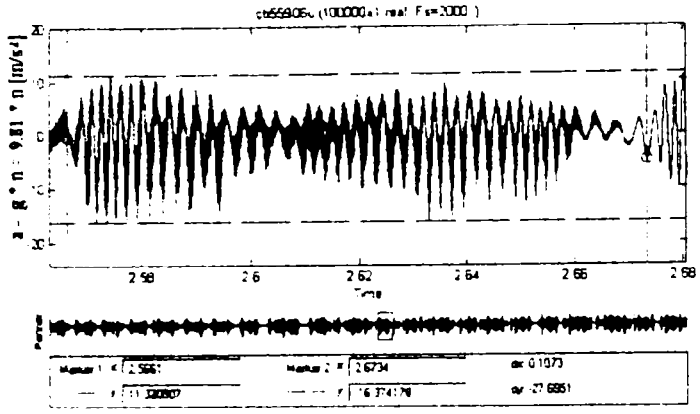


Fig. A.29. 17. (cb. 559. 06u)

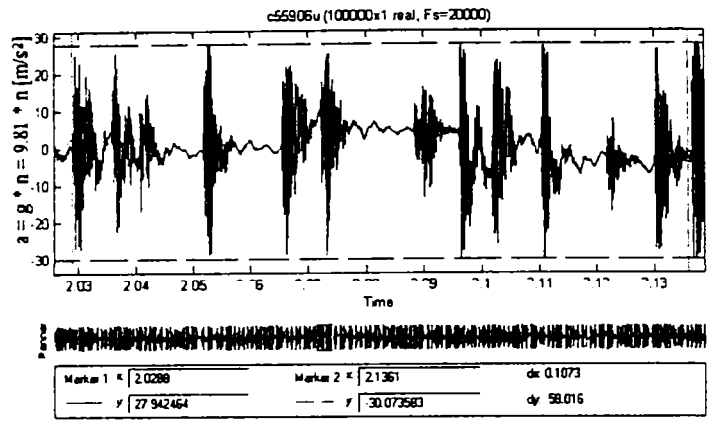


Fig. A.29. 18. (c.559. 06u)

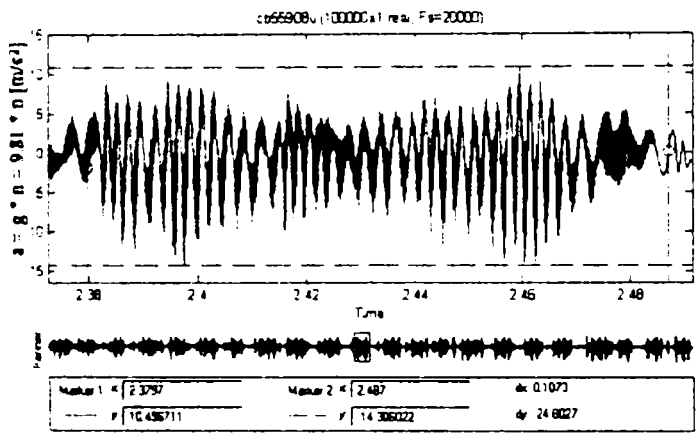


Fig. A.29. 19. (cb. 559. 08u)

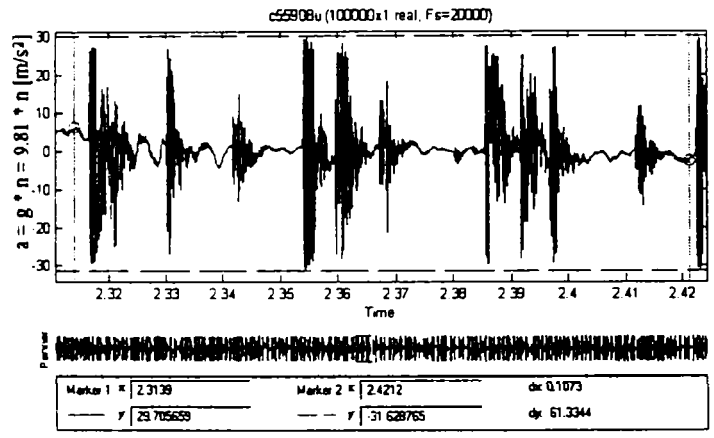


Fig. A.29. 20. (c.559. 08u)

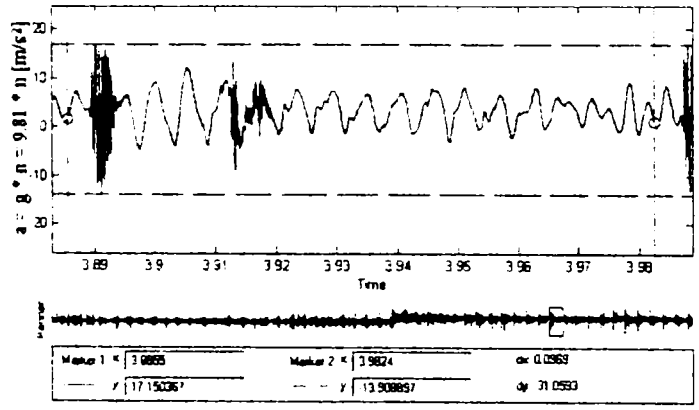


Fig. A.29. 21. (cb. 619. 01u)

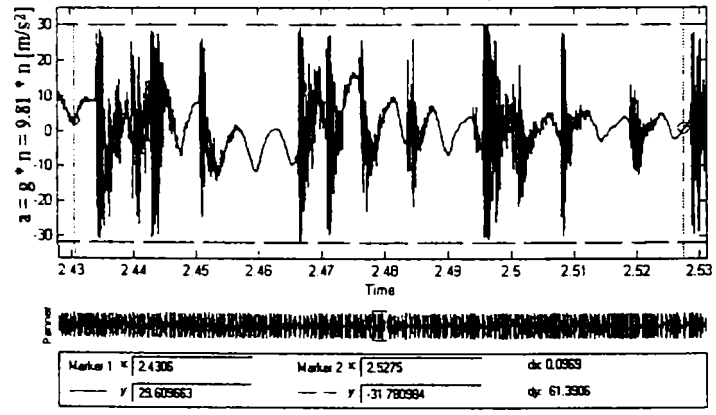


Fig. A.29. 22. (c.619. 01u)

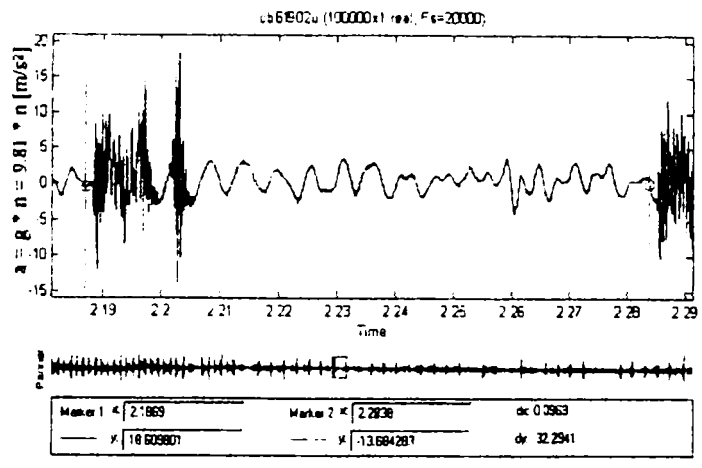


Fig. A.29. 23. (cb. 619. 02u)

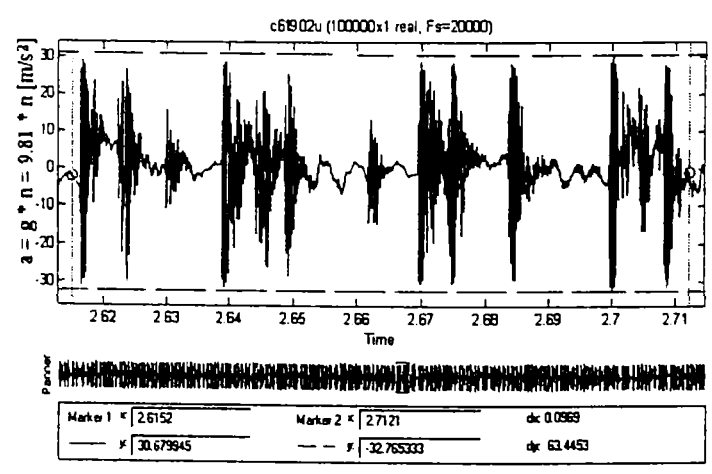


Fig. A.29. 24. (c.619. 02u)

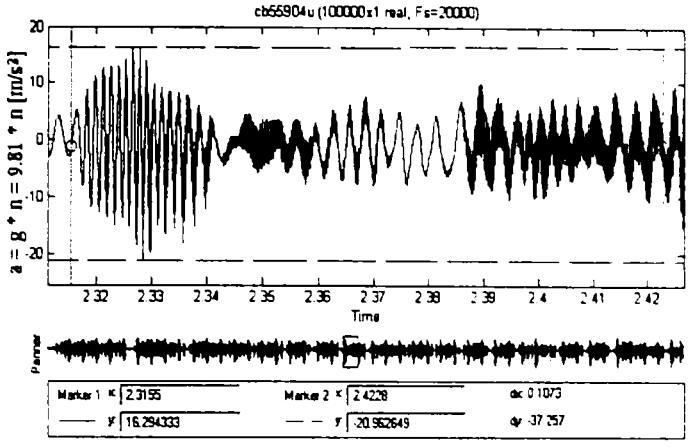


Fig. A.29. 25. (cb. 619. 04u)

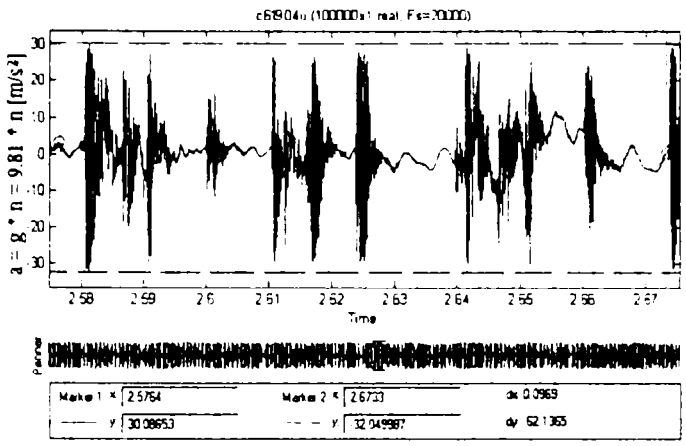


Fig. A.29. 26. (c.619. 04u)

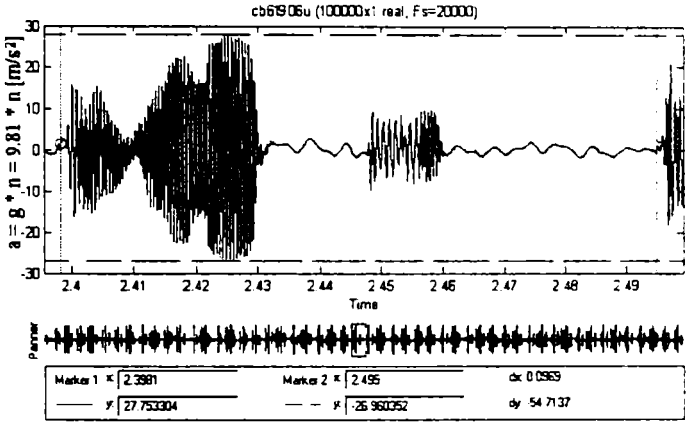


Fig. A.29. 27. (cb. 619. 06u)

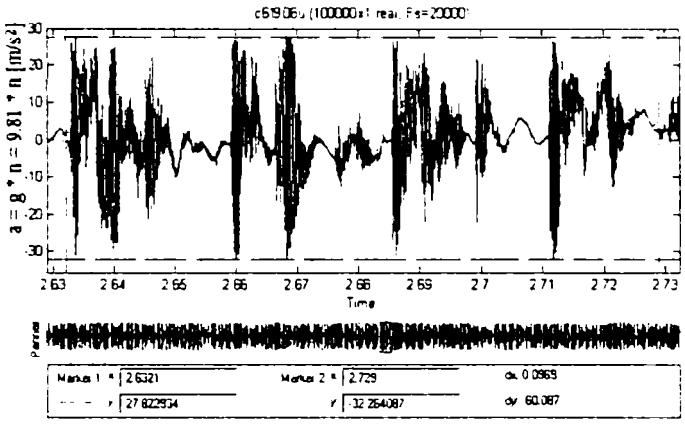


Fig. A.29. 28. (c.619. 06u)

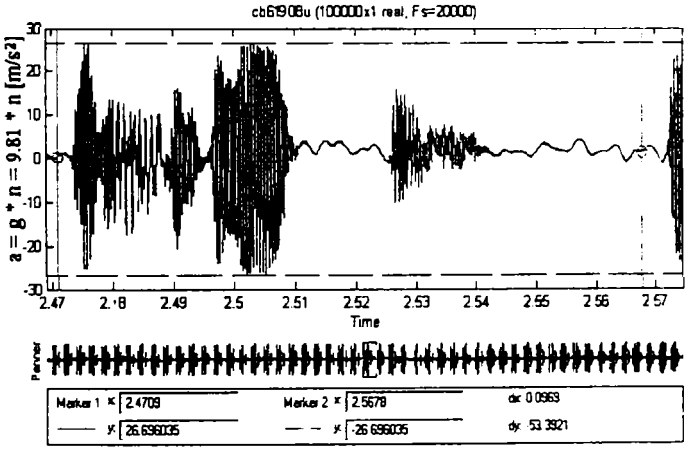


Fig. A.29. 29. (cb. 619. 08u)

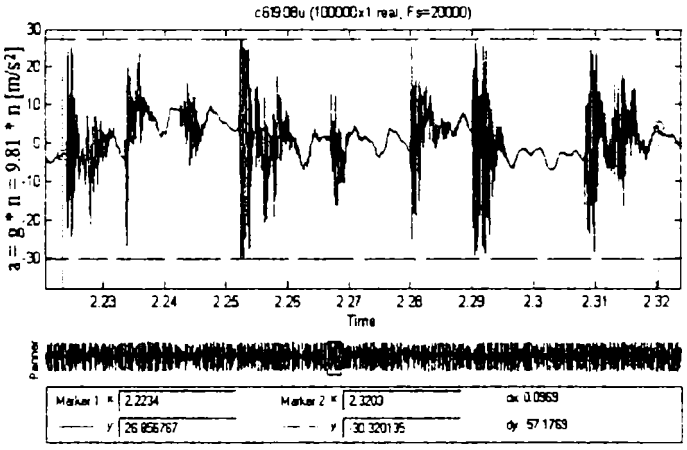


Fig. A.29. 30. (c.619. 08u)

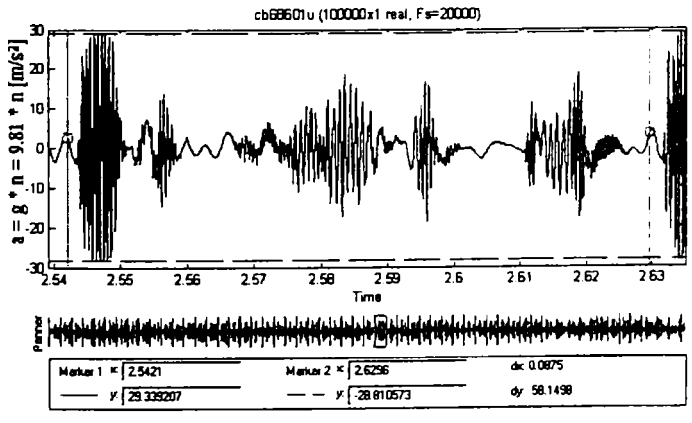


Fig. A.29. 31. (cb.686. 01u)

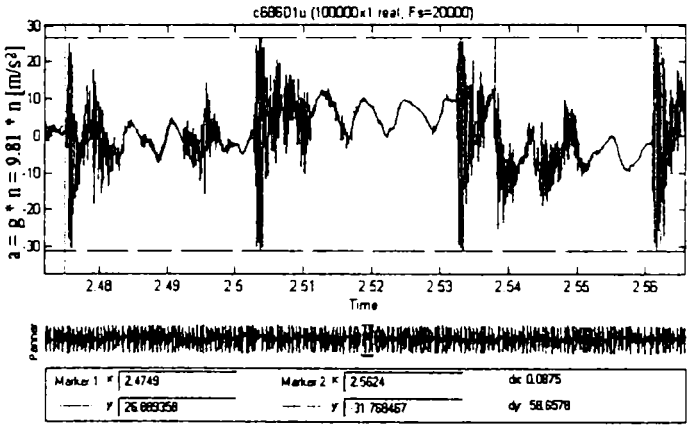


Fig. A.29. 32. (c. 686. 01u)

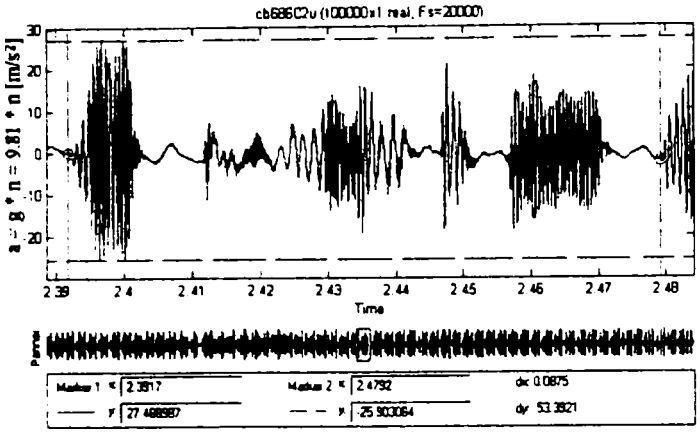


Fig. A.29. 33. (cb.686. 02u)

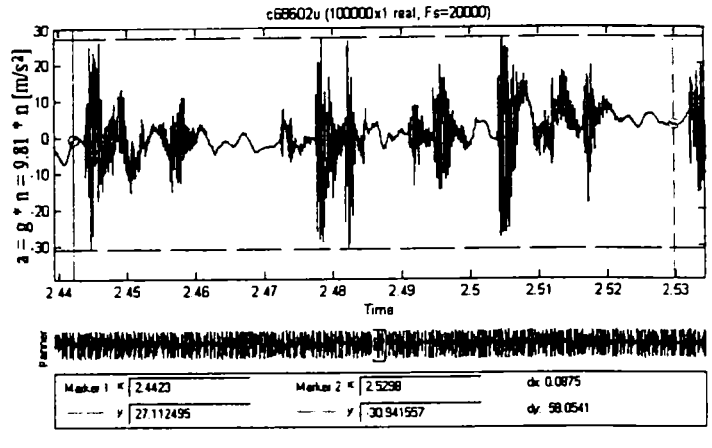


Fig. A.29. 34. (c. 686. 02u)

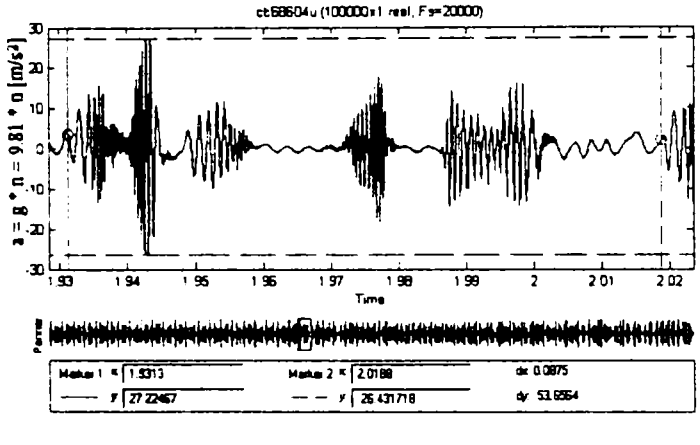


Fig. A.29. 35. (cb.686. 04u)

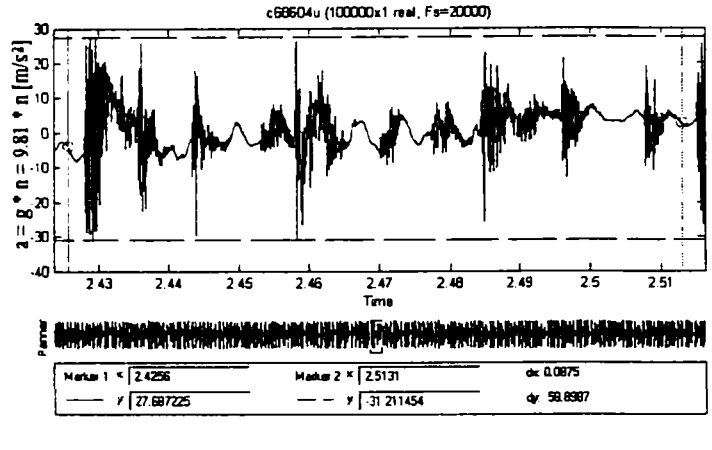


Fig. A.29. 36. (c. 686. 04u)

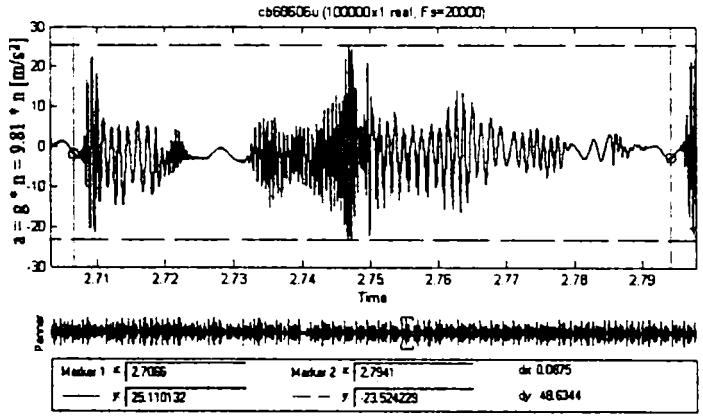


Fig. A.29. 37. (cb.686. 06u)

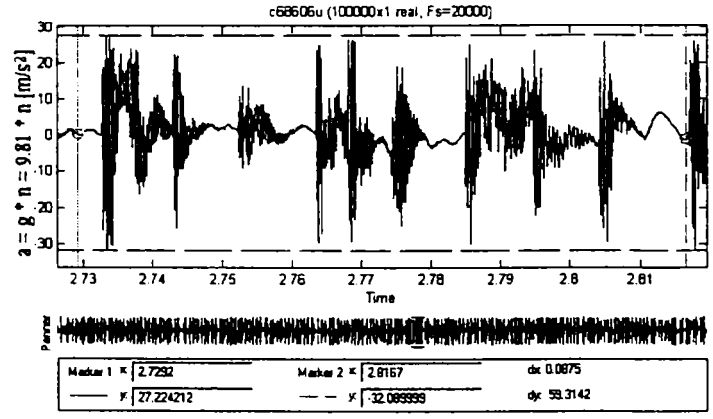


Fig. A.29.38. (c. 686. 06u)

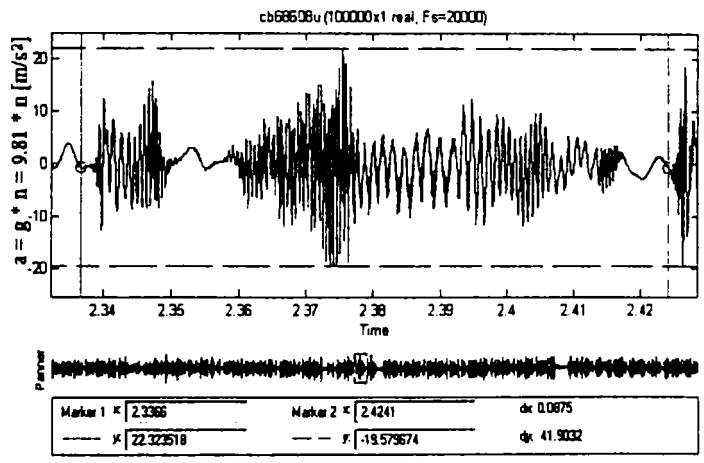


Fig. A.29. 39. (cb.686. 08u)

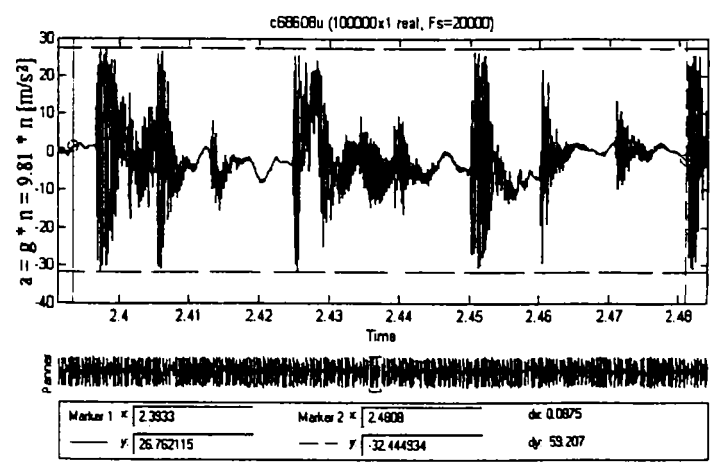


Fig. A.29. 40. (c. 686. 08u)

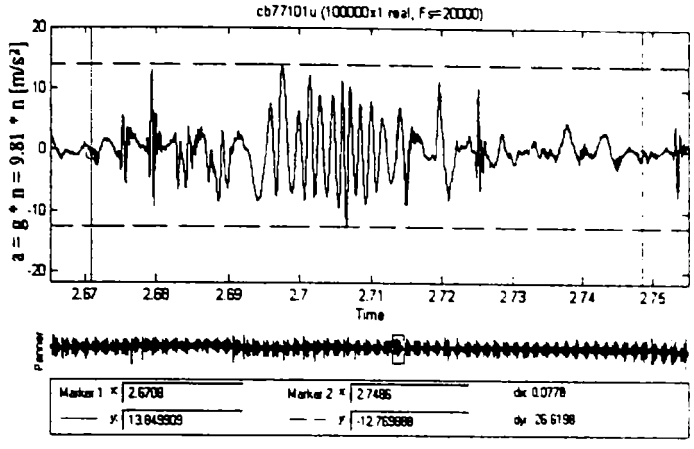


Fig. A.29. 41. (cb. 771. 01u)

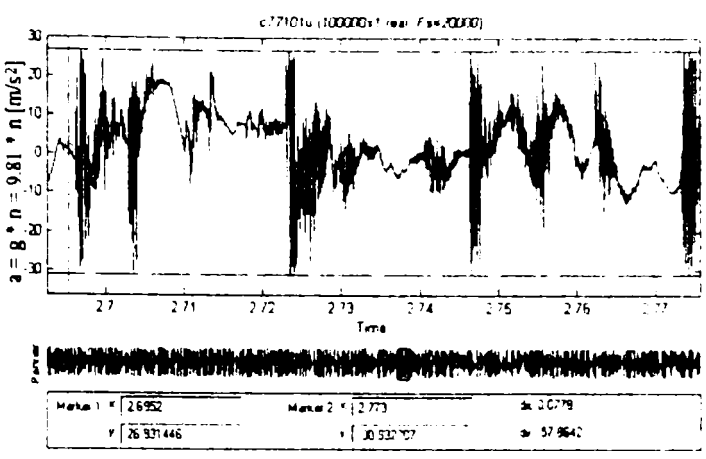


Fig. A.29. 42. (c.771. 01u)

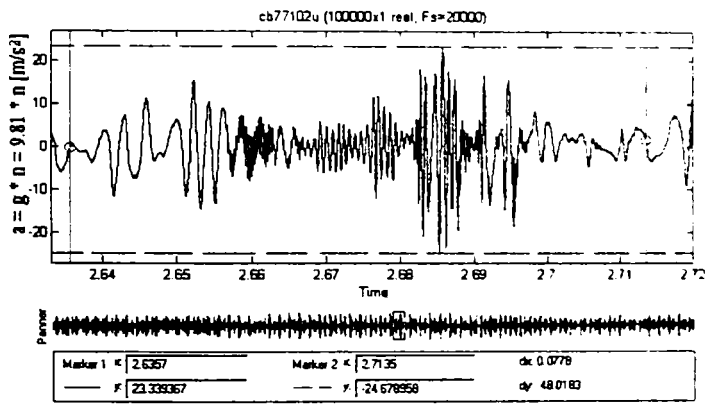


Fig. A.29. 43. (cb. 771. 02u)

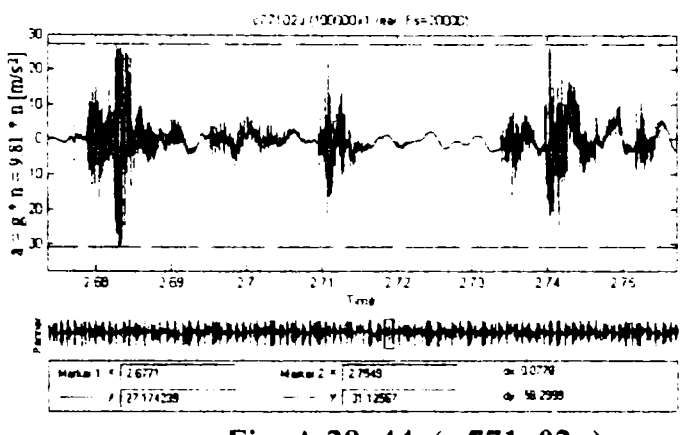


Fig. A.29. 44. (c.771. 02u)

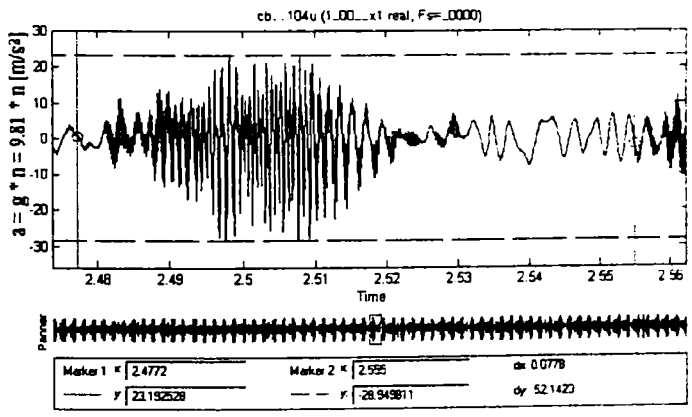


Fig. A.29. 45. (cb. 771. 04u)

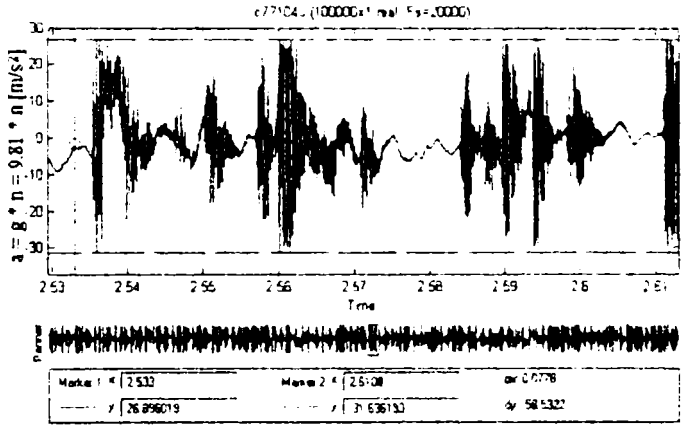


Fig. A.29. 46. (c.771. 04u)

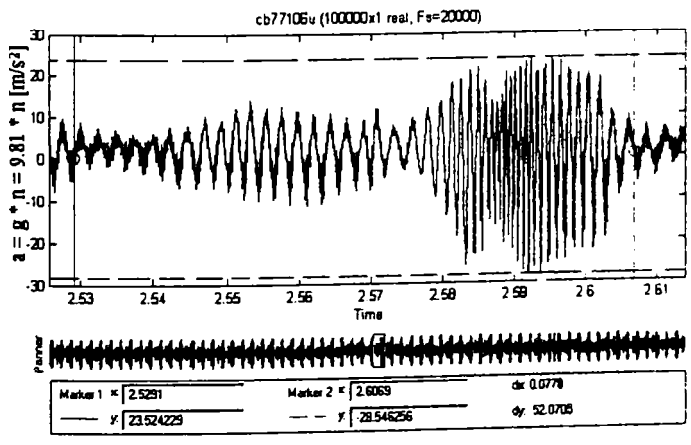


Fig. A.29. 47. (cb. 771. 06u)

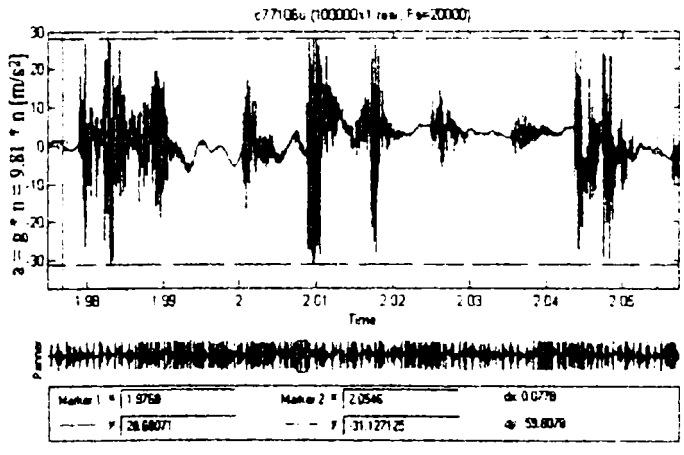


Fig. A.29. 48. (c.771. 06u)

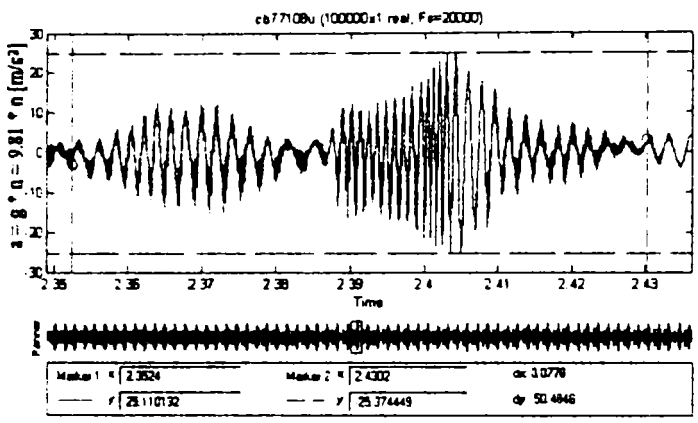


Fig. A.29. 49. (cb. 771. 08u)

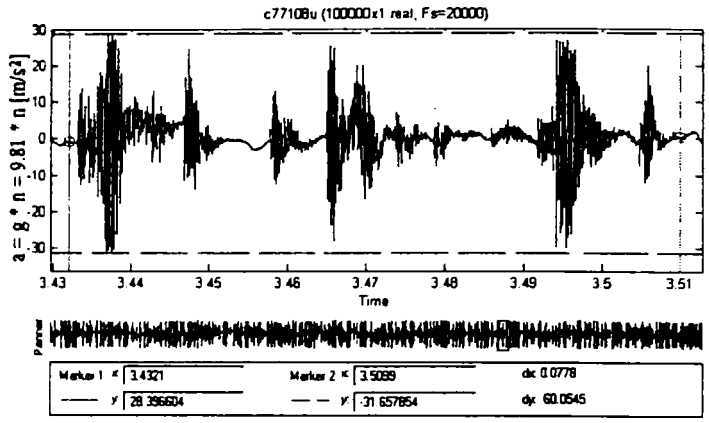


Fig. A.29. 50. (c.771. 08u)

Notă. Semnificaţia notaţiilor din parenteză:

- primele trei cifre reprezintă turaţiile arborelui maşinii-unelte , [rot/min];
- următoarele două cifre reprezintă viteza de avans, [mm/min];
- cb - aşchiere obişnuită;
- c – aşchiere vibropercutantă;
- u – măsurare acceleraţii pe universal.

Anexa 30.

Diagrame ale accelerațiilor măsurate pe arborele mașinii (pinolă) cu dispozitivul blocat (așchiere normală) și neblocat (așchiere vibropercutantă).

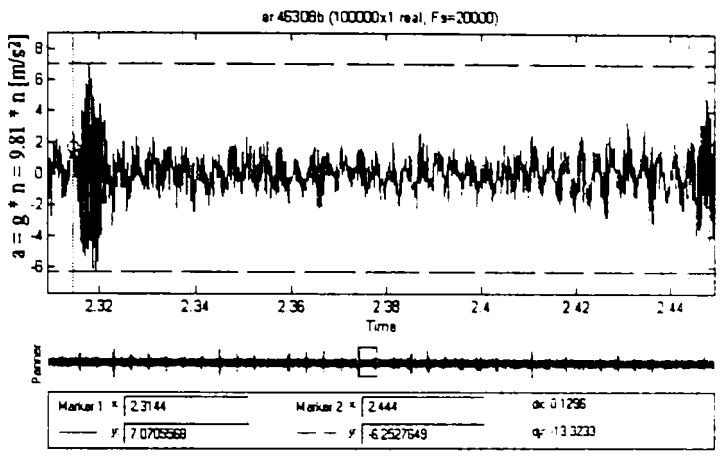


Fig. A30. 1. (ar.463. 08.b)

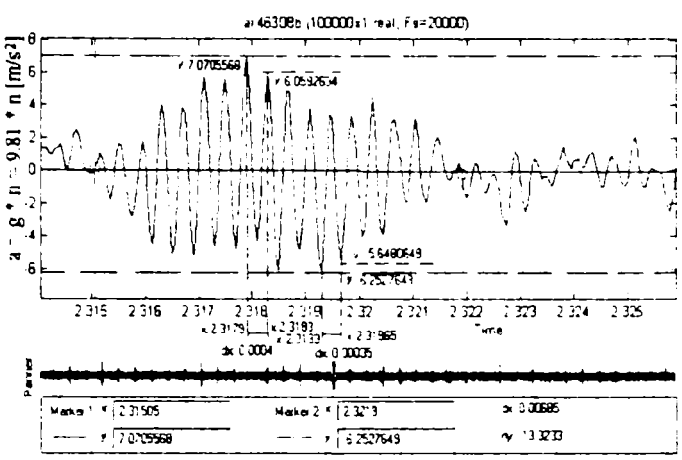


Fig. A30. 2. (ar. 463. 08.b detaliu)

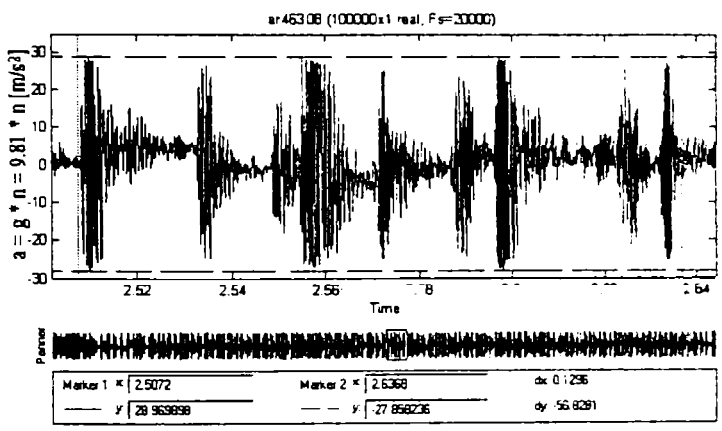


Fig. A30. 3. (ar.463. 08.c)

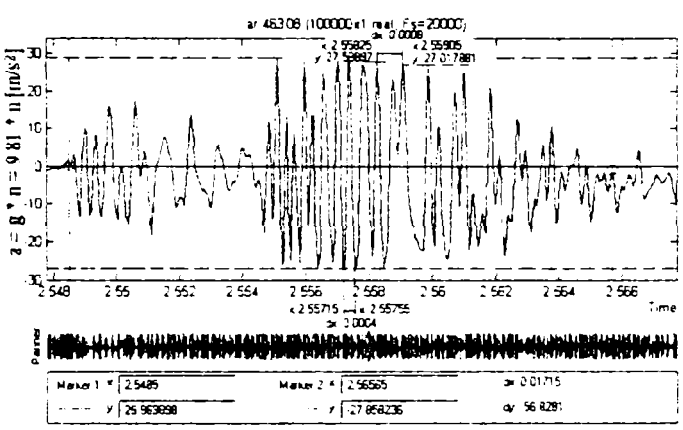


Fig. A30. 4. (ar. 463. 08.c detaliu)

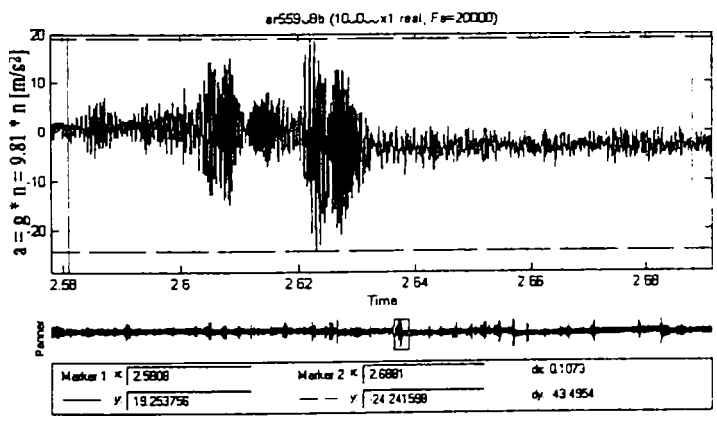


Fig. A30. 5. (ar. 559. 08.b)

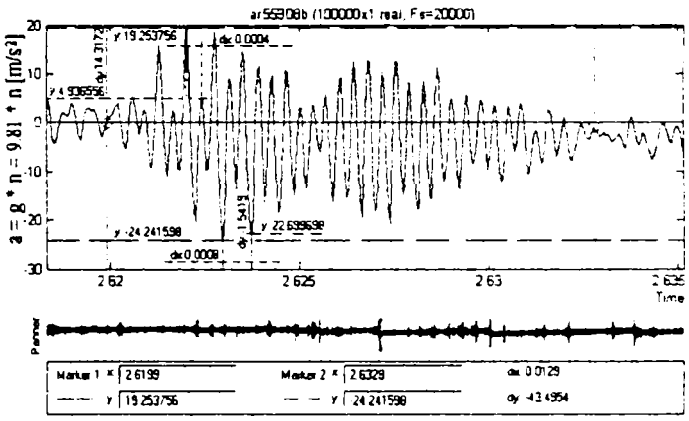


Fig. A30. 6. (ar. 559.08.b detaliu)

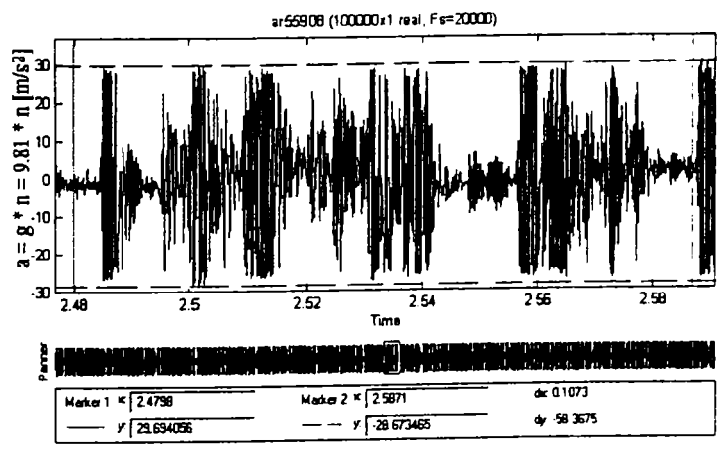


Fig. A30. 7. (ar.559.08.c)

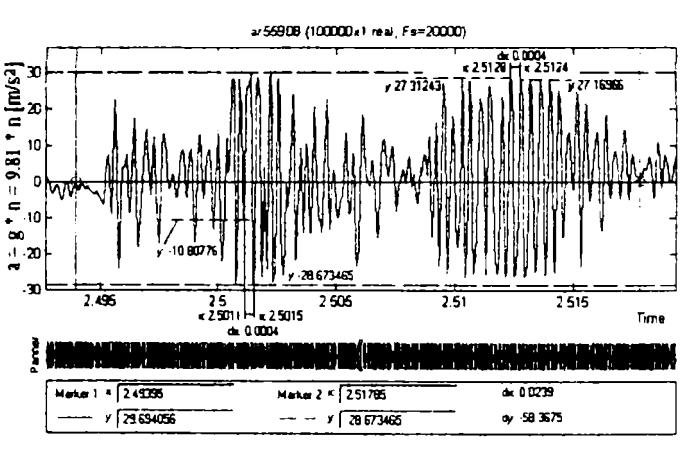


Fig. A30. 8. 9. (ar.559.08.c detaliu)

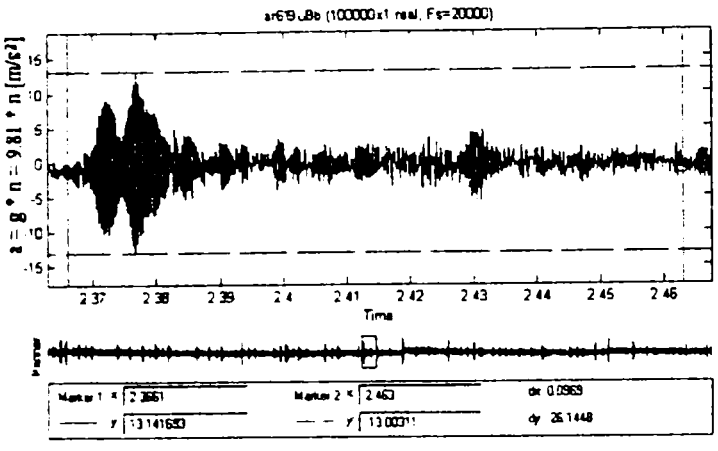


Fig. A30. 9. (ar. 619.08.b)

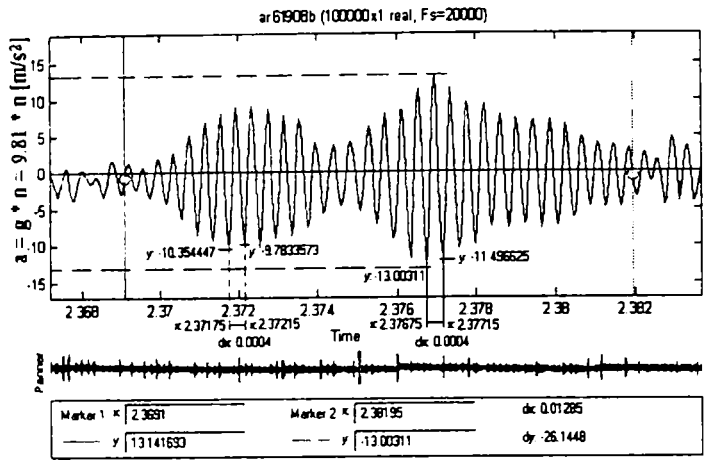


Fig. A30. 10. (ar. 619.08.b detaliu)

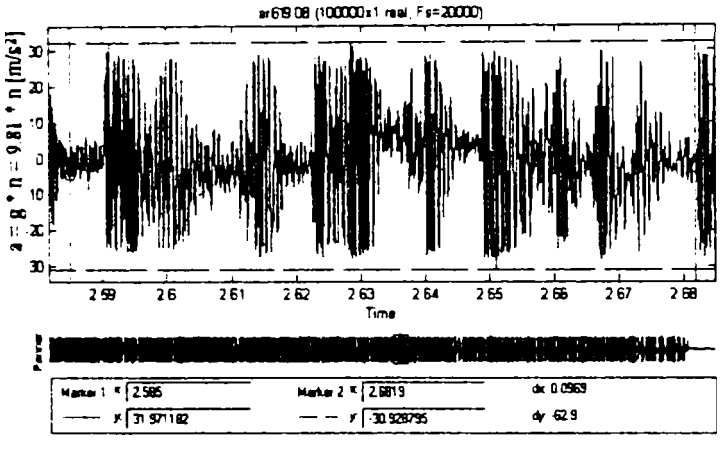


Fig. A30. 11. (ar .619. 08.c)

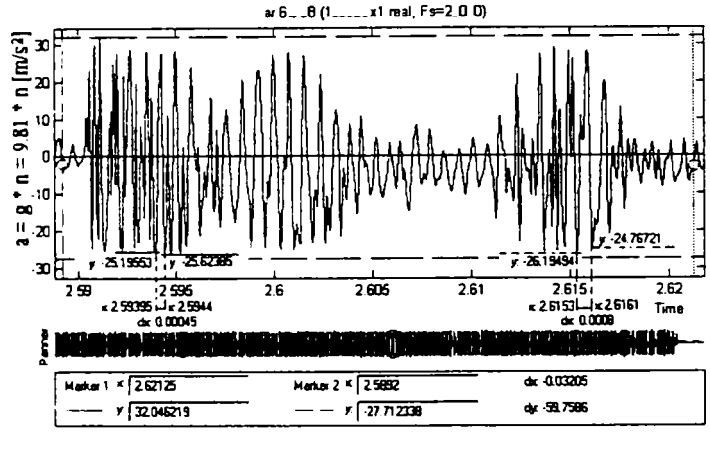


Fig. A30. 12. (ar. 619. 08.c detaliu)

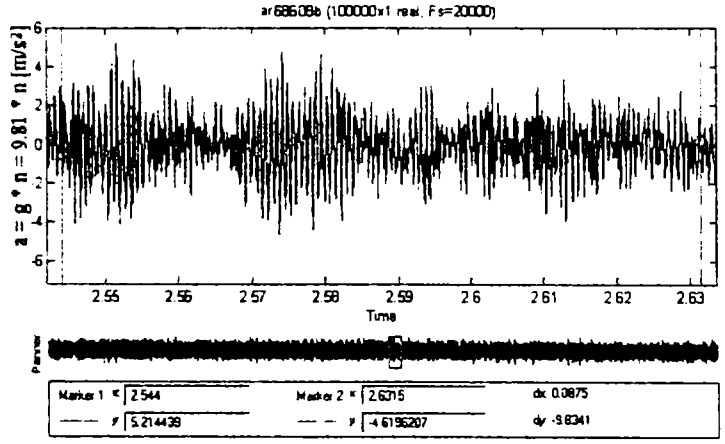


Fig. A30. 13. (ar.686. 08.b)

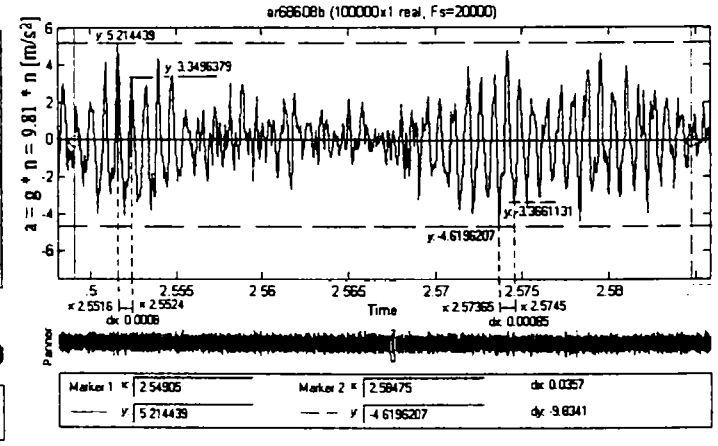


Fig. A30. 14. (ar.686. 08.b detaliu)

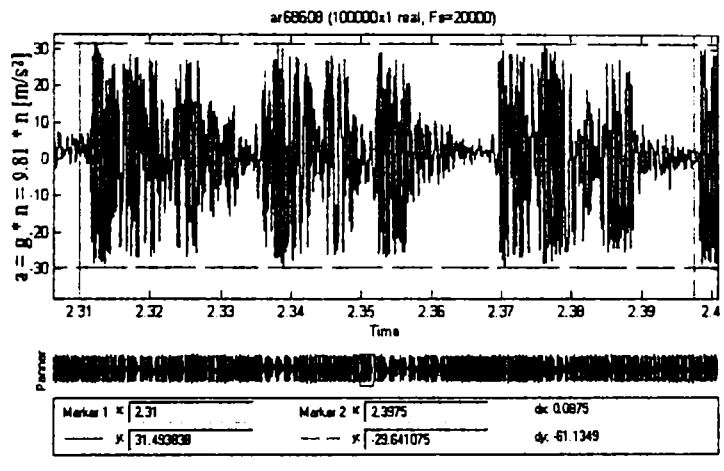


Fig. A30. 15. (ar. 686. 08.c)

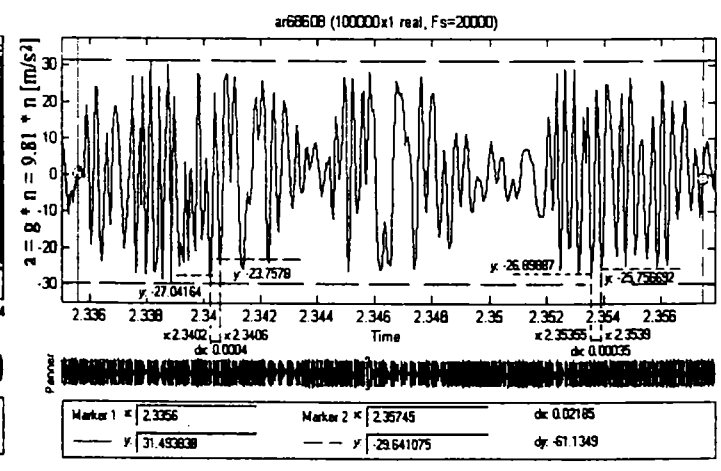


Fig. A30. 16. (ar. 686. 08.c detaliu)

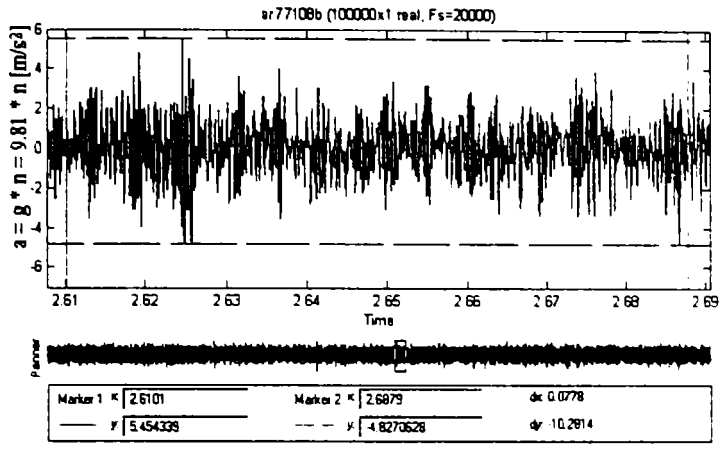


Fig. A30. 17. (ar. 771. 08.b)

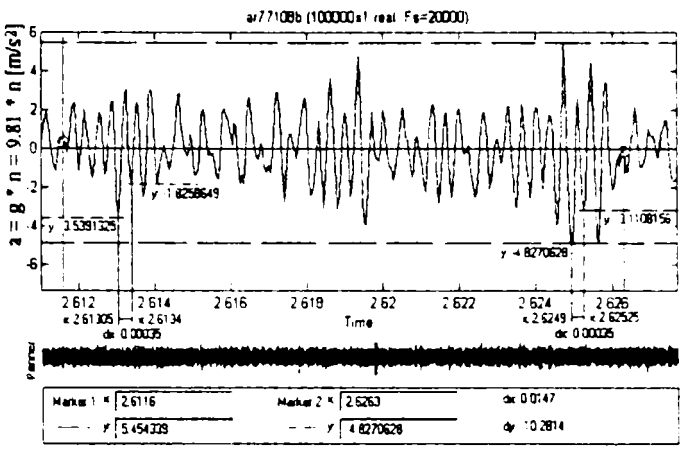


Fig. A30. 18. (ar. 771. 08.b detaliu)

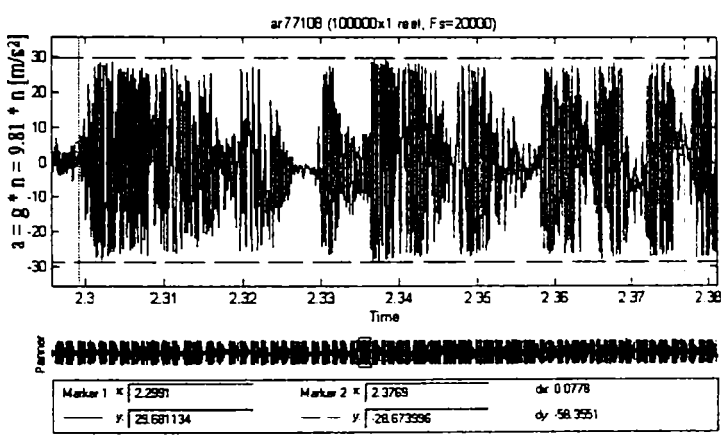


Fig. A30. 19. (ar.771. 08.c)

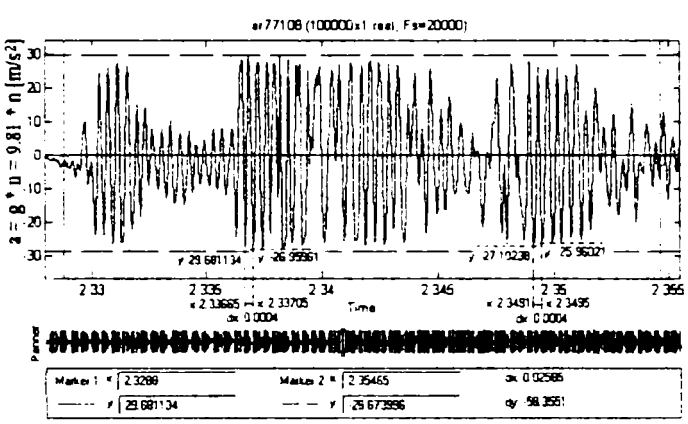


Fig. A30. 20. (ar. 771. 08.c detaliu)

Notă. Semnificația notațiilor din paranteză:

- ar – accelerații măsurate pe pinola arborelui mașinii unelte;
- primele trei cifre reprezintă turațiile arborelui mașinii-unelte , [rot/min];
- următoarele două cifre reprezintă viteza de avans, [mm/min];
- b - așchiere obișnuită;
- c – așchiere vibropercutantă;

Anexa 31.

Diagramele forței axiale de așchiere pentru dispozitiv blocat și dispozitiv neblocat

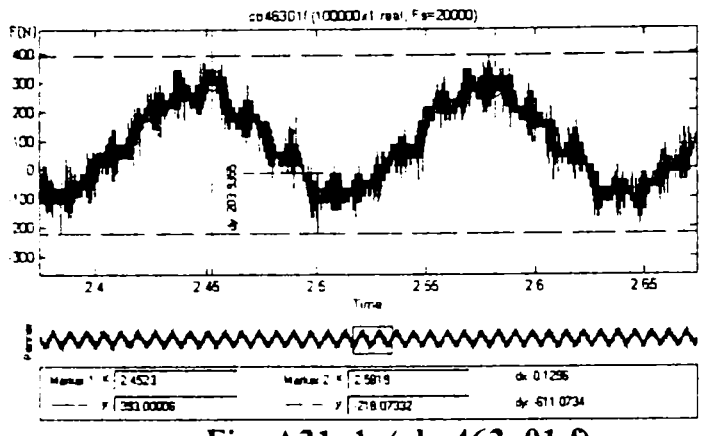


Fig. A31. 1. (cb. 463. 01.f)

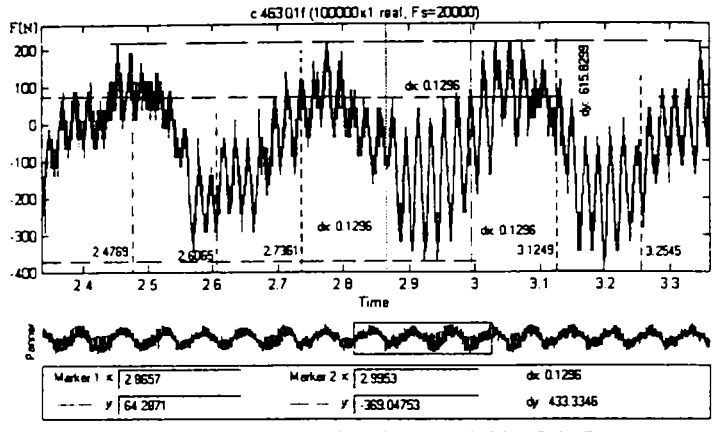


Fig. A31.2. (c. 463. 01.f)

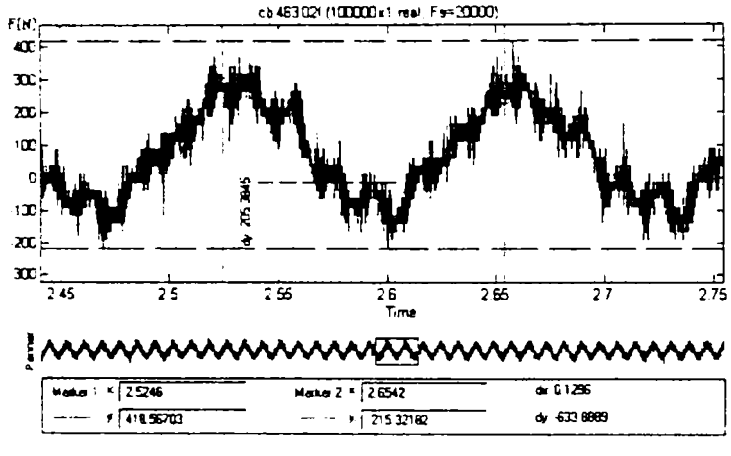


Fig. A31. 3. (cb. 463. 02.f)

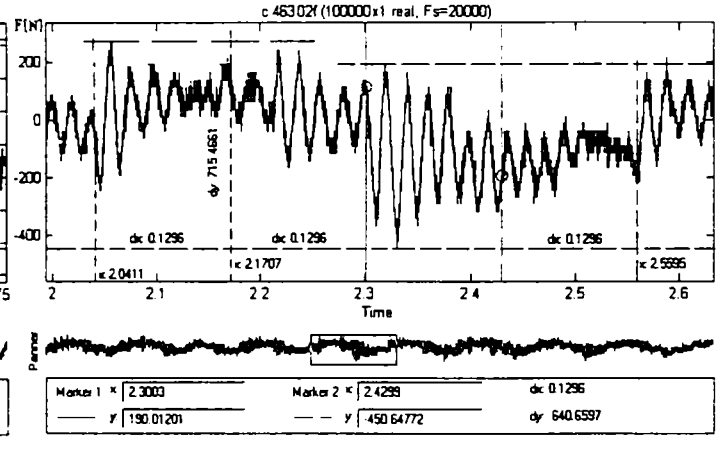


Fig. A31. 4. (c. 463. 02.f)

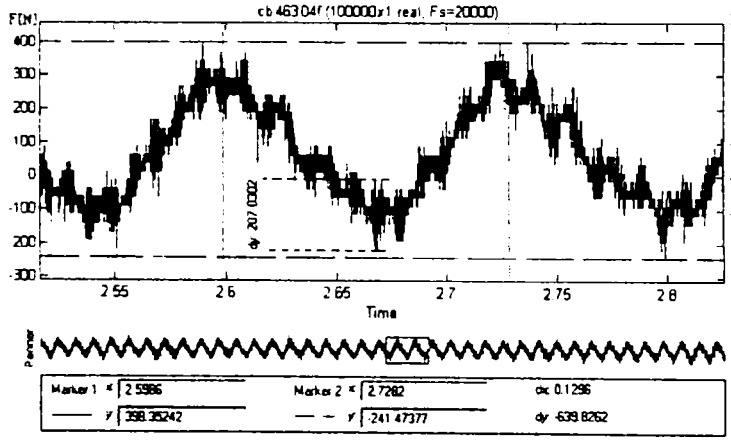


Fig. A31. 5. (cb. 463. 04.f)

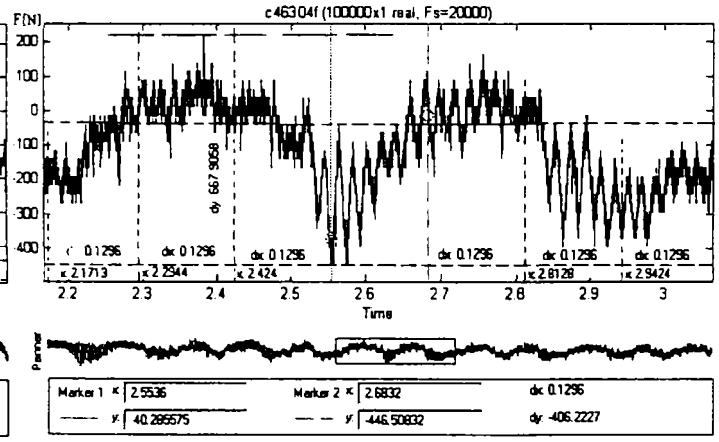


Fig. A31. 6. (c. 463. 04.f)

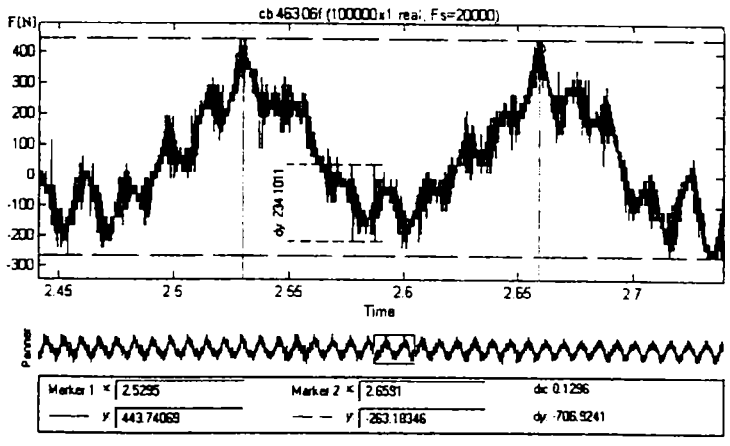


Fig. A31. 7. (cb. 463. 06.f)

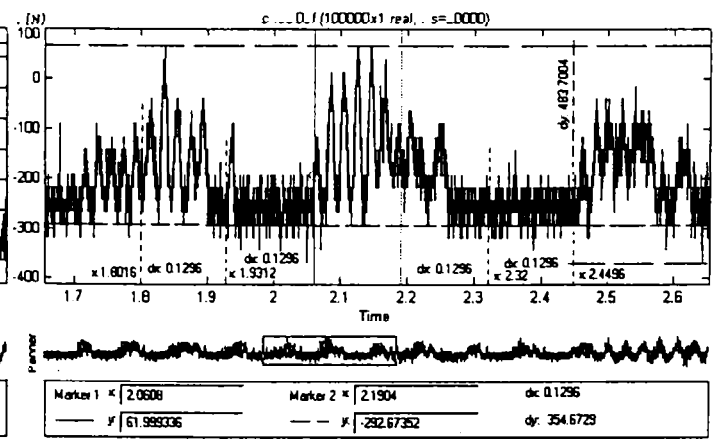


Fig. A31. 8. (c. 463. 06.f)

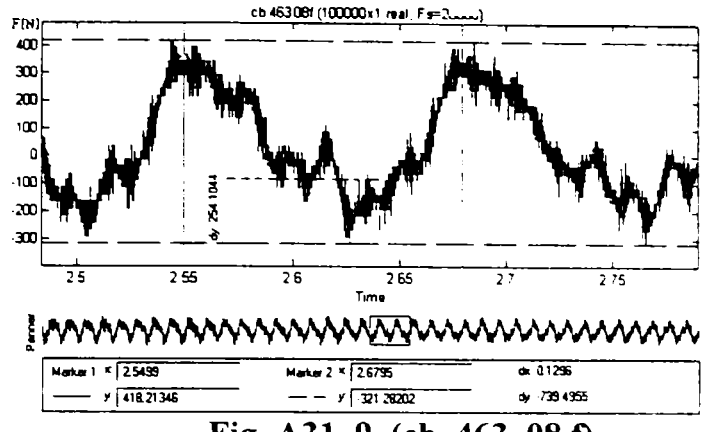


Fig. A31. 9. (cb. 463. 08.f)

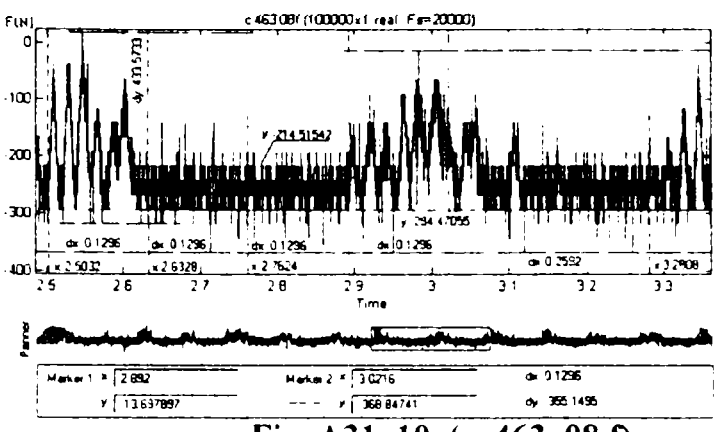


Fig. A31. 10. (c. 463. 08.f)

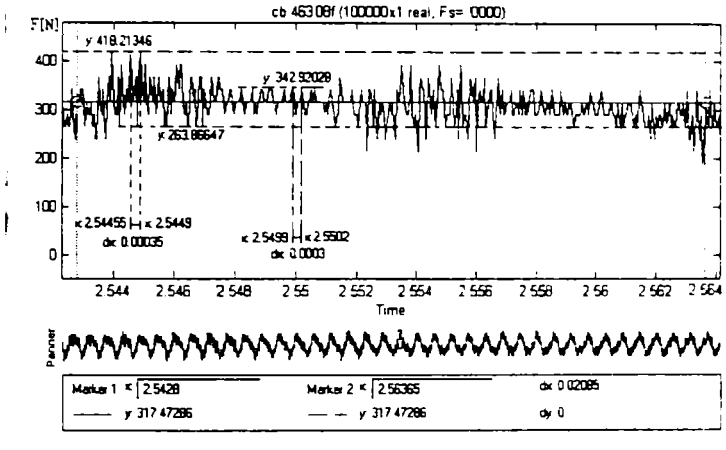


Fig. A31. 11. (cb. 463. 08.f detaliu)

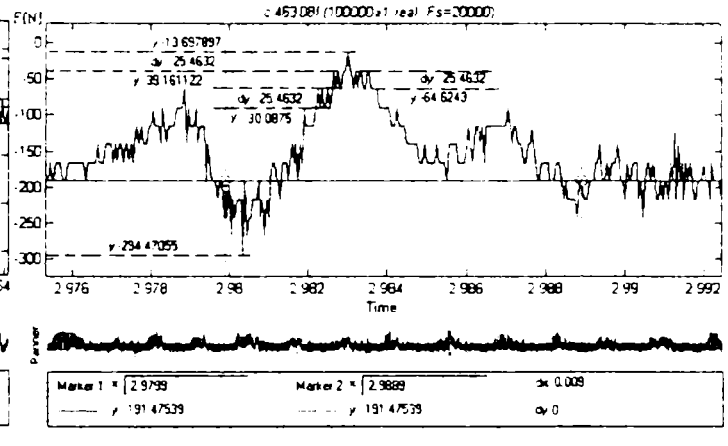


Fig. A31. 12. (c. 463. 08.f detaliu)

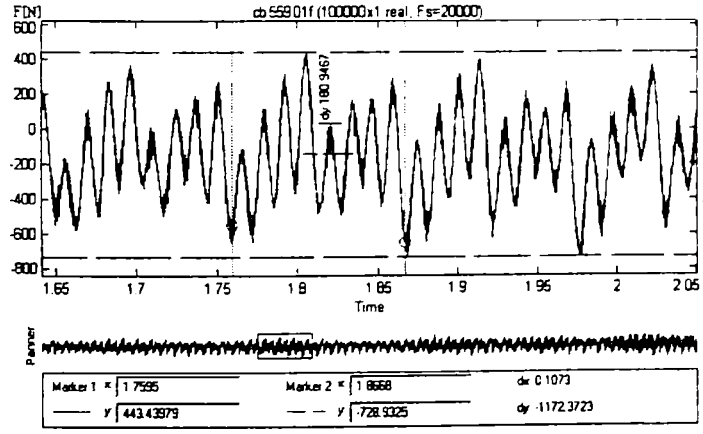


Fig. A31. 13. (cb.559.01.f)

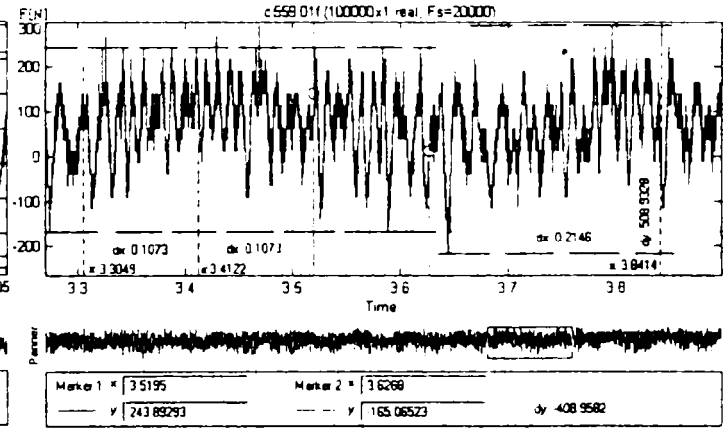


Fig. A31. 14. (c. 559. 01.f)

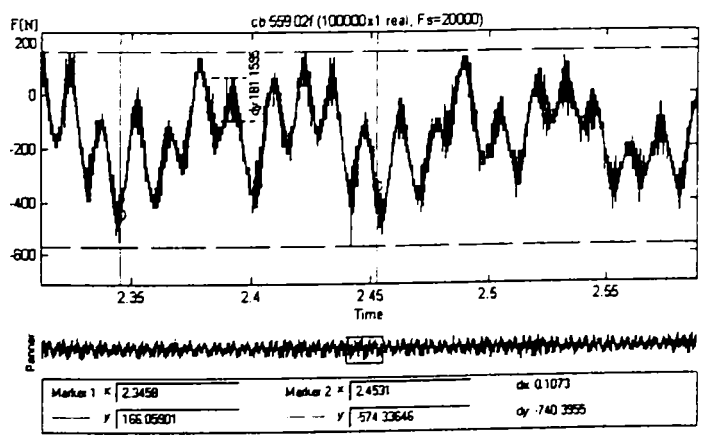


Fig. A31. 15. (cb.559.02.f)

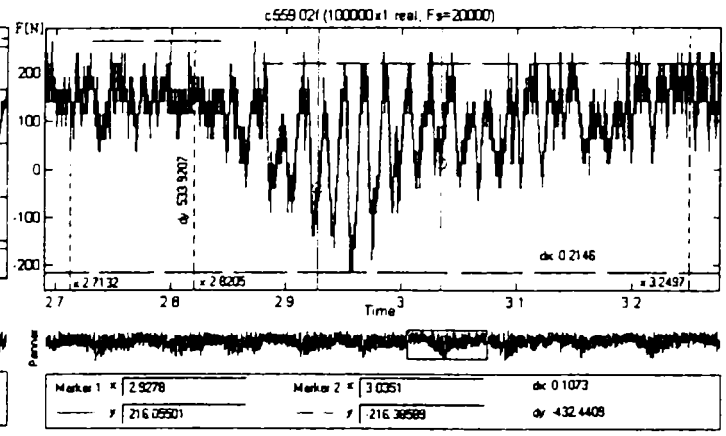


Fig. A31. 16. (c. 559. 02.f)

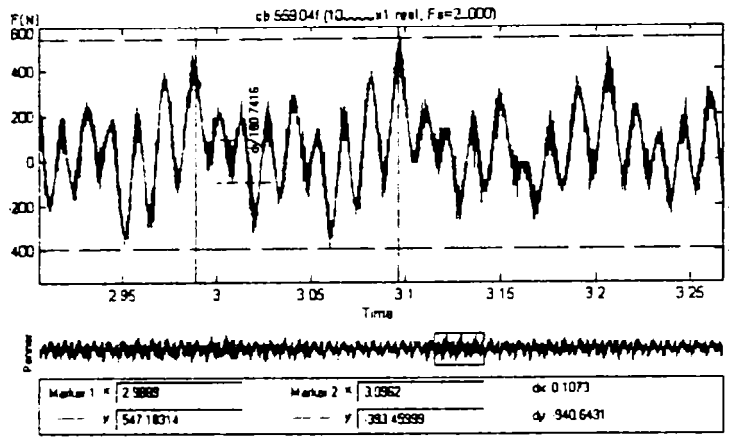


Fig. A31. 17. (cb.559.04.f)

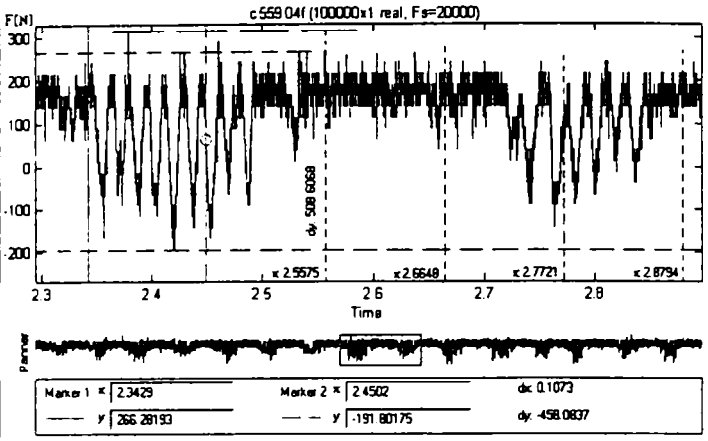


Fig. A31. 18. (c. 559. 04.f)

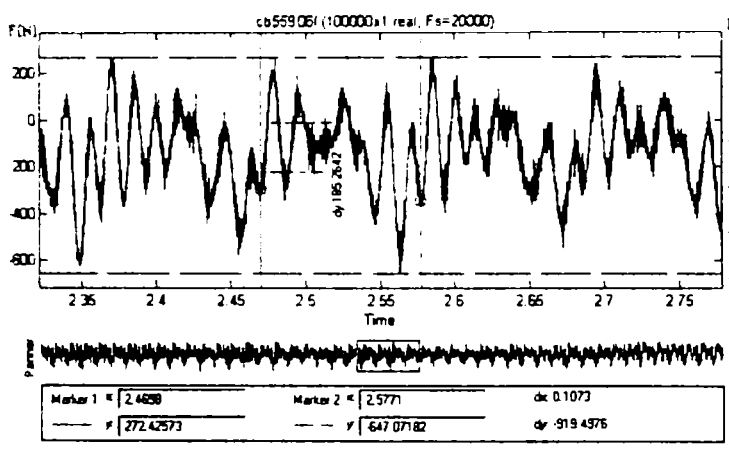


Fig. A31. 19. (cb.559.06.f)

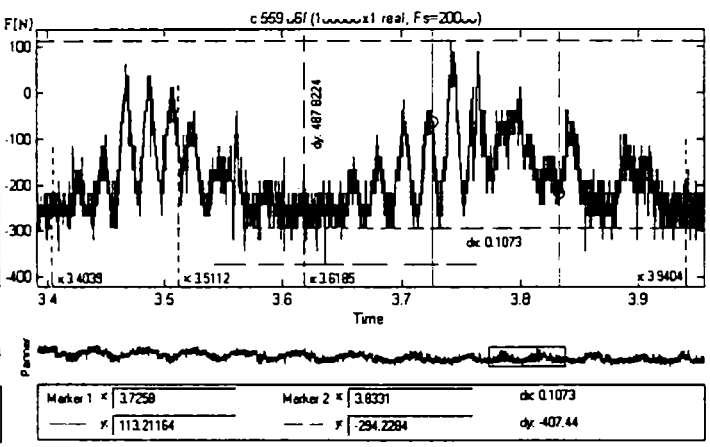


Fig. A31. 20. (c. 559. 06.f)

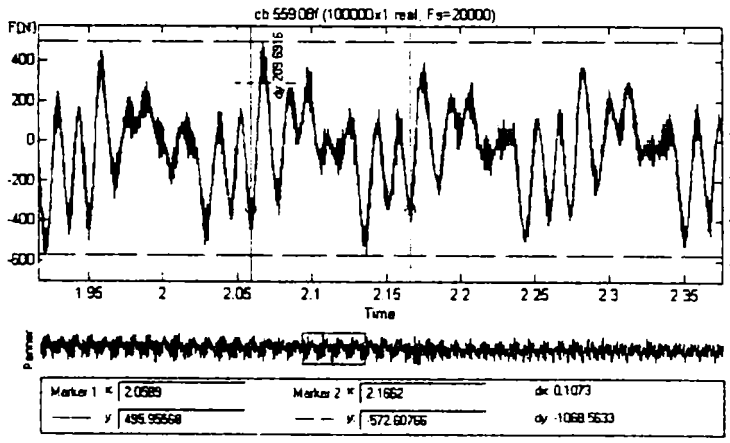


Fig. A31. 21. (cb.559.08.f)

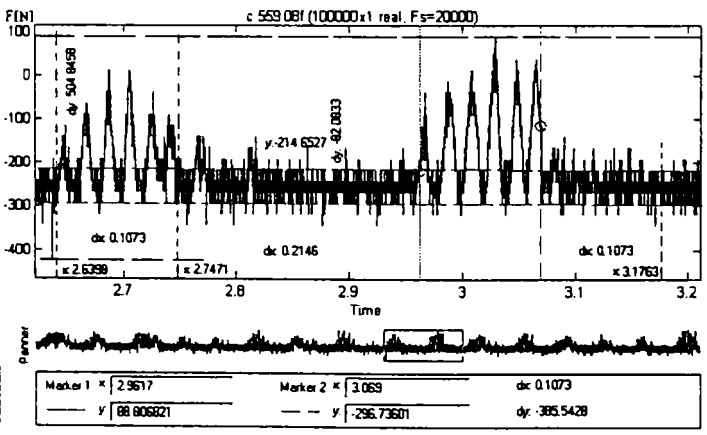


Fig. A31. 22. (c. 559. 08.f)

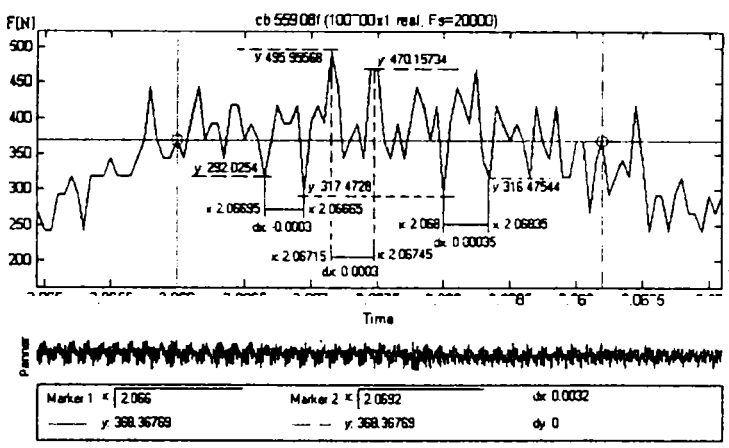


Fig. A31. 23. (cb.559.08.f detaliu)

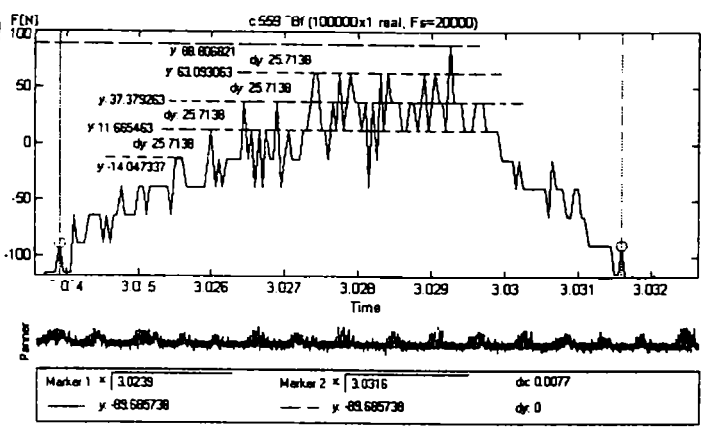


Fig. A31. 24. (c. 559. 08.f detaliu)

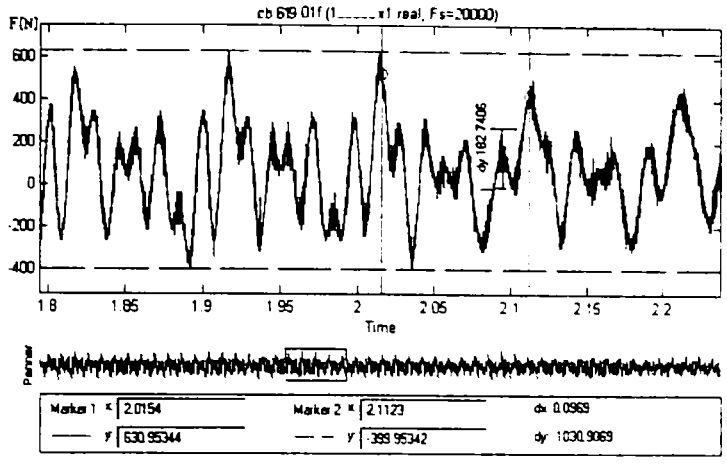


Fig. A31. 25. (cb.619. 01.f)

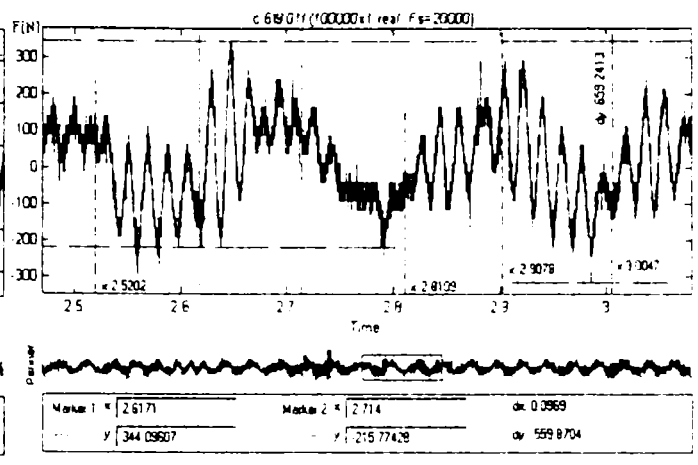


Fig. A31. 26. (c.619. 01.f)

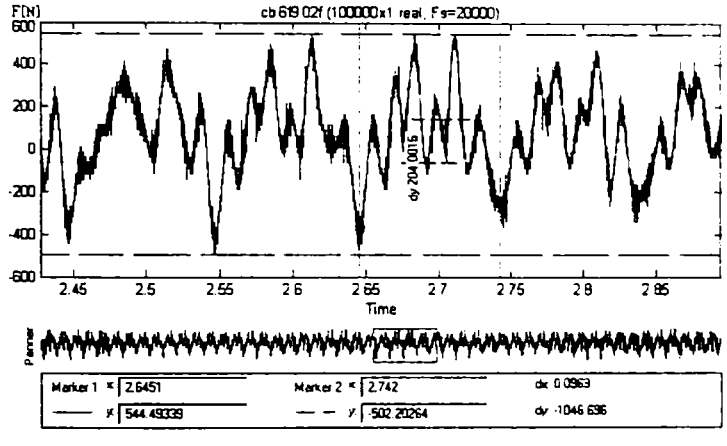


Fig. A31. 27. (cb.619. 02.f)

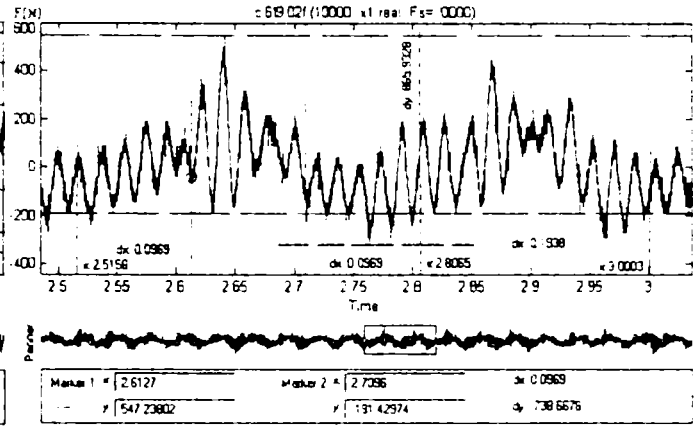


Fig. A31. 28. (c.619. 02.f)

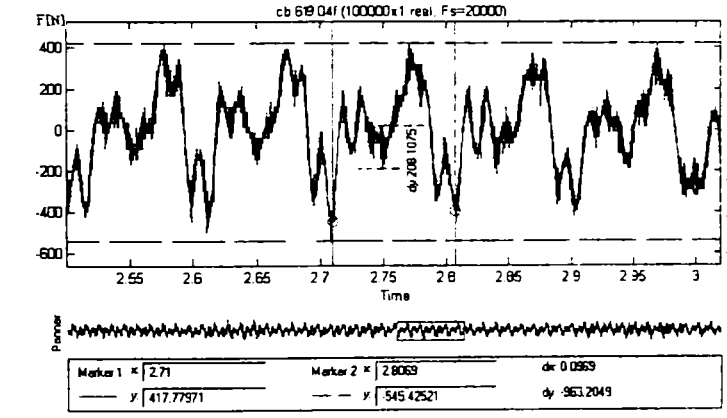


Fig. A31. 29. (cb.619. 04.f)

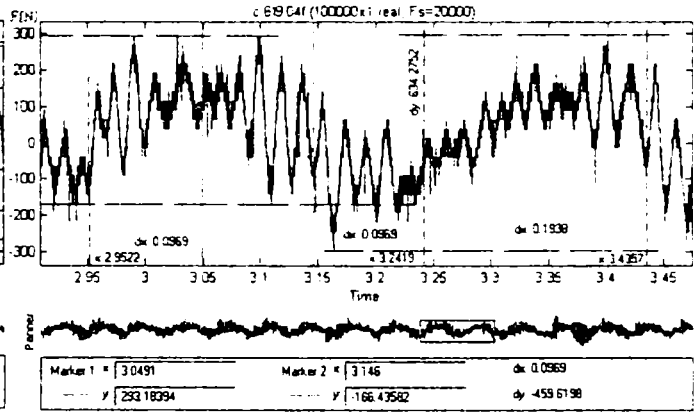


Fig. A31. 30. (c.619. 04.f)

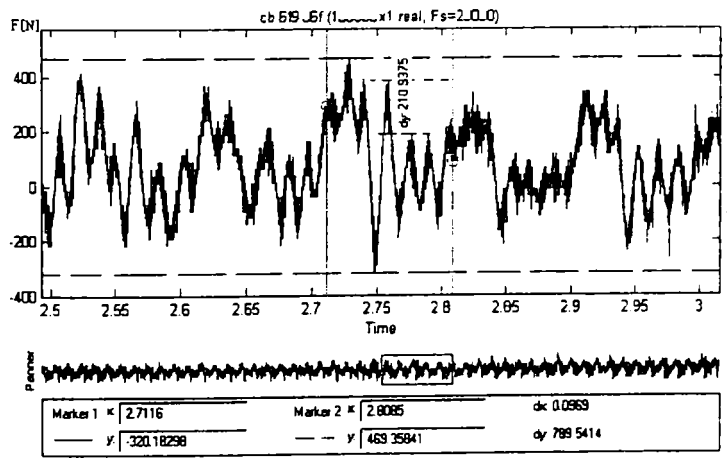


Fig. A31. 31. (cb.619. 06.f)

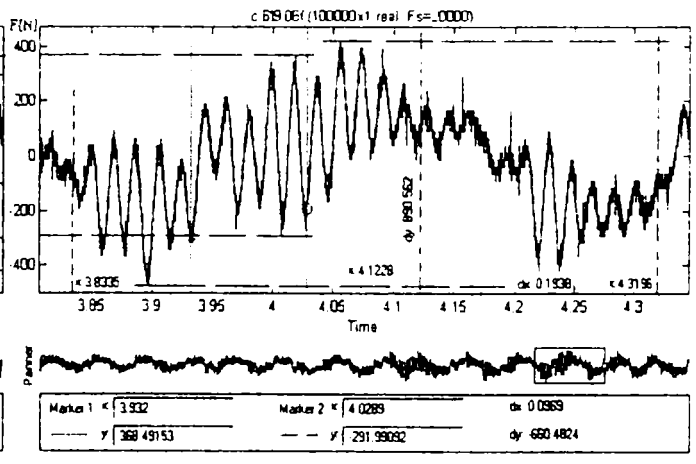


Fig. A31. 32. (c.619. 06.f)

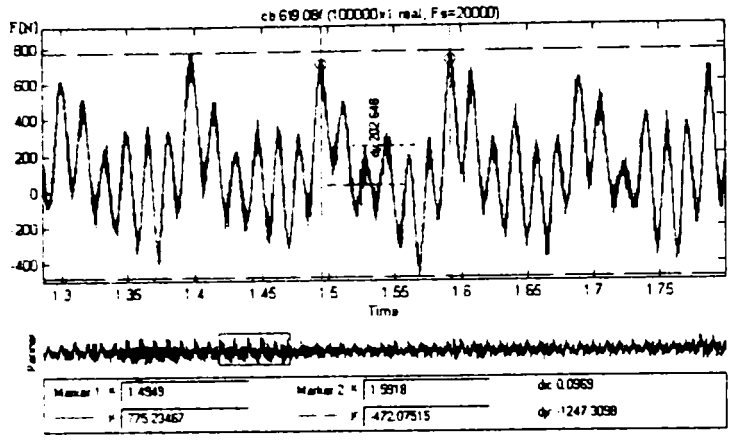


Fig. A31. 33. (cb.619. 08.f)

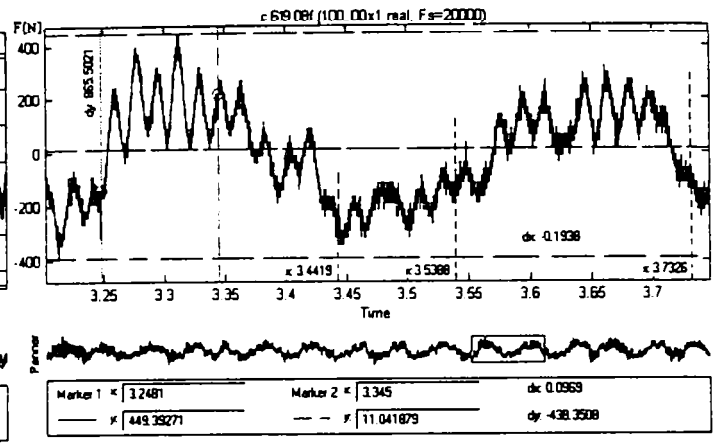


Fig. A31. 34. (c.619. 08.f)

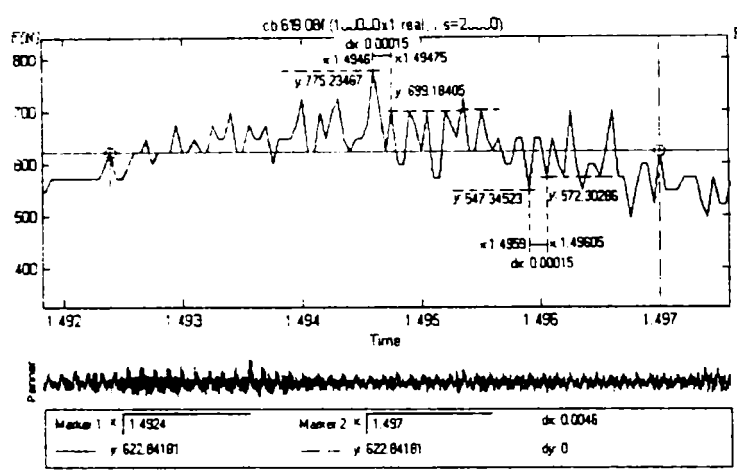


Fig. A31. 35. (cb.619. 08.f detaliu)

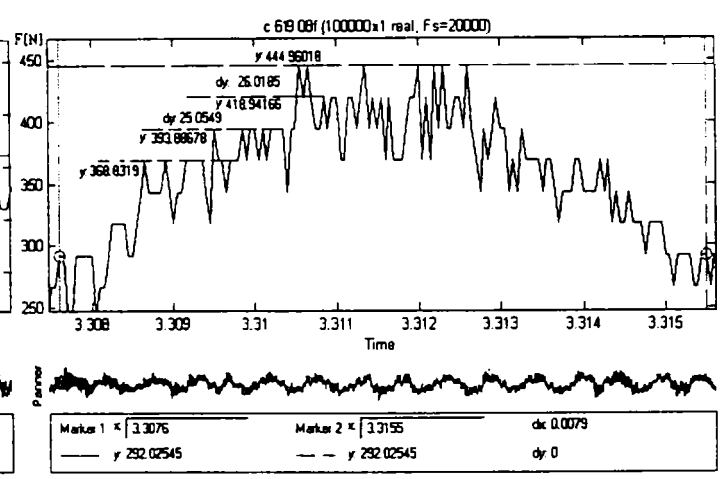


Fig. A31. 36. (c.619. 08.f detaliu)

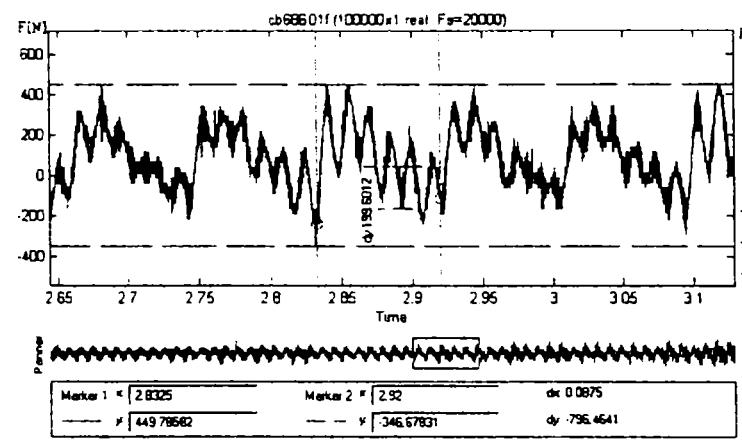


Fig. A31. 37. (cb. 686. 01.f)

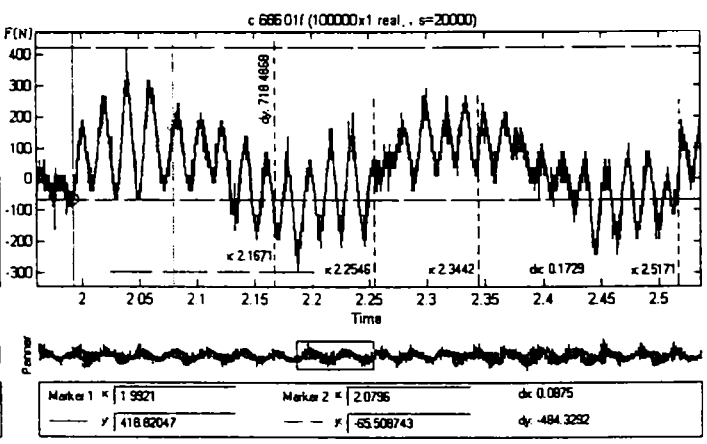


Fig. A31. 38. (c. 686. 01.f)

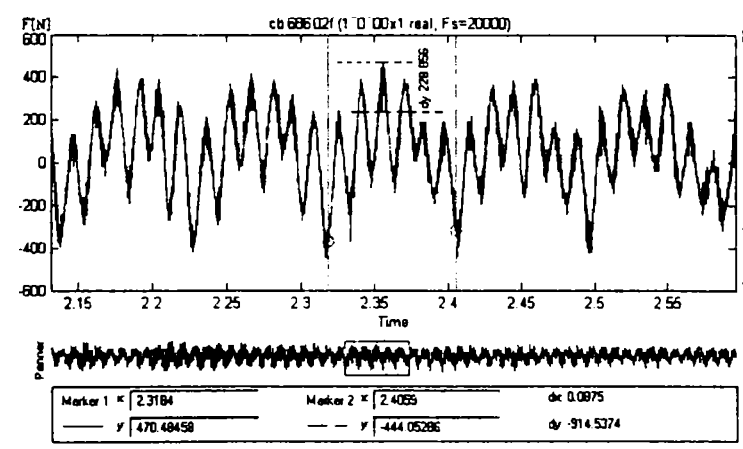


Fig. A31. 39. (cb. 686. 02.f)

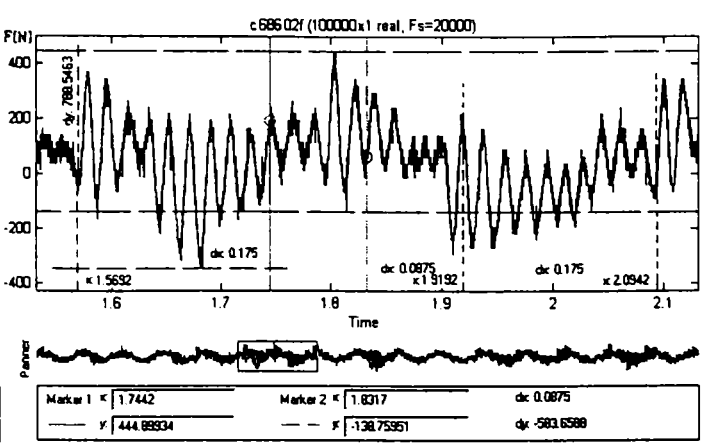


Fig. A31. 40. (c. 686. 02.f)

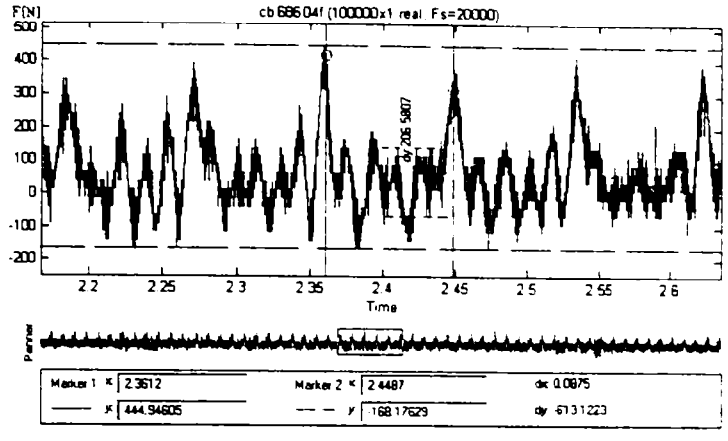


Fig. A31. 41. cb. 686. 04f

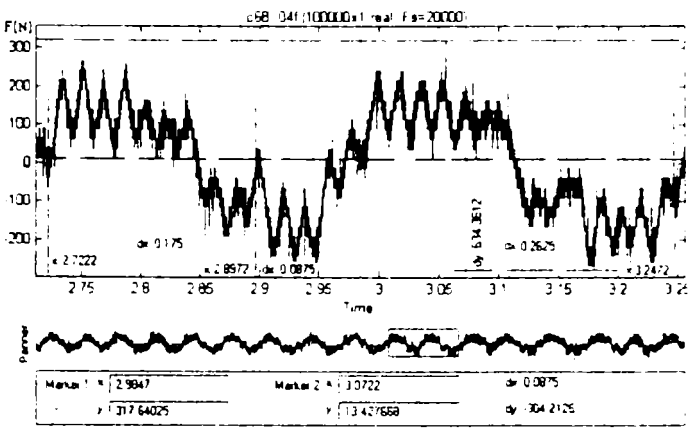


Fig. A31. 42. (c. 686. 04.f)

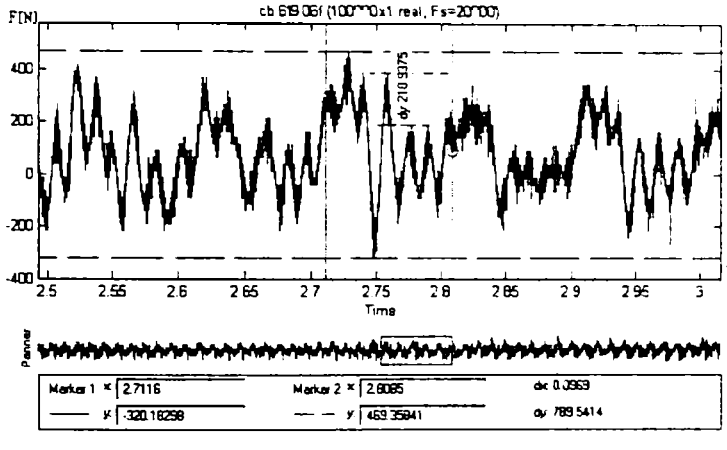


Fig. A31. 43. (cb. 686. 06.f)

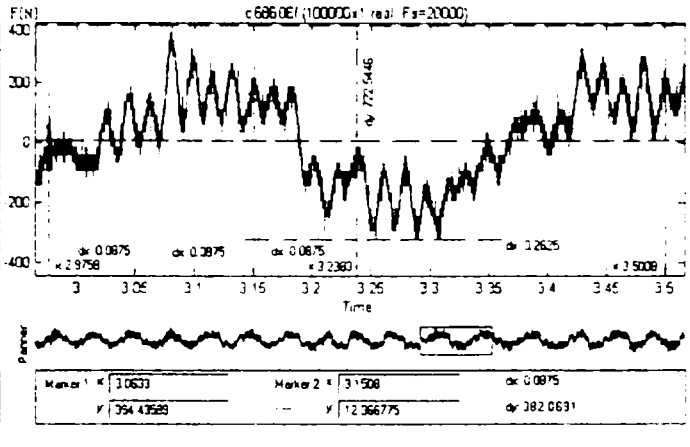


Fig. A31. 44. (c. 686. 06.f)

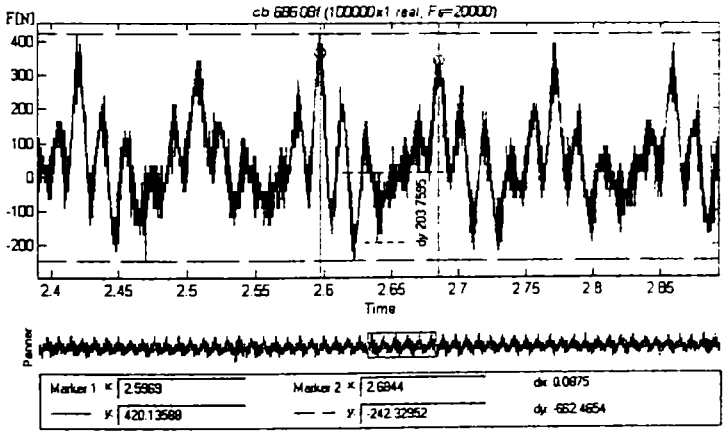


Fig. A31. 45. (cb. 686. 08.f)

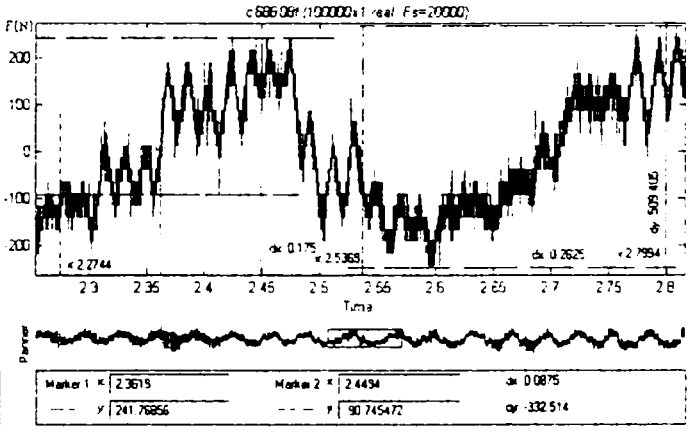


Fig. A31. 46. (c. 686. 08.f)

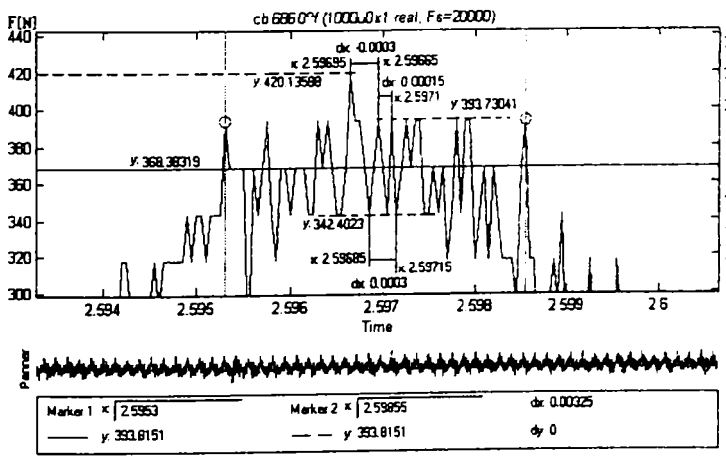


Fig. A31. 47. (cb. 686. 08.f detaliu)

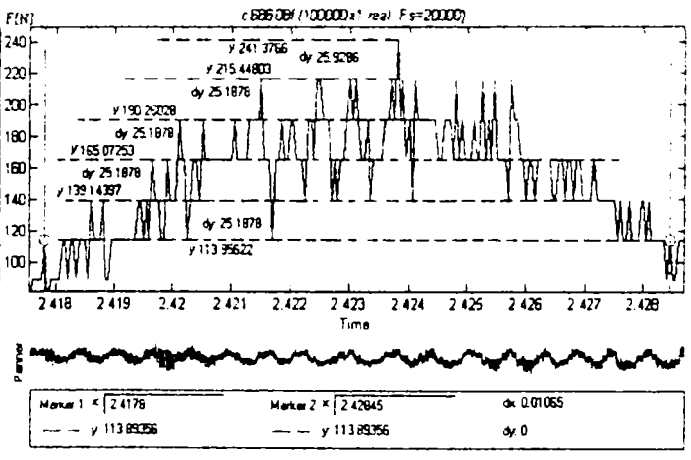


Fig. A31. 48. (c. 686. 08.f detaliu)

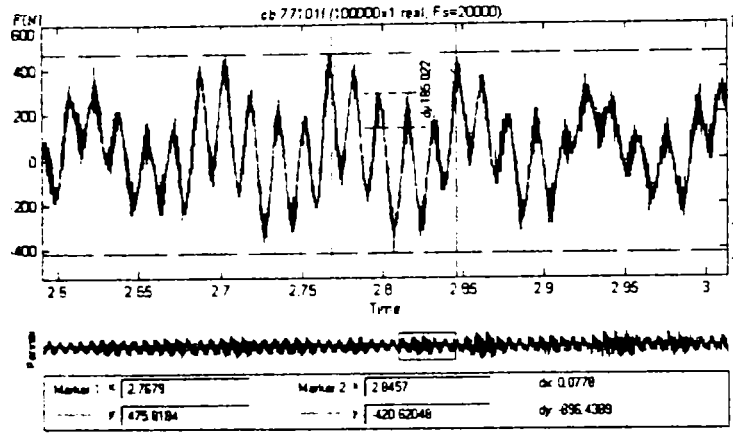


Fig. A31. 49. (cb.771. 0.1.f)

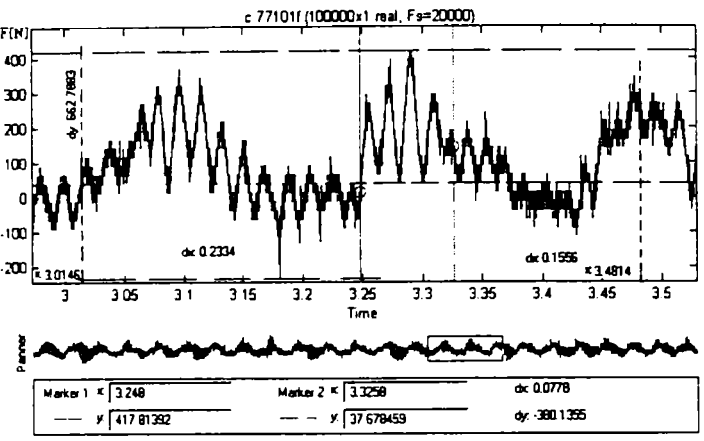


Fig. A31. 50. (c.771. 01.f)

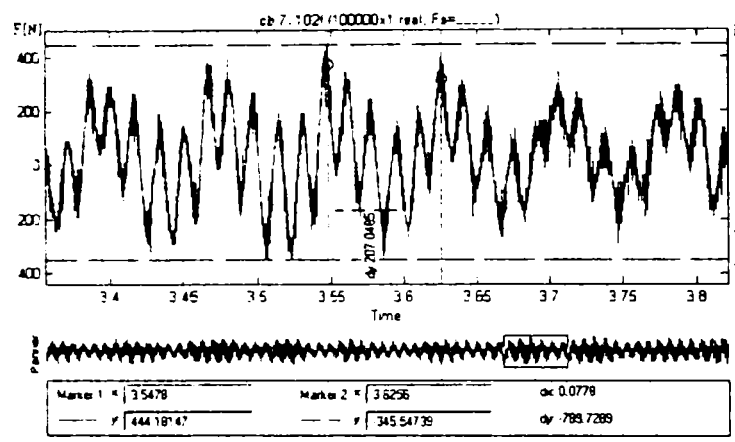


Fig. A31. 51. (cb.771. 02.f)

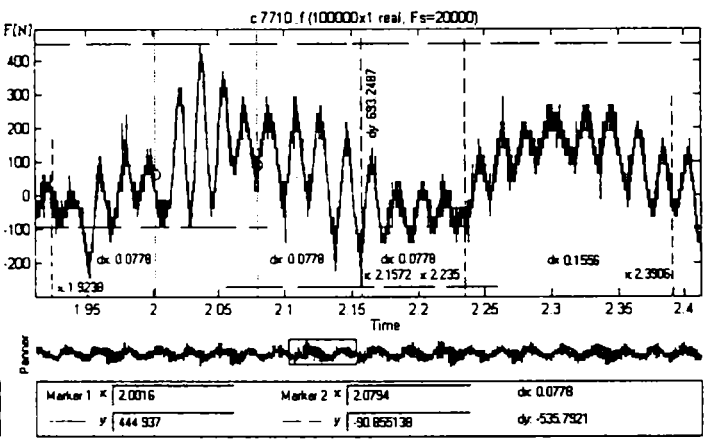


Fig. A31. 52. (c.771. 02.f)

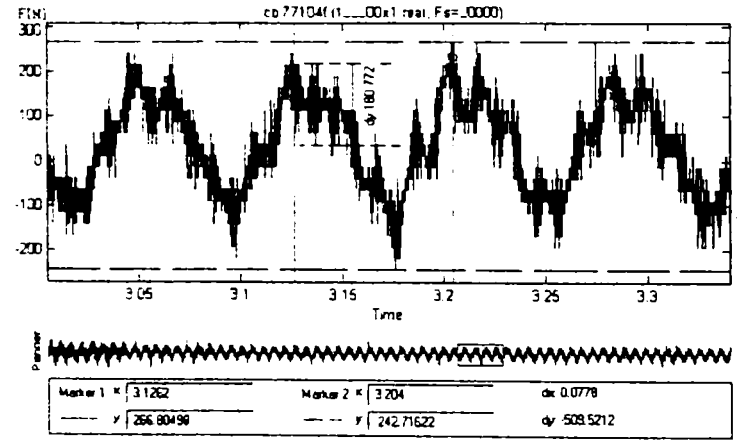


Fig. A31. 53. (cb.771. 04.f)

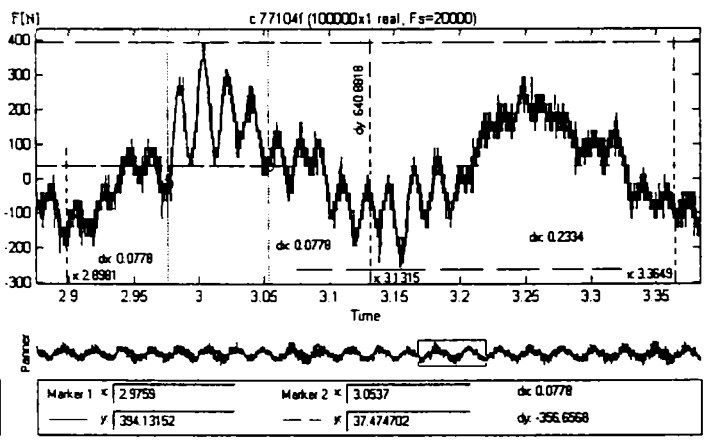


Fig. A31. 54. (c.771. 04.f)

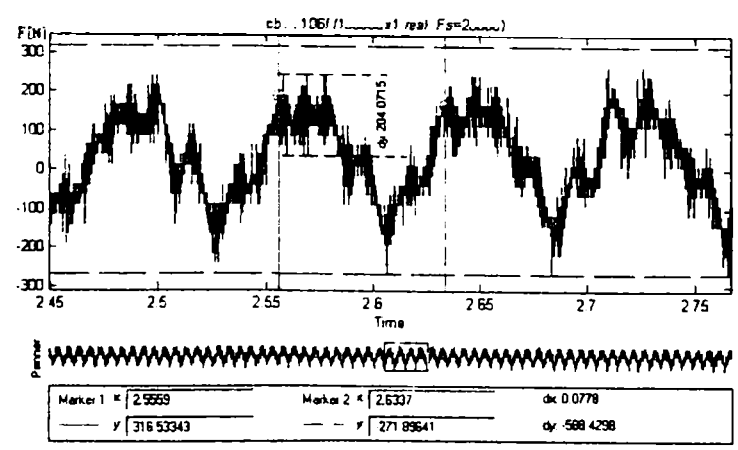


Fig. A31. 55. (cb.771. 06.f)

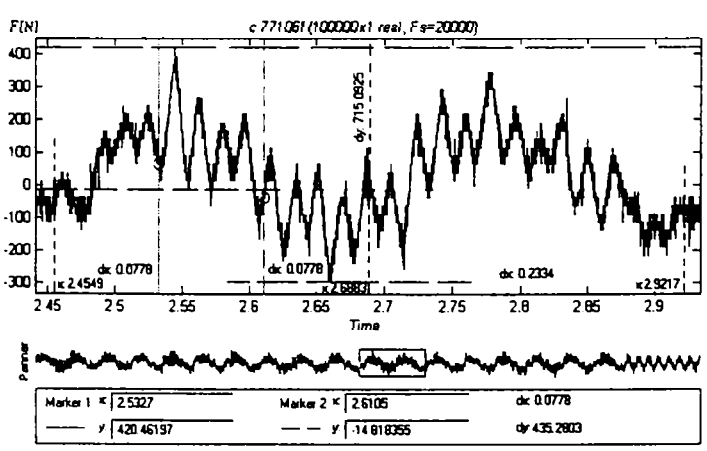


Fig. A31. 56. (c.771. 06.f)

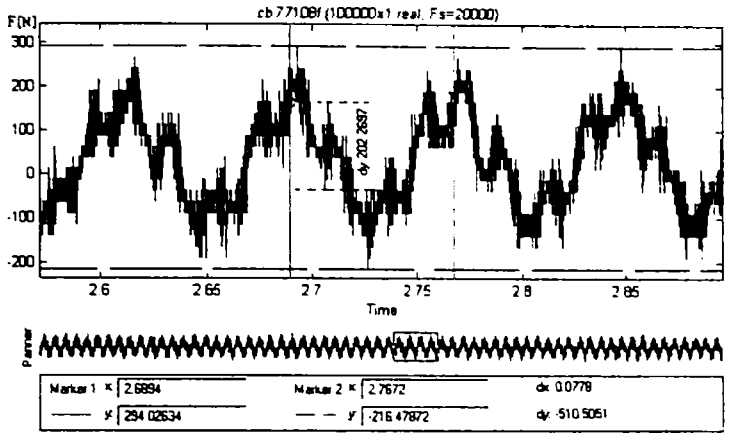


Fig. A31. 57. (cb.771. 08.f)

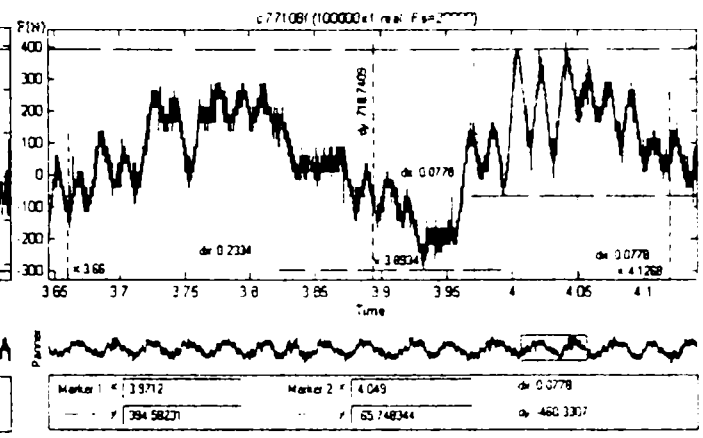


Fig. A31. 58. (c.771. 08.f)

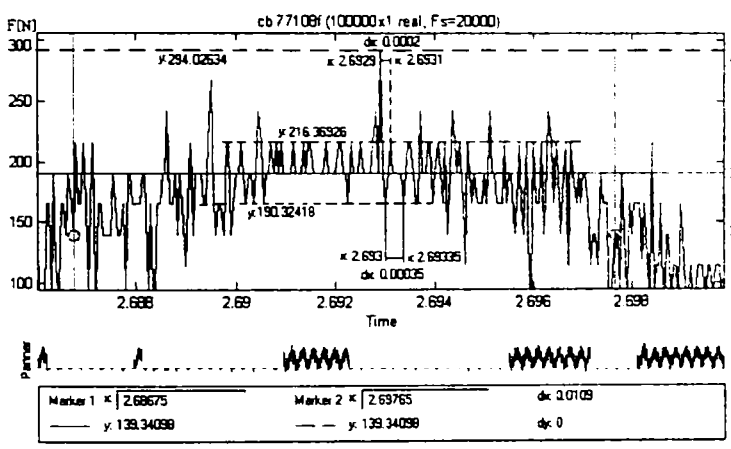


Fig. A31. 59. (cb.771. 08.f detaliu)

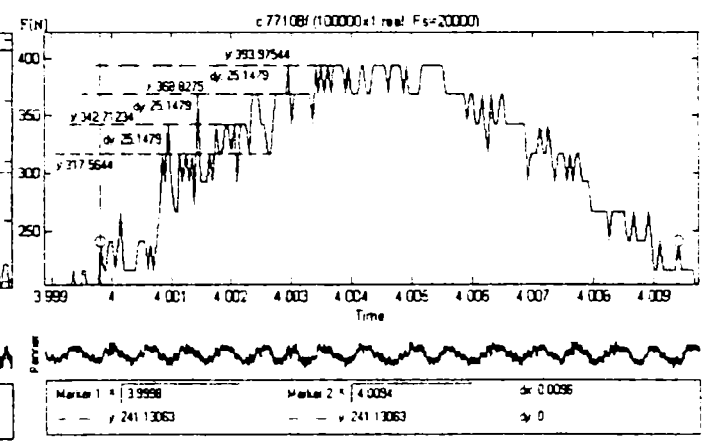


Fig. A31. 60. (c.771. 08.f detaliu)

Notă. Semnificația notărilor din paranteză:

- cb– dispozitivul a fost blocat (așchiere obișnuită);
- c - așchiere cu vibropercuții;
- primele trei cifre reprezintă turația utilizată la arborele mașinii-unelte;
- următoarele două cifre reprezintă viteza de avans utilizată;
- f – forța axială la găurire;
- detaliu – se prezintă aspectul variației forței axiale pe intervale de timp foarte mici.

Anexa 32.

Diagramele momentului de aşchiere cu dispozitiv blocat şi dispozitiv neblocat

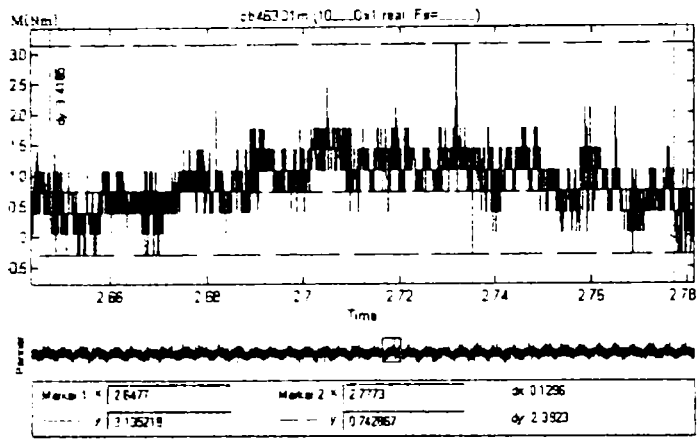


Fig. A32. 1. (cb. 463. 01.m)

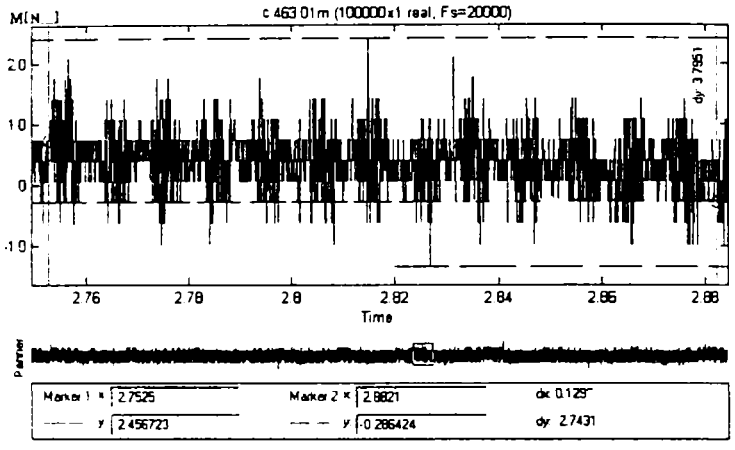


Fig. A32. 2. (c. 463. 01.m)

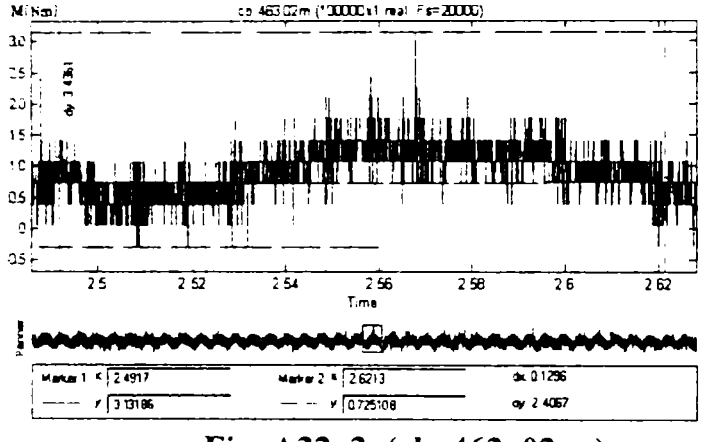


Fig. A32. 3. (cb. 463. 02.m)

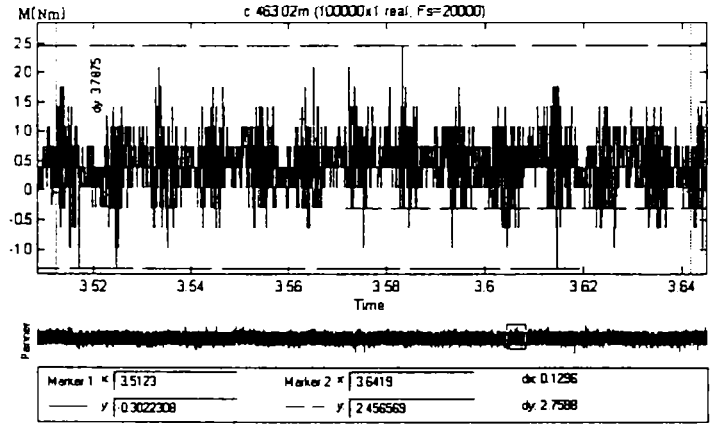


Fig. A32. 4. (c. 463. 02.m)

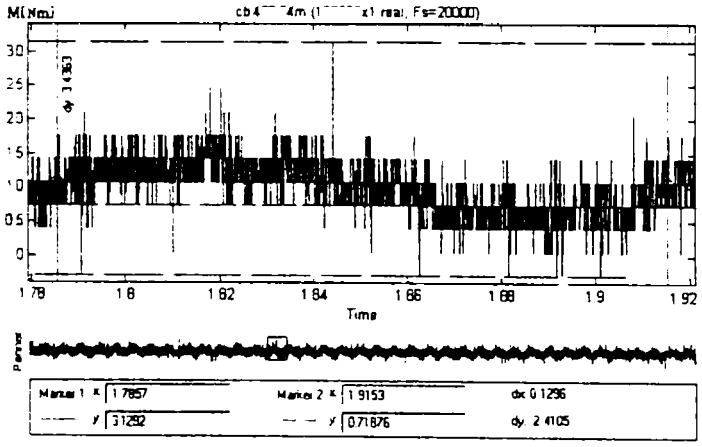


Fig. A32. 5. (cb. 463. 04.m)

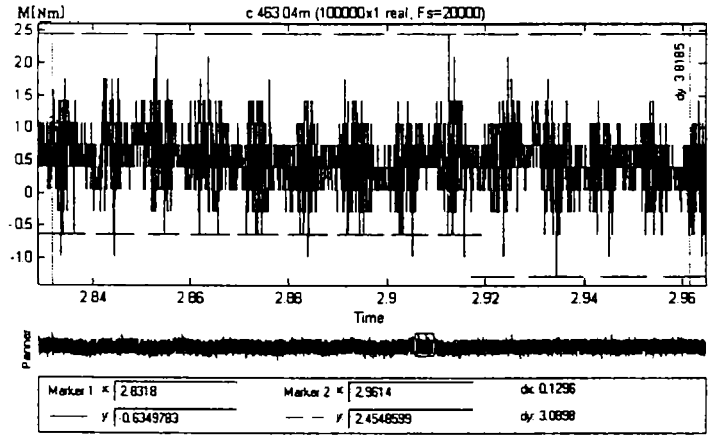


Fig. A32. 6. (c. 463. 04.m)

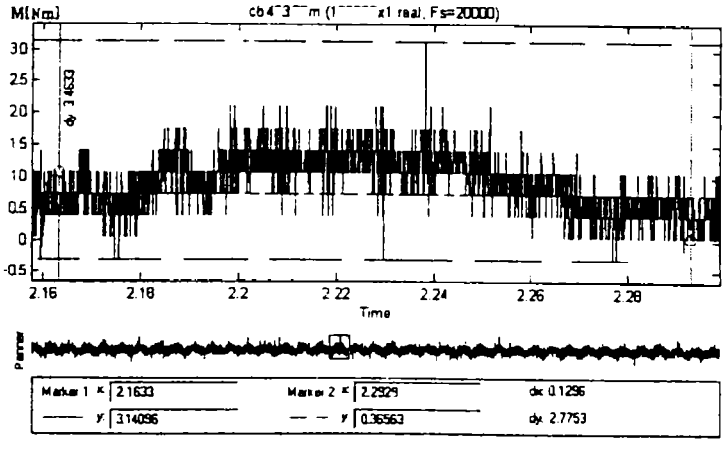


Fig. A32. 7. (cb. 463. 06.m)

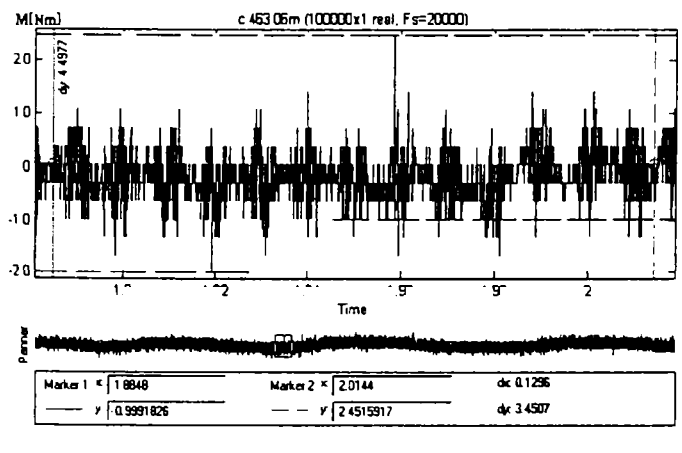


Fig. A32. 8. (c. 463. 06.m)

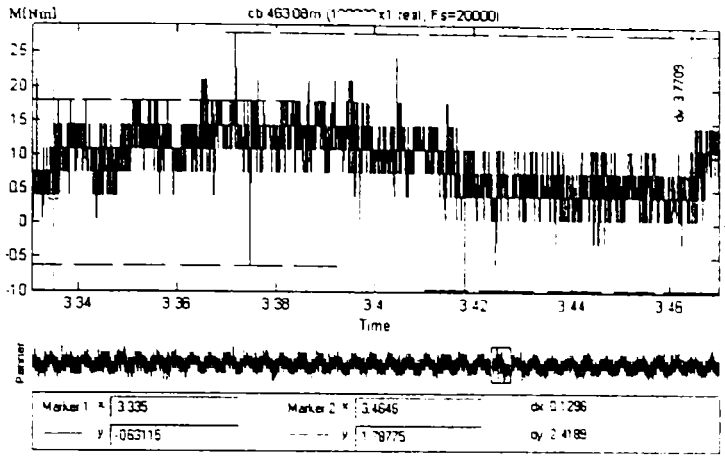


Fig. A32. 9. (cb. 463. 08.m)

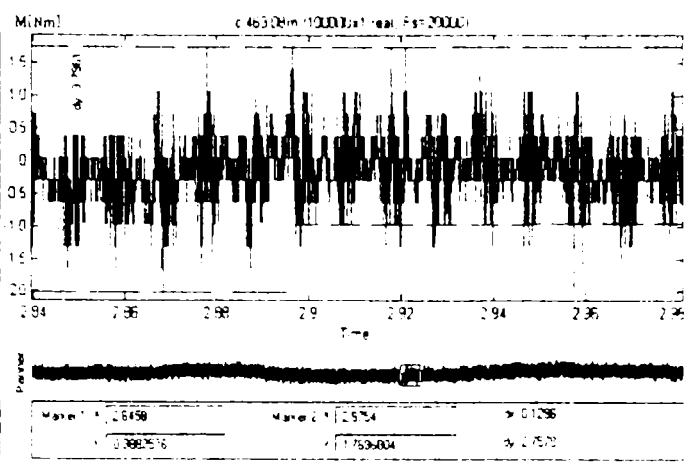


Fig. A32. 10. (c. 463. 08.m)

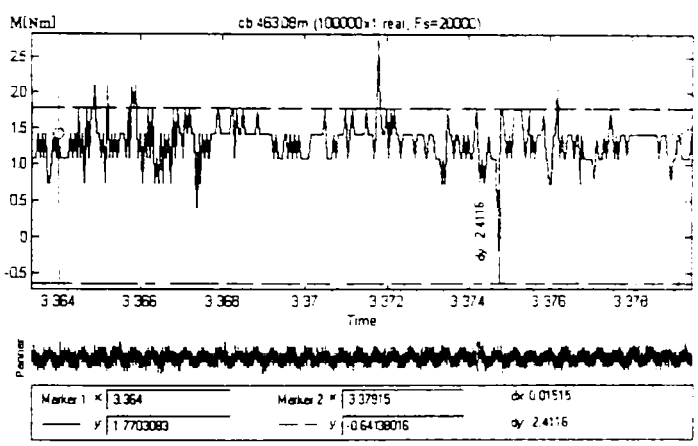


Fig. A32. 11. (cb. 463. 08.m detaliu)

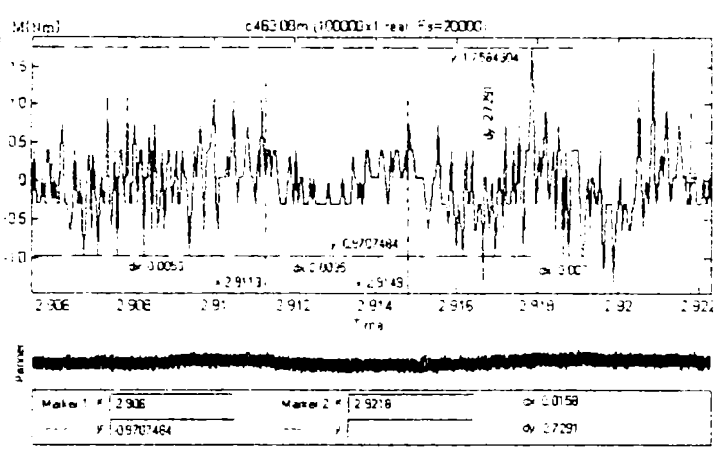


Fig. A32. 12. (c. 463. 01.m detaliu)

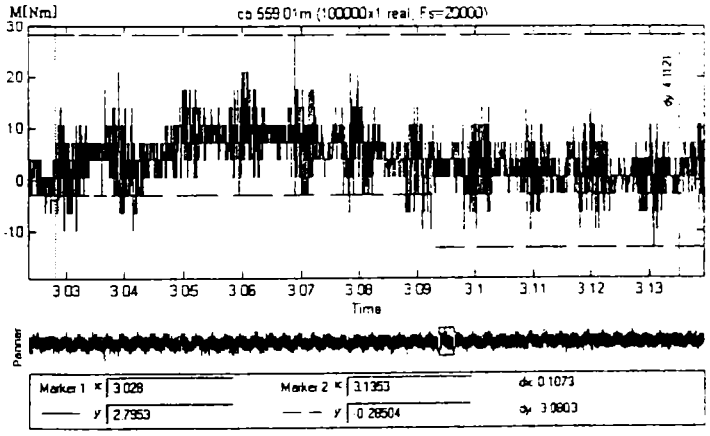


Fig. A32. 13. (cb. 559. 01.m)

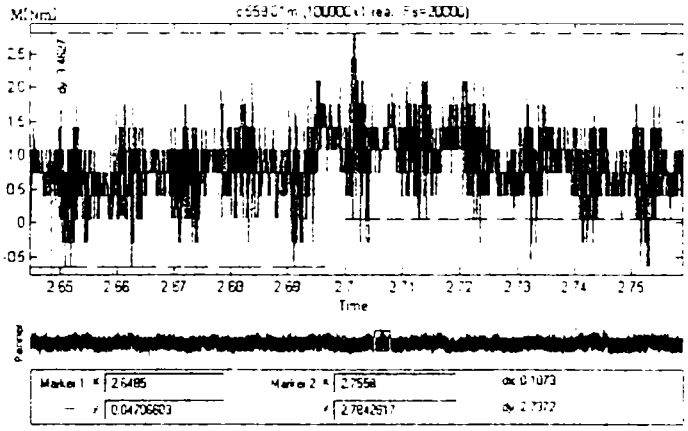


Fig. A32. 14. (c. 559. 01.m)

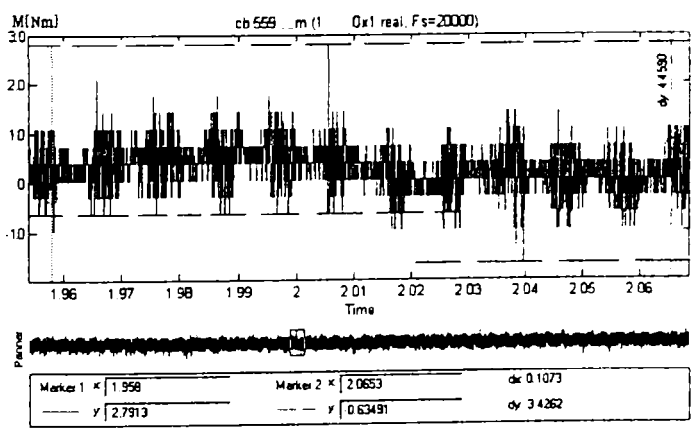


Fig. A32. 15. (cb. 559. 02.m)

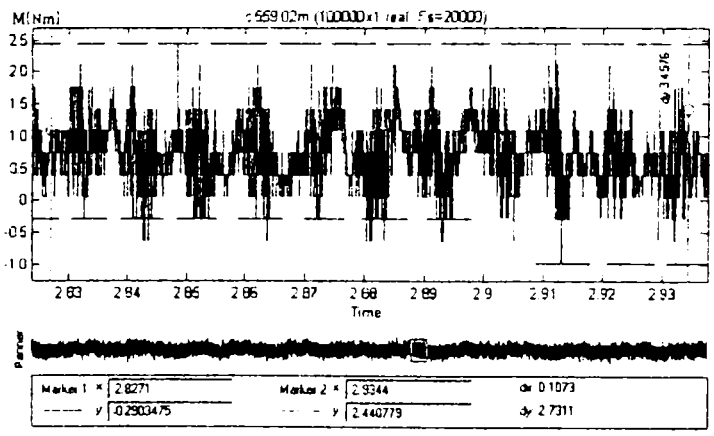


Fig. A32. 16. (c. 559. 02.m)

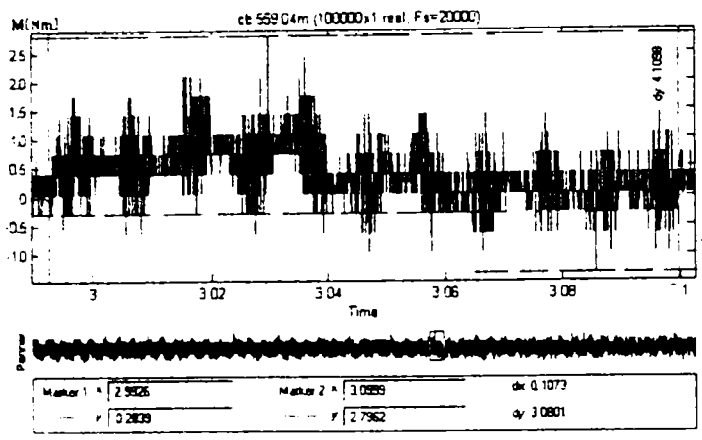


Fig. A32. 17. (cb. 559. 04.m)

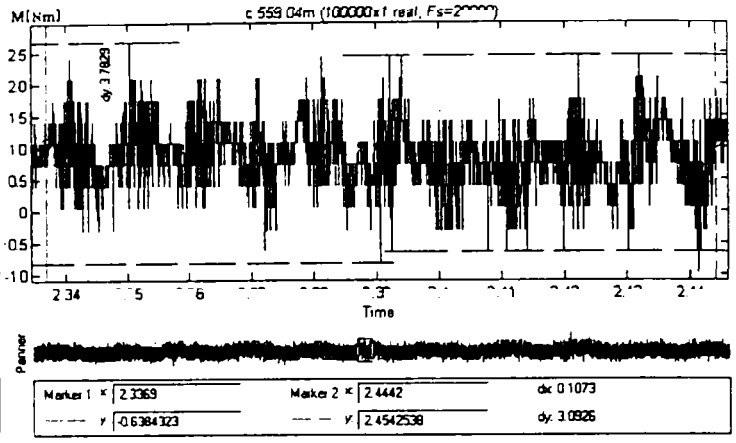


Fig. A32. 18. (c. 559. 04.m)

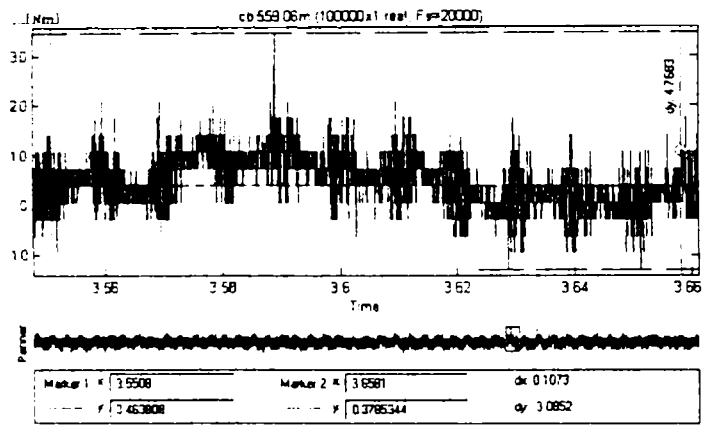


Fig. A32. 19. (cb. 559. 06.m)

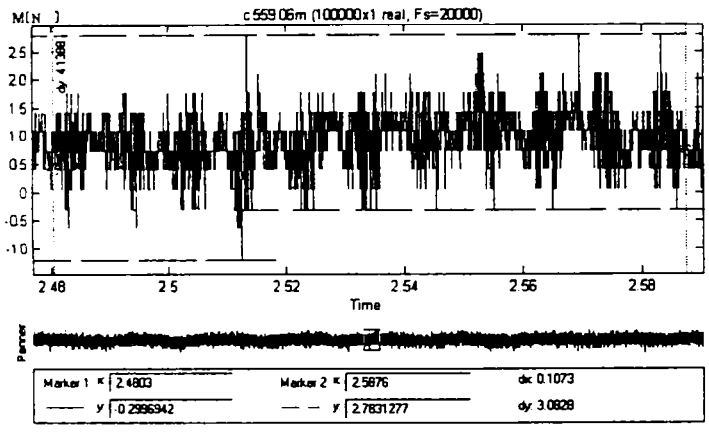


Fig. A32. 20. (c. 559. 06.m)

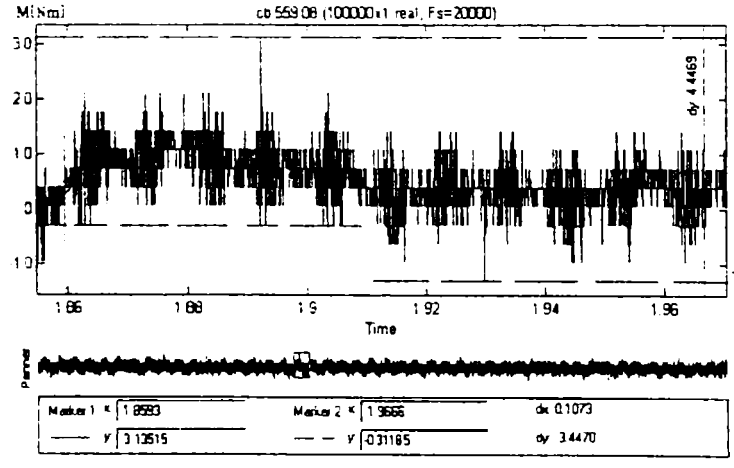


Fig. A32. 21. (cb. 559. 08.m)

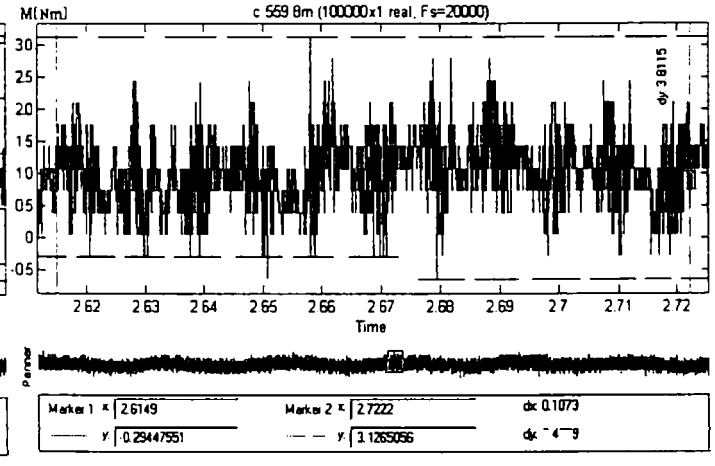


Fig. A32. 22. (c. 559. 08.m)

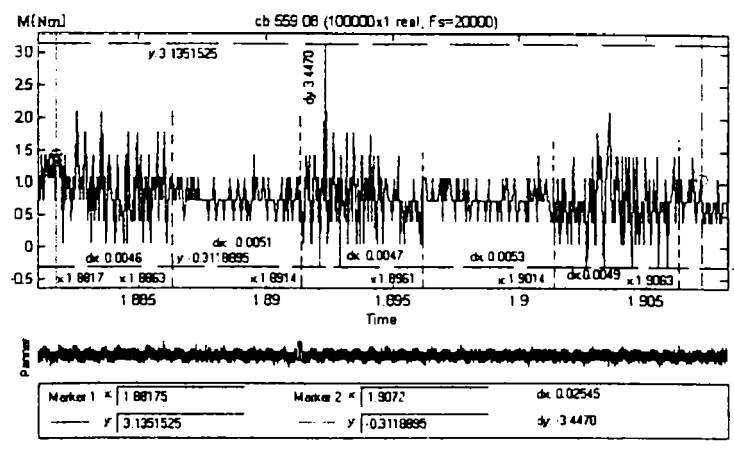


Fig. A32. 23. (cb. 559. 08.m detaliu)

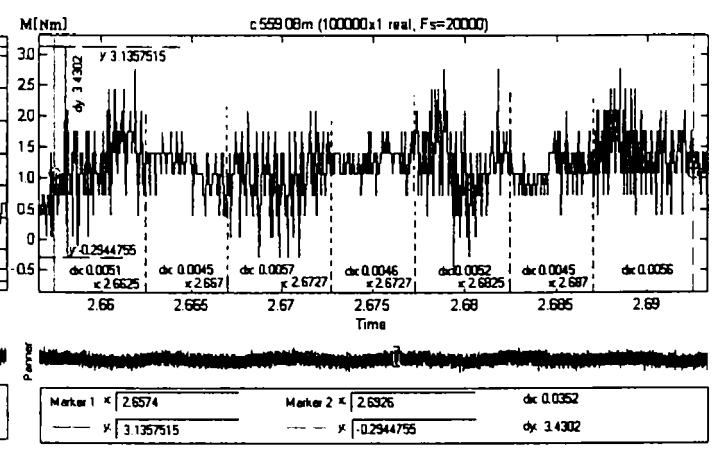


Fig. A32. 24. (c. 559. 08.m detaliu)

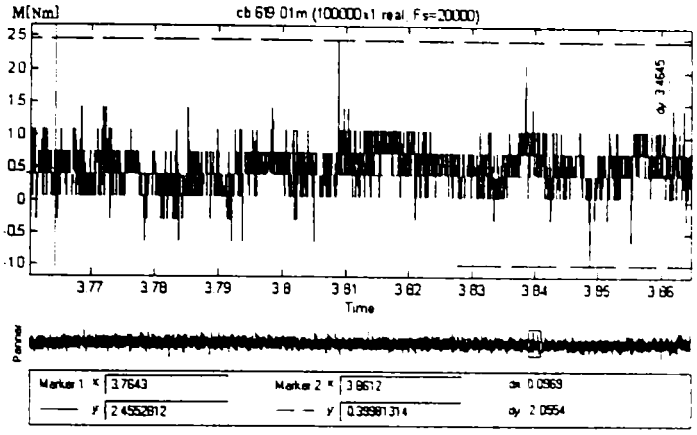


Fig. A32. 25. (cb. 619. 01.m)

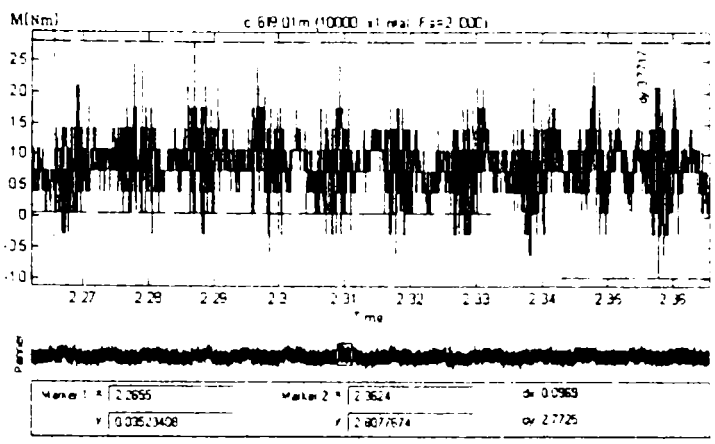


Fig. A32. 26. (c. 619. 01.m)

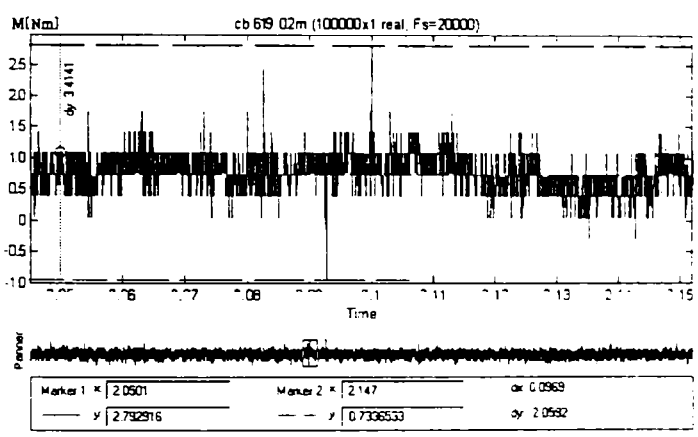


Fig. A32. 27. (cb. 619. 02.m)

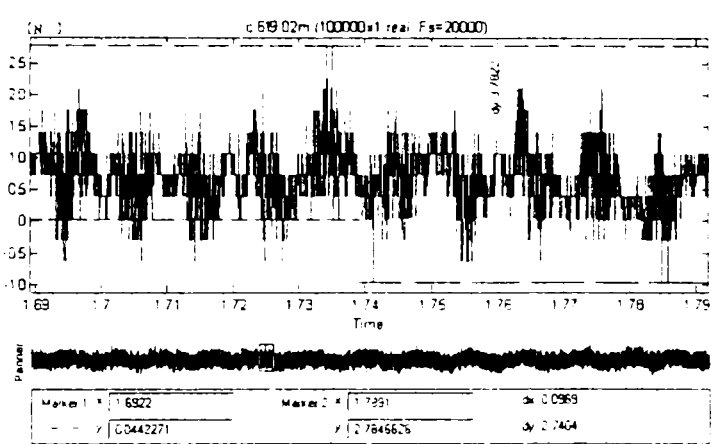


Fig. A32. 28. (c. 619. 02.m)

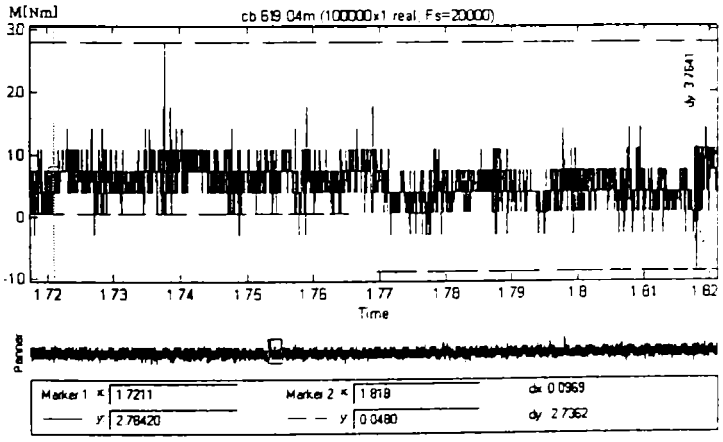


Fig. A32. 29. (cb. 619. 04.m)

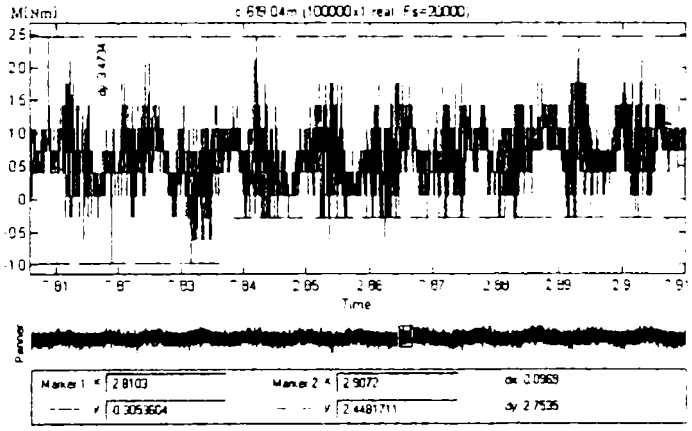


Fig. A32. 30. (c. 619. 04.m)

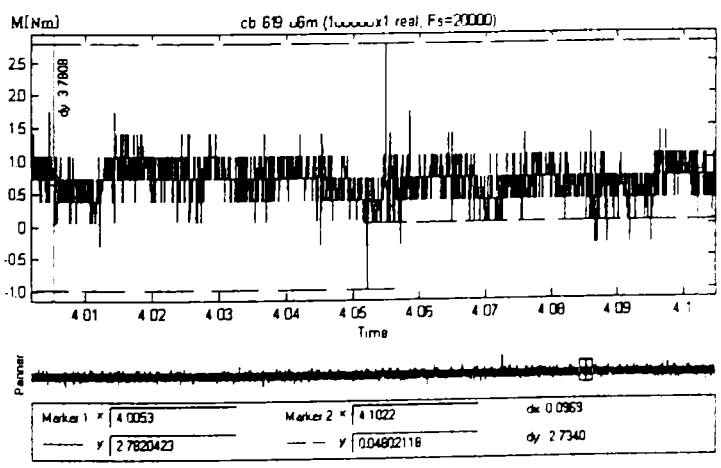


Fig. A32. 31. (cb. 619. 06.m)

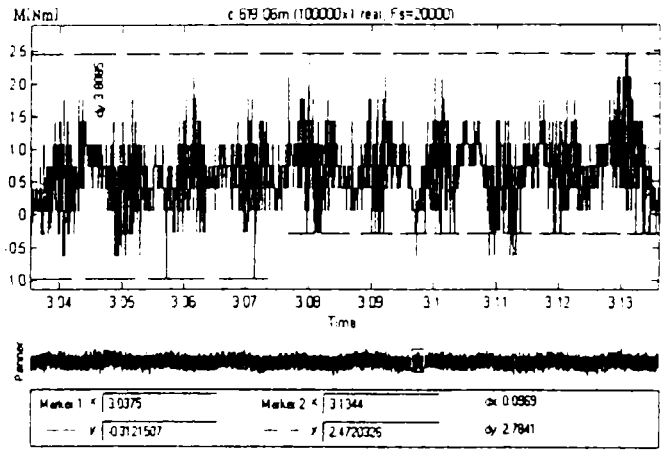


Fig. A32. 32. (c. 619. 06.m)

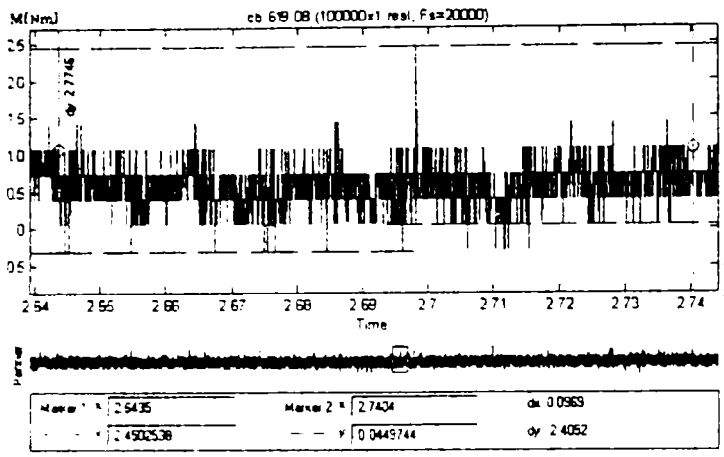


Fig. A32. 33. (cb. 619. 08.m)

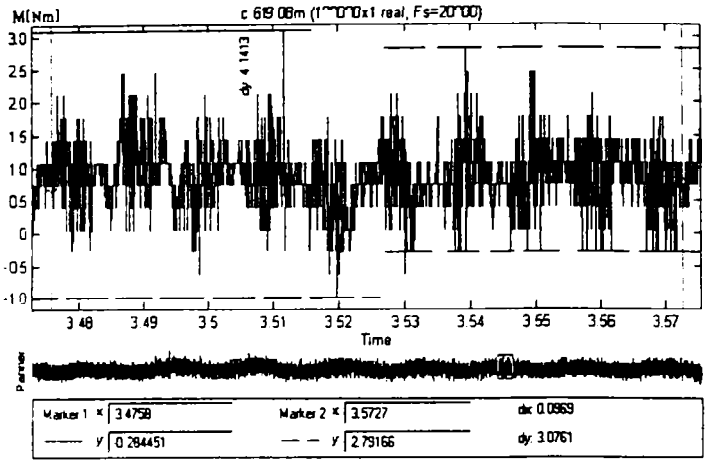


Fig. A32. 34. (c. 619. 08.m)

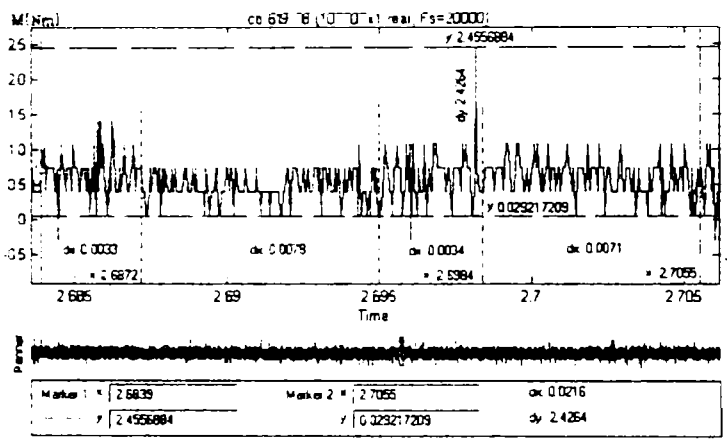


Fig. A32. 35. (cb. 619. 08.m detaliu)

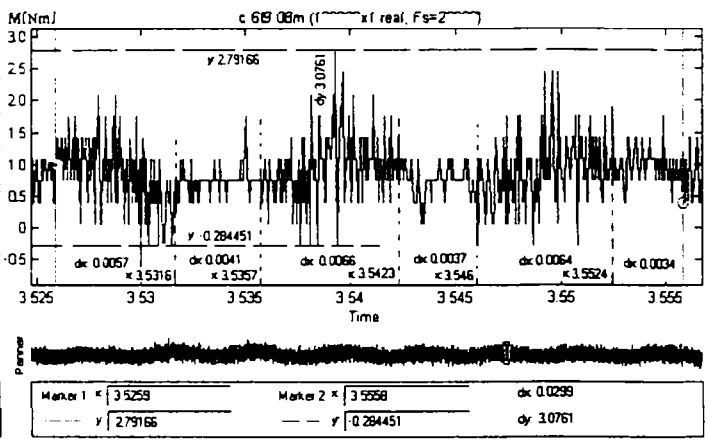


Fig. A32. 36. (c. 619. 08.m detaliu)

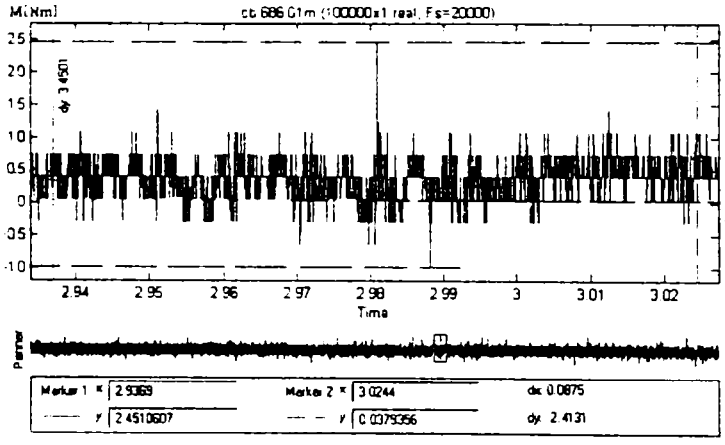


Fig. A32. 37. (cb. 686. 01.m)

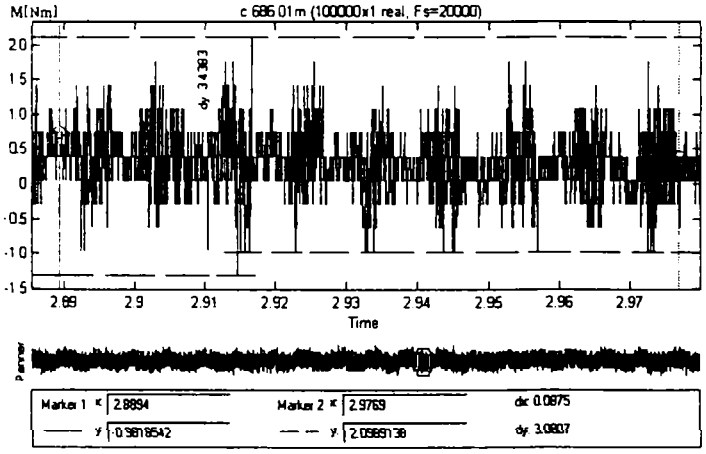


Fig. A32. 38. (c. 686 01.m)

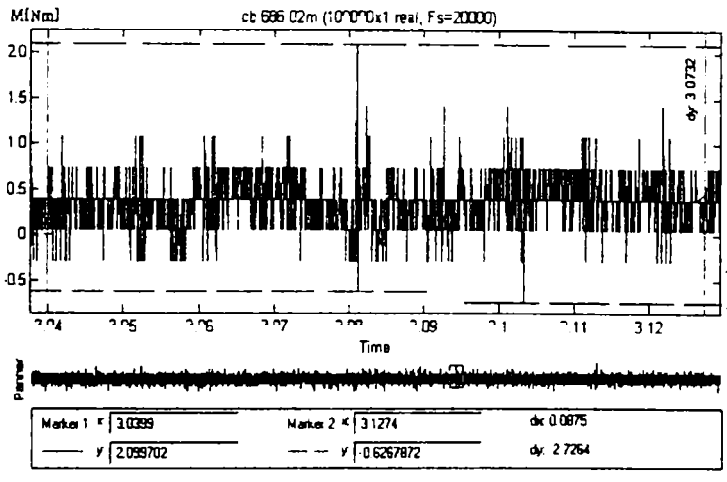


Fig. A32. 39. (cb. 686. 02.m)

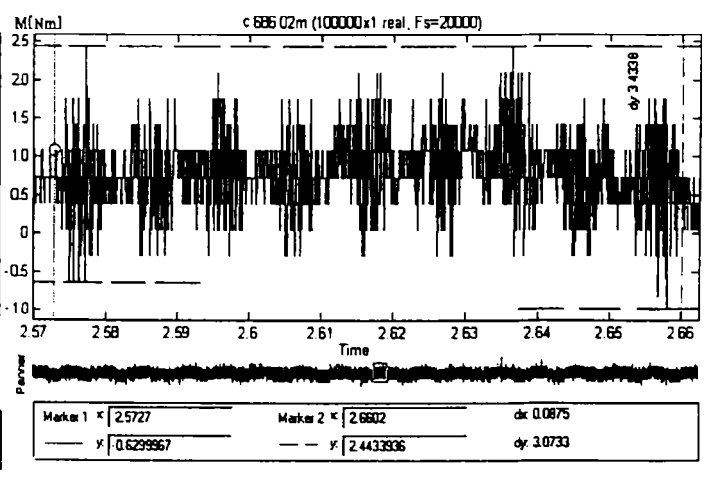


Fig. A32. 40. (c. 686 02.m)

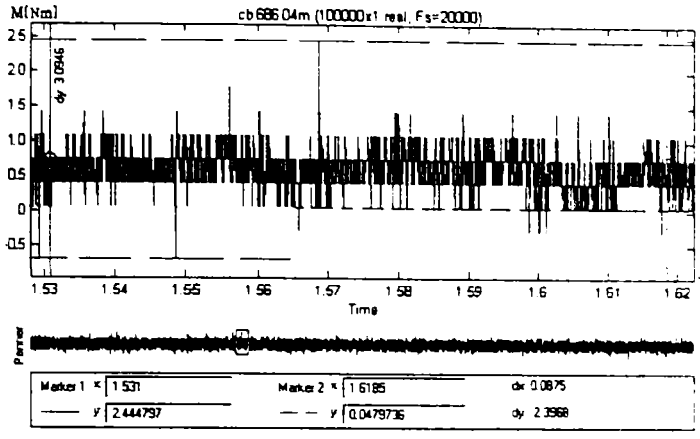


Fig. A32. 41. (cb. 686. 04.m)

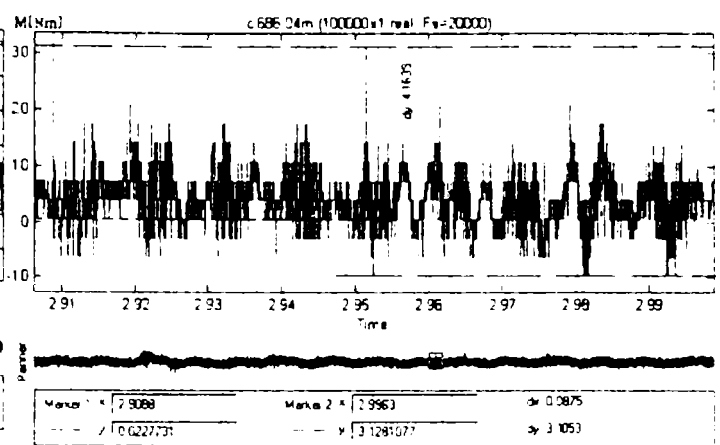


Fig. A32. 42. (c. 686 04.m)

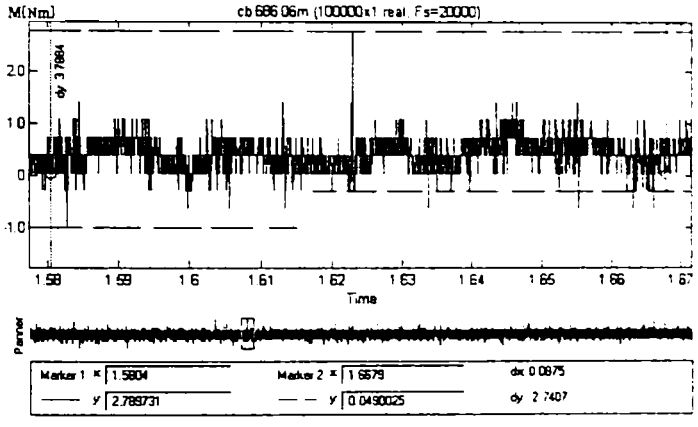


Fig. A32. 43. (cb. 686. 06.m)

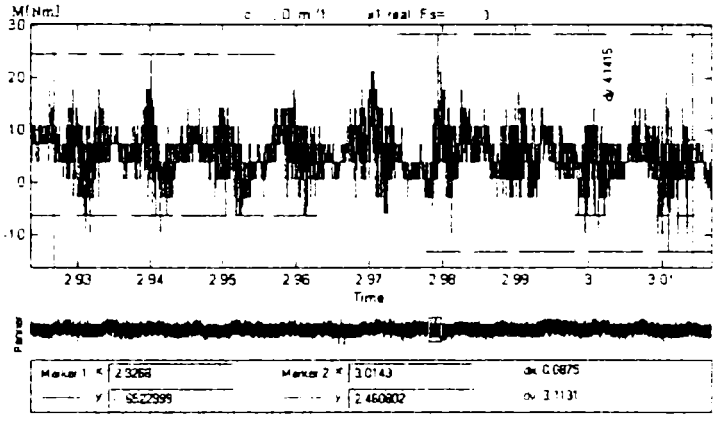


Fig. A32. 44. (c. 686 06.m)

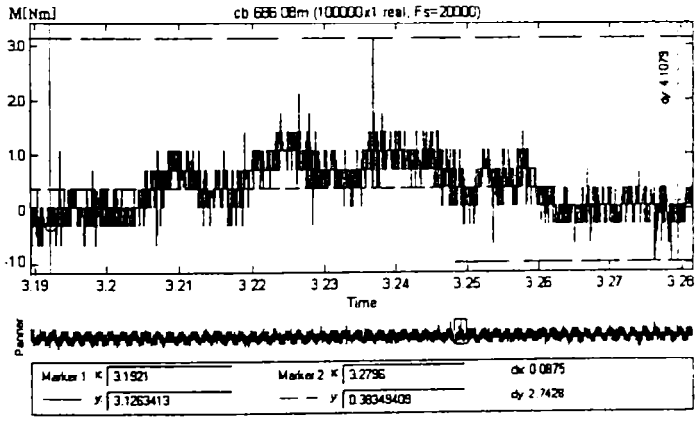


Fig. A32. 45. (cb. 686. 08.m)

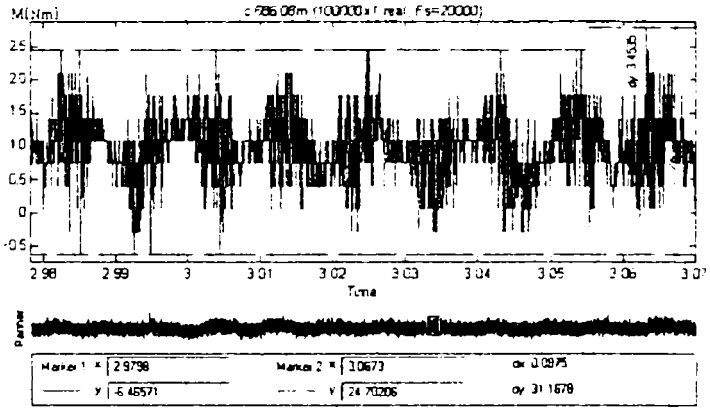


Fig. A32. 46. (c. 686 08.m)

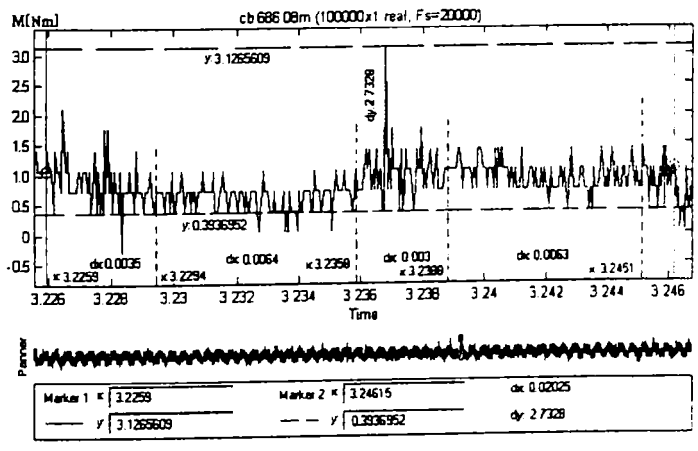


Fig. A32. 47. (cb. 686. 08.m detaliu)

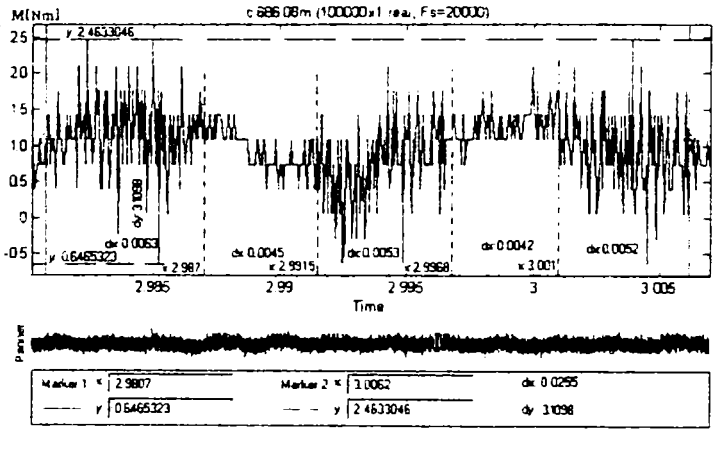


Fig. A32. 48. (c. 686 08.m detaliu)

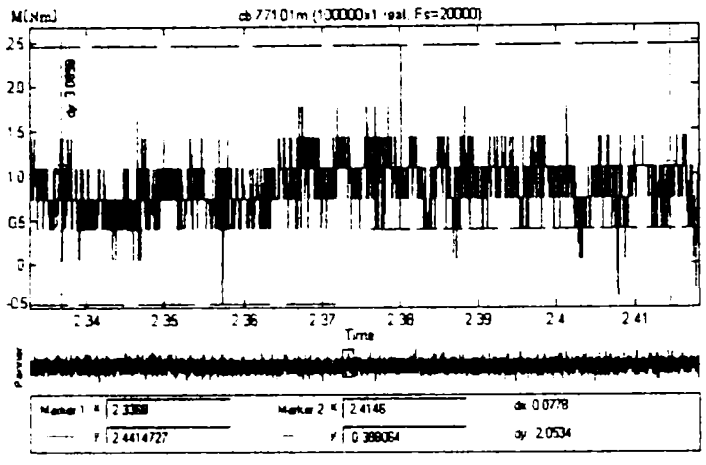


Fig. A32. 49. (cb. 771. 01.m)

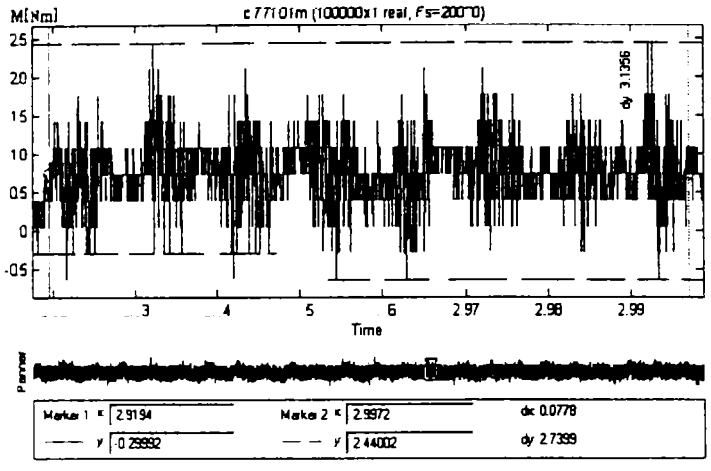


Fig. A32. 50. (c. 771 01.m)

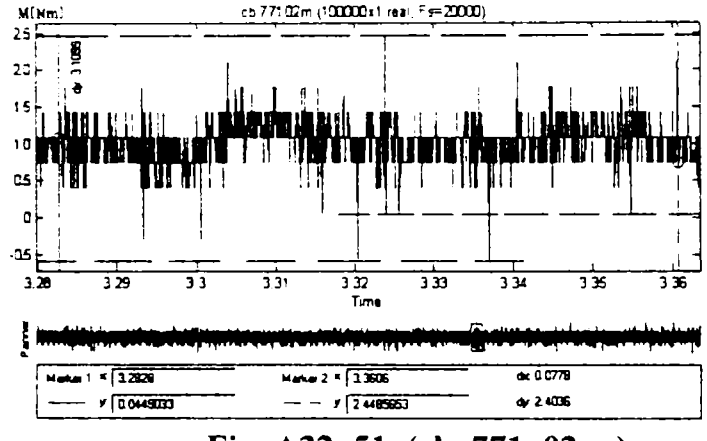


Fig. A32. 51. (cb. 771. 02.m)

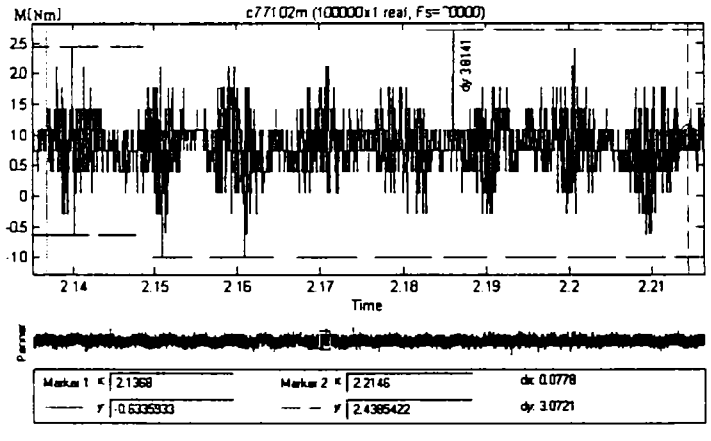


Fig. A32. 52. (c. 771 02.m)

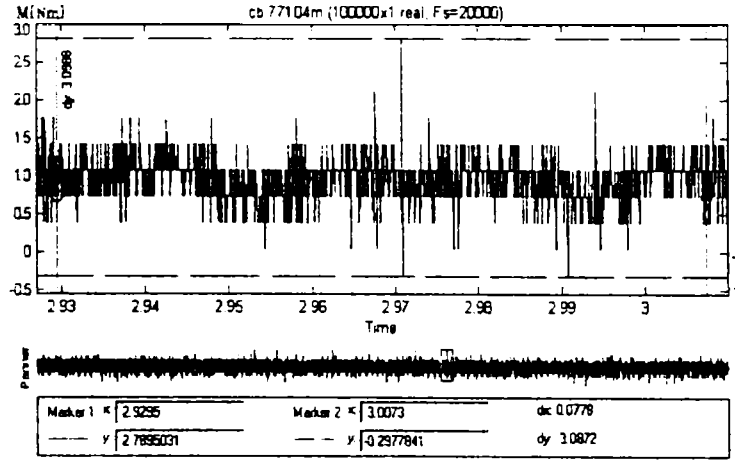


Fig. A32. 53. (cb. 771. 04.m)

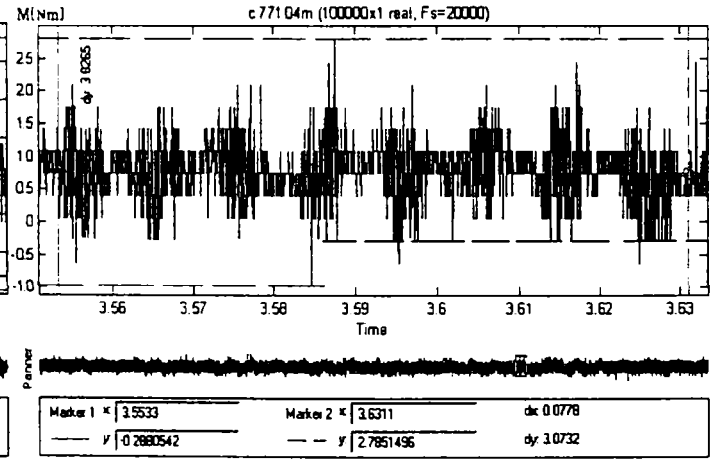


Fig. A32. 54. (c. 771 04.m)

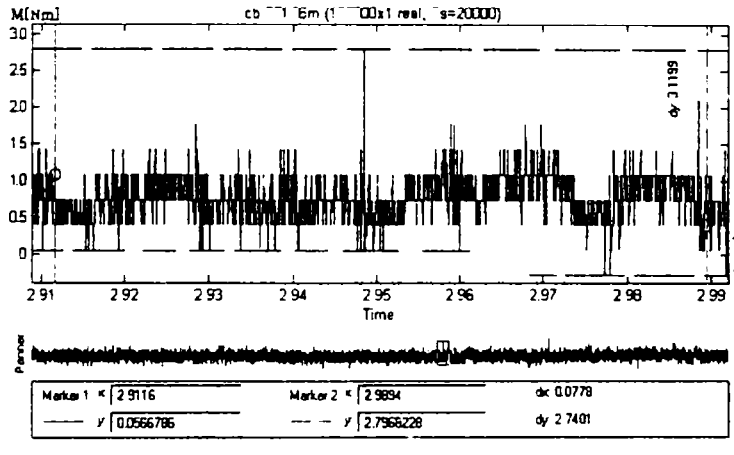


Fig. A32. 55. (cb. 771. 06.m)

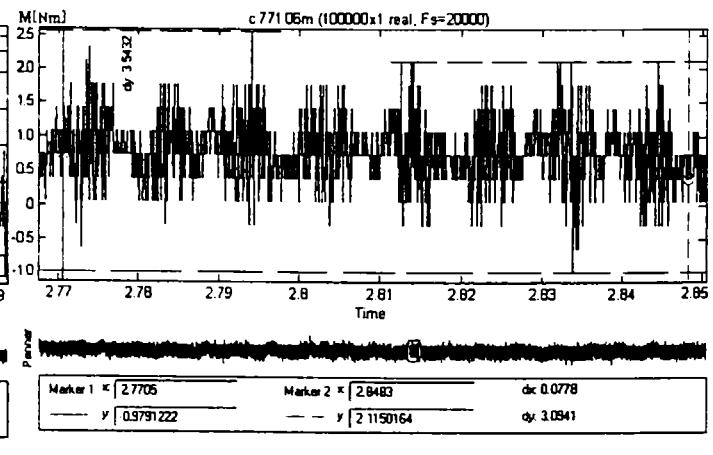


Fig. A32. 56. (c. 771 06.m)

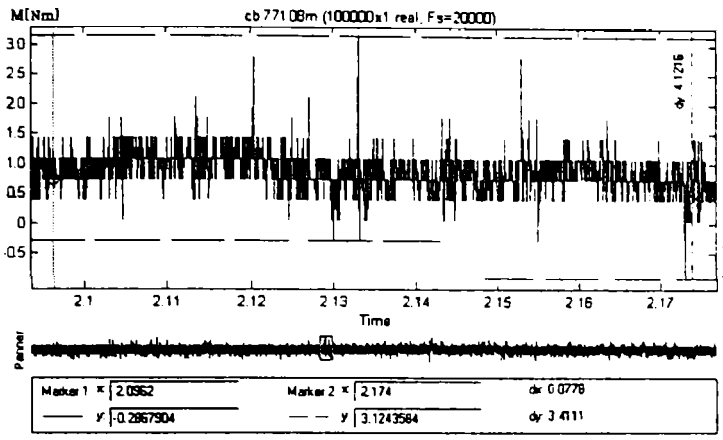


Fig. A32. 57. (cb. 771. 08.m)

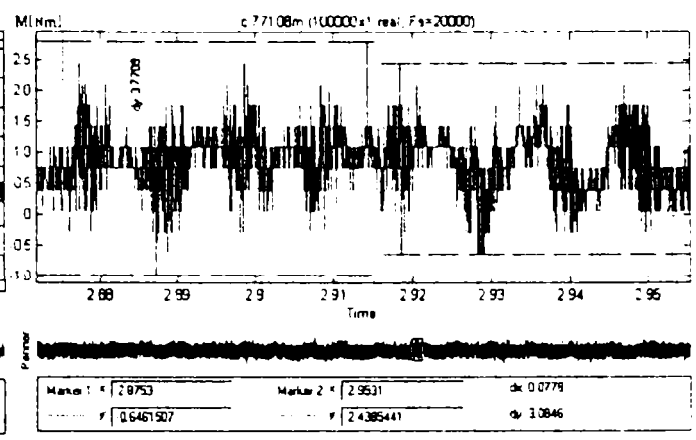


Fig. A32. 58. (c. 771 08.m)

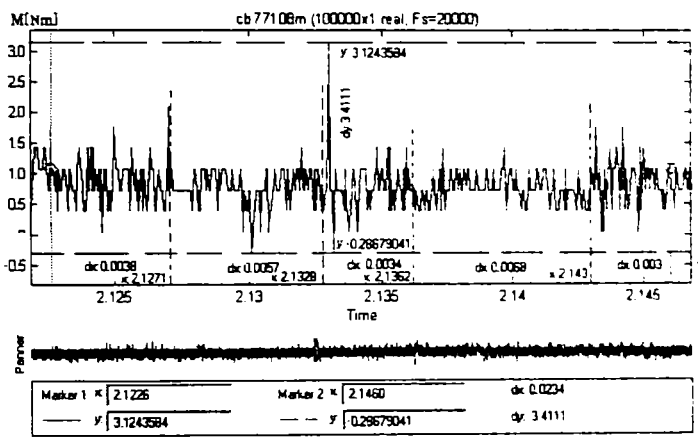


Fig. A32. 59. (cb. 771. 08.m detaliu)

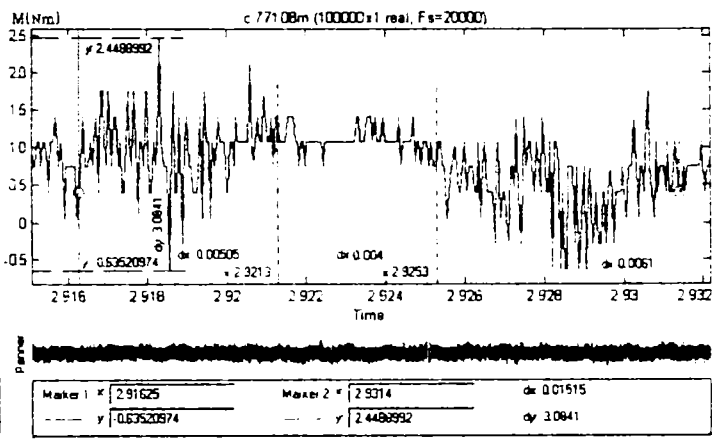


Fig. A32. 60. (c. 771 08.m detaliu)

Notă.

Semnificația notațiilor din paranteză:

- cb– dispozitivul pentru așchierea cu vibropercuții a fost blocat (așchiere obișnuită);
- c - așchiere cu vibropercuții;
- primele trei cifre reprezintă turația utilizată la arborele mașinii-unelte;
- următoarele două cifre reprezintă avansul utilizat;
- m –moment.

Anexa 33. A

Diagramele variațiilor vitezelor unghiulare ale burghiului la așchierea vibropercutantă

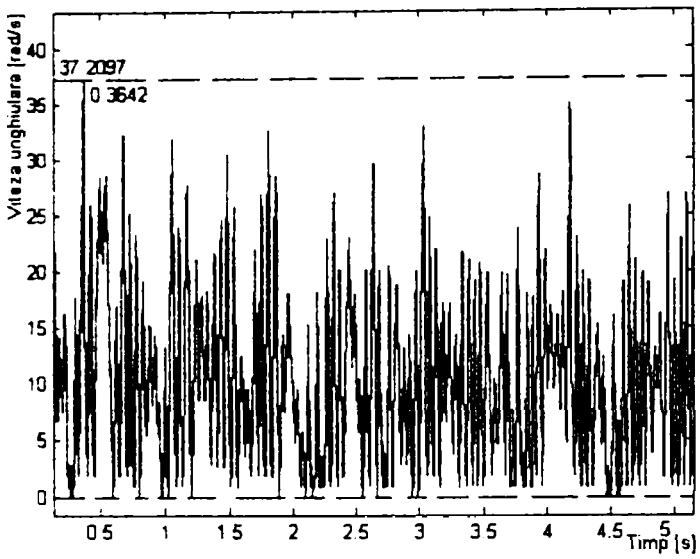


Fig. A.33.A. 1. (c. 463.01)

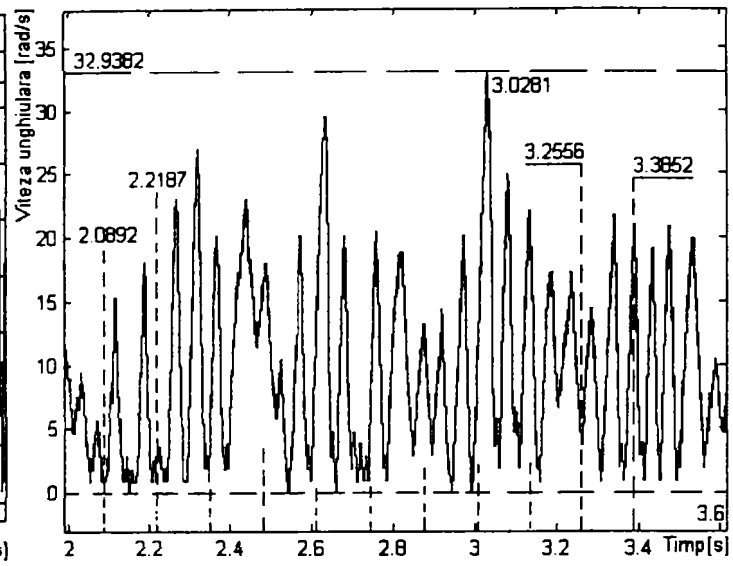


Fig. A.33.A. 2. (c. 463.01.det x 10turații)

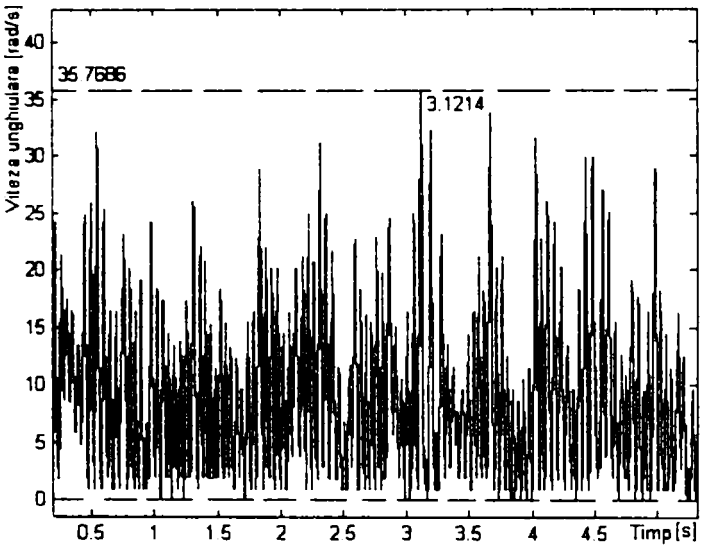


Fig. A.33.A. 3. (c. 463.02)

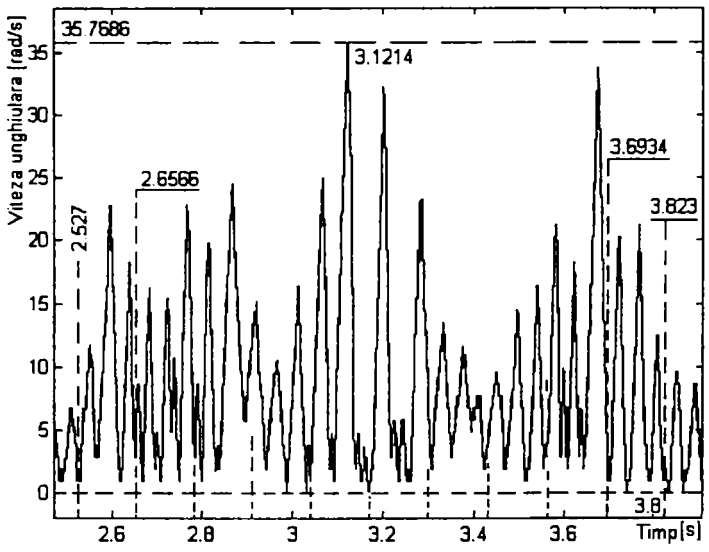


Fig. A.33.A. 4. (c. 463.02.det x 10turații)

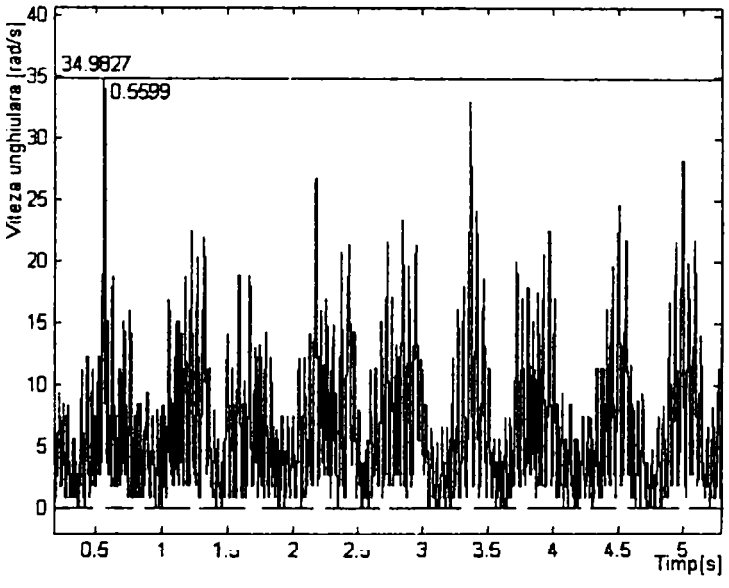


Fig. A.33.A. 5. (c. 463.04)

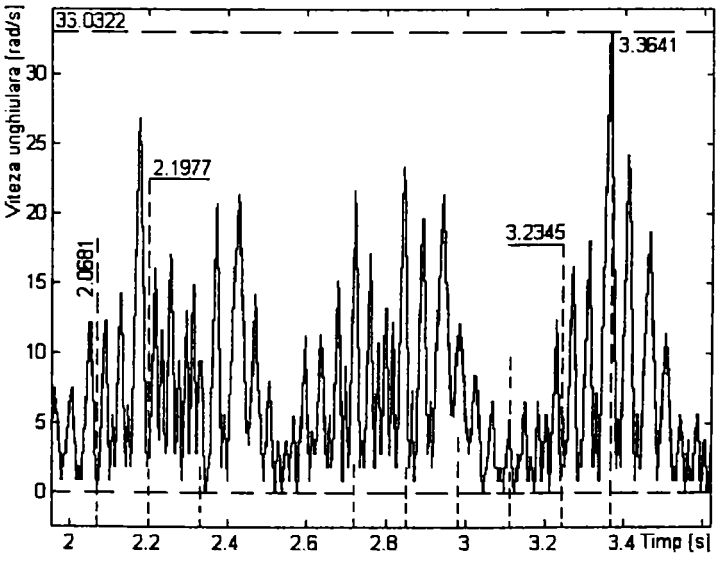


Fig. A.33.A. 6. (c. 463.04.det x 10turații)

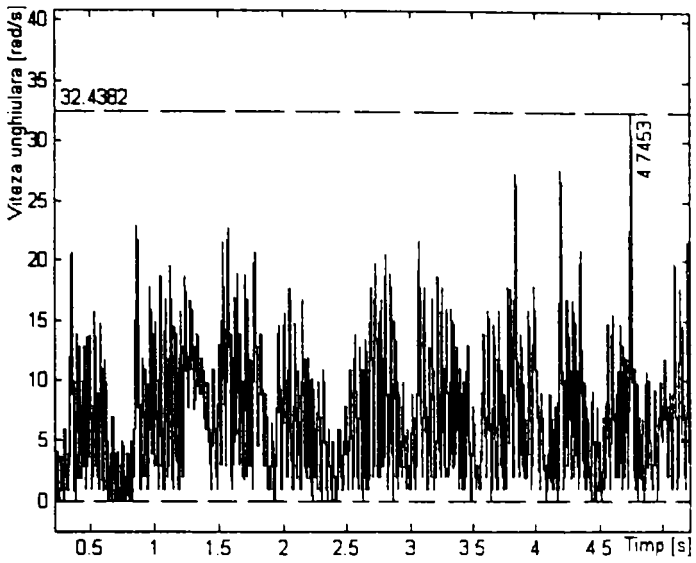


Fig. A.33.A. 7. (c. 463.06)

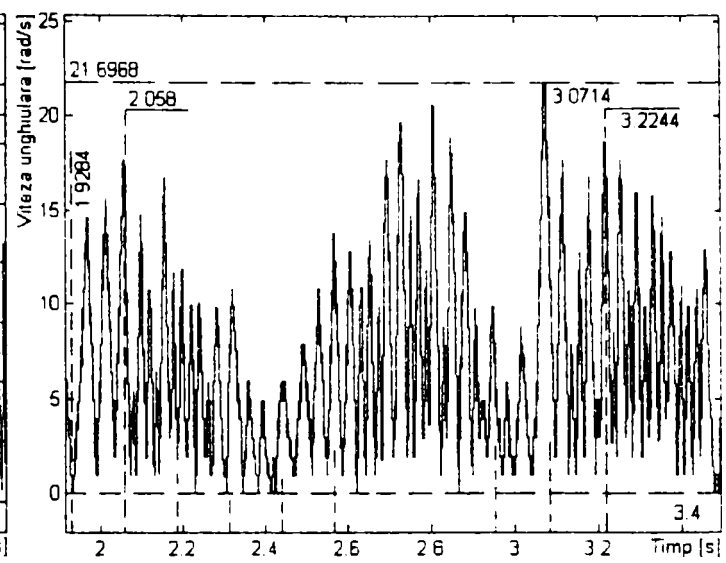


Fig. A.33.A. 8. (c. 463.06.det x 10turații)

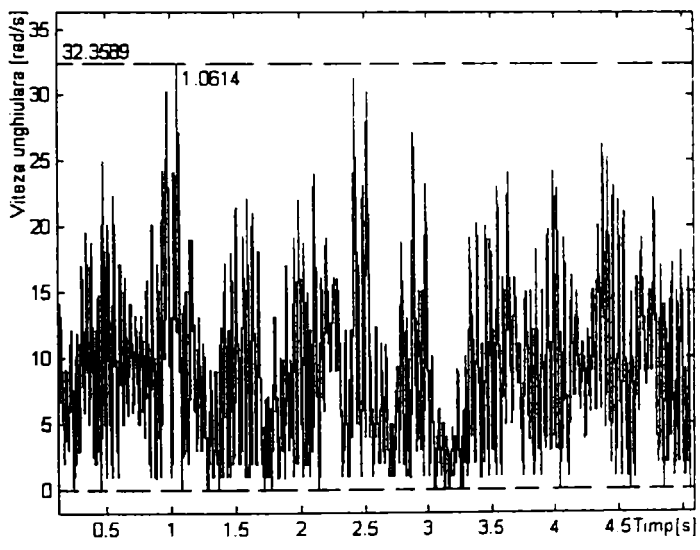


Fig. A.33.A. 9. (c. 463.08)

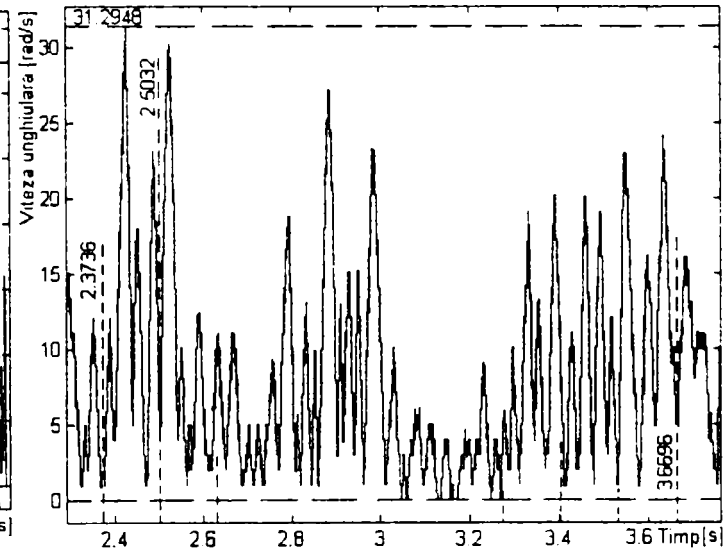


Fig. A.33.A. 10. (c. 463.08.det x 10turații)

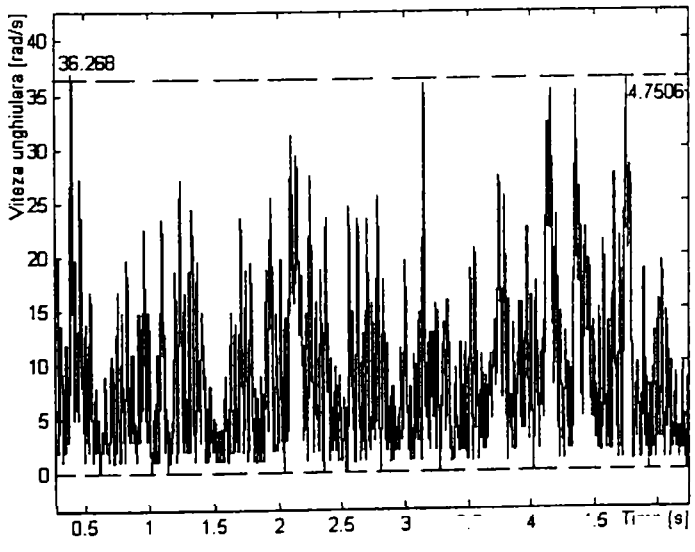


Fig. A.33.A. 11. (c. 559.01)

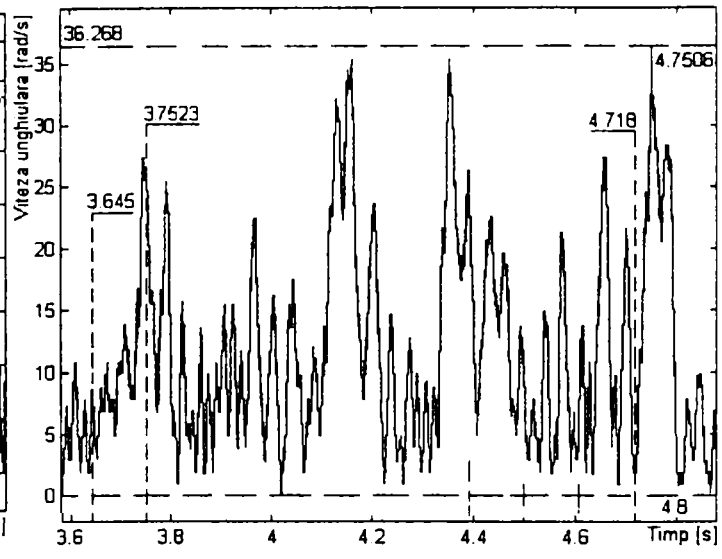


Fig. A.33.A. 12. (c. 559.01.det x 10turații)

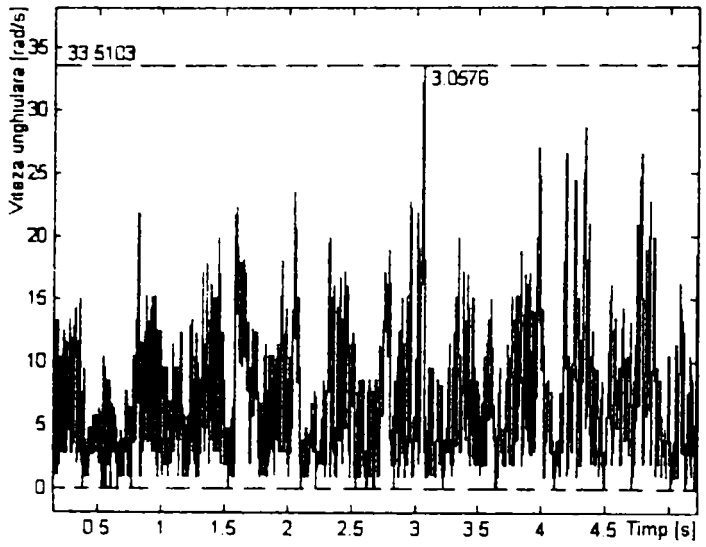


Fig. A.33.A. 13. (c. 559.02)

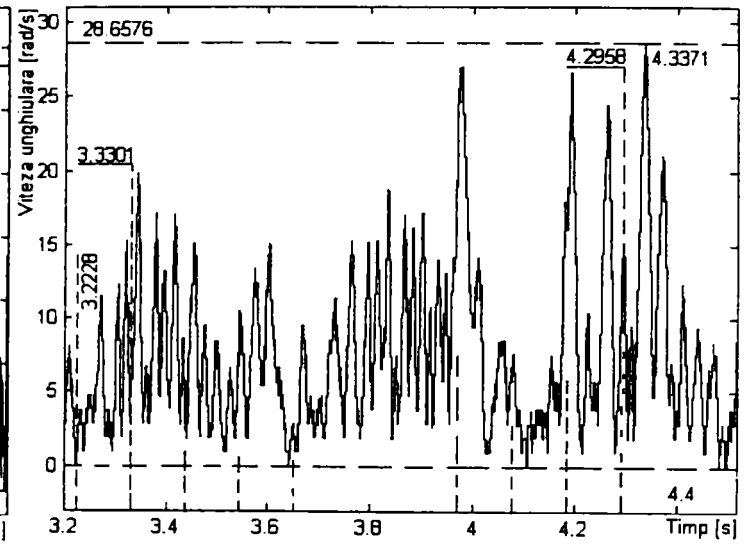


Fig. A.33.A. 14. (c. 559.02.det x 10turații)

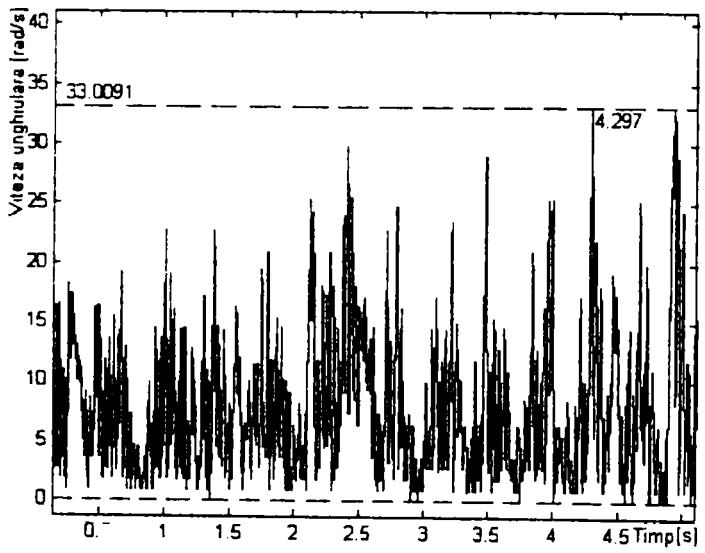


Fig. A.33.A. 15. (c. 559.04)

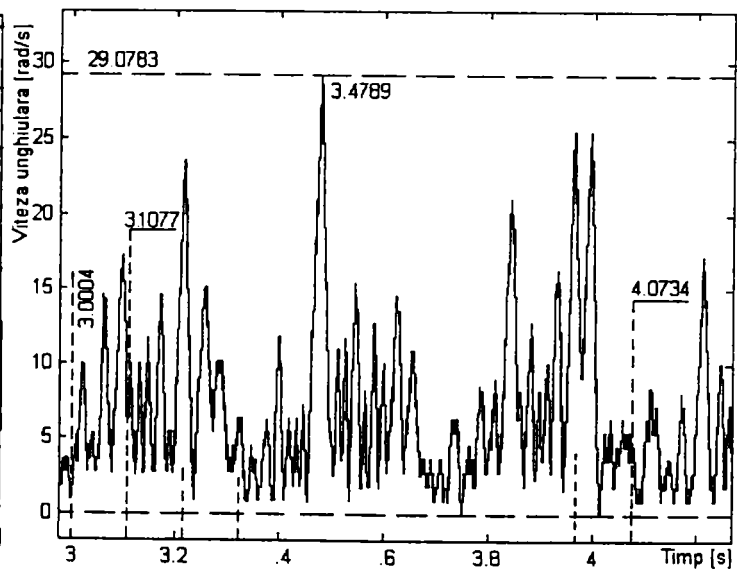


Fig. A.33.A. 16. (c. 559.04.det x 10turații)

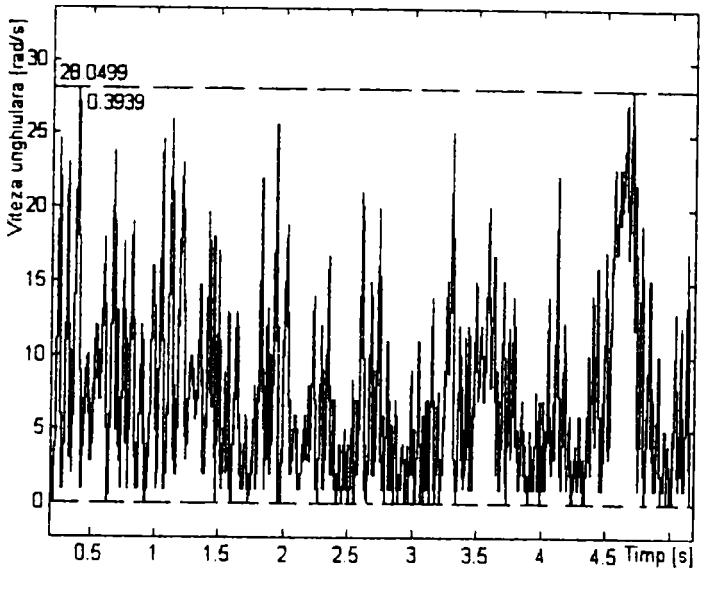


Fig. A.33.A. 17. (c. 559.06)

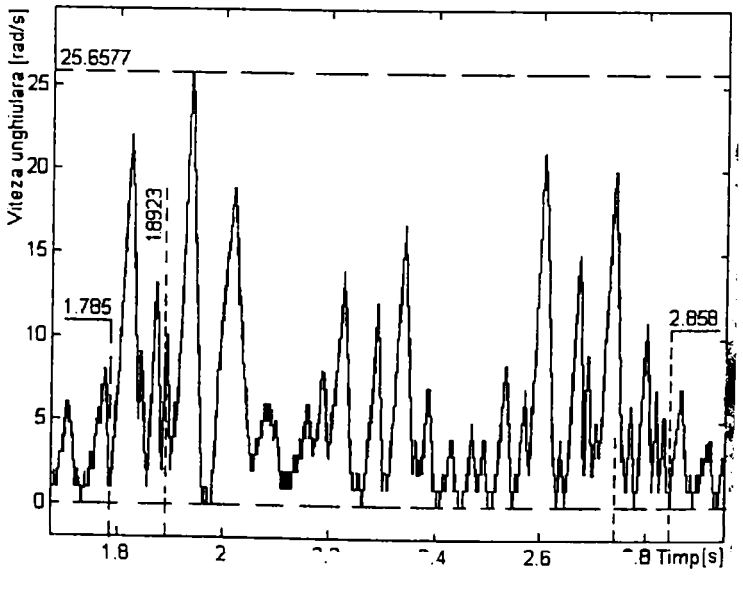


Fig. A.33.A. 18. (c. 559.06.det x 10turații)

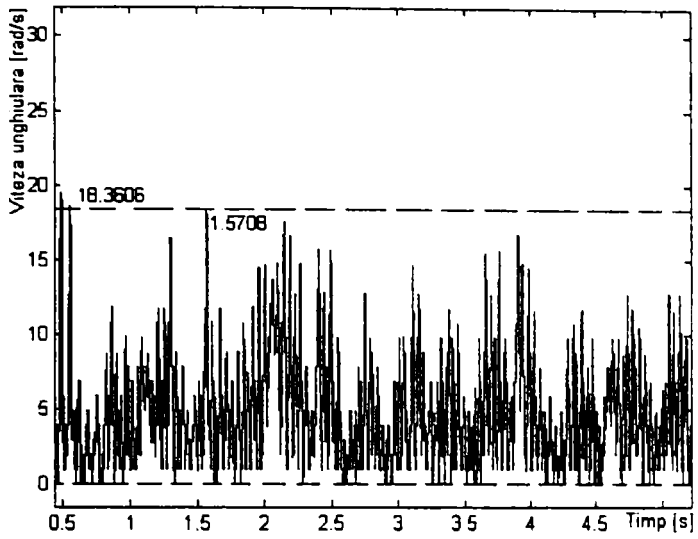


Fig. A.33.A. 19. (c. 559.08)

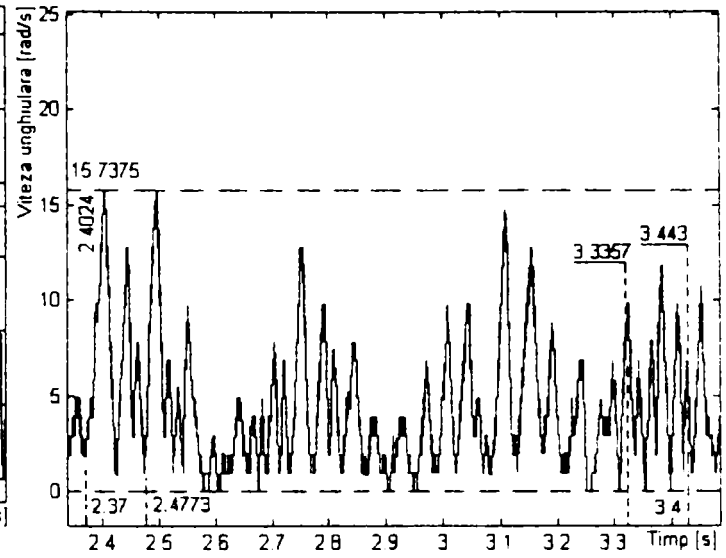


Fig. A.33.A. 20. (c. 559.08.det x 10turații)

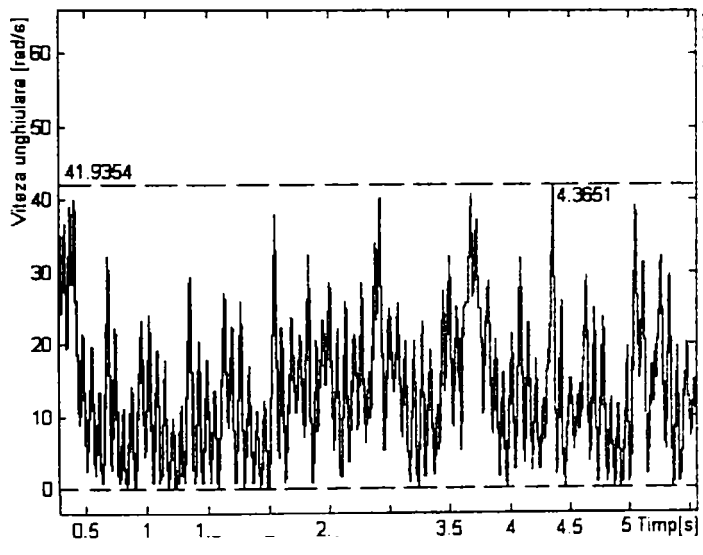


Fig. A.33.A. 21. (c. 619.01)

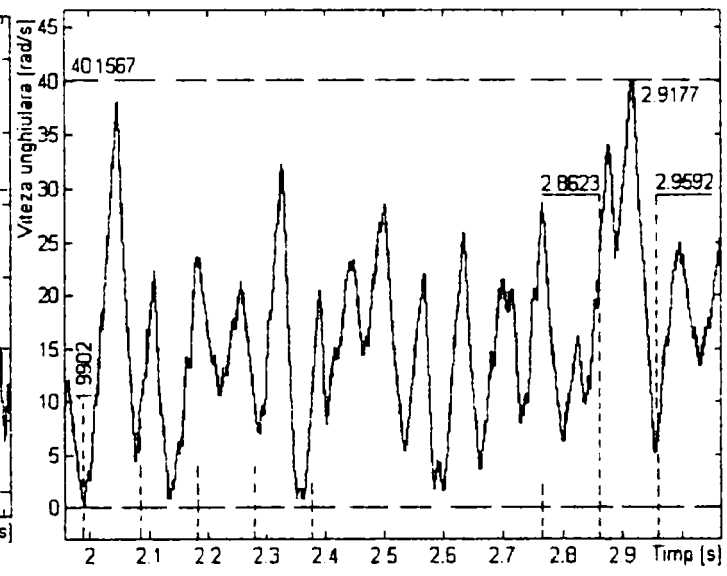


Fig. A.33.A. 22. (c. 619.01.det x 10turații)

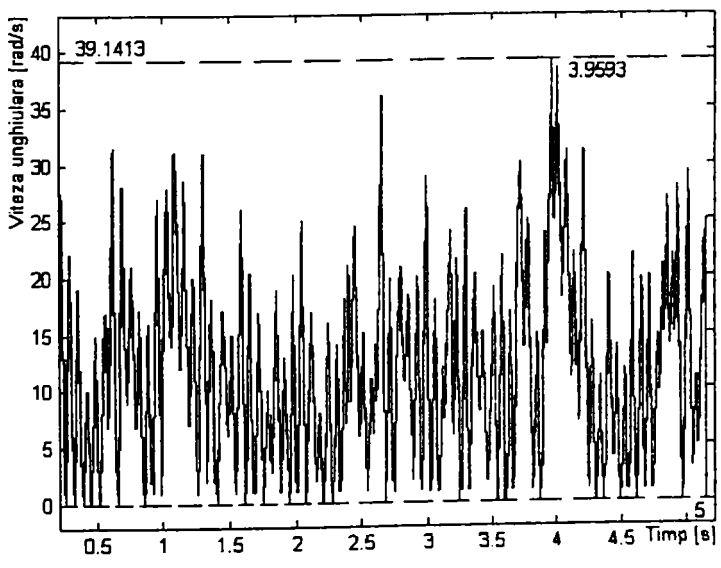


Fig. A.33.A. 23. (c. 619.02)

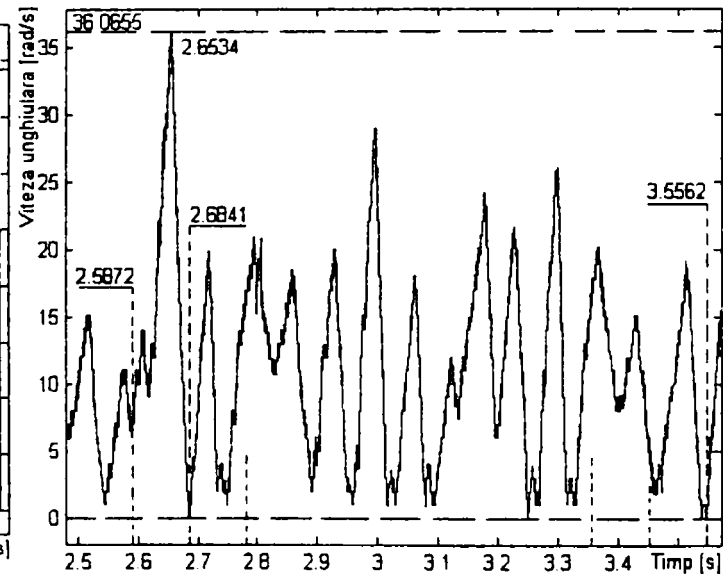


Fig. A.33.A. 24. (c. 619.02.det x 10turații)

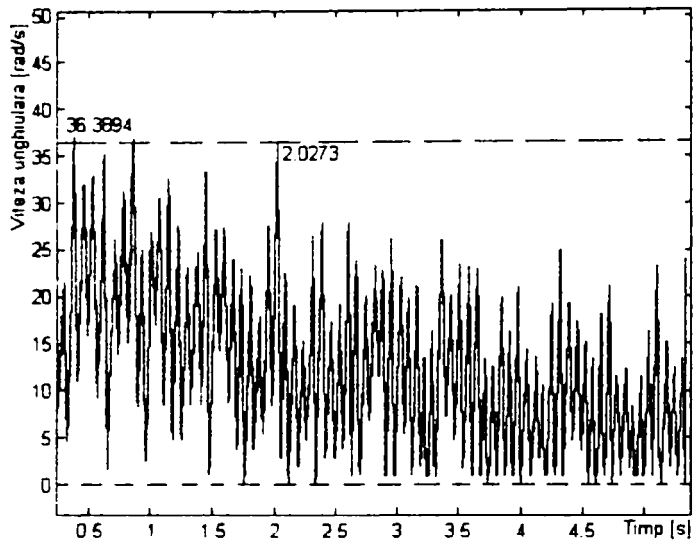


Fig. A.33.A. 25. (c. 619.04)

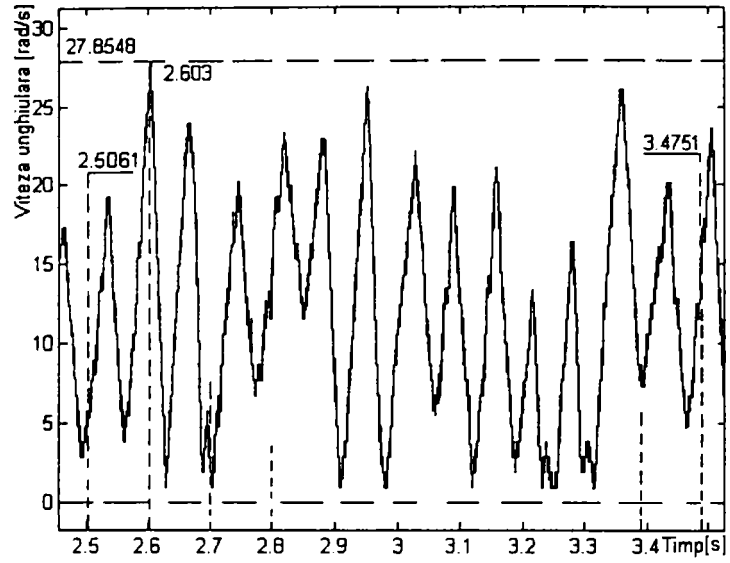


Fig. A.33.A. 26. (c. 619.04.det x 10turații)

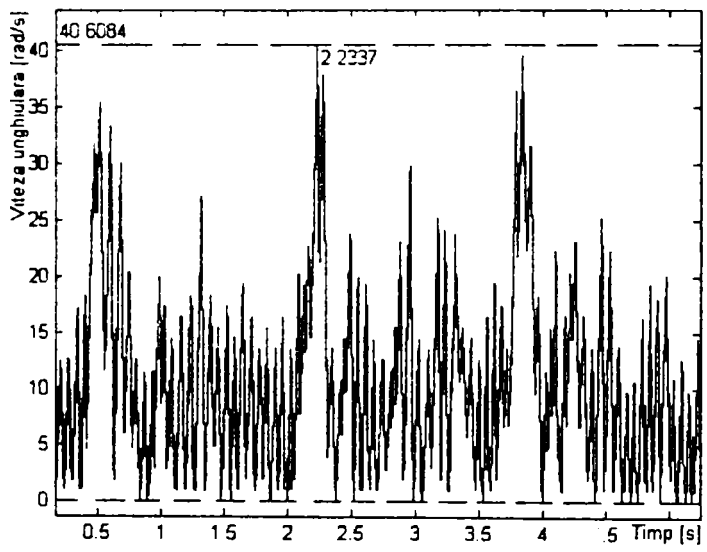


Fig. A.33.A. 27. (c. 619.06)

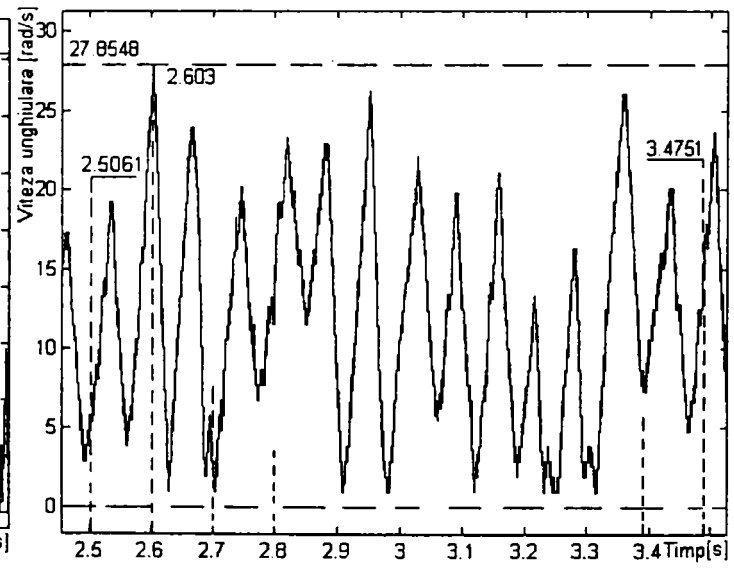


Fig. A.33.A. 28. (c. 619.06.det x 10turații)

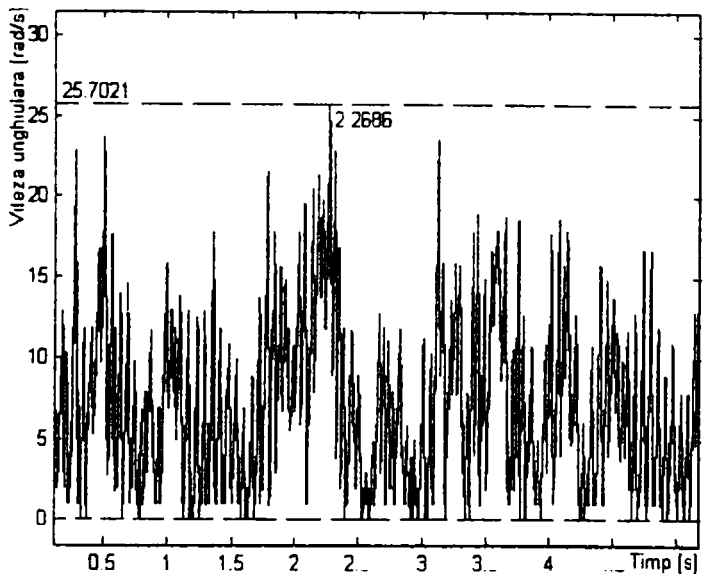


Fig. A.33.A. 29. (c. 619.08)

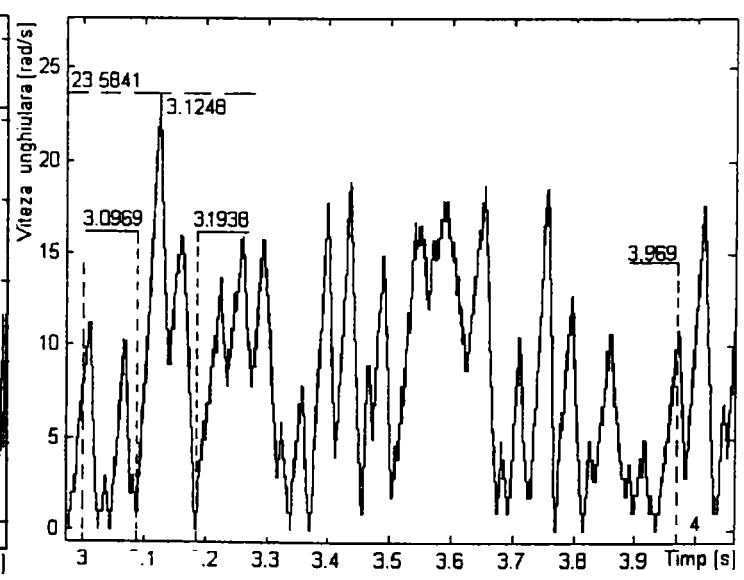


Fig. A.33.A. 30. (c. 619.08.det x 10turații)

Notă: În notarea figurilor primele trei cifre din paranteze reprezintă turația arborelui mașinii-unelte, iar următoarele două avansul de lucru, iar „det x 10turații"-reprezintă un detaliu din diagrama vitezelor unghiulare ale burghiului corespunzător la zece turații ale arborelui mașinii-unelte.

Anexa 33. B

Diagrame ale vitezelor unghiulare la așchierea cu dispozitivul vibropercutant blocat (așchierea obișnuită)

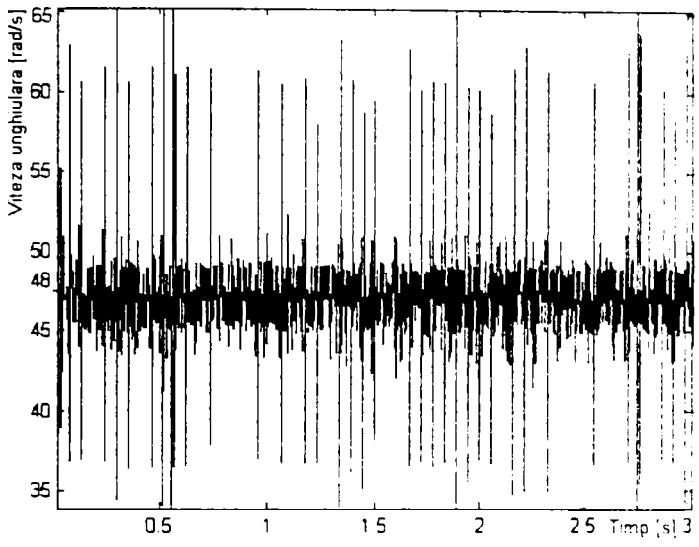


Fig. A.33.B. 1. (463.bl)

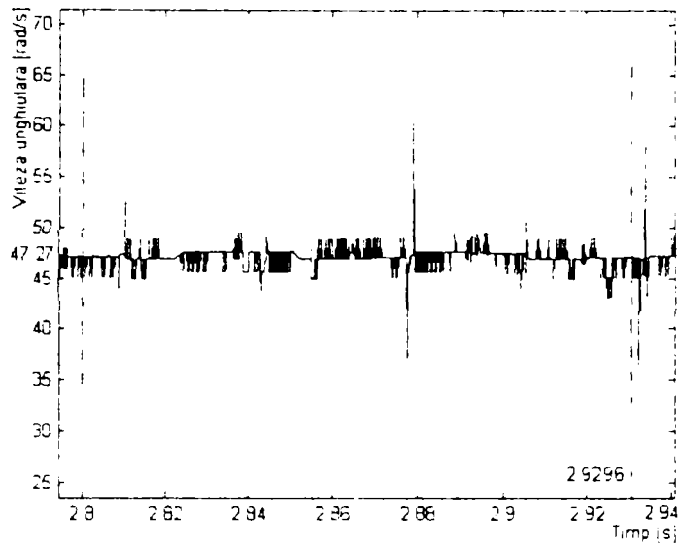


Fig. A.33.B. 2. (463.bl. det)

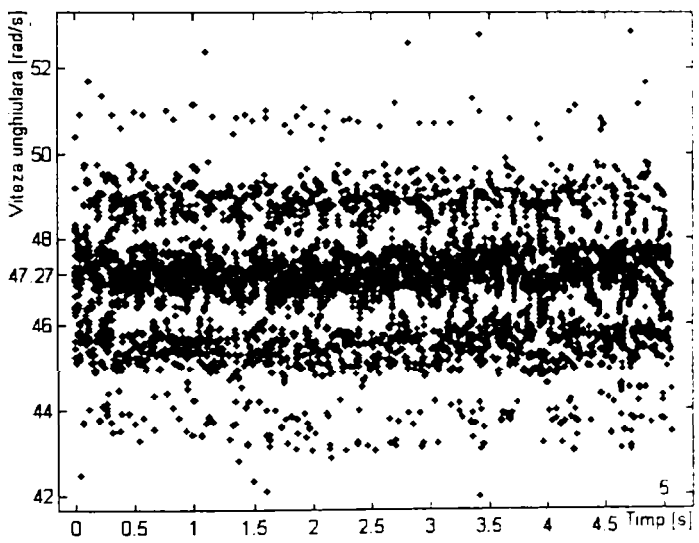


Fig. A.33.B. 3. (463.bl. puncte)

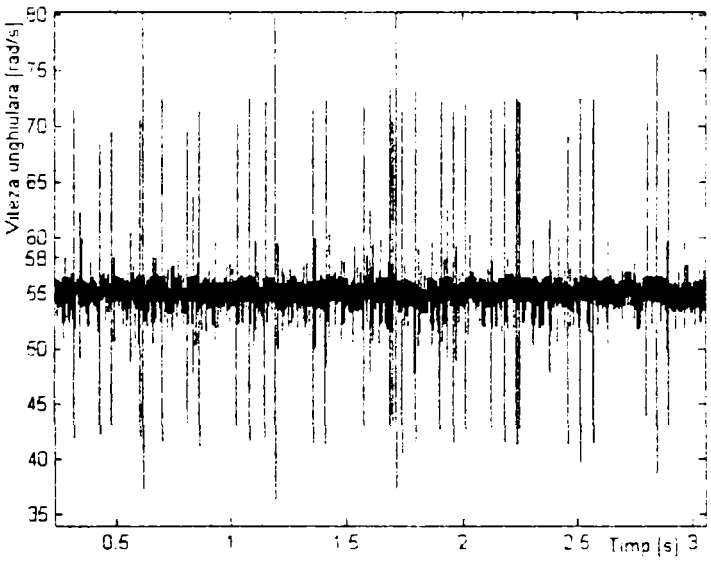


Fig. A.33.B. 4. (559.bl)

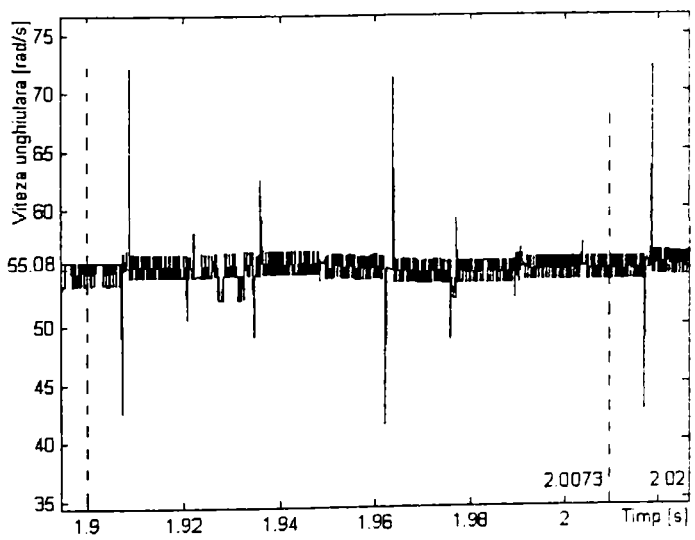


Fig. A.33.B. 5. (559.bl.det)

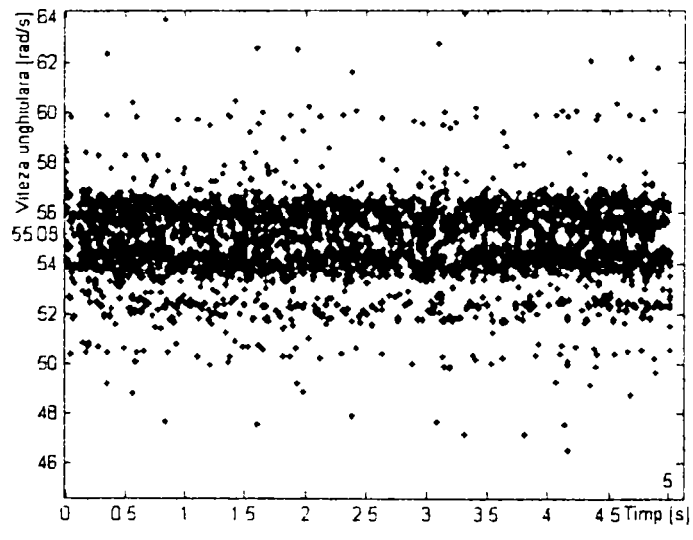


Fig. A.33.B. 6. (559.bl. puncte)

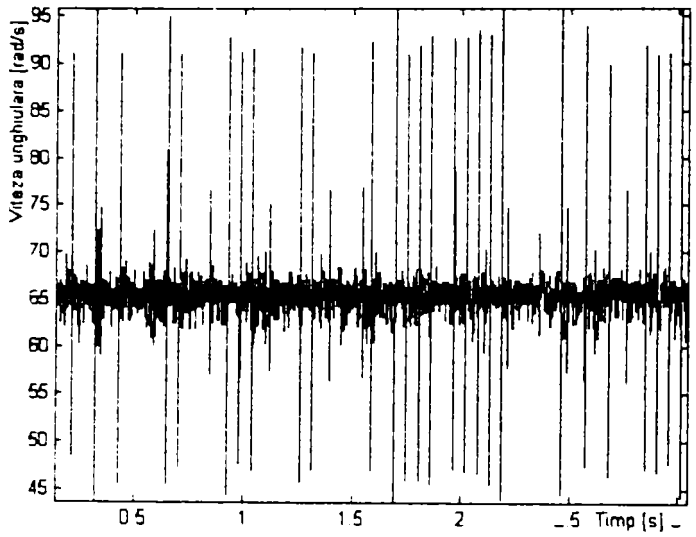


Fig. A.33.B. 7. (619.bl)

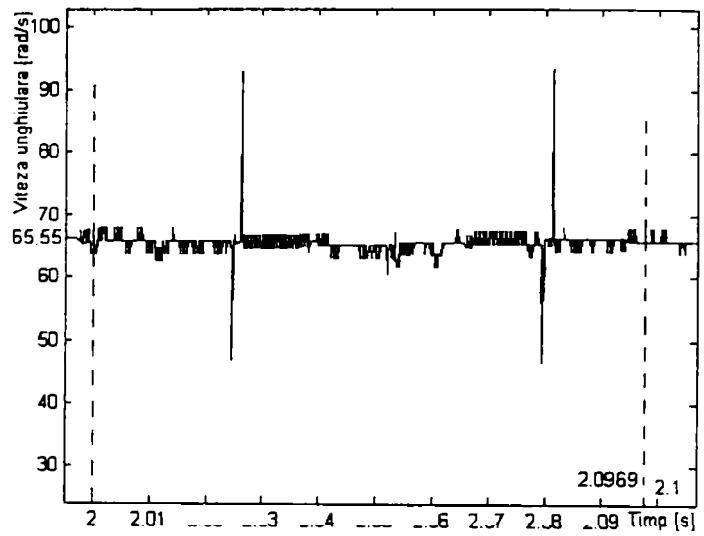


Fig. A.33.B. 8. (619.bl. det)

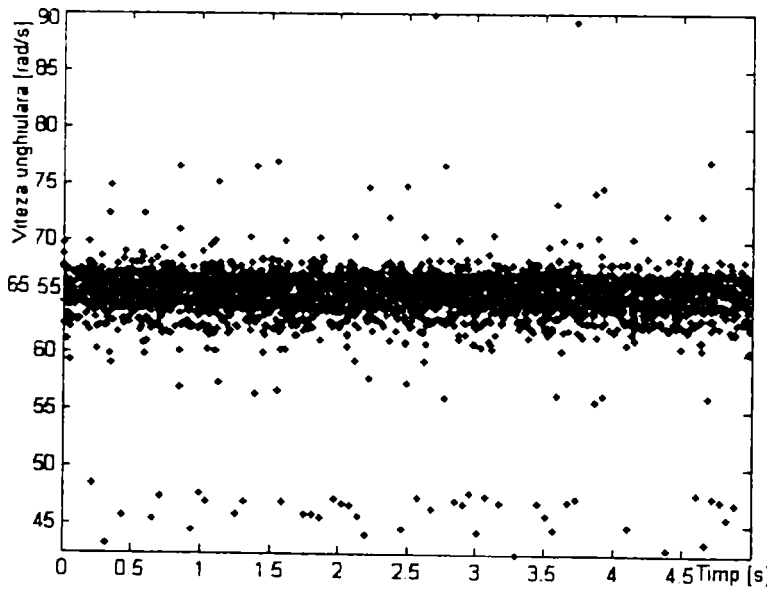


Fig. A.33.B. 9. (619.bl. puncte)

Notă. Diagramele notate cu puncte sunt reprezentări ale semnalului achiziționat prin valori singulare, iar cele notate cu „det" reprezintă semnalul achiziționat pe timpul unei turații.

Anexa 34.

Aşchii obținute cu masa rigidizată în cazul aşchierii vibropercutante şi în cazul aşchierii obişnuite

Fig. A34. 1. (463. 01.c)

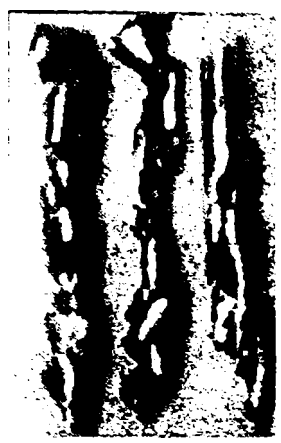


Fig. A34. 2. (463. 01.bl)

Fig. A34. 3. (463. 02.c)

Fig. A34. 4. (463. 02.bl)

Fig. A34. 5.(463. 04.c)

Fig. A34. 6.(463. 04.bl)

Fig. A34. 7. (463. 06.c)

Fig. A34. 8. (463. 06.bl)

Fig. A34. 9. (463. 08.c)

Fig. A34. 10. (463. 08.bl)

Fig. A34. 11. (559. 01.c)

Fig. A34. 12. (559. 01.bl)

Fig. A34. 13. (559. 02.c)

Fig. A34. 14. (559. 02.bl)

Fig. A34. 15. (.559. 04.c)

Fig. A34. 16. (559. 04.bl)

Fig. A34. 17. (559. 06.c)

Fig. A34. 18. (559. 06.bl)

Fig. A34. 19. (559. 08.c)

Fig. A34. 20. (559. 08.bl)

Fig. A34. 21. (619. 01.c)

Fig. A34. 22. (619. 01.bl)

Fig. A34. 23. (619. 02.c)

Fig. A34. 24. (619. 02.bl)

Fig. A34. 25. (619. 04.c)

Fig. A34. 26. (619. 04.bl)

Fig. A34. 27. (619. 06.c)

Fig. A34. 28. (619. 06.bl)

Fig. A34. 29.(619. 08.c)

Fig. A34. 30. (619. 08.bl)

Fig. A34. 31. (686. 01.c)

Fig. A34. 32.686. 01.bl)

Fig. A34. 33. (686. 02.c)

Fig. A34. 34.(686. 02.bl)

Fig. A34. 35. (686. 04.c)

Fig. A34. 36. (686. 04.bl)

Fig. A34. 37. (686. 06.c)

Fig. A34. 38. (686. 06.bl)

Fig. A34. 39. (686. 08.c)

Fig. A34. 40. (686. 08.bl)

Fig. A34. 41. (771. 01.c)

Fig. A34. 42. (771. 01.bl)

Fig. A34. 43. (771. 02.c)

Fig. A34. 44. (771. 02.bl)

Fig. A34. 45. (771. 04.c

Fig. A34. 46. (771. 04.bl)

Fig. A34. 47. (771. 06.c)

Fig. A34. 48. (771. 06.bl)

Fig. A34. 49. (771. 08.c)

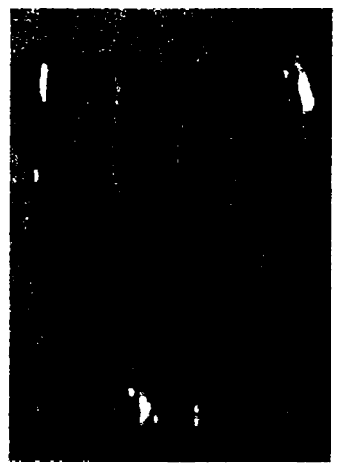


Fig. A34. 50. (771. 08.bl)

Notă. Semnificația notărilor din paranteză:

- primele trei cifre reprezintă turația utilizată la arborele mașinii-unelte;
- următoarele două cifre reprezintă avansul utilizat;
- c - aşchiere cu vibropercuții;
- bl– dispozitivul vibropercutant a fost blocat (aşchiere obişnuită).

Anexa 35.

1.Rugozitatea suprafețelor prelucrate prin găurire cu vibroșocuri (fig.a) și obișnuit (fig.b). Mărire 25x.

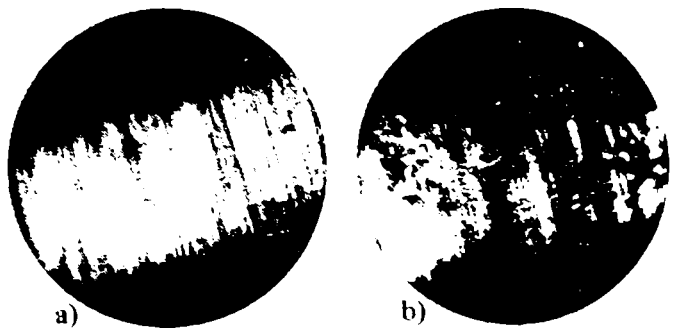


Fig. A.35. 1. (771. 08)

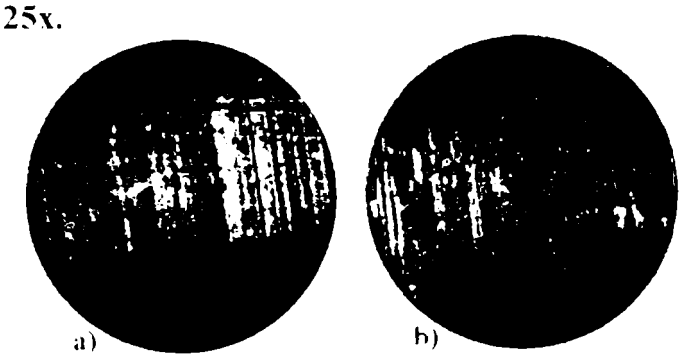


Fig. A.35. 2. (771. 06)

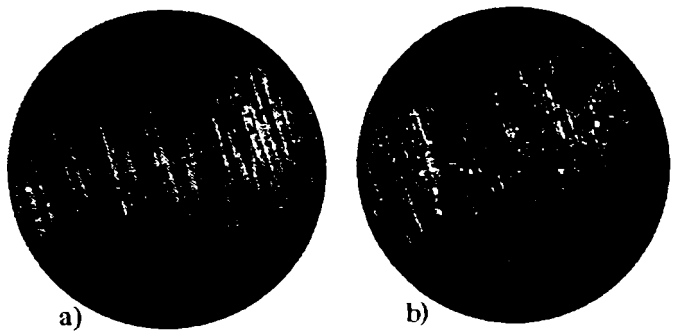


Fig. A.35. 3 (771. 04)

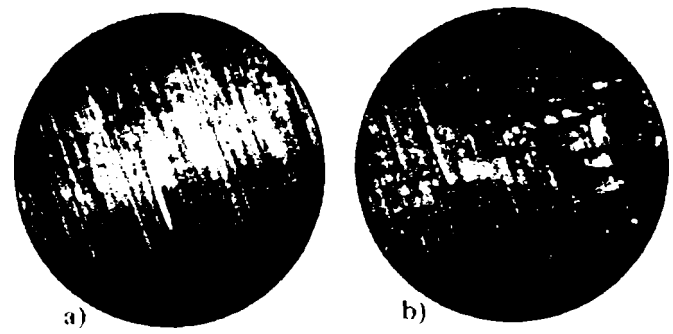


Fig. A.35. 4. (771. 02)

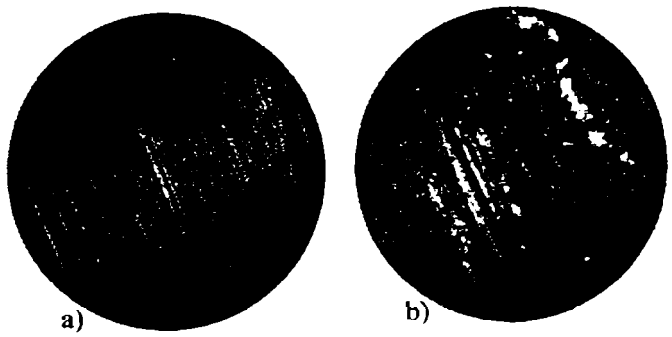


Fig. A.35. 5. (771. 01)

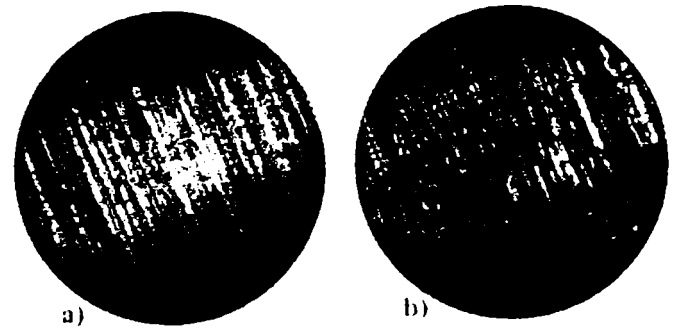


Fig. A.35. 6. (686. 08)

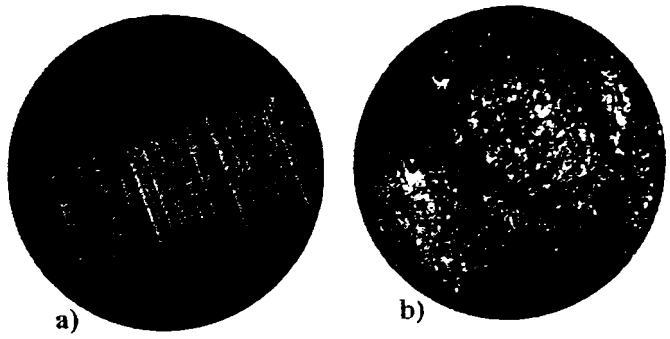


Fig. A.35. 7. (686. 06)

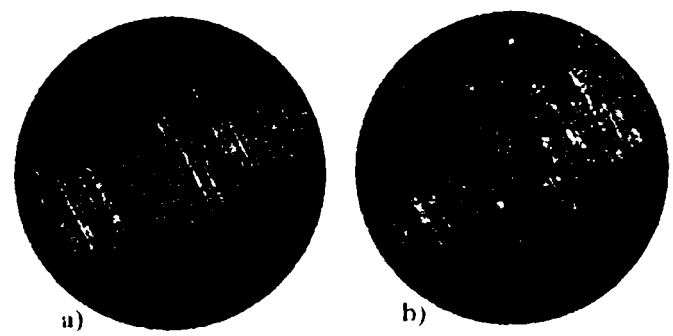


Fig. A.35. 8. (686. 04)

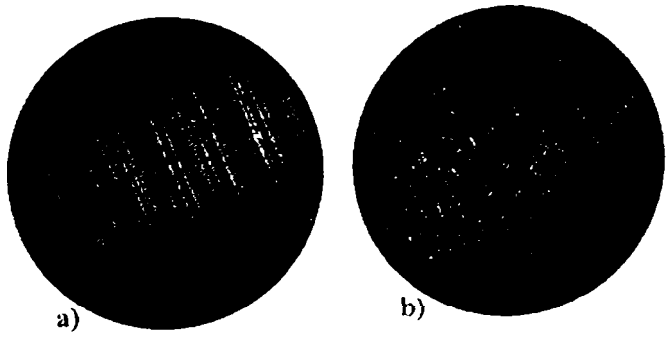


Fig. A.35. 9. (686. 02)

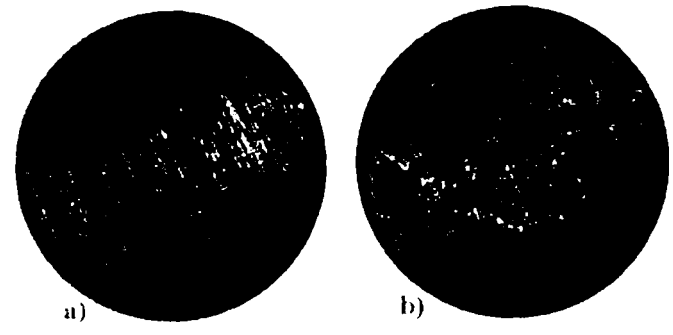


Fig. A.35. 10. (686.01)

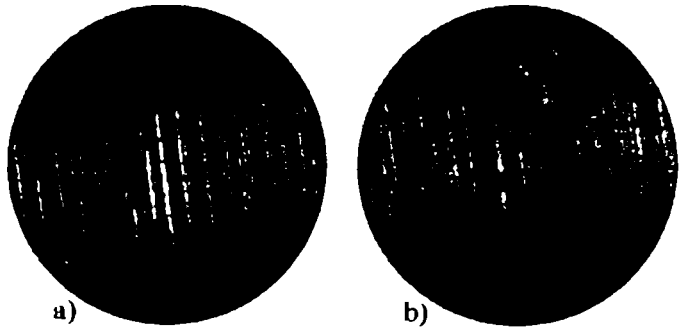


Fig. A.35. 11. (619. 08)

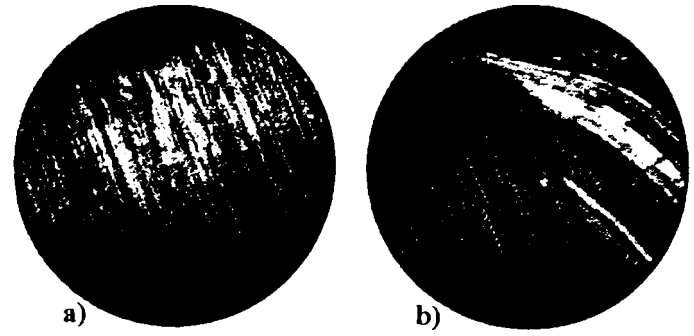


Fig. A.35. 12. (619. 06)

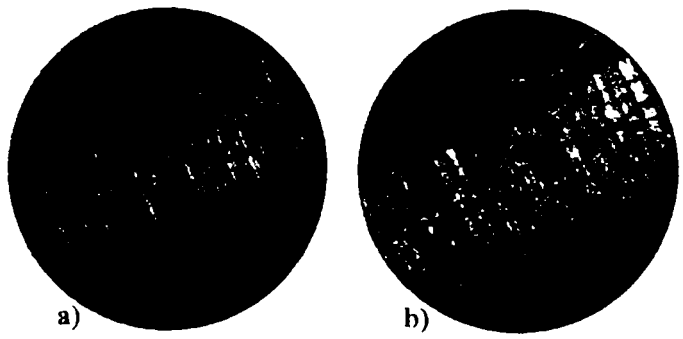


Fig. A.35. 13. 619. 04)

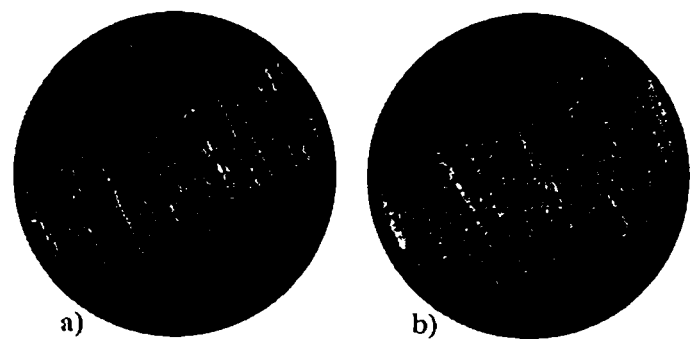


Fig. A.35. 14. (619. 02)

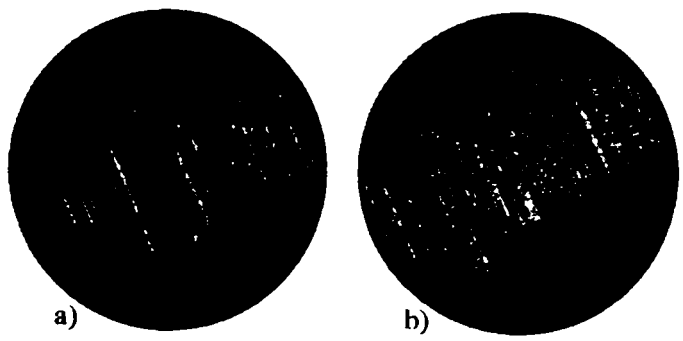


Fig. A.35. 15. (619. 01

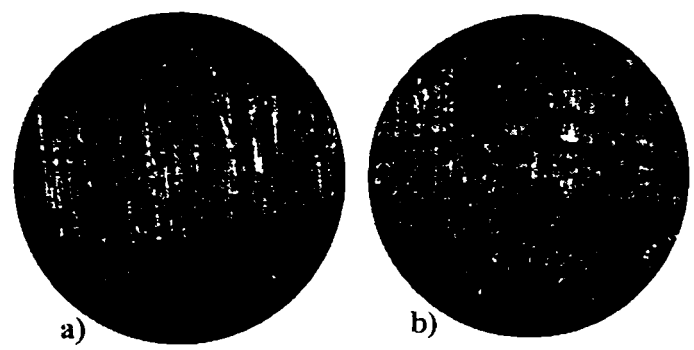


Fig. A.35. 16. (559. 08)

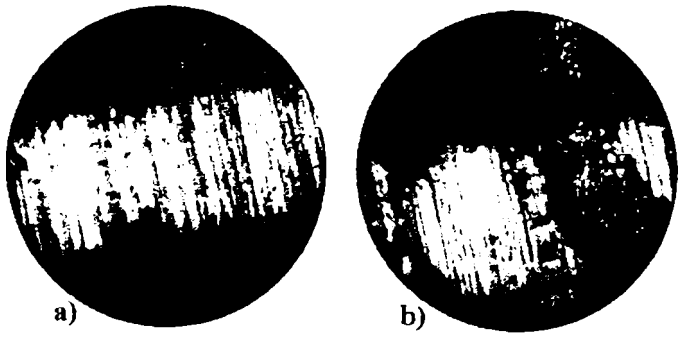


Fig. A.35. 17. (559. 06)

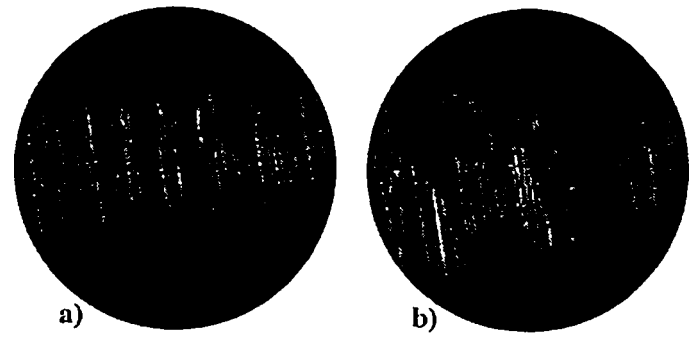


Fig. A.35. 18. (559. 04)

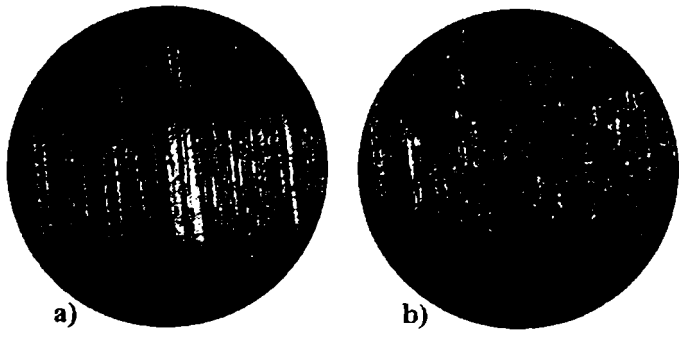


Fig. A.35. 19. (559. 02)

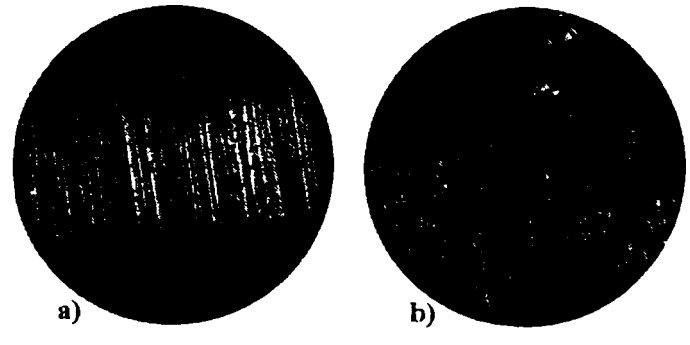


Fig. A.35. 20. (559. 01)

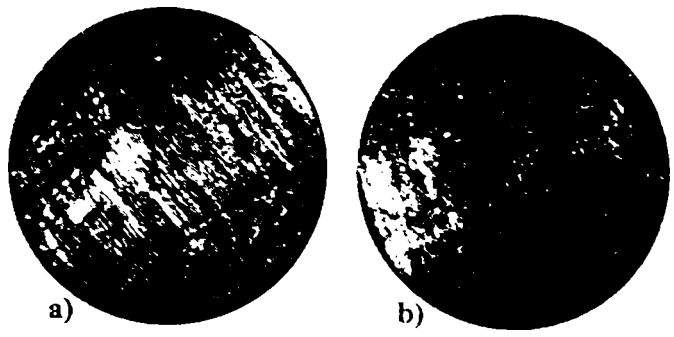


Fig. A.35. 21. (463. 08)

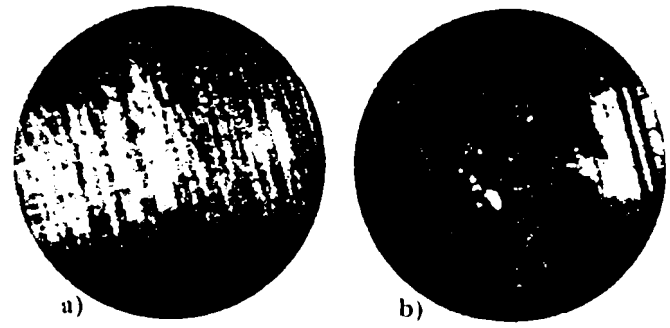


Fig. A.35. 22. (463. 06)

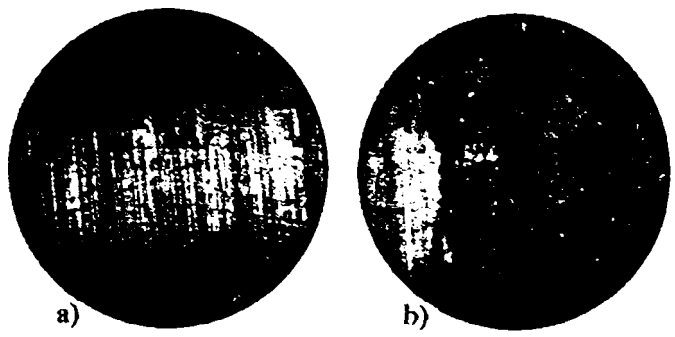


Fig. A.35. 23. (463. 04)

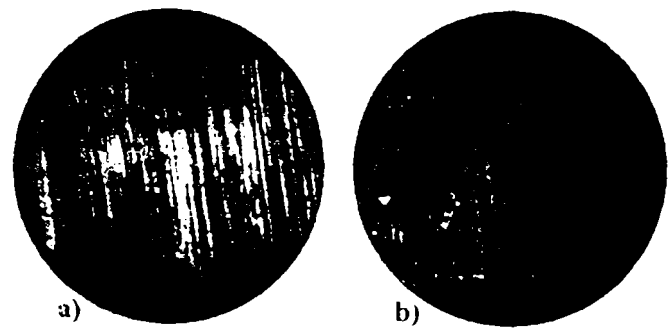


Fig. A.35. 24. (463. 02)

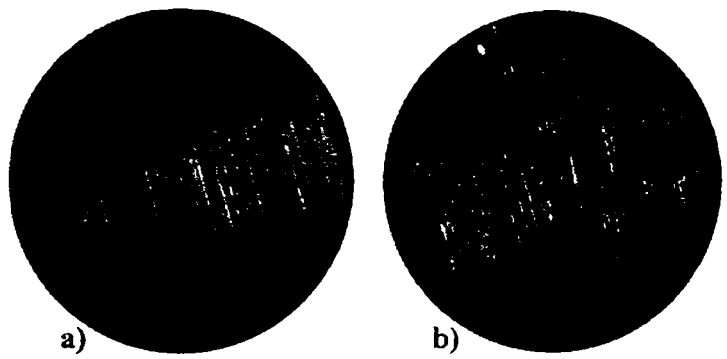


Fig. A.35. 25. (463. 01)

2. Etaloane pentru rugozitate tip STAL DOVODKA E15718 (Gost 9378-80)

Fig.A.35.26.Rci.R_a=1,6÷0,8[μm]

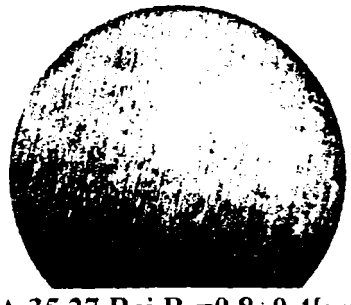


Fig.A.35.27.Rci.R_a=0,8÷0,4[μm]

Fig.A.35.28.Rci.R_a=0,4÷0,2[μm]

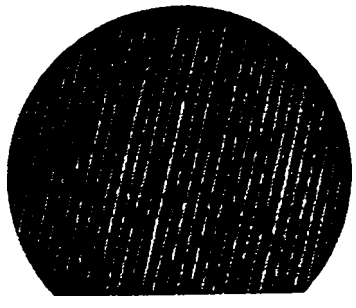


Fig.A.35.29.Si R_a= 3,2÷1,6 [μm]

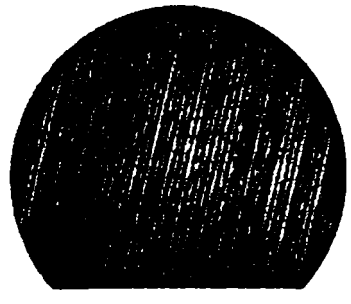


Fig.A.35.30.Si.R_a=1,6÷0,8 [μm]

Fig.A.35.31.Si.R_a= 0,8÷0,4[μm]

Notă. Primele trei cifre din paranteze reprezintă turația arborelui mașinii-unelte, iar următoarele două cifre viteza de avans.Rci - rectificare interioară: Si - strunjire interioară.

Anexa 36.

Microstructura stratului aşchiat cu vibropercuții (fig. a) şi a stratului aşchiat obişnuit (fig. b.)
Scara de mărire 500x

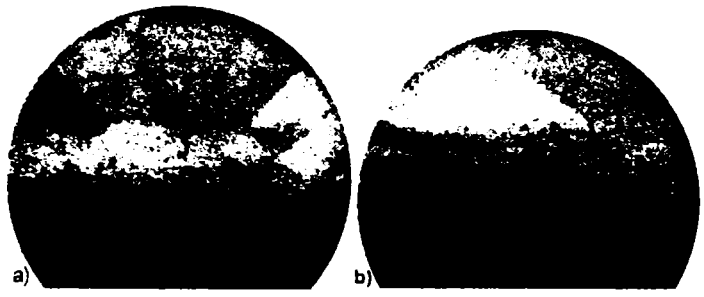


Fig. A.36. 1. (771. 08)

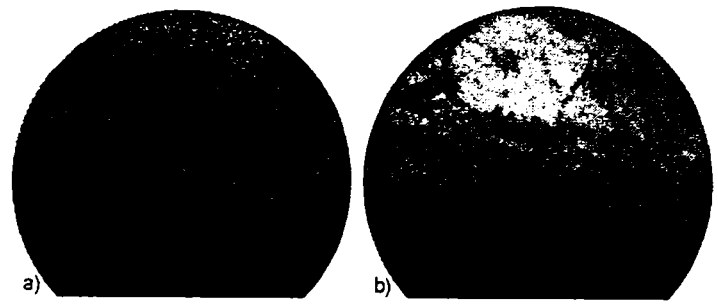


Fig. A.36. 2. (771. 06)

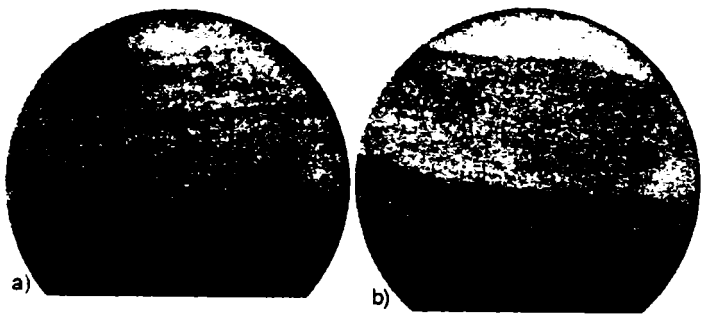


Fig. A.36. 3. (771. 04)

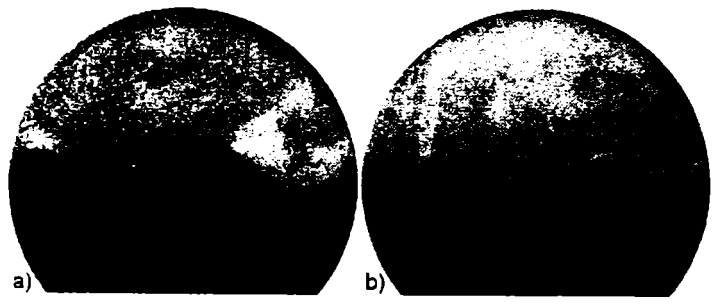


Fig. A.36. 4. (771. 02)

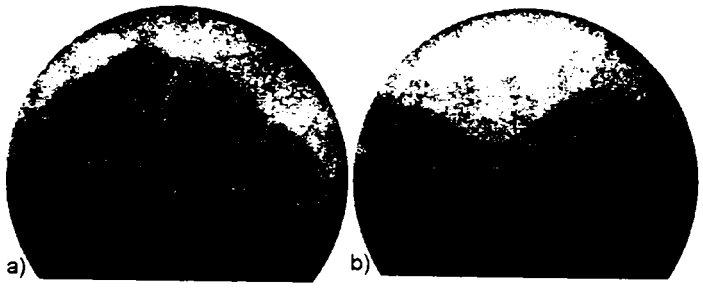


Fig. A.36. 5. (771. 01)

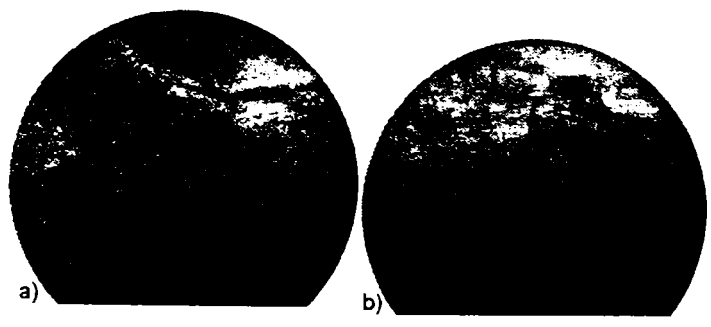


Fig. A.36. 6. (686. 08)

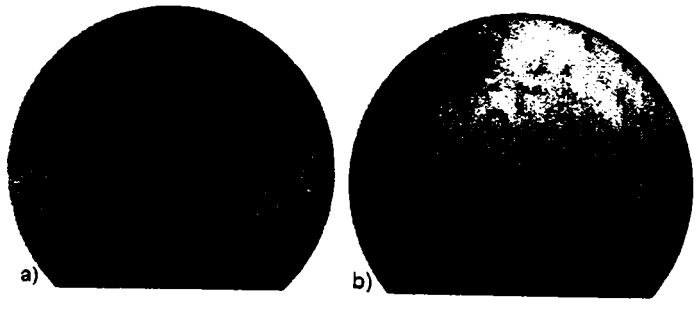


Fig. A.36. 7. (686. 06)

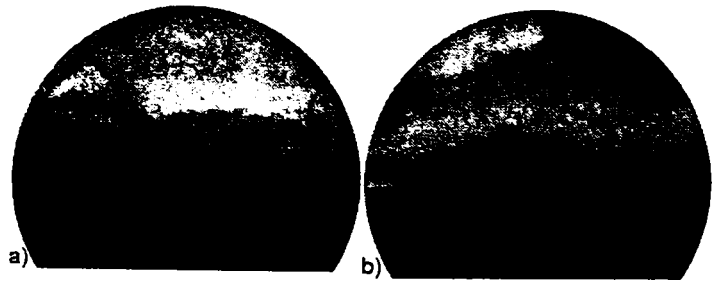


Fig. A.36. 8. (686. 04)

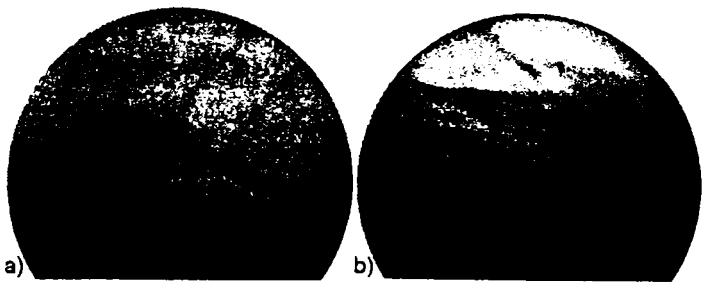


Fig. A.36. 9. (686. 02)

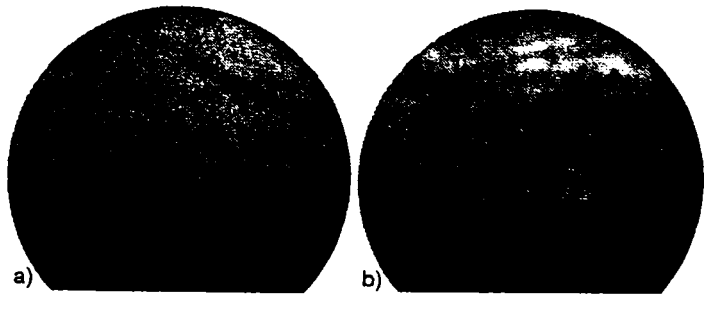


Fig. A.36. 10. (686. 01)

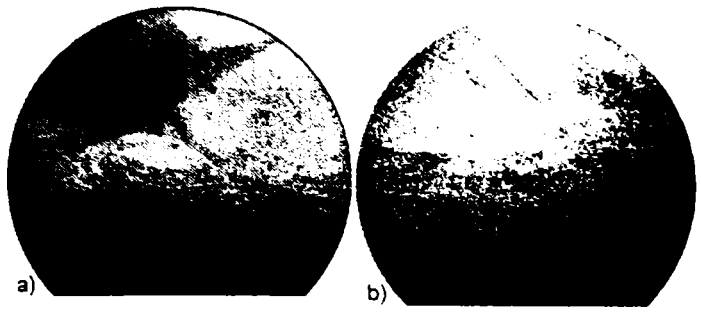


Fig. A.36. 11. (619. 08)

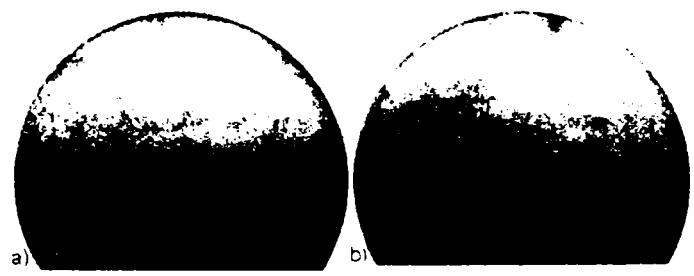


Fig. A.36. 12. (619. 06)

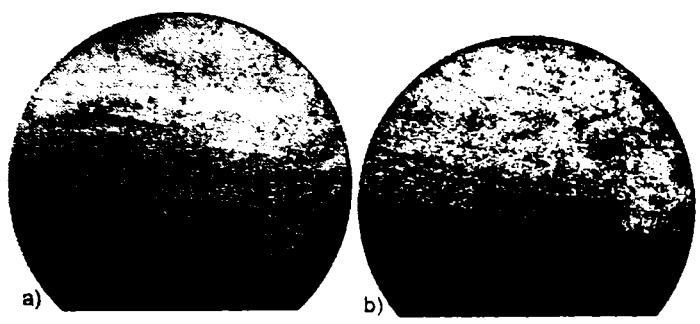


Fig. A.36. 13. (619. 04)

Fig. A.36. 14. (619. 02)

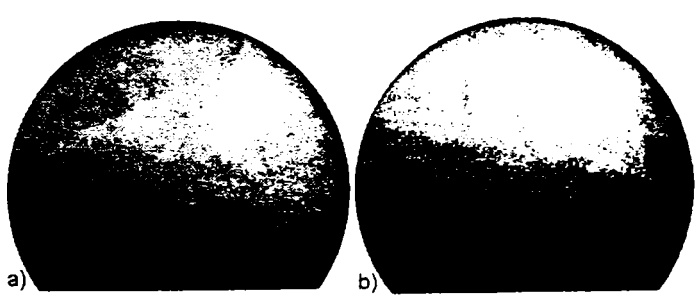


Fig. A.36. 15. (619. 01)

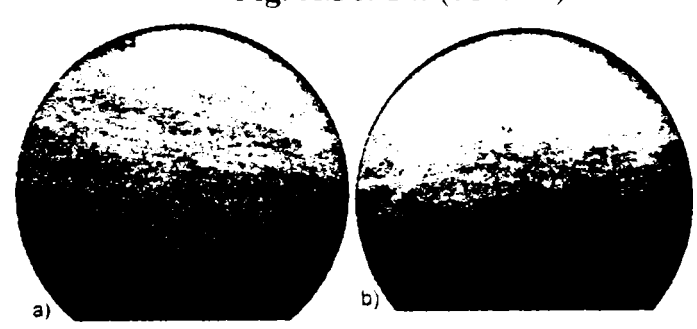


Fig. A.36. 16. (559. 08)

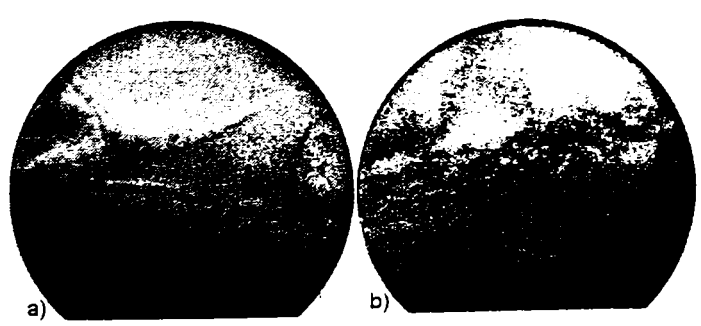


Fig. A.36. 17. (559. 06)

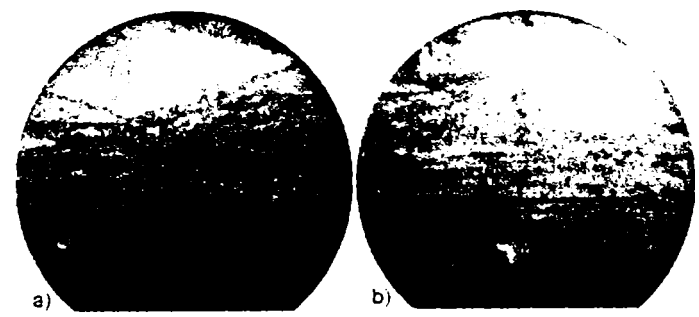


Fig. A.36. 18. (559. 04)

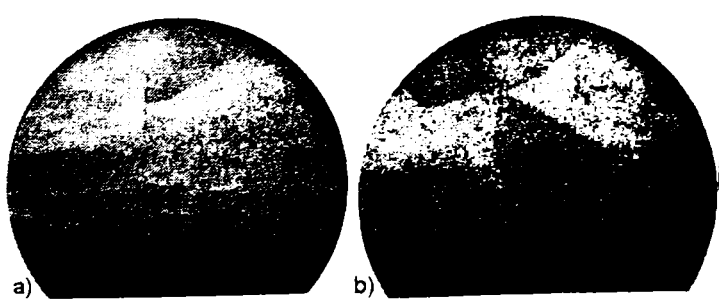


Fig. A.36. 19. (559. 02)

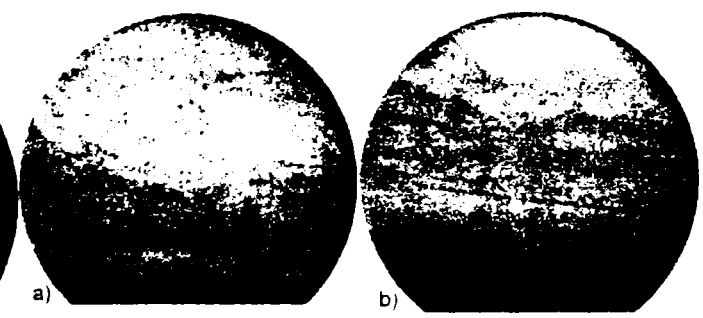


Fig. A.36. 20. (559. 01)

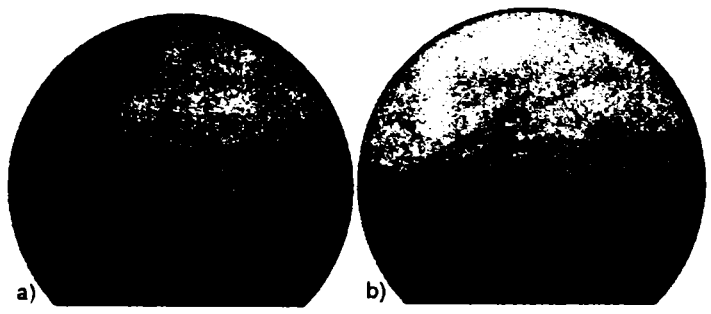


Fig. A.36. 21. (463. 08)

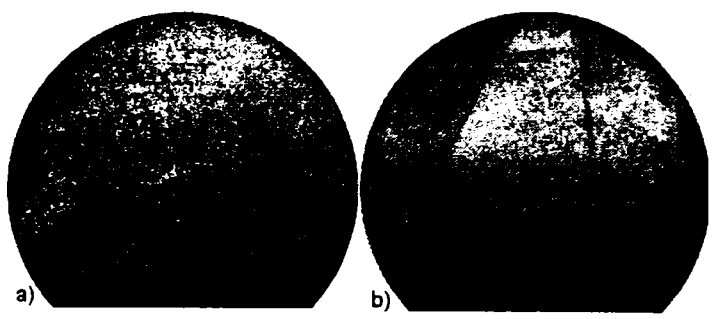


Fig. A.36. 22. (463. 06)

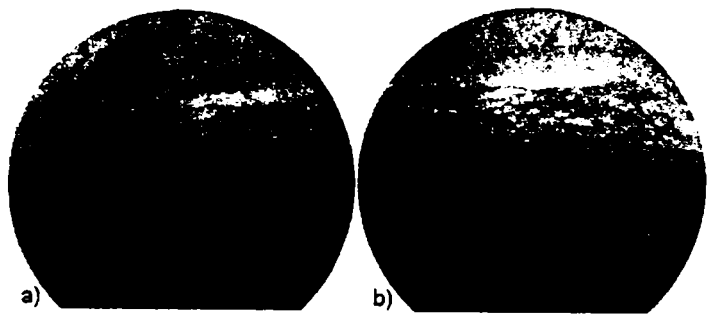


Fig. A.36. 23. (463. 04)

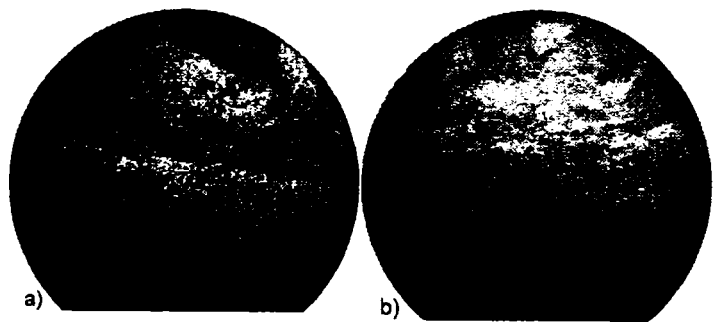


Fig. A.36. 24. (463. 02)

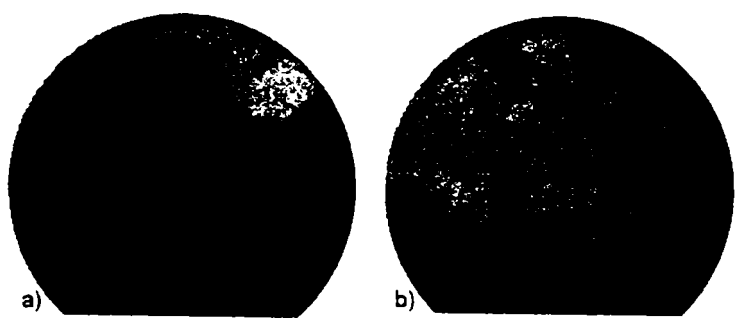


Fig. A.36. 25. (463. 01)

Notă. Primele trei cifre din paranteze reprezintă turaţia arborelui maşinii-unelte, iar următoarele două cifre viteza de avans.

a) - structuri la aşchierea vibropercutantă;

b - structuri la aşchierea obişnuită. structuri la aşchierea

Anexa 37.

Caracteristicile plăcii de achiziție de date PCI 1200

Caracteristicile plăcii de achiziție de date PCI 1200 date de catalogul firmei National Instruments sunt:

1. Moduri de funcționare :
   - cu nul de referință (referenced single ended);
   - cu nul fără referință (nonreferenced single ended);
   - diferențial.
2. Amplificarea internă : 2,5,10,20,50,100.
3. Domeniul de măsurare (în funcție de amplificare) :
   - unipolar : 0-10-V (neamplificat);
   - bipolar : ±5V (neamplificat.
4. Intrări analogice :
   - 16 canale – pentru modul de lucru cu nul;
   - 8 canale – pentru modul de lucru diferențial.
5. Ieşiri analogice : 2 cu un domeniu de funcționare de 0-10V
6. Rata de achiziție, programabilă: până la 100.000 eșantioane / sec (100 kHz);
7. Cuantificarea convertorului Analog/Digital: 12 bit.
8. Integrare în sistemul de calcul : slot PCI.

Anexa 38.

Programul de achiziţie „C++_disp.cpp" a semnalelor accelerometrului şi timbrelor tensometrice şi înscrierea datelor în fişiere pe hard disc

```
#include "nidaqex.h"
void main(void)
{
   /*
    * Variabile locale:
    */
   i16 iStatus = 0;
   i16 iRetVal = 0;
   i16 iDevice = 1;
   i32 lTimeout = 600;
   i16 iChan = 1;
   i16 iGain = 1;
   f64 dSampRate = 20000.0;
   f64 dScanRate = 0;
   u32 ulCount = 100000;
   char* strFilename = "etrf2120.d1";
   i16 iIgnoreWarning = 0;
   /* Setarea limitei de timp :nr. secunde * 18tacturi/sec.)
    */
   iStatus = Timeout_Config(iDevice, lTimeout);
   iRetVal = NIDAQErrorHandler(iStatus, "Timeout_Config",
    iIgnoreWarning);
   /* Achizitioneaza date pe un canal analogic, diferential si
          * scrie pe disc. Datele sunt stocate pe 16 biti.
          * Numele fisierului este dat de 'strFilename',
          * Traductorul se alimenteaza cu tensiune de 9V.
          * Achizitia se face pe canalul 0 diferntial (pin 1 si 2)
          * Se leaga cate o rezistenta pe fiecare canal ( pin 1 - 11, 2 - 11)
          * si se citeste in paralel cu alimentarea.
    */
         printf("Start!");
         /* Start achizitie pe disc.
         */
   iStatus = Lab_ISCAN_to_Disk (iDevice, iChan, iGain, strFilename, ulCount, dSampRate,
dScanRate, 0);
         /* Verifica erori.
         */
   iRetVal = NIDAQErrorHandler(iStatus, "Lab_ISCAN_to_Disk",
    iIgnoreWarning);
         /* Daca totul este OK, afişează 'Datele au fost achizitionate!" si termina programul.
         */
   if (iStatus == 0) {
      printf("Datele au fost achizitionate!");
   }
   /* Reseteaza limita de timp. */
   iStatus = Timeout_Config(iDevice, -1);
}
/* Sfarsit program */
```

Programele de prelucrare a datelor achizitionate la etalonare

Programul de prelucrare și convertire a semnalelor în tensiune „ medie_tenso.m"

```
global contor x1;
[filename,pathname] = uigetfile('*.*','Selectati fisierul de date',300,100);
if filename ~=0
  fid = fopen(strcat(pathname,filename),'r');
  [x1,count1] = fread(fid,inf,'int16');
  fclose('all');

  count1 = 80000;

  x1 = x1(1:count1);
  timp  = [1:count1]/20000;
  x2 = -1*x1.*(10/4096);
  medie = mean(x2(1:count1));
  medie_str = num2str(medie);


  f_1 = figure('Name','Semnal traductor tensometric',...
  'NumberTitle','off');
  plot(timp,x2,'k');
  title(strcat('Medie semnal = ', medie_str,'[V]'));
  xlabel('Timp [s]');
  ylabel('Tensiune [V]');
elseif filename == 0
  disp('Nu s-a selectat nici un fisier');
end
```

Programele de calcul „etalonare.m"a forțelor și momentelor corespunzătoare tensiunii medii calculate la etalonare

% Forta axiala

```
clear all
load et_fax
y = fax*10;
x = etfax;
%x = x.*10/4096;
px = polyfit(x,y,1);
f = polyval(px,x);
tabelx = [x' y' f' (y-f)']
f_1 = figure('Name','Diagrama de etalonare traductor tensometric',...
  'NumberTitle','off');
   plot(x,y,'or',x,f,'-k')
   %axis([0  0.5  0  41])
   title(strcat('Diagrama de etalonare a fortei pe directie axiala'));
   ylabel('Forta [N]');
   xlabel('Tensiune [V]');
```

```
disp('Dreapta de aproximare pentru Fx: ')
disp('<Fax = Forta pe directie axiala in [N]; U = tensiune masurata in [V]>')
disp(strcat('Fax = ',num2str(px(1)),'*U + ',num2str(px(2))))
```

% Moment

```
clear all
load et_mom
y = mom;
x = etmom;
%x = x.*10/4096;
py = polyfit(x,y,1);
f = polyval(py,x);
tabely = [x' y' f' (y-f)']
f_1 = figure('Name','Diagrama de etalonare traductor tensometric',...
  'NumberTitle','off');
   plot(x,y,'or',x,f,'-k')
   %axis([0  0.5  0  41])
   title(strcat('Diagrama de etalonare a momentului'));
   ylabel('Moment [Nm]');
   xlabel('Tensiune [V]');
   disp('Dreapta de aproximare pentru Fy: ')
   disp('<M = Moment in [Nm]; U = tensiune masurata in [V]>')
   disp(strcat('M = ',num2str(py(1)),'*U + ',num2str(py(2))))
```

Anexa 39

Programe utilizate la calculul şi afişarea diagramelor vitezelor unghiulare ale burghiului

Programul „viteza.m"

```
% Meniu viteza
%global fd_a0 fd_a1;
global viteza_rot timp_sec;
viteza_rot = 0;
timp_sec = 0;
%fd_a0 = 1/100000;
%fd_a1 = 5;

f60_1_1 = figure('Name','Viteza unghiulara',...
  'NumberTitle','off','Position', [250 250 310 90]);

% Butoane
f60_1_4_2  = uicontrol('Style', 'pushbutton', 'String', 'Citire fisier',...
  'Position', [30 60 110 20], 'Callback', 'viteza_fis');
f60_1_4_2a  = uicontrol('Style', 'pushbutton', 'String', 'Citire fisier mat',...
  'Position', [30 30 110 20], 'Callback', 'viteza_mat');
f60_1_4_3  = uicontrol('Style', 'pushbutton', 'String', 'Diagrama timp',...
  'Position', [170 30 110 20], 'Callback', 'viteza_diag');
%f60_1_4_7  = uicontrol('Style', 'pushbutton', 'String', 'Stabilire domeniu',...
%  'Position', [170 60 110 20], 'Callback', 'prel_dat1');

% Iesire
f60_1_6  = uicontrol('Style', 'pushbutton', 'String', 'Iesire',...
  'Position', [170 5 110 20], 'Callback', 'delete(f60_1_1)');
```

Programul „viteza_fis.m"

```
    global viteza_rot timp_sec x1
% Citire din fisier
[filename,pathname] = uigetfile('*.*','Selectati fisierul de date',300,100);
if filename ~=0
   fid = fopen(strcat(pathname,filename));
   j = 1;
   h_bara = waitbar(0,'Citire valori din fisier...');
   while 1
      line = fgetl(fid);
      if ~isstr(line),break,end
      tiro(j) = str2num(line);
      line = fgetl(fid);
      if ~isstr(line),break,end
      timp(j) = str2num(line);
      j = j + 1;
      waitbar(j/40000)
   end
   fclose(fid);
```

```
close(h_bara);
nr_int1 = 1;
nr_int2 = 40000;
nr_conv = 50;
tiro         = tiro(nr_int1:nr_int2);
timp         = timp(nr_int1:nr_int2);
timp_cont    = zeros(1,nr_int2-nr_int1);
tiro_cont    = zeros(1,nr_int2-nr_int1);
suma_timp_cont = zeros(1,nr_int2-nr_int1);
suma_tiro_cont = zeros(1,nr_int2-nr_int1);
viteza_rot   = zeros(1,nr_int2-1-nr_int1);

% Calcul delta t si delta x
h_bara1 = waitbar(0,'Conversie date...');
for i = 1:nr_int2-nr_int1-1
   if timp(i+1) <= timp(i)
      timp_cont(i)   = abs(timp(i) - timp(i+1));
   else
      timp_cont(i)   = abs(timp(i) + (65535 - timp(i+1)));
   end
   if tiro(i+1) <= tiro(i)
      tiro_cont(i)   = abs(tiro(i) - tiro(i+1));
   else
      tiro_cont(i)   = abs(tiro(i) + (65535 - tiro(i+1)));
   end

   %suma_timp_cont(i) = sum(timp_cont(1:i));
      %suma_tiro_cont(i) = sum(tiro_cont(1:i));
      %viteza_rot(i)    = 4000*pi*suma_tiro_cont(i)/suma_timp_cont(i)*(50.3/128.7);
      %timp_sec(i)      = suma_timp_cont(i)./2000000;
      waitbar(i/nr_int2);
   end
   %%%%%%%
   ferestra_conv = zeros(1,nr_conv)+1/nr_conv;
   tiro_conv = conv(tiro_cont,ferestra_conv);
   timp_conv = conv(timp_cont,ferestra_conv);
   viteza_rot = 4000*pi*tiro_conv./timp_conv*(50.3/128.7); % rad/sec
   timp_sec = cumsum(timp_conv)./2000000; % secunde
   %%%%%%%%%
   %timp_sec1 = timp_sec;
   %ti1 = 0:0.00005:max(timp_sec);
   %x1 = interp1(timp_sec,viteza_rot,ti1,'linear');
   %fd_a0 = min(timp_sec);
   %fd_a1 = max(timp_sec);
   %save(filename,'ti1','x1','timp_sec','fd_a0','fd_a1')
   save(filename,'viteza_rot','timp_sec');
   close(h_bara1);
end
```

Programul „viteza_mat.m"

```
global viteza_rot timp_sec x1
% Citire din fisier
[filename,pathname] = uigetfile('*.*','Selectati fisierul tip .mat',300,100);
if filename ~=0
   load(strcat(pathname,filename));
end
```

Programul „viteza_diag.m",

```
  %count1 = find(timp_sec == fd_a0);
  %count2 = find(timp_sec == fd_a1);
  %ti2 = timp_sec(count1:count2);
  %x2  = x1(count1:count2);

% Afisare
   f_1 = figure('Name','Viteza unghiulara a sculei','NumberTitle','off');
   %plot(ti2,x2,'k');
   plot(timp_sec,viteza_rot,'k');
   %title(strcat('Viteza unghiulara a sculei in functie de timp'));
   xlabel('Timp [s]');ylabel('Viteza unghiulara [rad/s]');
```

Anexa 40

Programe de prelucrare a semnalelor înregistrate de la accelerometru și timbre tensometrice

Programul „acc_mat.m" de prelucrare a semnalelor pentru determinarea accelerațiilor

```
% Deschidere fisier
[filename,pathname] = uigetfile('*.*','Acceleratii',300,100);
if filename ~=0
  fid = fopen(strcat(pathname,filename),'r');
  [x1,count1] = fread(fid,inf,'int16');
  fclose('all');
  %Transformare in volti
  x1 = (x1).*(10/4096);
  % Se scade valoarea offset
  x1 = x1 - mean(x1);
  % Constanta traductorului KD 35
  kd35 = 0.0171;
  % Calibrare
  x1 = x1./kd35;
  Fs = 20000;
  save(strcat(pathname,filename),'x1','Fs');
  clear x1;
end
```

Programul „forta_mat.m" de prelucrare a semnalelor pentru determinarea forței axiale

```
% Deschidere fisier
[filename,pathname] = uigetfile('*.*','Forta axiala',300,100);
if filename ~=0
  fid = fopen(strcat(pathname,filename),'r');
  [x1,count1] = fread(fid,inf,'int16');
  fclose('all');

  %Transformare in volti
  x2 = (x1).*(10/4096);

  % Etalonare Volt -> Newton
  fax =10423.2602*x2  - 13.3435;

  Fs = 20000;
  save(strcat(pathname,filename),'fax','Fs');
  clear x1;
  clear x2;
elseif filename == 0
  disp('Nu s-a selectat nici un fisier');
end
```

Programul „moment_mat.m" de prelucrare a semnalelor pentru determinarea momentului la șchiere

```
% Deschidere fisier
[filename,pathname] = uigetfile('*.*','Momente',300,100);
if filename ~=0
  fid = fopen(strcat(pathname,filename),'r');
  [x1,count1] = fread(fid,inf,'int16');
  fclose('all');

  % Transformare Unitati Digitale -> Volt
  x2 = (x1).*(10/4096);


  % Etalonare Volt -> Newton*Metru
  m_asc =140.16132*x2 + 0.051902;

  Fs = 20000;
  save(strcat(pathname,filename),'m_asc','Fs');
  clear x1;
  clear x2;
end
```

Anexa 41.

Program de trasare diagrame „peak to peak" pentru accelerații, forțe, momente și viteze unghiulare în funcție de turații și avans

```
y0n = [1 2 4 6 8]';
x0s = [463 559 619 686 771]';
[X,Y] = meshgrid(x0s,y0n);
[XI,YI] = meshgrid(463:1:771,1:0.1:8);
% %Amplitudinile acceleratiilor masurate „peak to peak" pe pinola masinii-unelte
% %Dispozitiv blocat
acc_bl  = [13.3233 43.4954 26.1448 9.8341 10.2814];
   x  = [463 559 619 686 771];
   xi = (463:1:771);
   yi = interp1(x,acc_bl,xi,'spline');
   figure; plot(xi,yi,'r');
% %Dispozitiv  neblocat
acc_nebl = [56.8281 58.3675 62.9000 61.1349 58.3551];
   x  = [463 559 619 686 771];
   xi = (463:1:771);
   yi = interp1(x,acc_nebl,xi,'spline');
   hold on; plot(xi,yi,'k');
   xlabel('Turatie [rot/min]')
   ylabel('Acceleratie x9,81 [m/s^2]')
   title('Amplitudinea acceleratiilor „peak to peak" masurate pe pinola, avans = 8 [mm/min]');
% Amplitudinile acceleratiilor masurate „peak to peak" pe dispozitivul vibropercutant
%Dispozitiv blocat
acc_bl1 = [46.2484 49.4273 55.2416 48.8035 17.5103
          51.9238 47.8414 55.5066 52.6306 21.8634
          52.5990 52.5991 54.9780 51.0083 18.4875
          25.7291 47.5013 56.8282 51.0426 15.2643
          15.0161 53.1278 52.5991 46.3646 15.6952];
 Z = acc_bl1;
 ZI = interp2(X,Y,Z,XI,YI,'spline');
 f = figure; mesh(XI,YI,ZI);
 title('Amplitudinea acceleratiilor „peak to peak" masurate pe dispozitiv - aschiere obisnuita');
% %Dispozitiv  neblocat
acc_nebl1 = [65.1982 70.4846 67.9811 64.3172 62.7313
            64.3172 68.2819 69.1184 66.2268 64.3715
            62.9956 65.5198 65.5132 65.9031 67.5484
            63.7895 64.4934 63.4498 71.3656 66.6894
            68.2819 67.6652 62.1145 64.3172 65.9088];
  Z = acc_nebl1;
  ZI = interp2(X,Y,Z,XI,YI,'spline');
  figure; mesh(XI,YI,ZI);
title('Amplitudinea acceleratiilor „peak to peak" masurate pe dispozitiv - aschiere vibropercutanta');
for i = 1:5
   x  = [463 559 619 686 771];
   xi = (463:1:771);
   yi = interp1(x,acc_bl1(i,:),xi,'spline');
   figure; plot(xi,yi,'r');
   yi = interp1(x,acc_nebl1(i,:),xi,'spline');
   hold on; plot(xi,yi,'k');
```

```
   xlabel('Turatie [rot/min]')
   ylabel('Acceleratie x9,81 [m/s^2]')
   title(strcat('Amplitudinea acceleratiilor „peak to peak" masurate pe dispozitiv, avans
',num2str(y0n(i)),' [mm/min]'));
end

% %Amplitudinile acceleratiilor masurate „peak to peak" pe universal
% %Dispozitiv blocat
 acc_bl2 = [33.0396 26.2419 31.0593 58.1498 26.6198
        46.5198 35.3717 32.2941 53.3921 48.0183
        50.2203 37.2570 37.2570 53.6564 52.1423
        49.9559 27.6951 54.7137 48.6344 52.0705
        44.6696 24.8027 53.3921 41.9032 50.4846];
 Z = acc_bl2;
 ZI = interp2(X,Y,Z,XI,YI,'spline');
 figure; mesh(XI,YI,ZI);
title('Amplitudinea acceleratiilor „peak to peak" masurate pe universal - aschiere obisnuita');
% %Dispozitiv neblocat
 acc_nebl2 = [59.8079 59.1119 61.3906 58.6578 57.8642
          59.7115 59.6691 63.4453 58.0541 58.2999
          58.4147 60.6951 62.1365 58.8987 58.5322
          60.0140 58.0160 60.0870 59.3142 59.8078
          58.7106 61.3344 57.1769 59.2070 60.0545];
 Z = acc_nebl2;
 ZI = interp2(X,Y,Z,XI,YI,'spline');
 figure; mesh(XI,YI,ZI);
 title('Amplitudinea acceleratiilor „peak to peak" masurate pe universal - aschiere vibropercutanta');
for i = 1:5
   x  = [463 559 619 686 771];
   xi = (463:1:771);
   yi = interp1(x,acc_bl2(i,:),xi,'spline');
   figure; plot(xi,yi,'r');
   yi = interp1(x,acc_nebl2(i,:),xi,'spline');
   hold on; plot(xi,yi,'k');
   xlabel('Turatie [rot/min]')
   ylabel('Acceleratie x9,81 [m/s^2]')
   title(strcat('Amplitudinea acceleratiilor „peak to peak" masurate pe universal, avans
',num2str(y0n(i)),' [mm/min]'));
end

% %Valorile fortelor masurate „peak to peak"utilizând masa tensometrică
% %Dispozitiv blocat
 f_bl = [203.9355 180.9467 182.7406 199.6012 185.0220
       205.3845 181.1595 204.0016 228.8560 207.0485
       207.0302 180.7416 208.1075 206.5807 180.7720
       234.1011 185.2642 210.9375 224.6696 204.0715
       254.1044 205.6916 202.6480 203.7595 202.2697];
 Z = f_bl;
 ZI = interp2(X,Y,Z,XI,YI,'spline');
 figure; mesh(XI,YI,ZI);
title('Ampl. fortelor axiale „peak to peak" masurate pe masa tensometrica - aschiere obisnuita');
```

```
% %Dispozitiv  neblocat
 f_nebl = [433.3346 408.9582 559.8704 484.3292 380.1355
        640.6597 432.4409 738.6678 583.6588 535.7921
        406.2227 458.0837 459.6198 304.2126 356.6560
        354.6729 407.4400 660.4824 382.0651 435.2803
        335.1495 385.5429 438.3508 332.5140 460.3307];
 Z = f_nebl;
 ZI = interp2(X,Y,Z,XI,YI,'spline');
 figure; mesh(XI,YI,ZI);
title('Ampl. fortelor axiale „peak to peak" masurate pe masa tensometrica - aschiere vibropercutanta');

for i = 1:5
    x  = [463 559 619 686 771];
    xi = (463:1:771);
    yi = interp1(x,f_bl(i,:),xi,'spline');
    figure; plot(xi,yi,'r');
    yi = interp1(x,f_nebl(i,:),xi,'spline');
    hold on; plot(xi,yi,'k');
    xlabel('Turatie [rot/min]')
    ylabel(' [N]')
    title(strcat('Ampl. fortelor axiale „peak to peak" masurate pe masa tensometrica, avans : ',num2str(y0n(i)),' [mm/min]'));
end
% %Valorile momentelor măsurate  „peak to peak"utilizând masa tensometrică
% %Dispozitiv blocat
 m_bl = [2.3923 3.0803 2.0554 2.4131 2.0534
        2.4067 3.4262 2.0592 2.7264 2.4036
        2.4105 3.0801 2.7362 2.3968 3.0872
        2.7753 3.0852 2.7340 2.7407 2.7401
        2.4189 3.4470 2.4052 2.7428 3.4111];
 Z = m_bl;
 ZI = interp2(X,Y,Z,XI,YI,'spline');
 figure; mesh(XI,YI,ZI);
title('Ampl. momentelor „peak to peak" masurate pe masa tensometrica - aschiere obisnuita');
% %Dispozitiv  neblocat
 m_nebl = [2.7431 2.7372 2.7725 3.0807 2.7399
        2.7588 2.7311 2.7404 3.0733 3.0721
        3.0898 3.0926 2.7535 3.1053 3.0732
        3.4507 3.0828 2.7841 3.1131 3.0941
        2.7579 3.4209 3.0761 3.1098 3.0846];
 Z = m_nebl;
 ZI = interp2(X,Y,Z,XI,YI,'spline');
 figure; mesh(XI,YI,ZI);
title('Ampl. momentelor „peak to peak" masurate pe masa tensometrica - aschiere vibropercutanta');

for i = 1:5
    x  = [463 559 619 686 771];
    xi = (463:1:771);
    yi = interp1(x,m_bl(i,:),xi,'spline');
    figure; plot(xi,yi,'r');
    yi = interp1(x,m_nebl(i,:),xi,'spline');
```

```
    hold on; plot(xi,yi,'k');
    xlabel('Turatie [rot/min]')
    ylabel('[N*m]')
    title(strcat('Ampl. momentelor „peak to peak" masurate pe masa tensometrica. avans
',num2str(y0n(i)),' [mm/min]'));
end

% % %Valorile vitezelor unghiulare măsurate „peak to peak"utilizând traductorul TIRO
y0n = [1 2 4 6 8]';
x0s = [463 559 619]';
[X,Y] = meshgrid(x0s,y0n);
[XI,YI] = meshgrid(463:1:619,1:0.1:8);
% % %Dispozitiv blocat
 vit_bl = [47.27 55.08 65.55
        47.27 55.08 65.55
        47.27 55.08 65.55
        47.27 55.08 65.55
        47.27 55.08 65.55];
 Z = vit_bl;
ZI = interp2(X,Y,Z,XI,YI,'linear');
figure; mesh(XI,YI,ZI);
title('Viteze unghiulare „peak to peak" masurate cu traductor TIRO - aschiere obisnuita');
%2.%Dispozitiv  neblocat
vit_nebl = [37.2087    36.2680 41.9354
         35.9827        33.5103 39.1413
         34.0322        33.0091 36.3894
         32.4382        28.0499 40.6084
         32.3589        18.3606 25.7021];
Z = vit_nebl;
ZI = interp2(X,Y,Z,XI,YI,'linear');
figure; mesh(XI,YI,abs(ZI));
title('Viteze unghiulare „peak to peak" masurate cu traductor TIRO - aschiere vibropercutanta');

for i = 1:5
    x  = [463 559 619];
    xi = (463:1:619);
    yi = interp1(x,vit_bl(i,:),xi,'linear');
    figure; plot(xi,yi,'r');
    yi = interp1(x,vit_nebl(i,:),xi,'linear');
    hold on; plot(xi,yi,'k');
    xlabel('Turatie [rot/min]')
    ylabel('[rad/s]')
    title(strcat('Viteze unghiulare „peak to peak" masurate cu traductor TIRO, avans
',num2str(y0n(i)),' [mm/min]'));
end
```

Anexa 42.

Diagrame ale amplitudinilor accelerațiilor, forțelor, momentelor și vitezelor unghiulare în funcție de turație și viteza de avans măsurate „peak to peak" trasate cu programul „diag2.m"

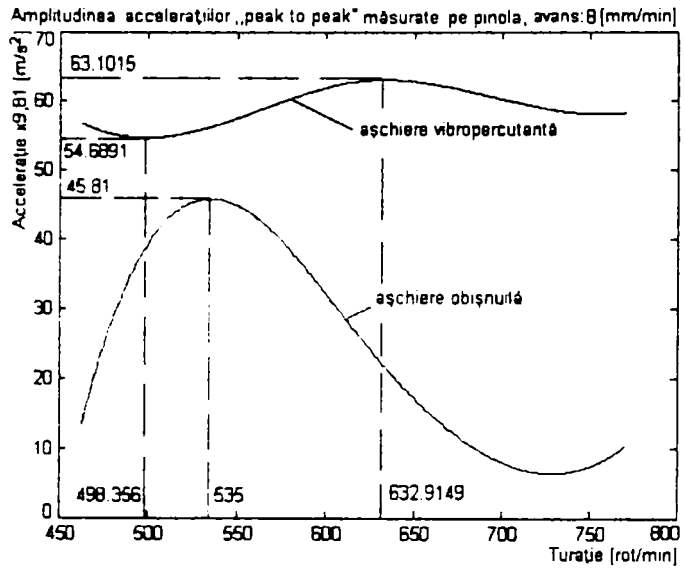


Fig. A.42.1.

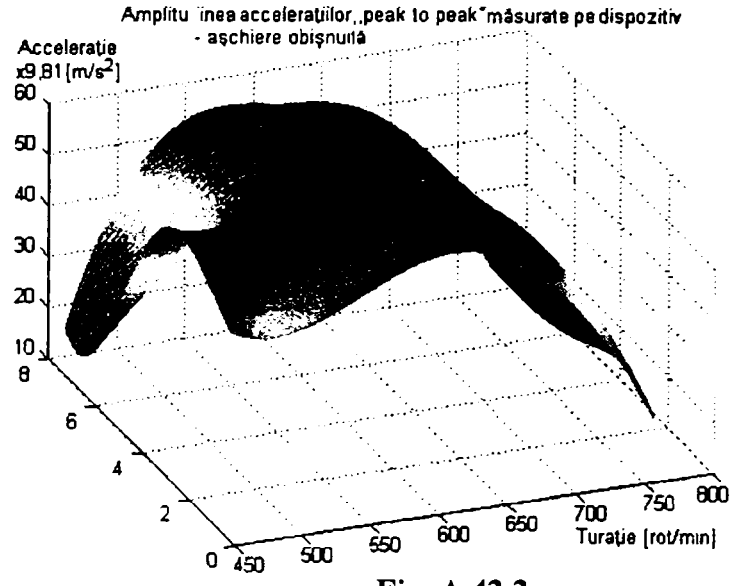


Fig. A.42.2.

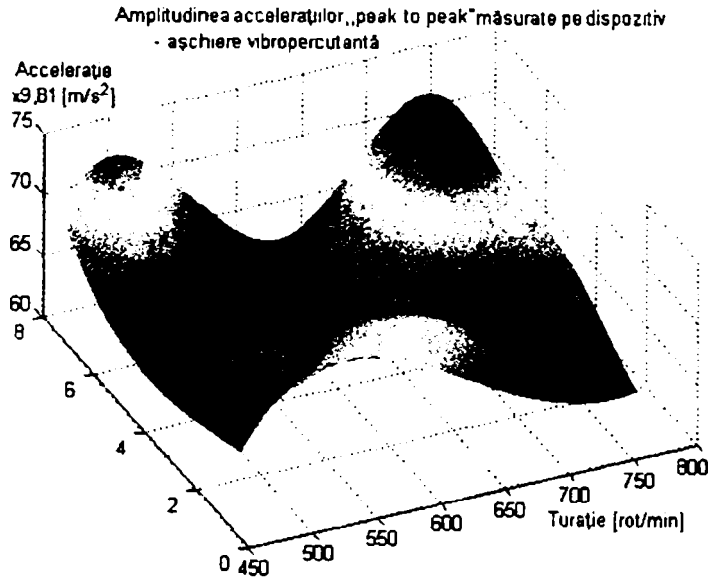


Fig. A.42.3.

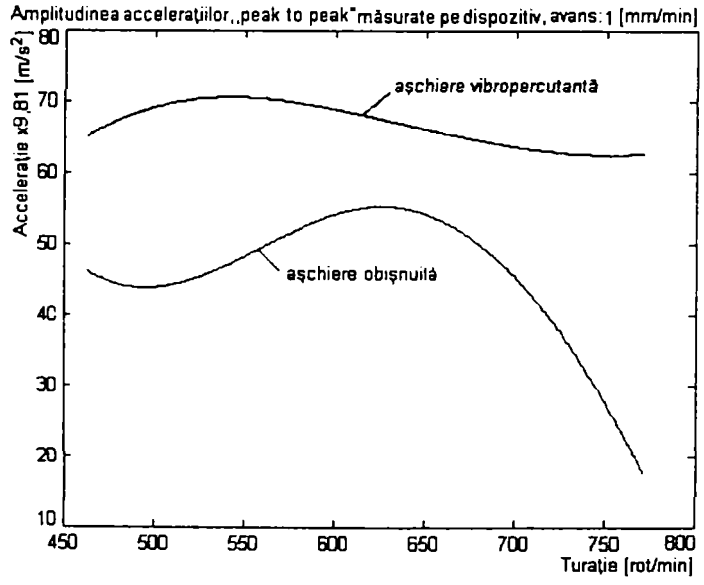


Fig. A.42.4.

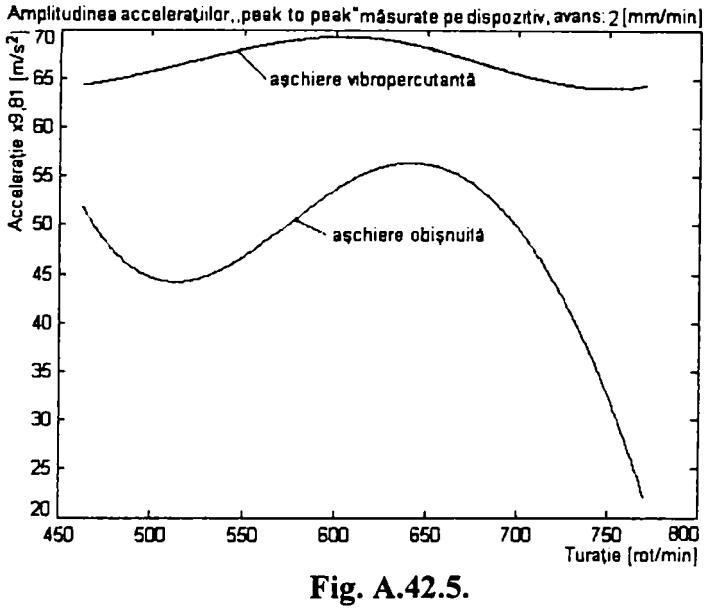


Fig. A.42.5.

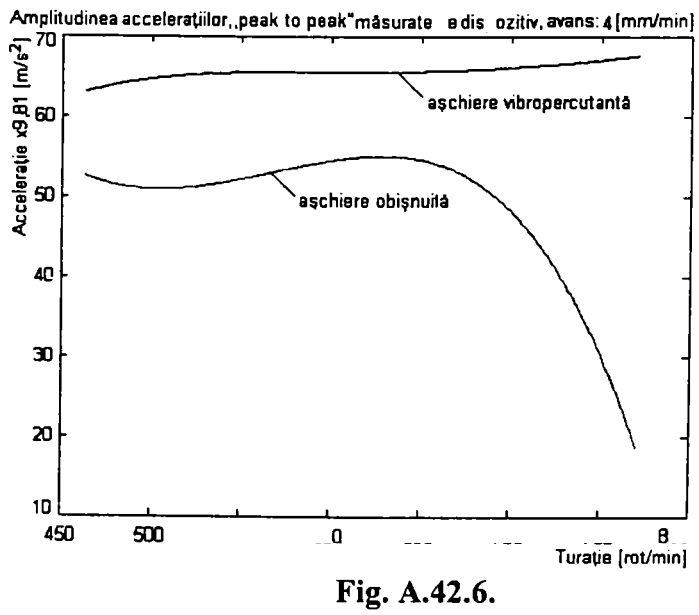


Fig. A.42.6.

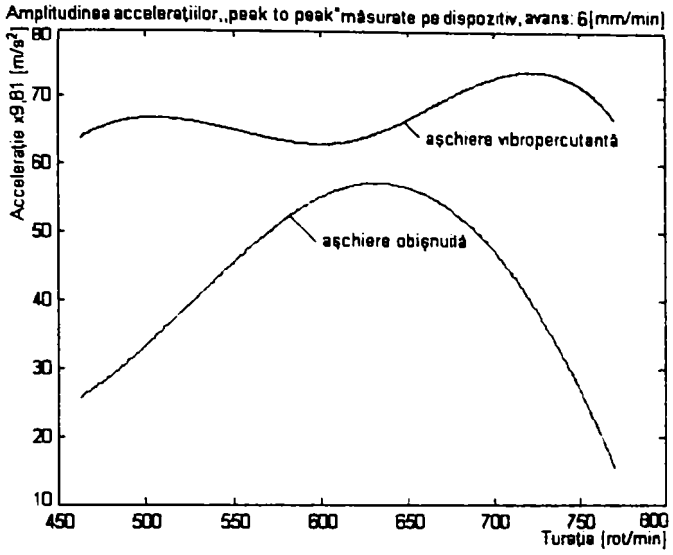


Fig. A.42.7.

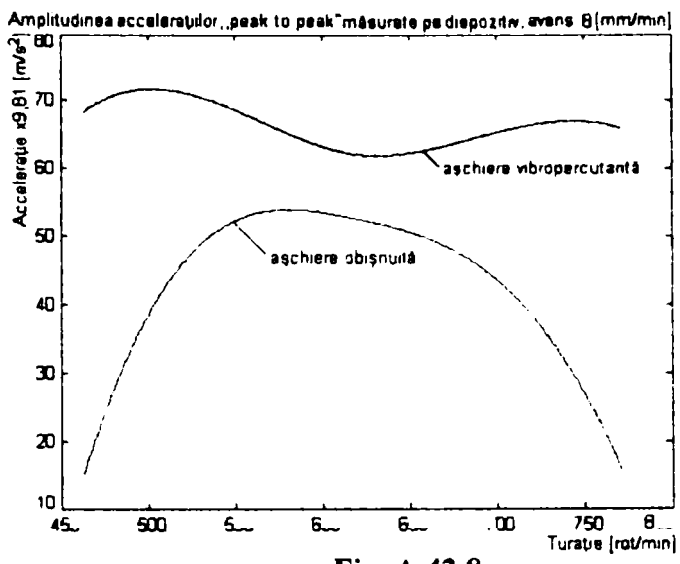


Fig. A.42.8.

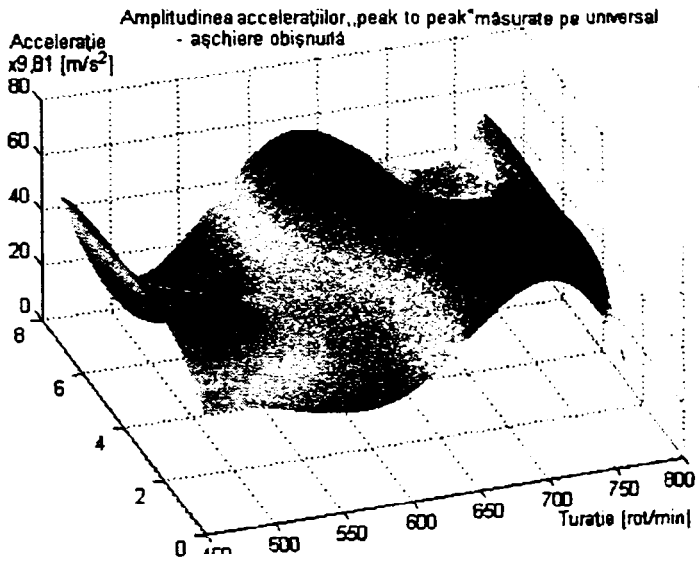


Fig. A.42.9.

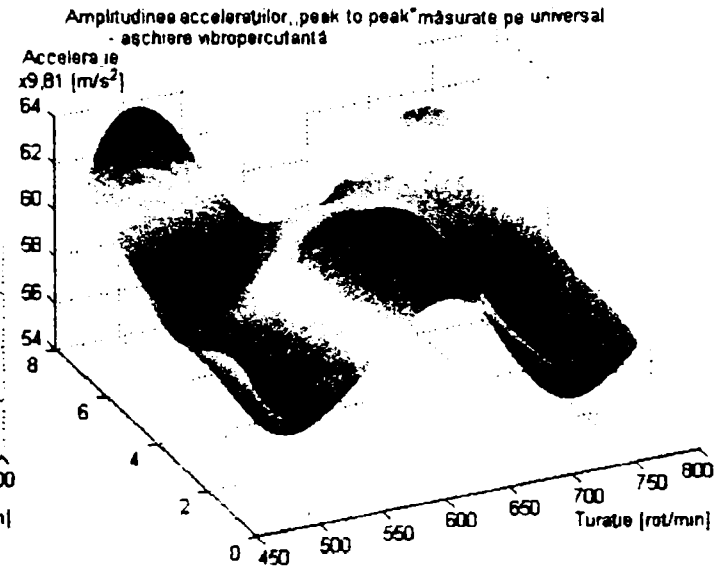


Fig. A.42.10.

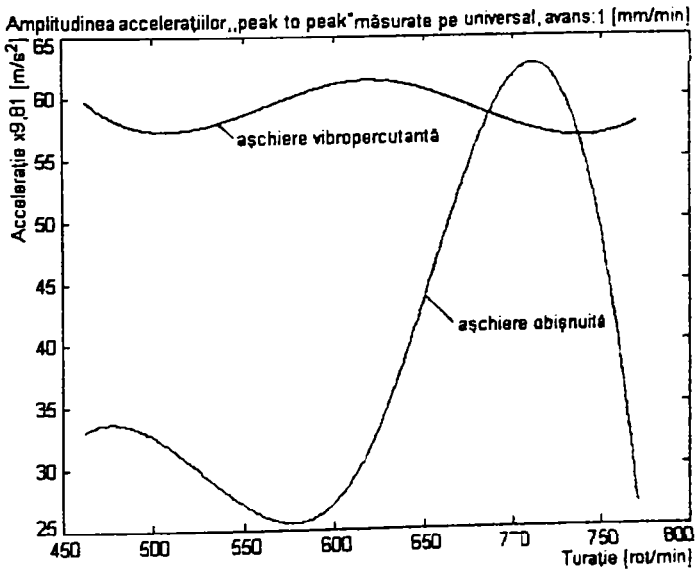


Fig. A.42.11.

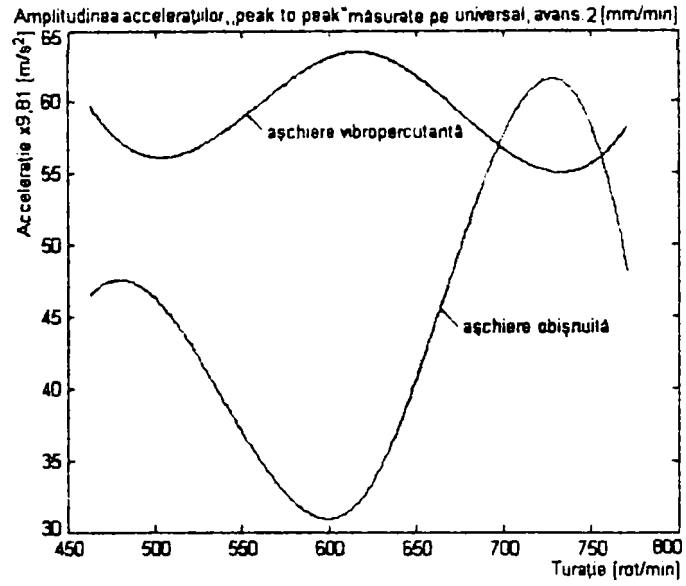


Fig. A.42.12.

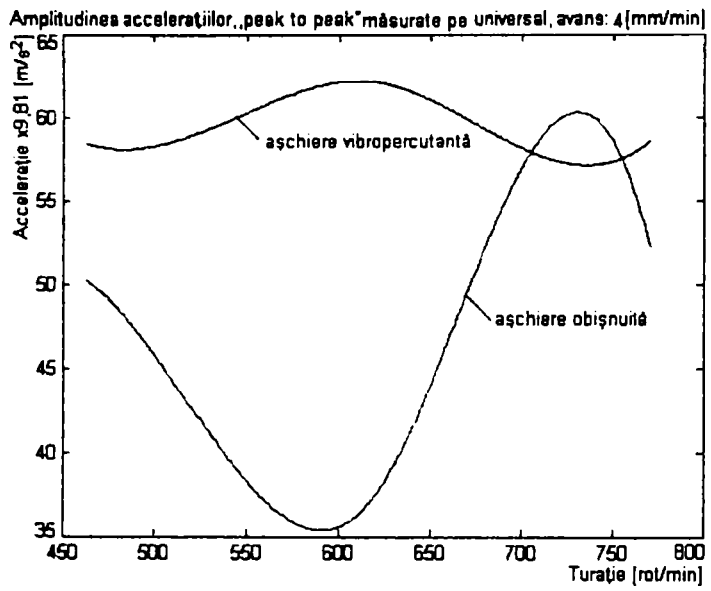


Fig. A.42.13.

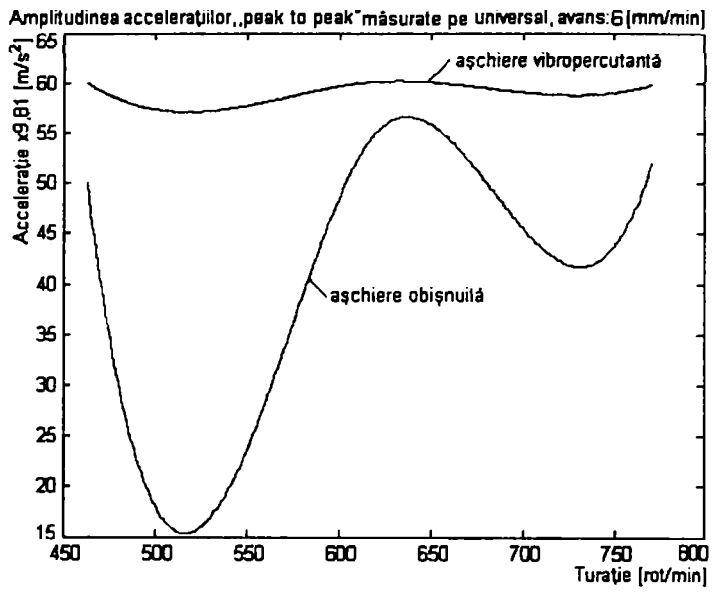


Fig. A.42.14.

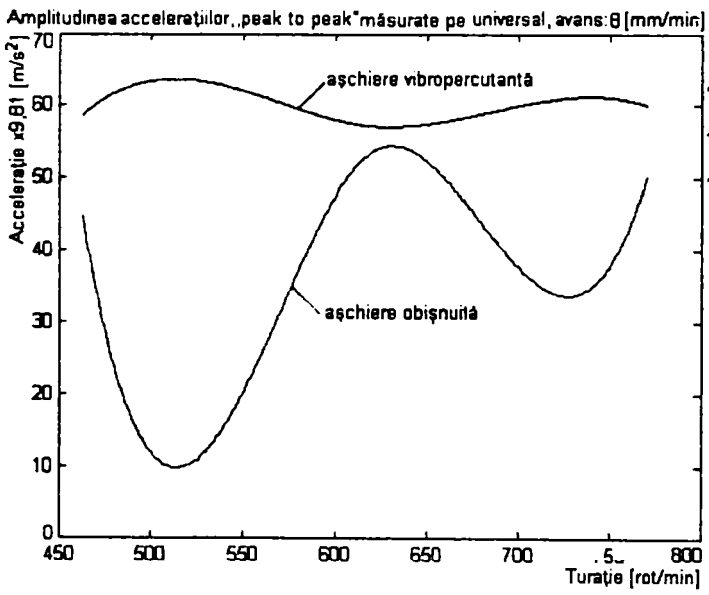


Fig. A.42.15.

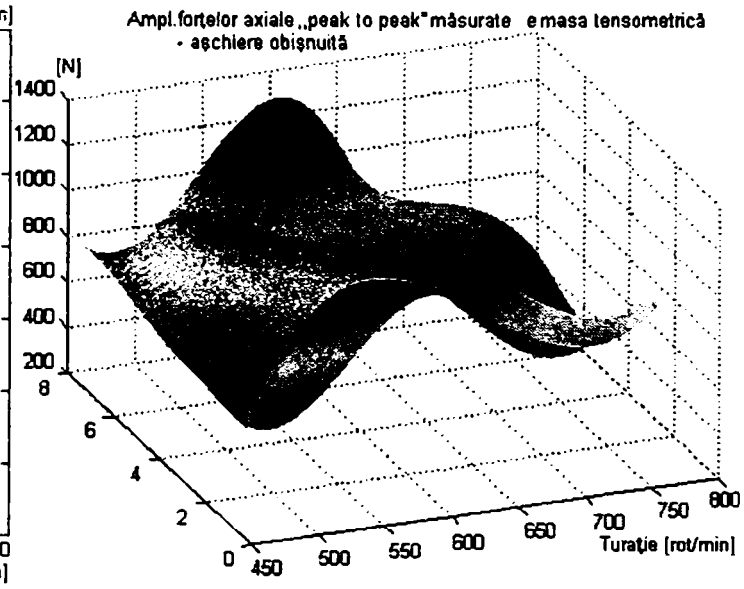


Fig. A.42.16.

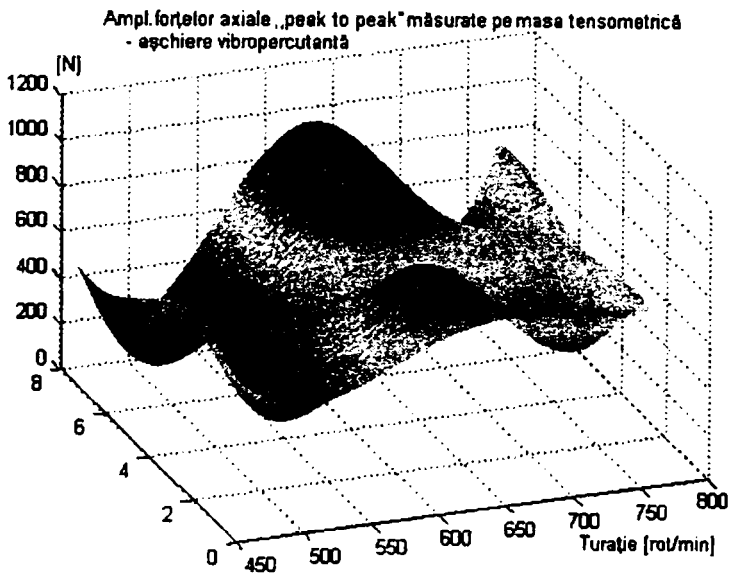


Fig. A.42.17.

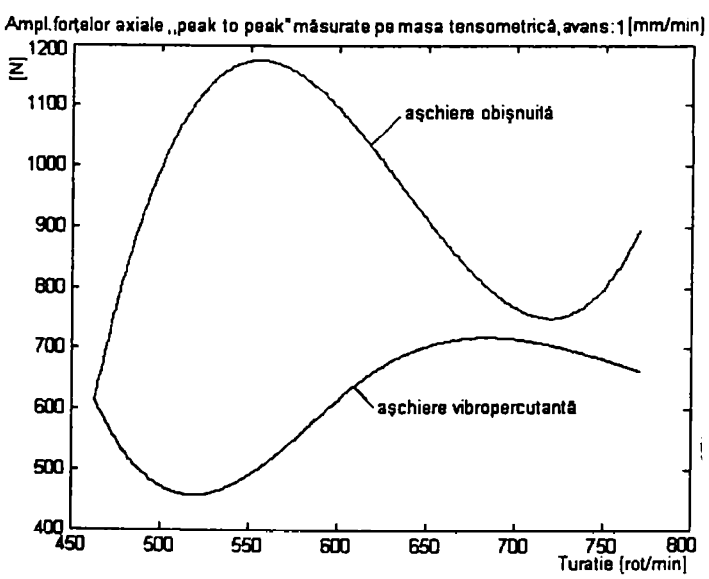


Fig. A.42.18.

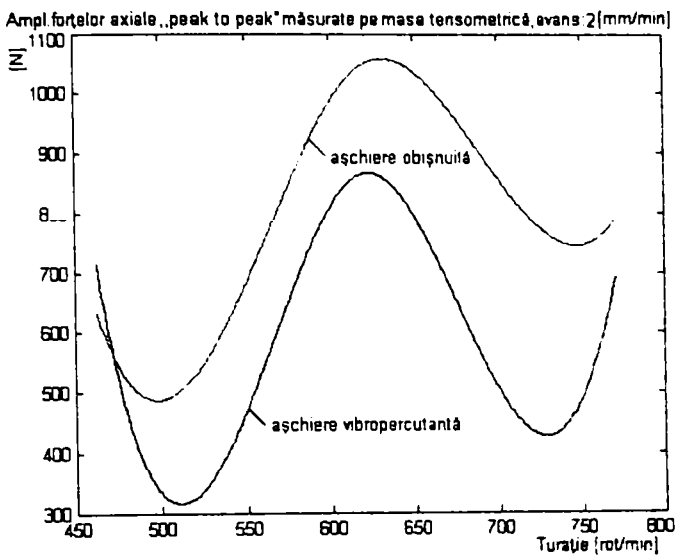


Fig. A.42.19.

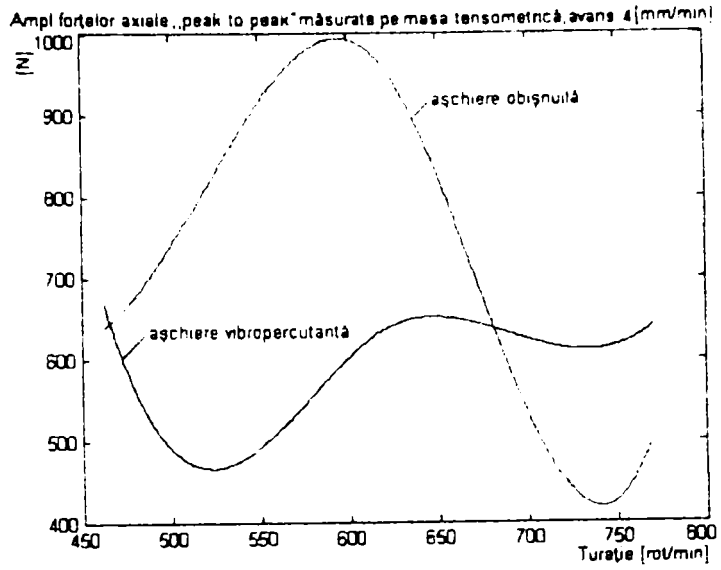


Fig. A.42.20.

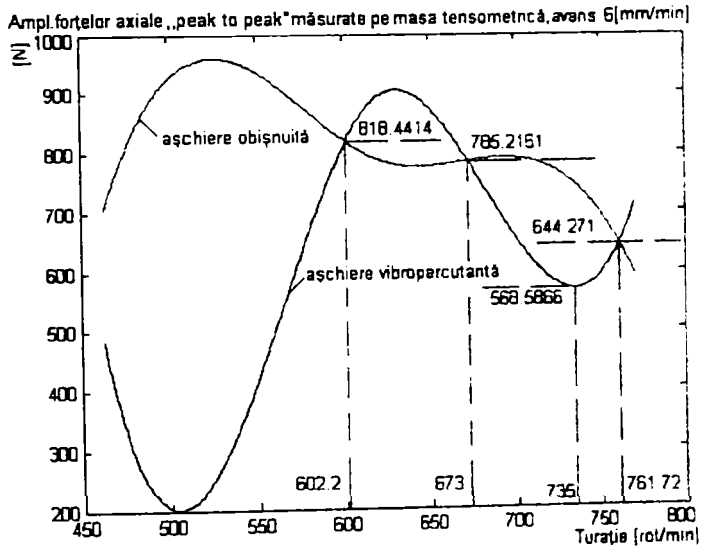


Fig. A.42.21.

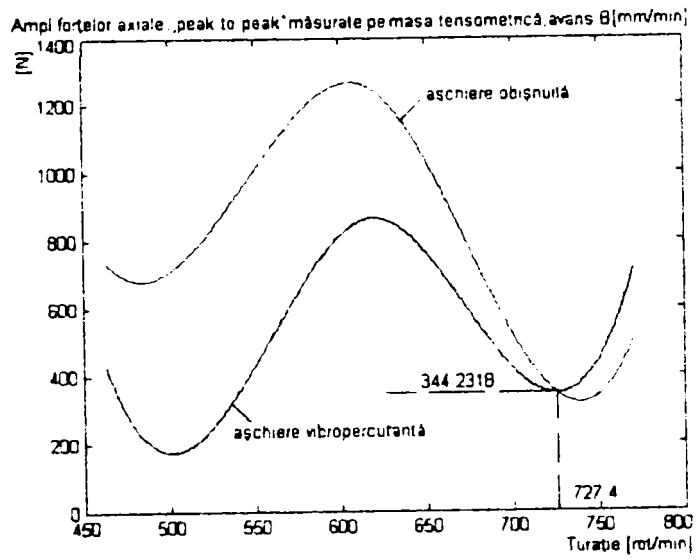


Fig. A.42.22.

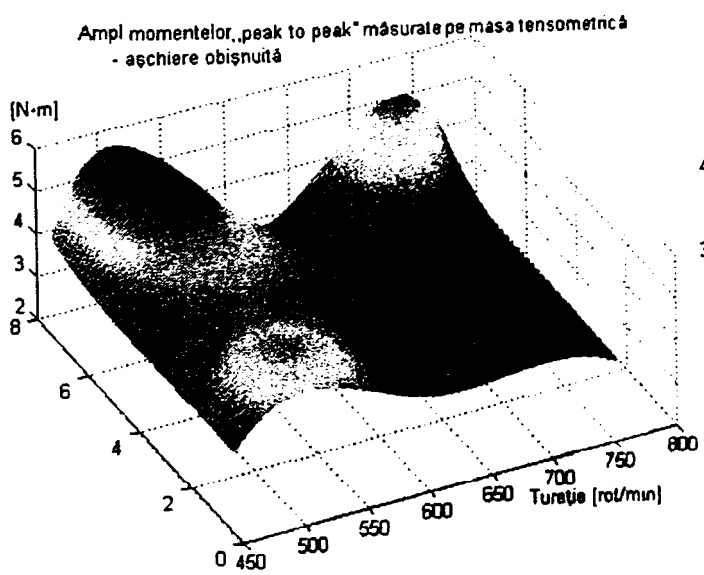


Fig. A.42.23.

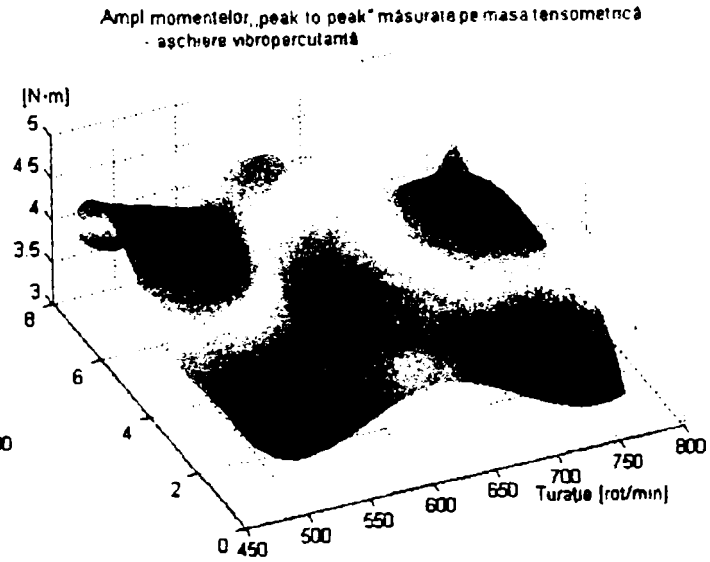


Fig. A.42.24.

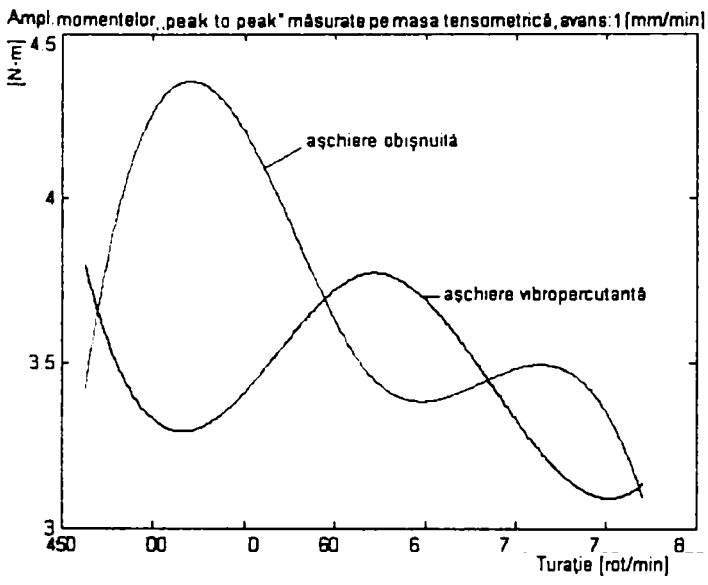


Fig. A.42.25.

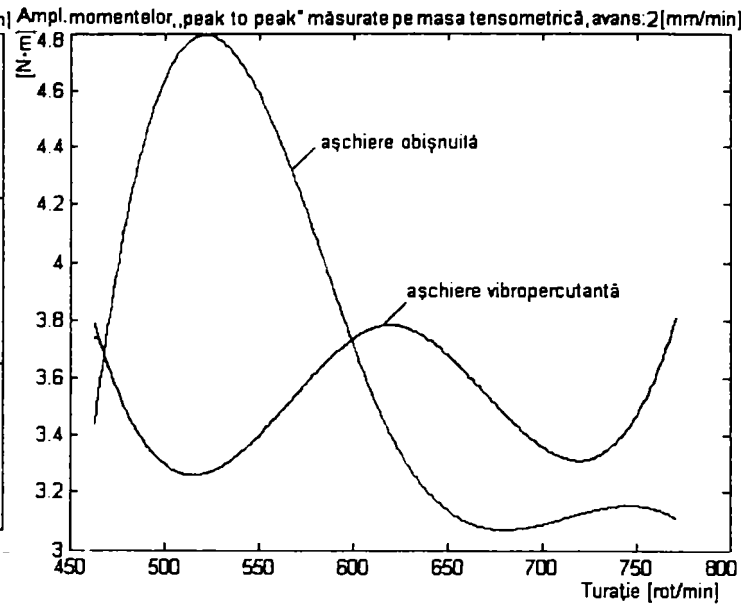


Fig. A.42.26.

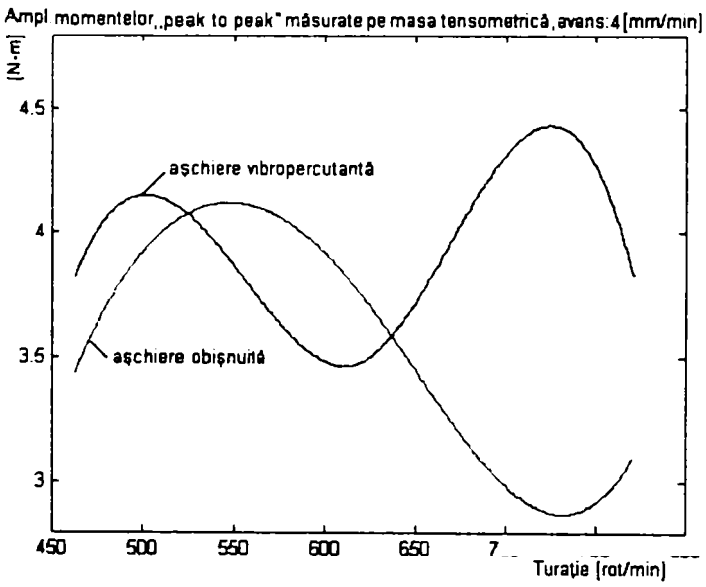


Fig. A.42.27.

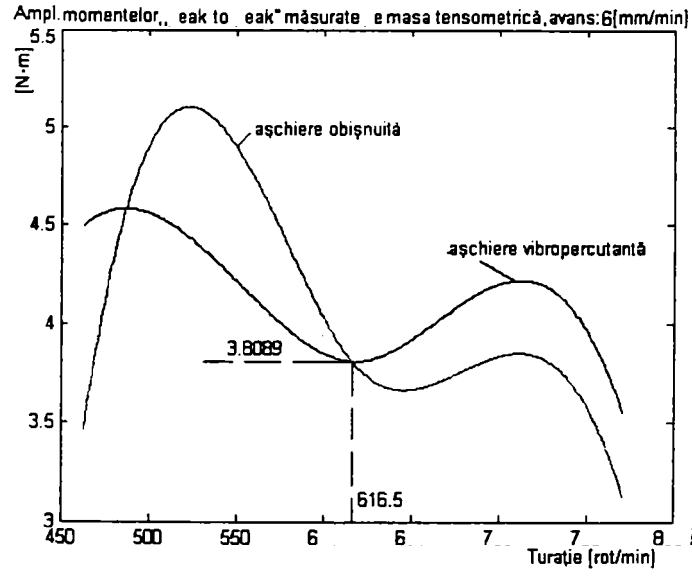


Fig. A.42.28.

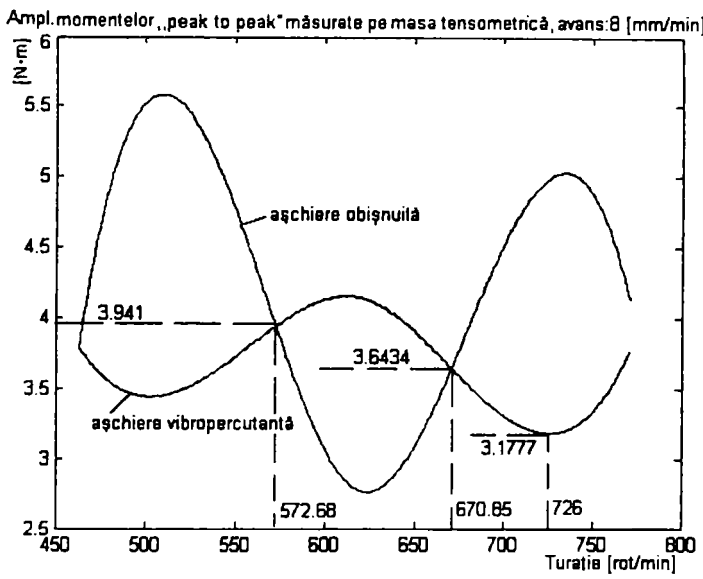


Fig. A.42.29.

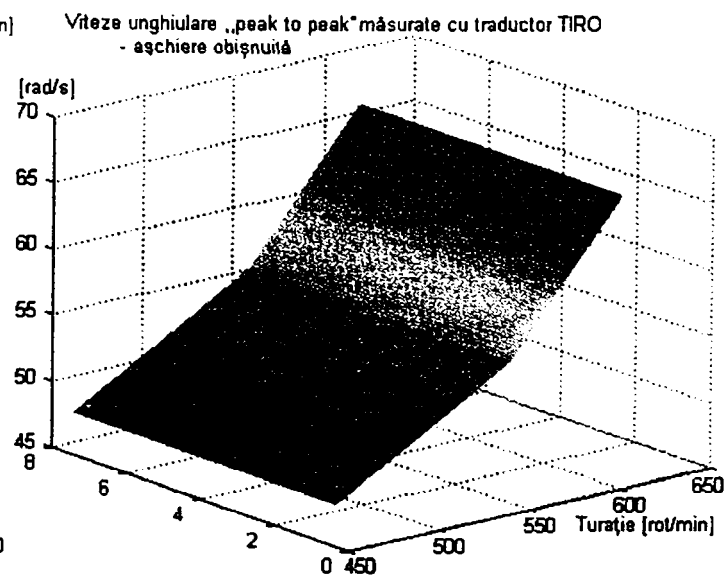


Fig. A.42.30.

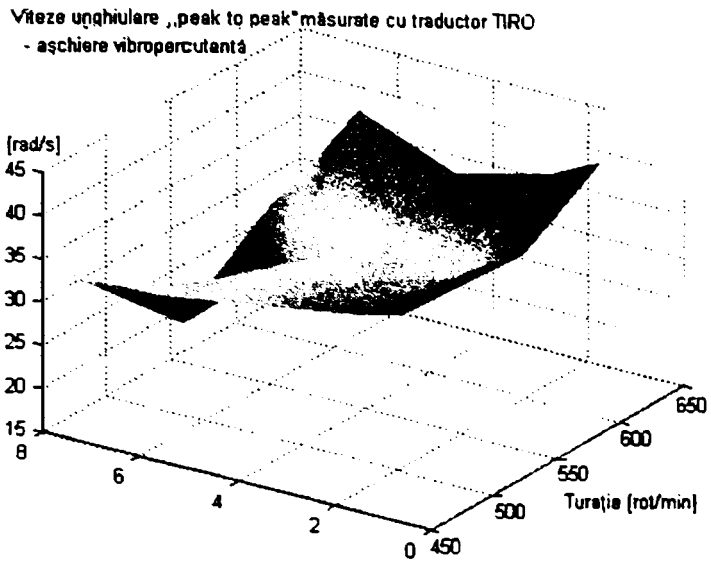


Fig. A.42.31.

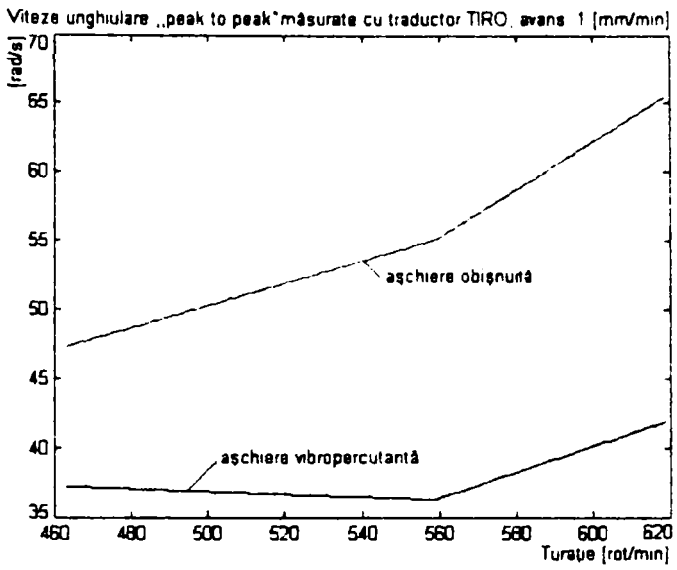


Fig. A.42.32.

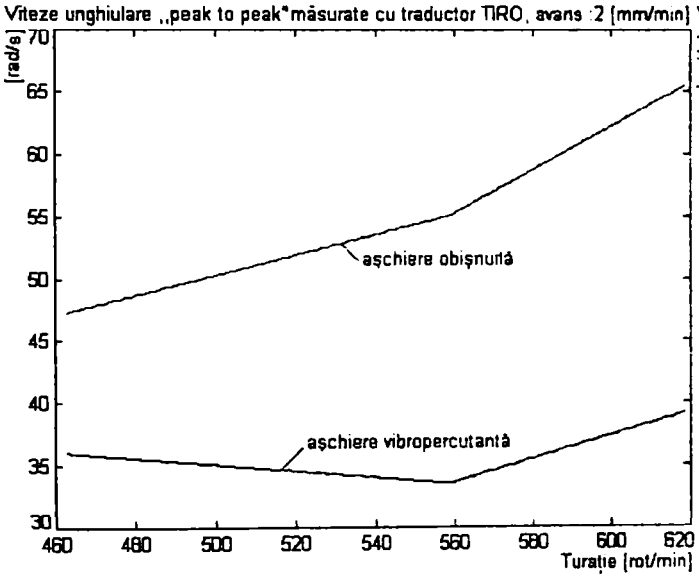


Fig. A.42.33.

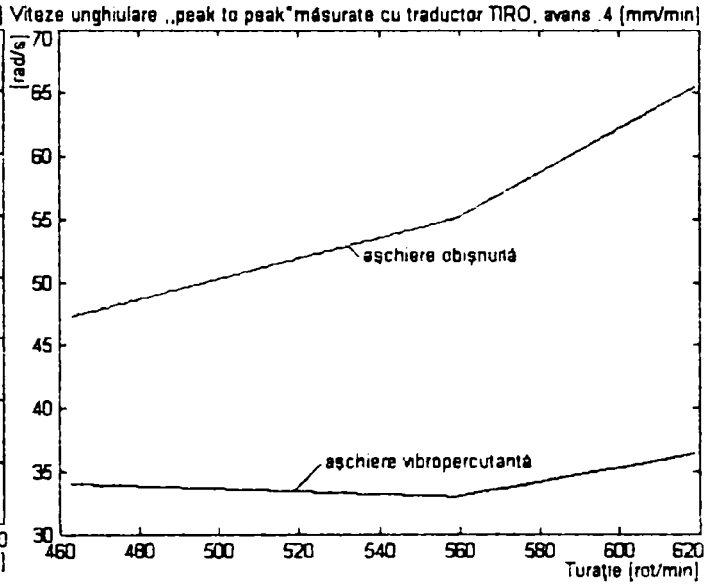


Fig. A.42.34.

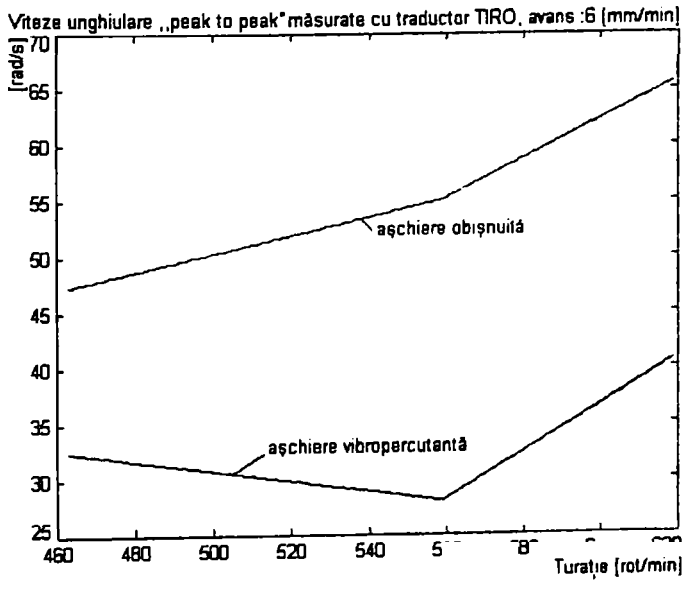


Fig. A.42.35.

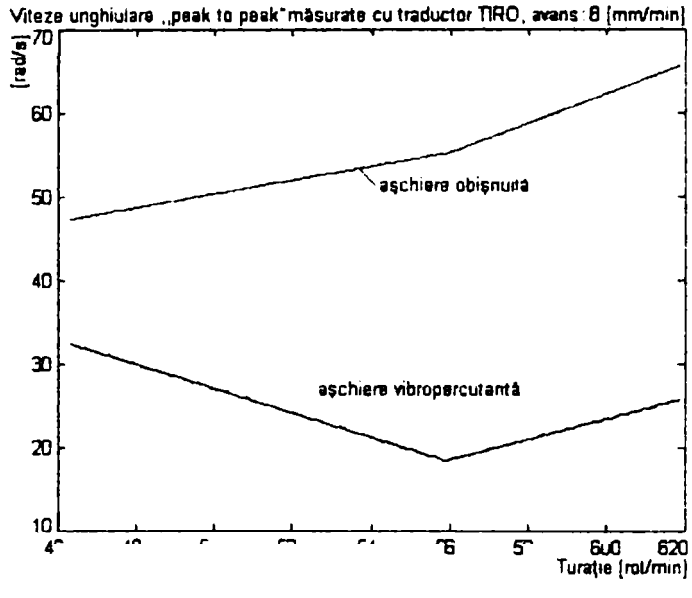


Fig. A.42.36.

Anexa 43.

Diagrame RMS şi spectrelor frecvenţelor în regim de aşchiere obişnuit şi regim vibropercutant

1.Măsurate pe dispozitivul vibropercutant

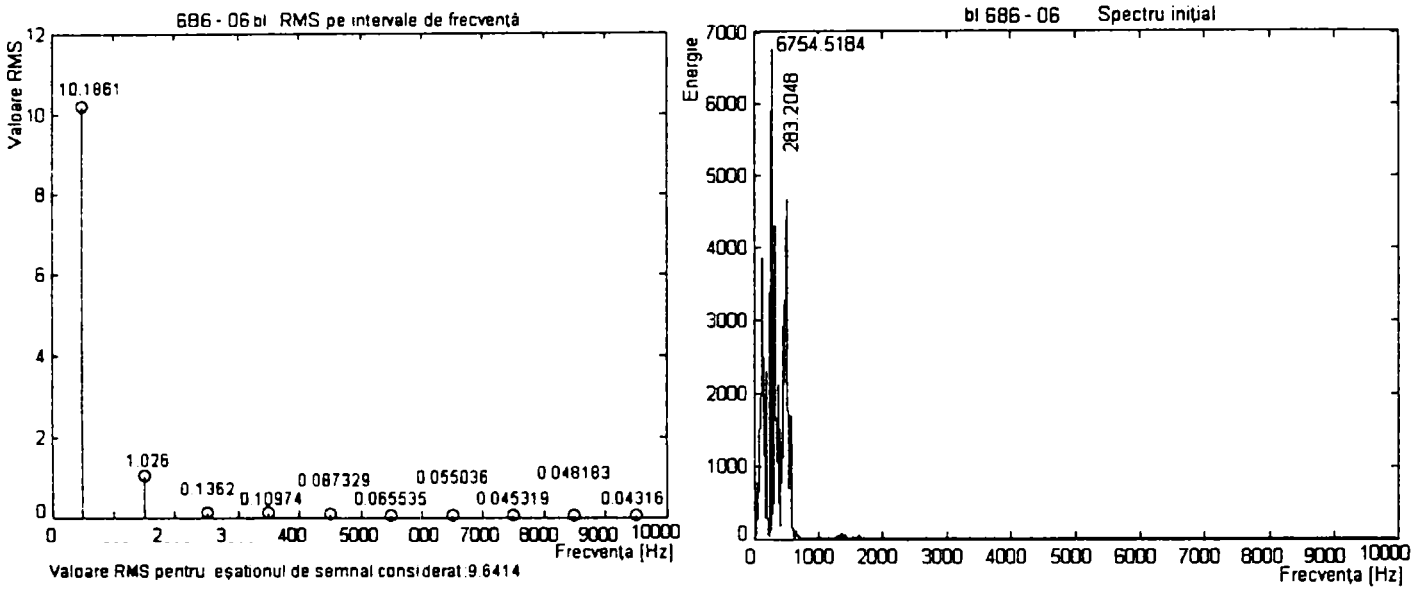


Fig. A.43.1.1.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=686 şi v_s=6.

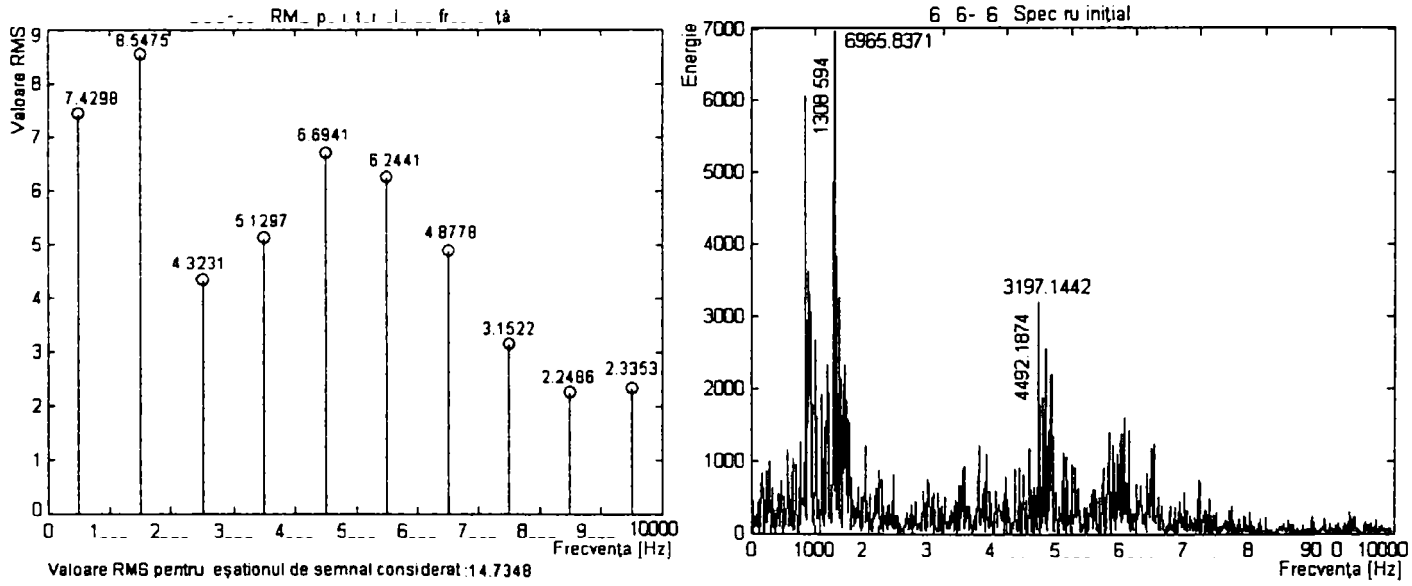


Fig. A.43.1.1.2. Diagrama RMS şi spectrul de frecvenţă la găurirea vibropercutantă: n=686 şi v_s=6.

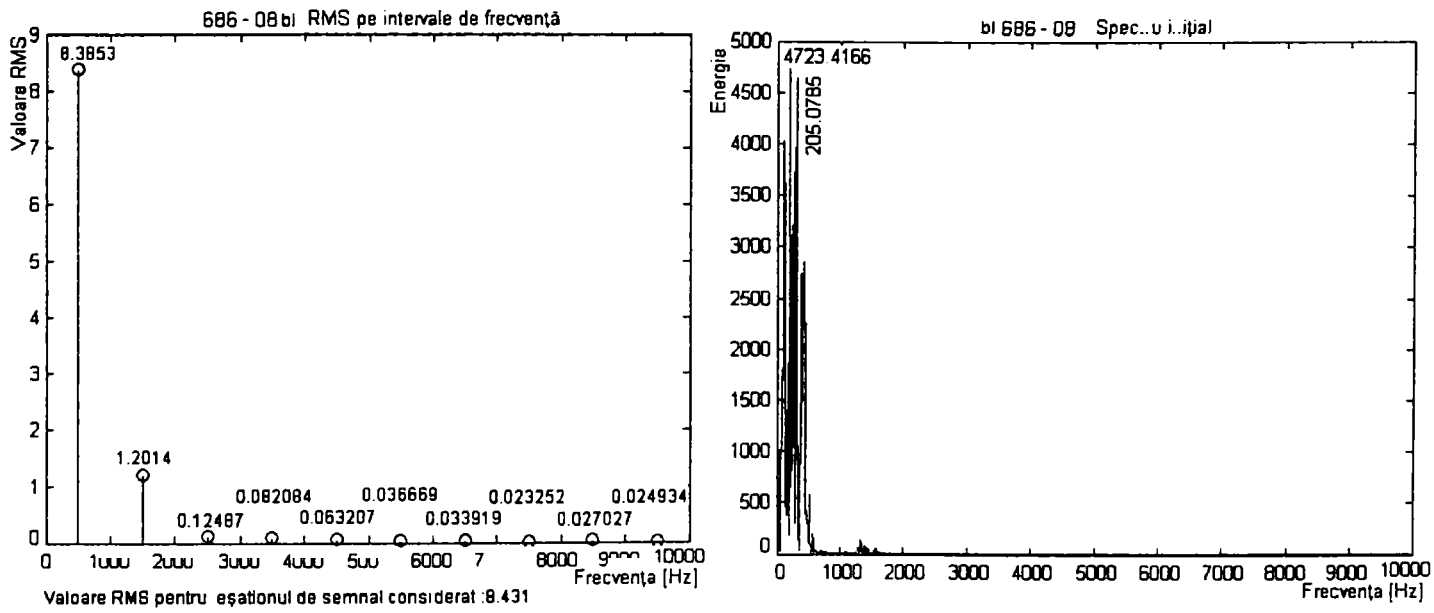


Fig. A.43.1.2.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=686 şi v_s=8

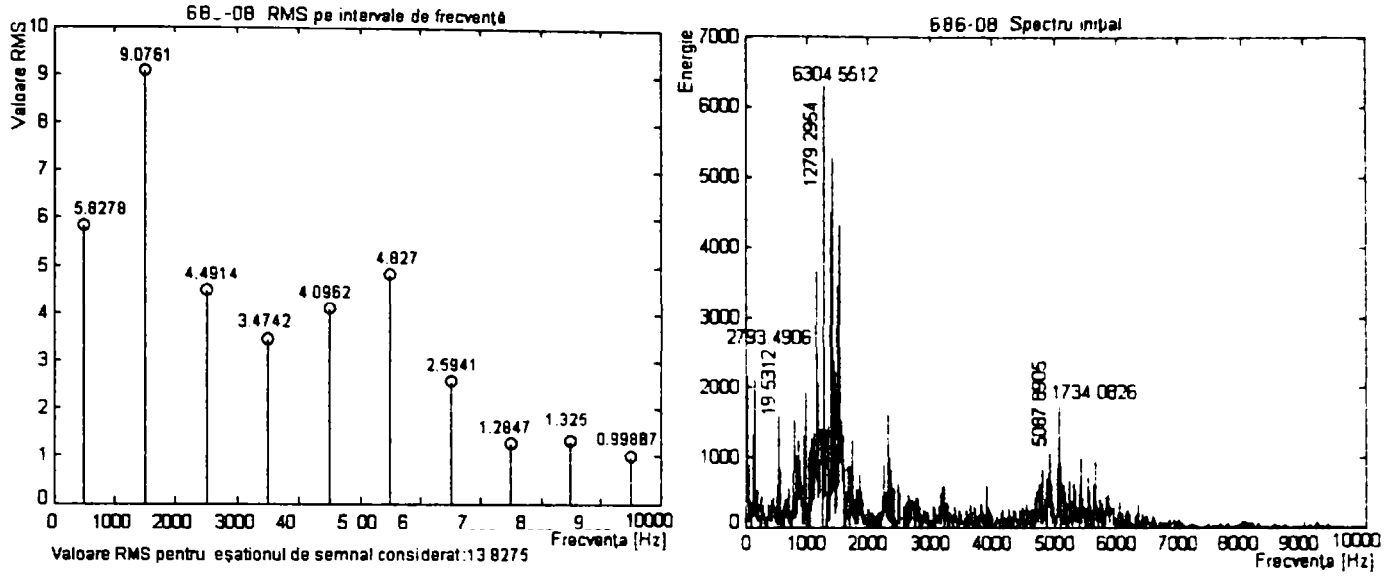


Fig. A.43.1.2.2. Diagrama RMS şi spectrul de frecvență la găurirea vibropercutantă: n=686 şi v_s=8.

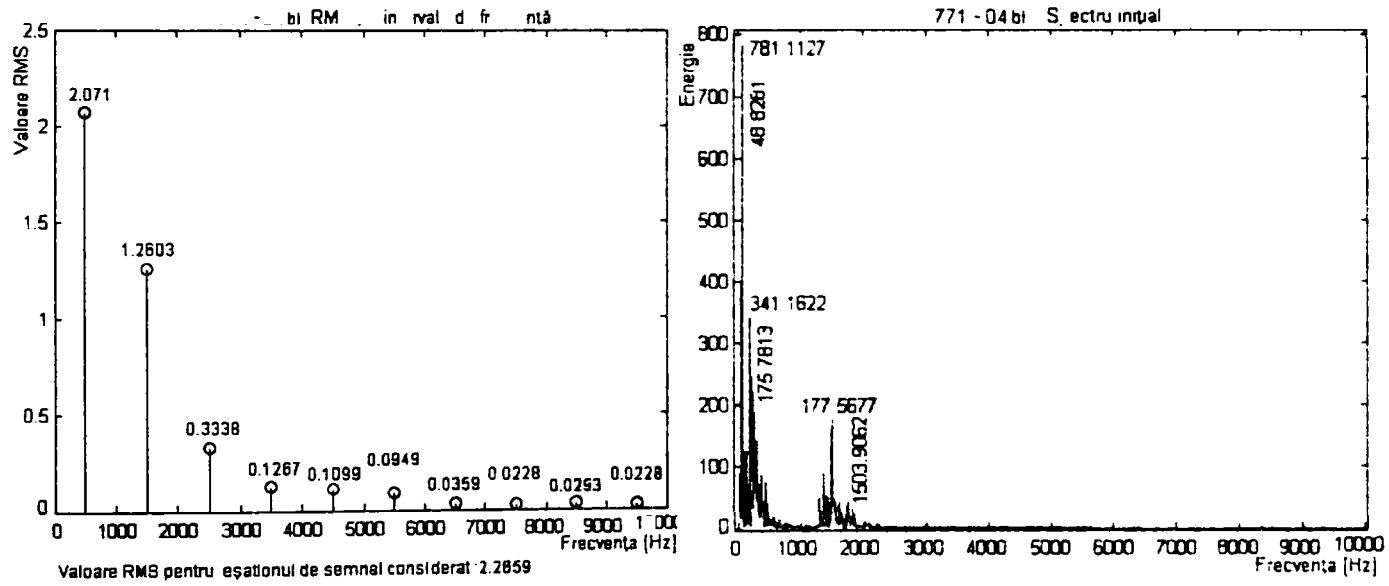


Fig. A.43.1.3.1. Diagrama RMS şi spectrul de frecvență la găurirea obișnuită: n=771 şi v_s=4

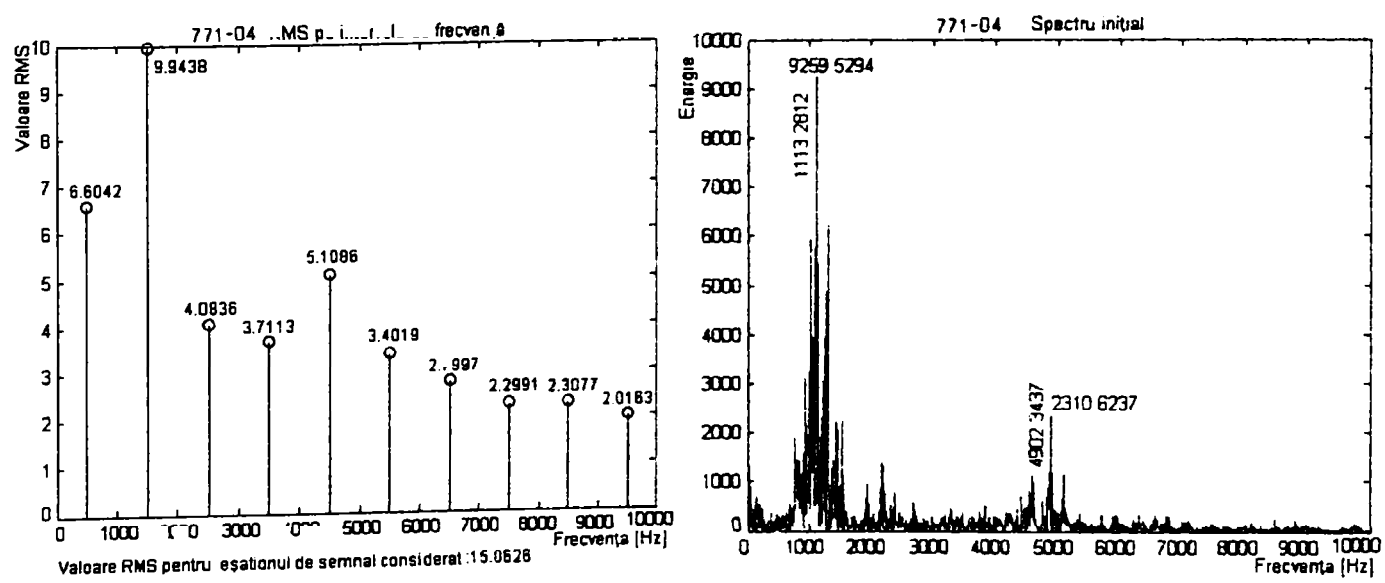


Fig. A.43.1.3.2. Diagrama RMS şi spectrul de frecvență la găurirea vibropercutantă: n=771 şi v_s=4.

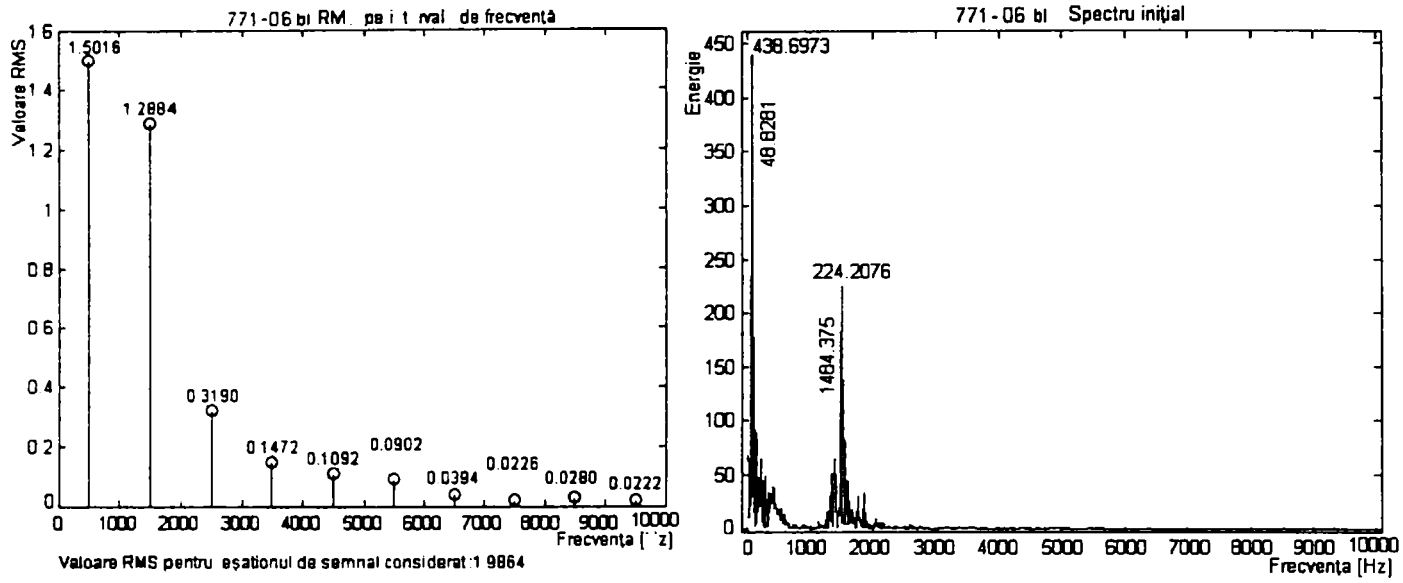


Fig. A.43.1.4.1. Diagrama RMS şi spectrul de frecvență la găurirea obişnuită: n=771 şi v_s=6

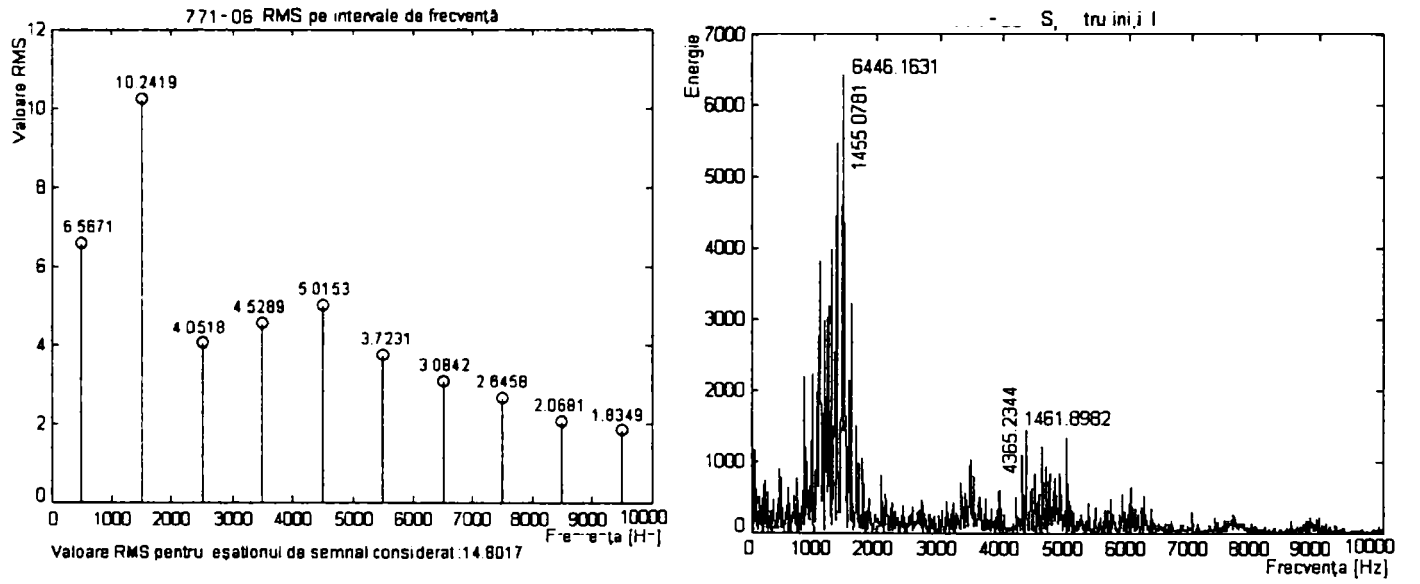


Fig. A.43.1.4.2. Diagrama RMS şi spectrul de frecvență la găurirea vibropercutantă: n=771 şi v_s=6.

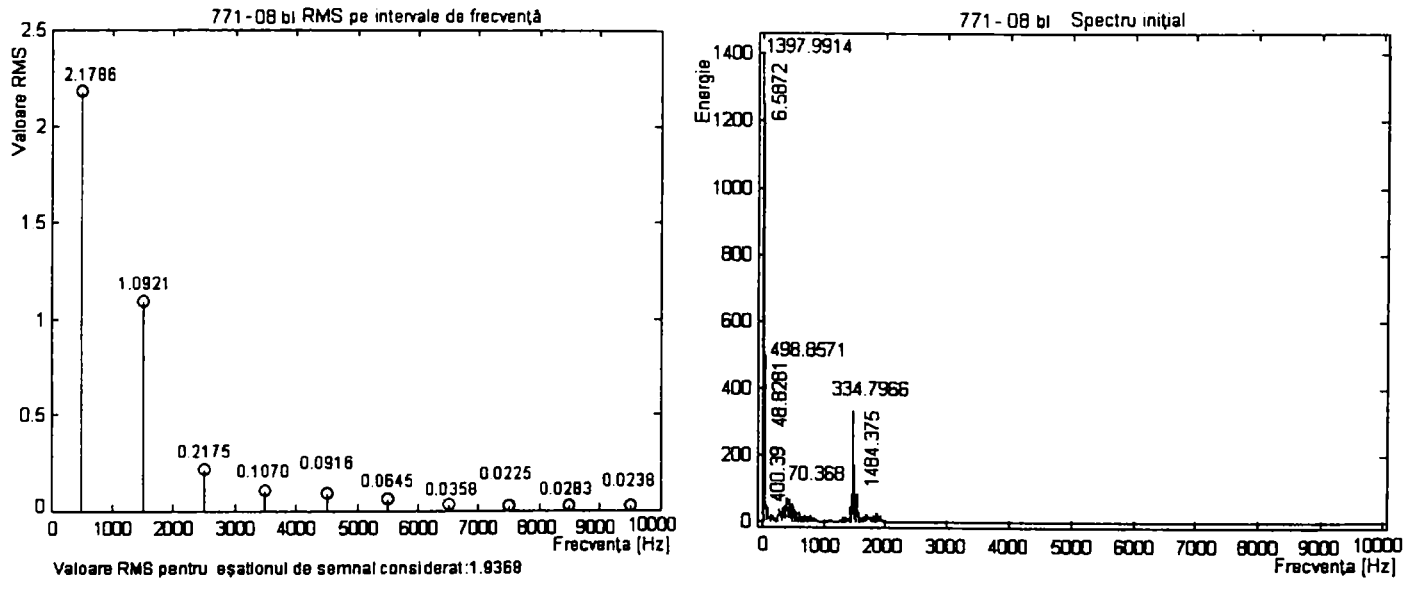


Fig. A.43.1.5.1. Diagrama RMS şi spectrul de frecvență la găurirea obişnuită: n=771 şi v_s=8

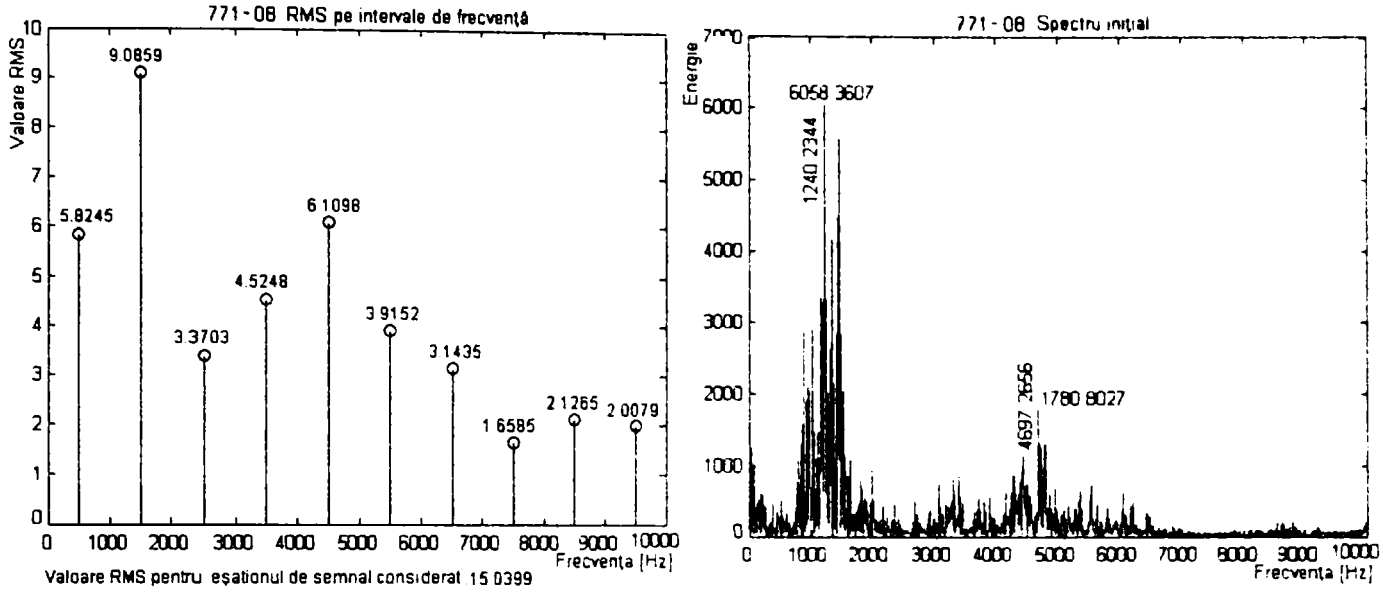


Fig. A.43.1.5.2. Diagrama RMS şi spectrul de frecvență la găurirea vibropercutantă: n=771 şi v_s=8.

2.Măsurate pe pinola maşinii-unelte CP20UO

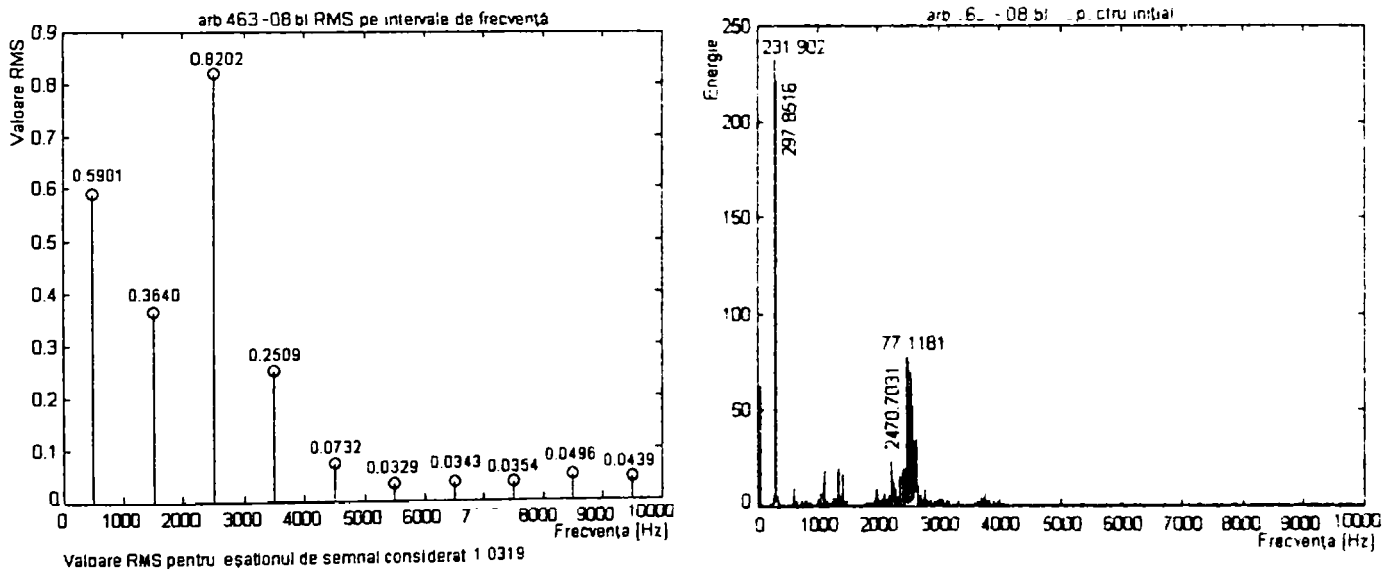


Fig. A.43.2.1.1. Diagrama RMS şi spectrul de frecvență la găurirea obişnuită: n=463 şi v_s=8

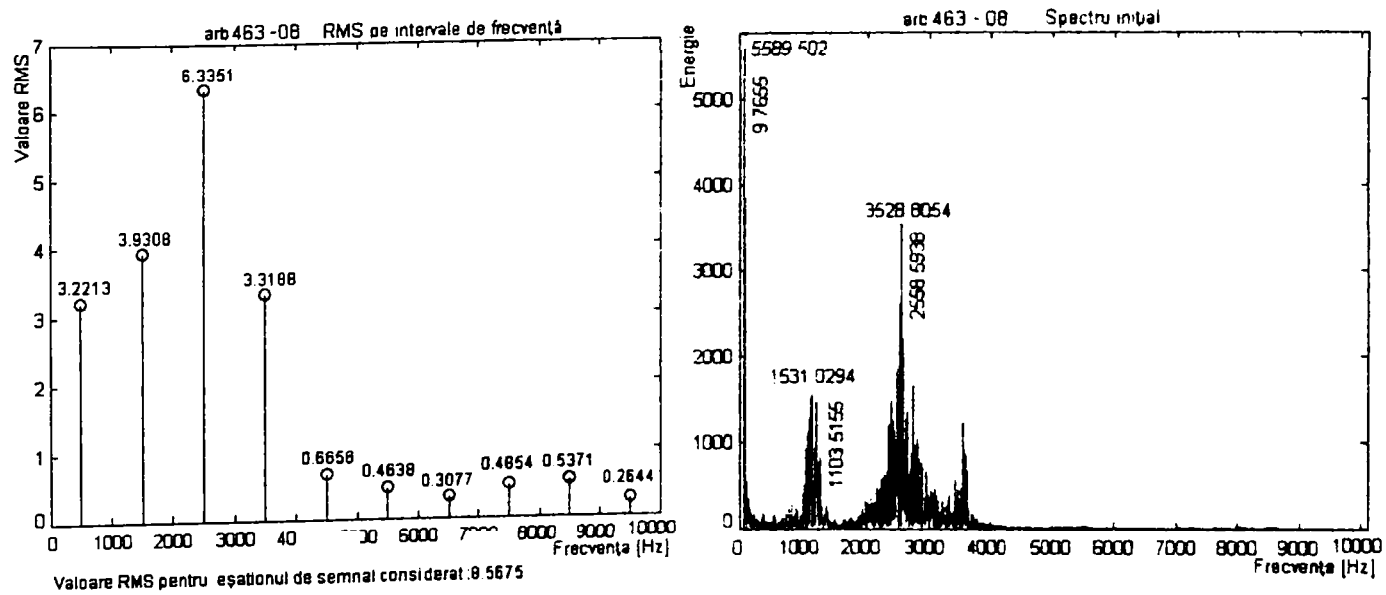


Fig. A.43.2.1.2. Diagrama RMS şi spectrul de frecvență la găurirea vibropercutantă: n=463 şi v_s=8.

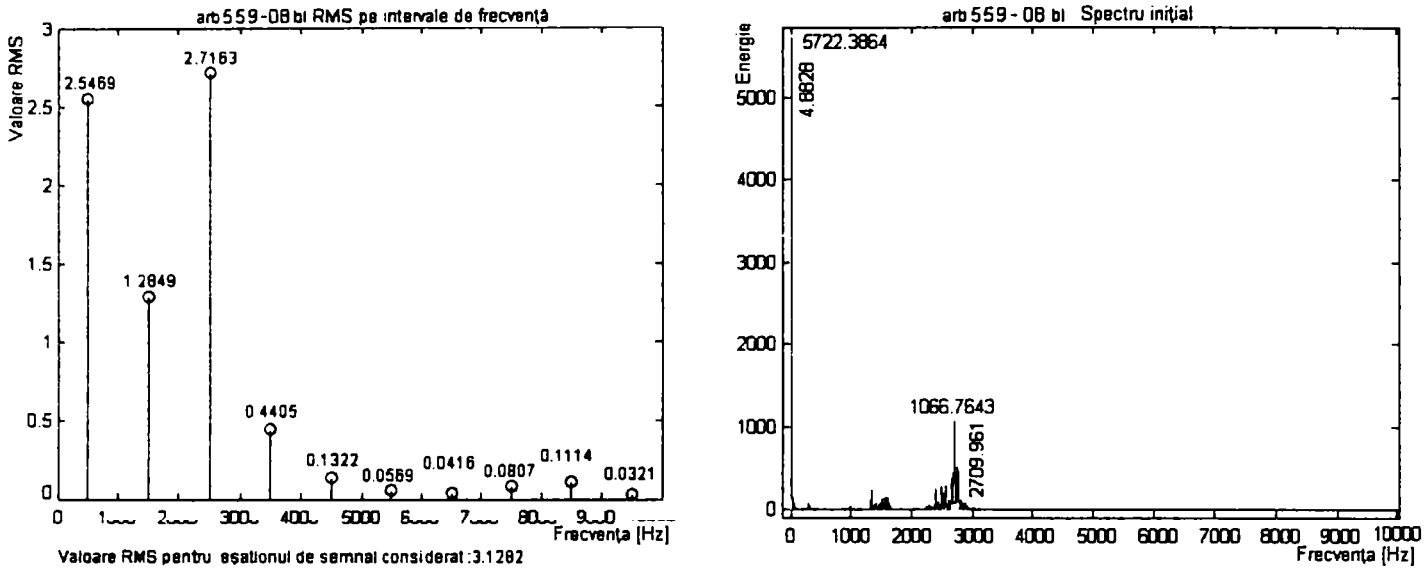


Fig. A.43.2.2.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=559 şi v_s=8

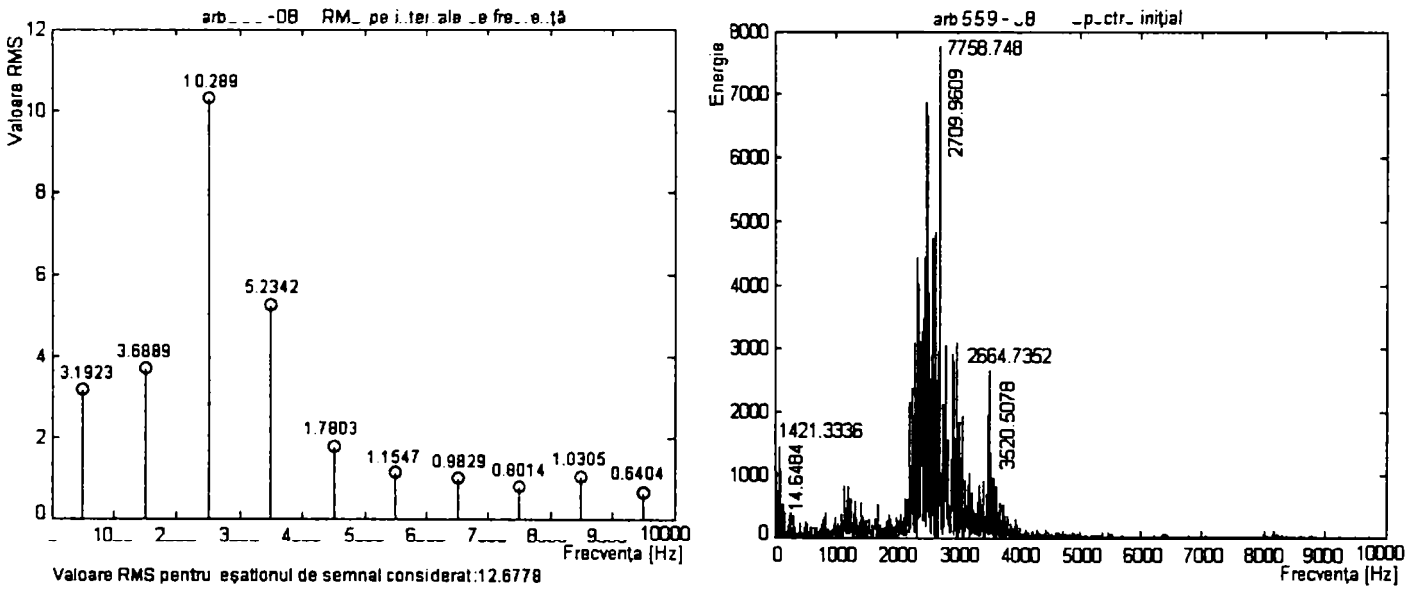


Fig. A.43.2.2.2. Diagrama RMS şi spectrul de frecvenţă la găurirea vibropercutantă: n=559 şi v_s=8.

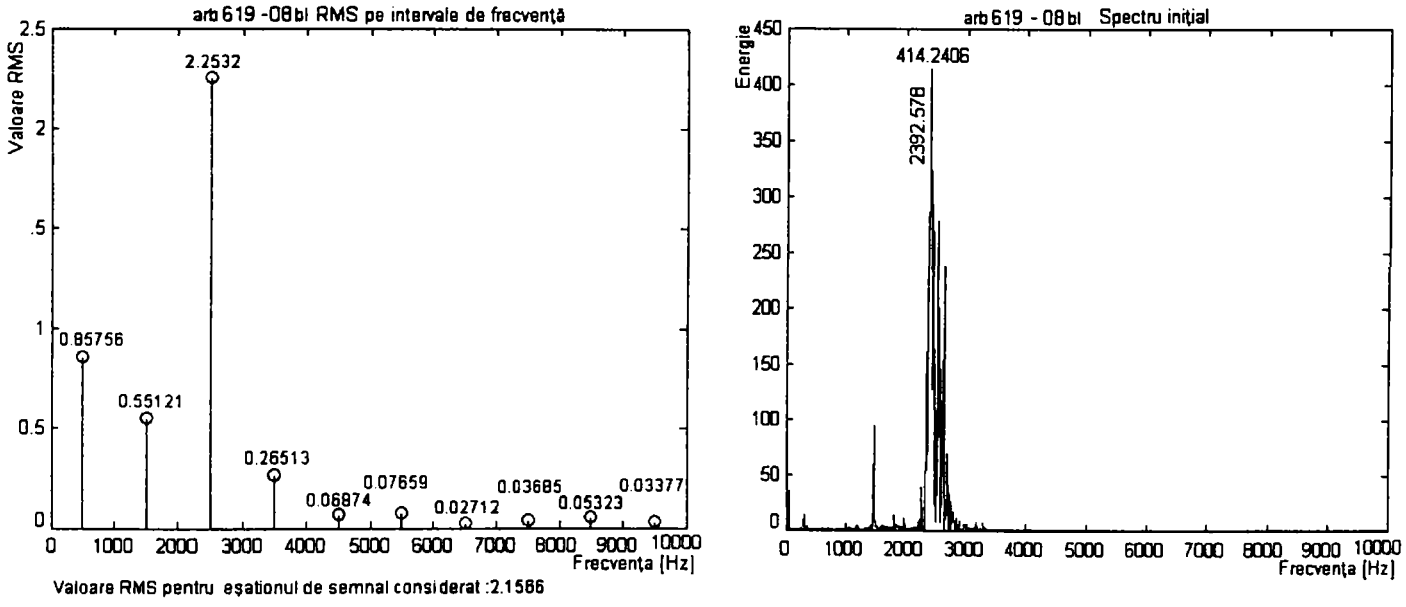


Fig. A.43.2.3.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=619 şi v_s=8

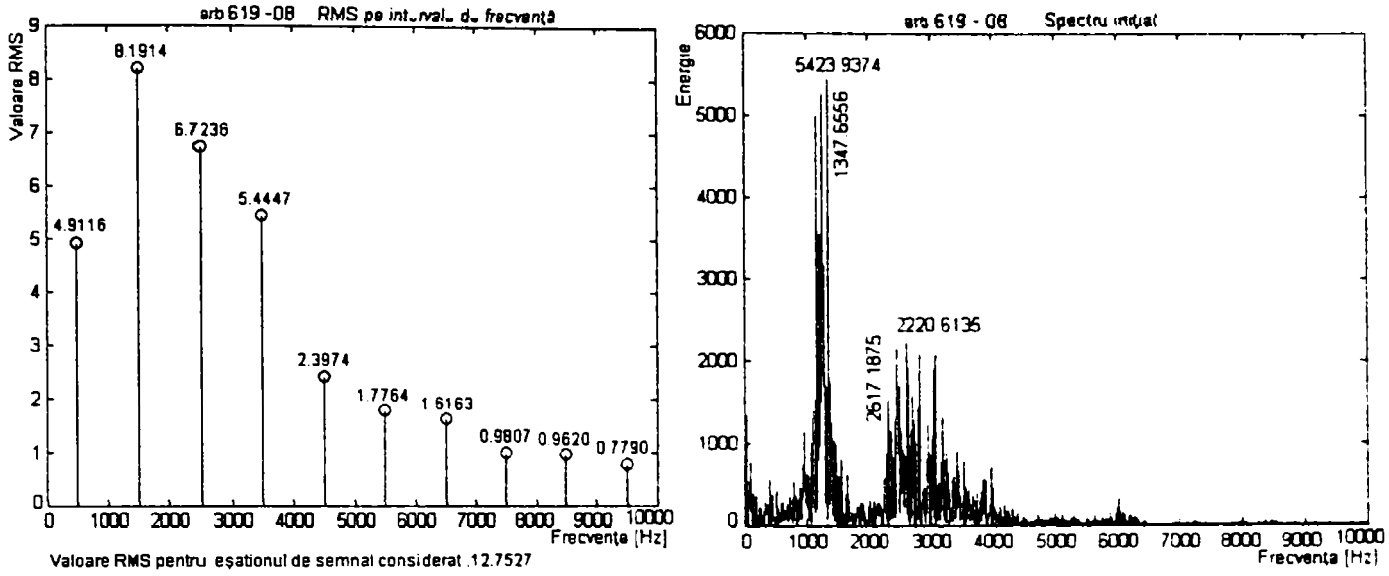


Fig. A.43.2.3.2. Diagrama RMS și spectrul de frecvență la găurirea vibropercutantă: n=619 și v_s=8.

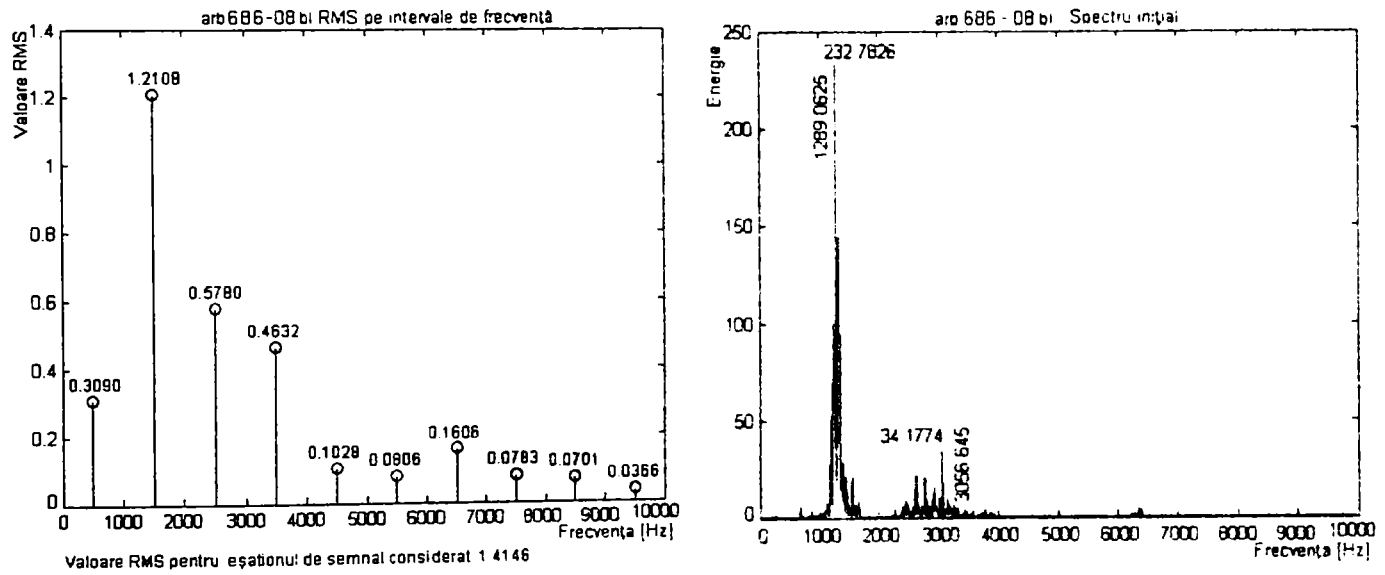


Fig. A.43.2.4.1. Diagrama RMS și spectrul de frecvență la găurirea obișnuită: n=686 și v_s=8.

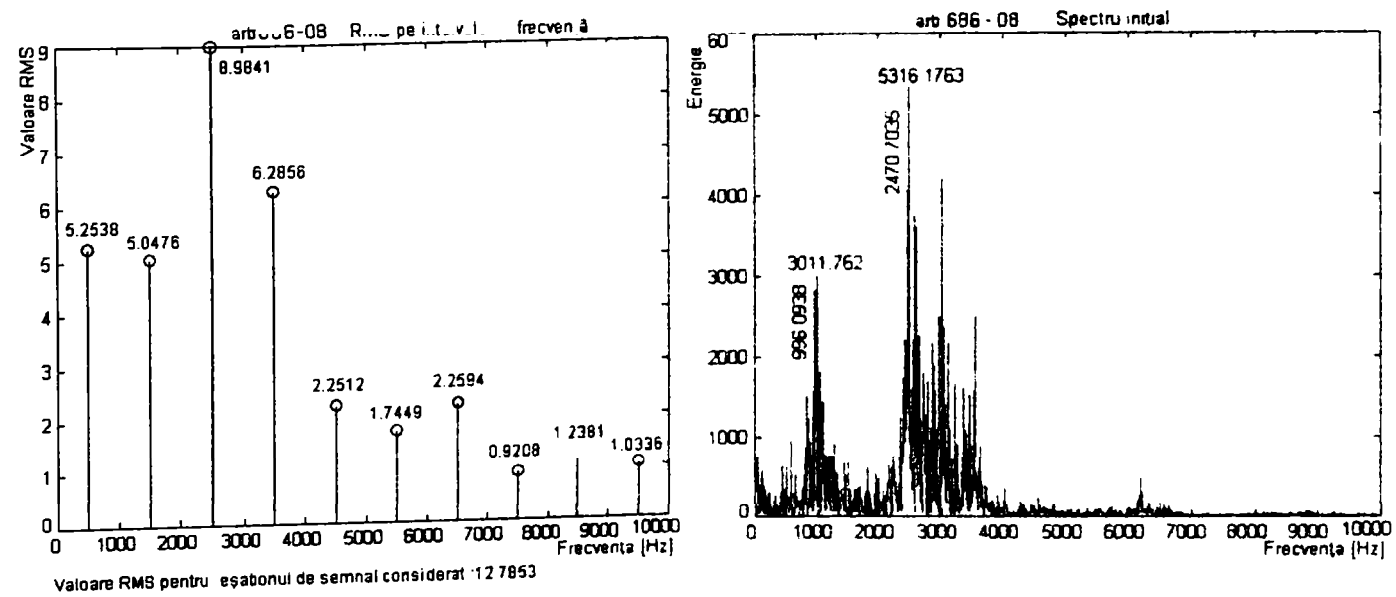


Fig. A.43.2.4.2. Diagrama RMS și spectrul de frecvență la găurirea vibropercutantă: n=686 și v_s=8.

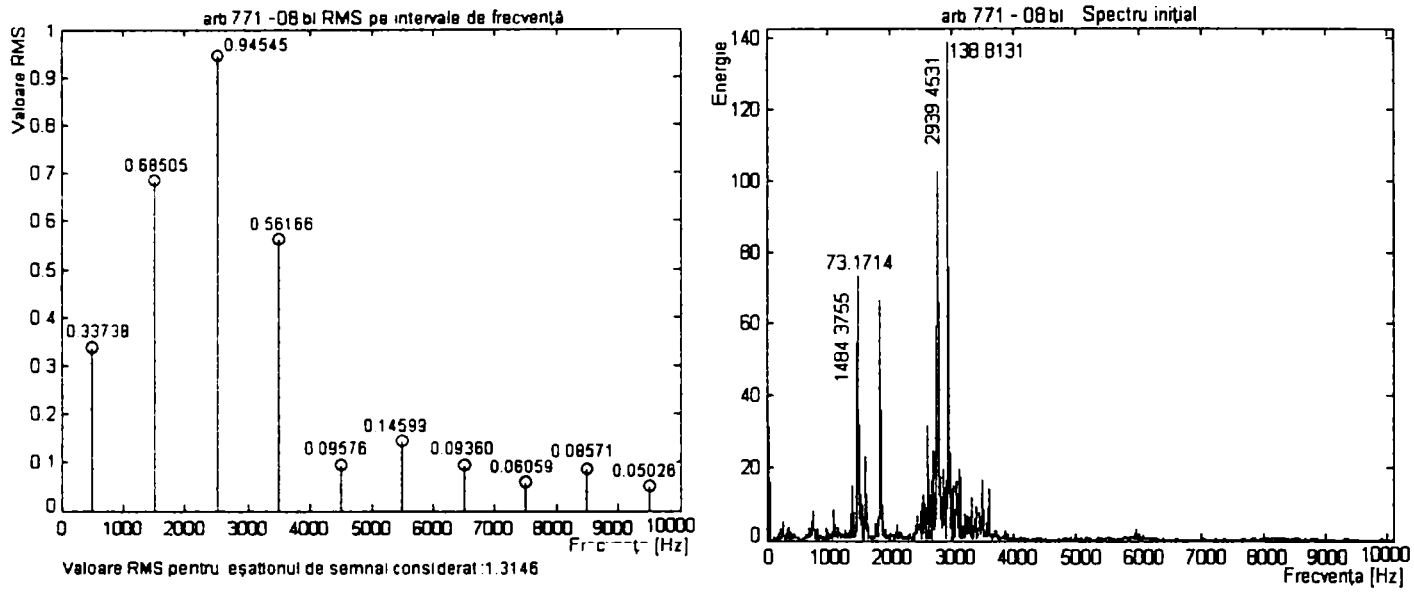


Fig. A.43.2.5.1. Diagrama RMS şi spectrul de frecvență la găurirea obişnuită: n=771 şi v_s=8.

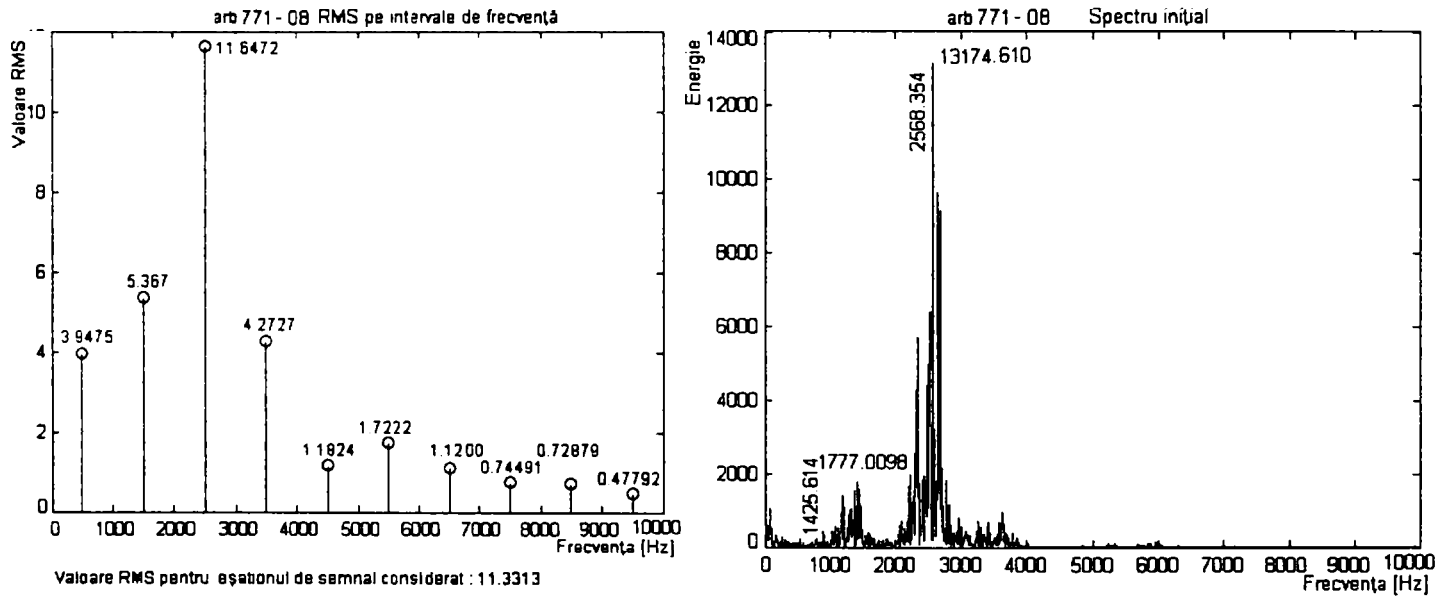


Fig. A.43.2.5.2. Diagrama RMS şi spectrul de frecvență la găurirea vibropercutantă: n=771 şi v_s=8.

3. Măsurate pe unul dintre bacurile universalului

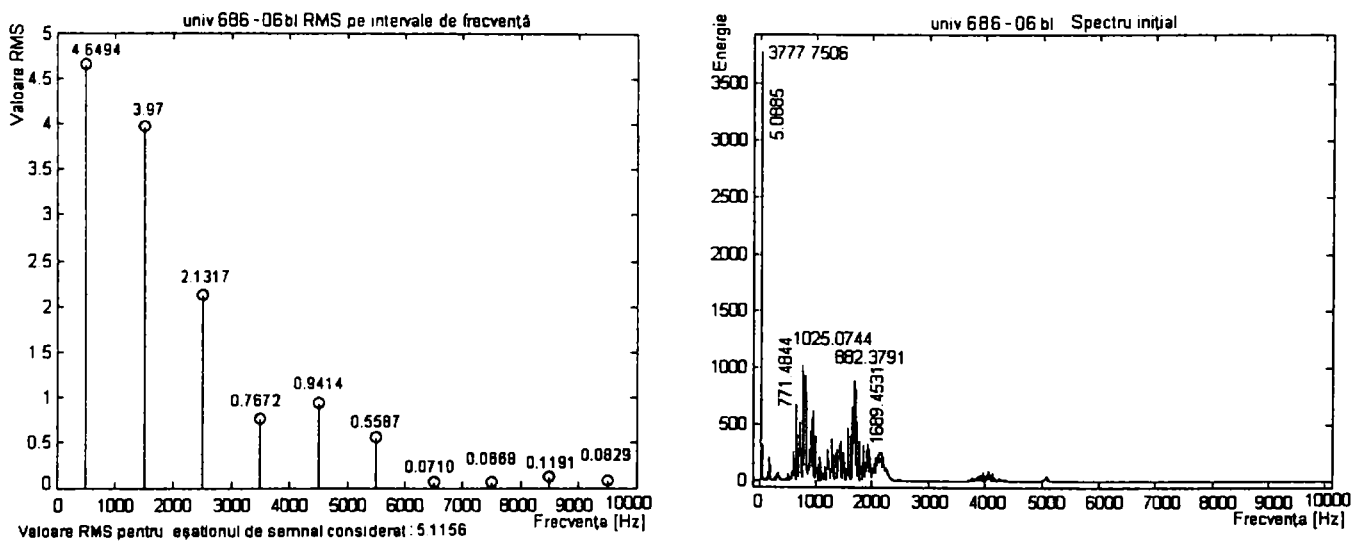


Fig. A.43.3.1.1. Diagrama RMS şi spectrul de frecvență la găurirea obişnuită: n=686 şi v_s=6.

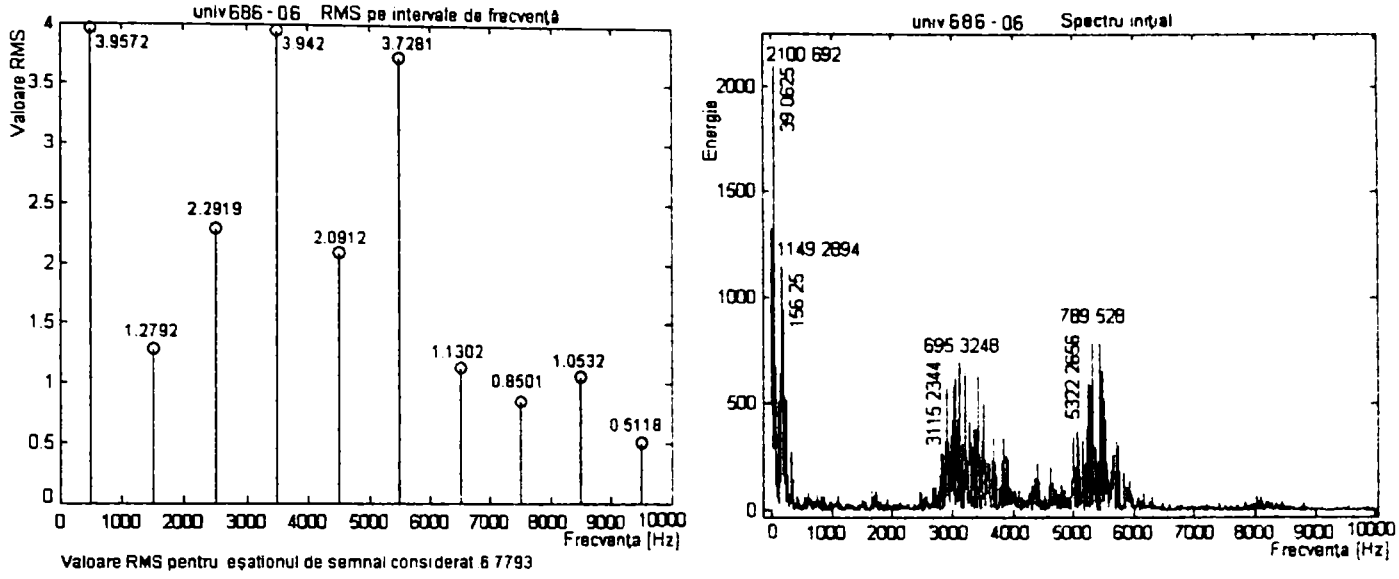


Fig. A.43.3.1.2. Diagrama RMS şi spectrul de frecvenţă la găurirea vibropercutantă: n=686 şi v_s=6.

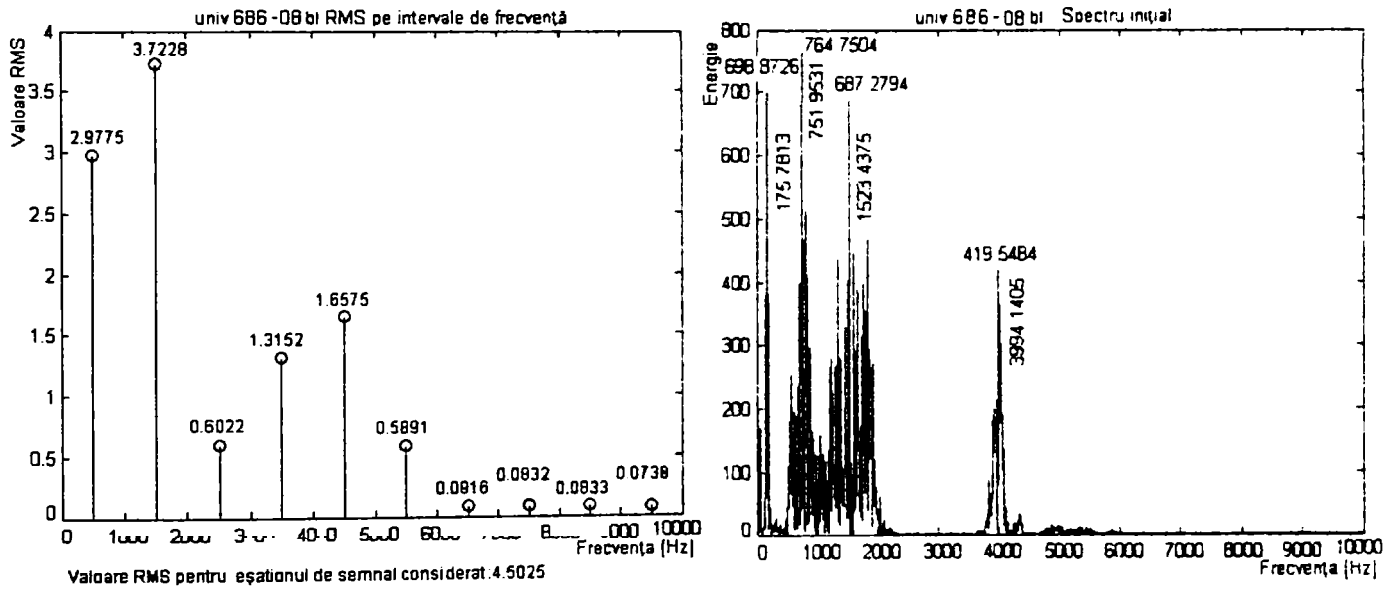


Fig. A.43.3.2.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=686 şi v_s=8.

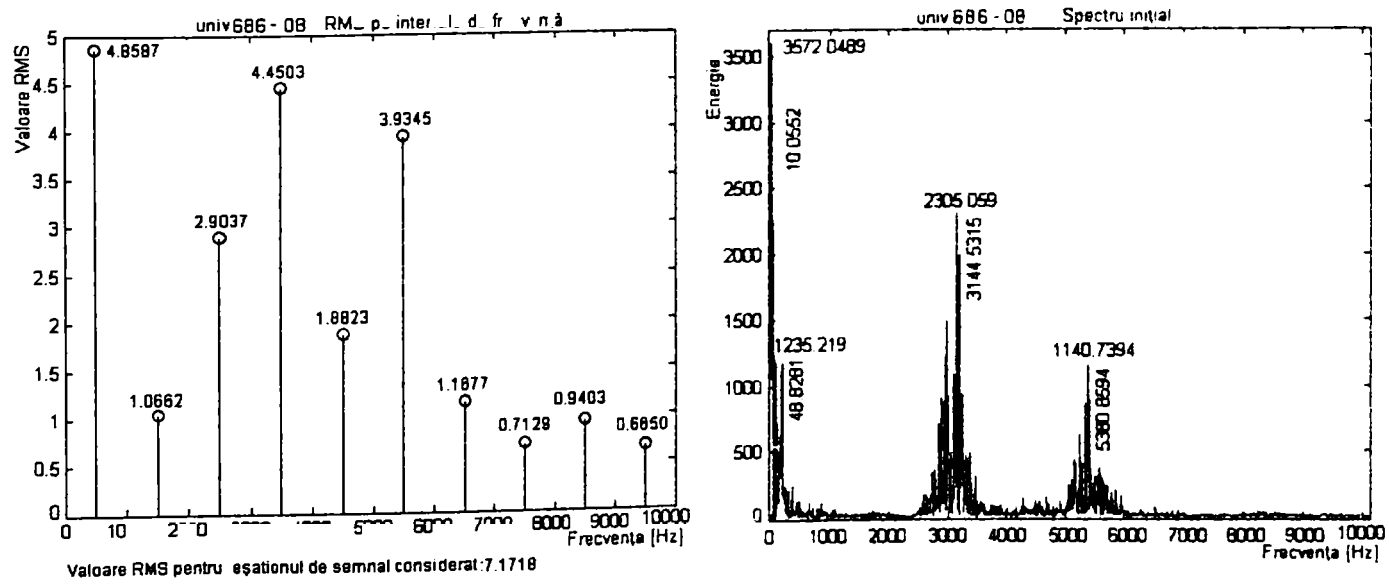


Fig. A.43.3.2.2. Diagrama RMS şi spectrul de frecvenţă la găurirea vibropercutantă: n=686 şi v_s=8.

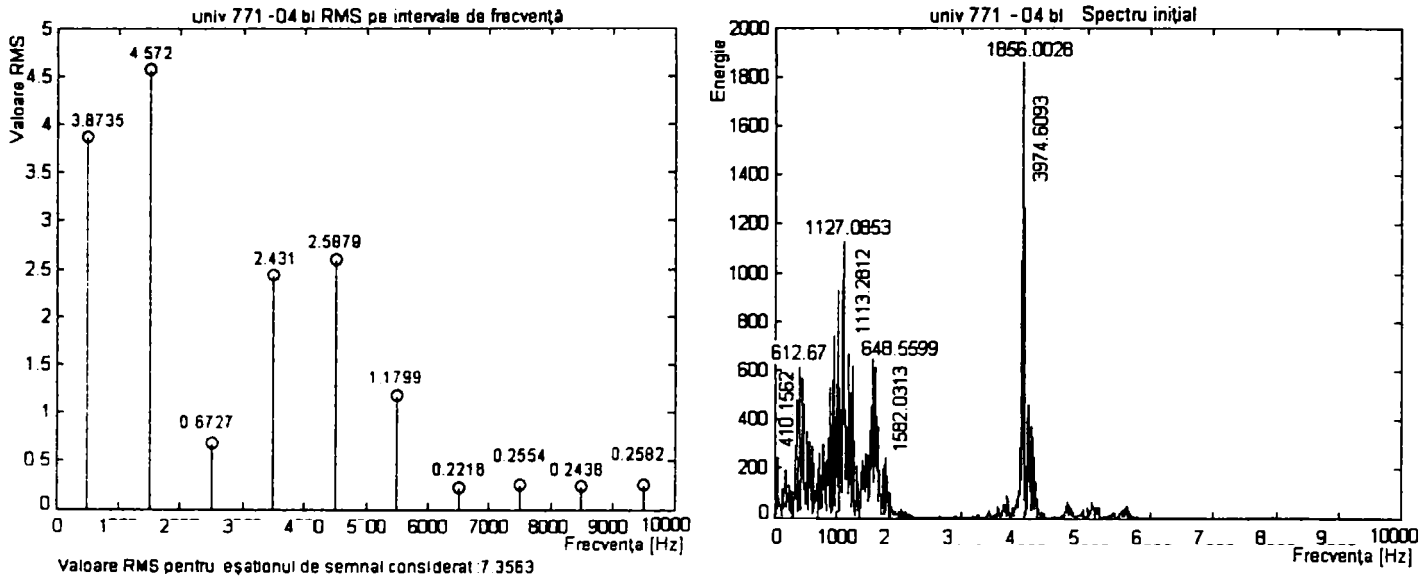


Fig. A.43.3.3.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=771 şi v_s=4.

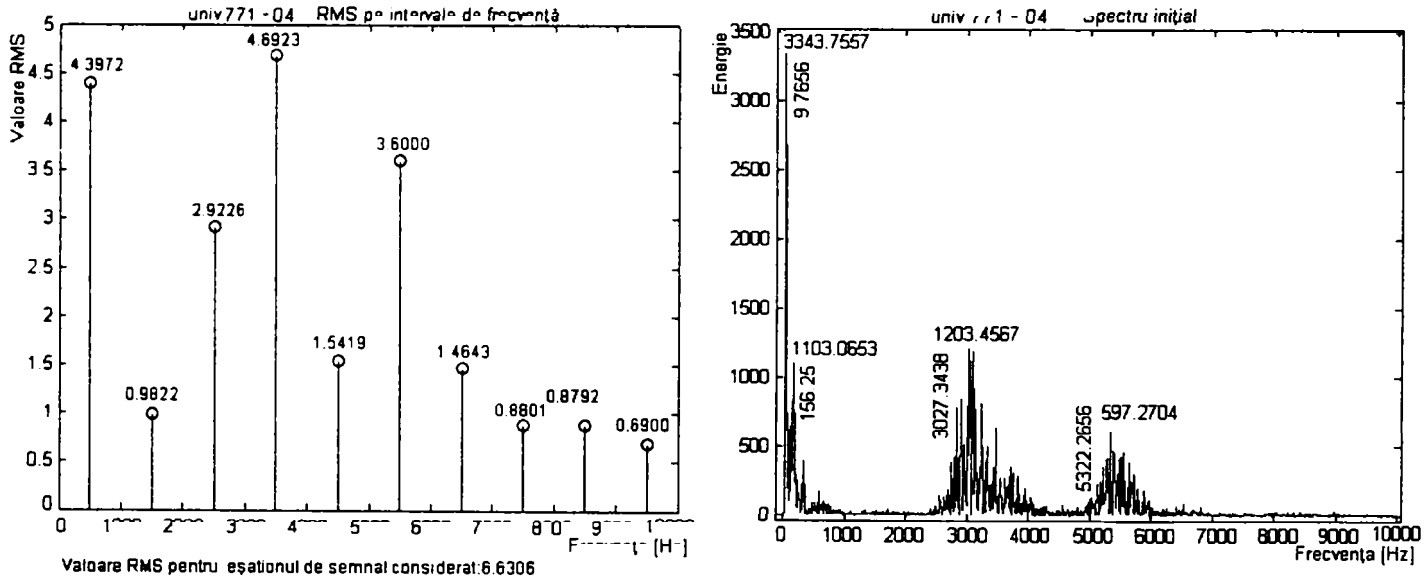


Fig. A.43.3.3.2. Diagrama RMS şi spectrul de frecvenţă la găurirea vibropercutantă: n=771 şi v_s=4.

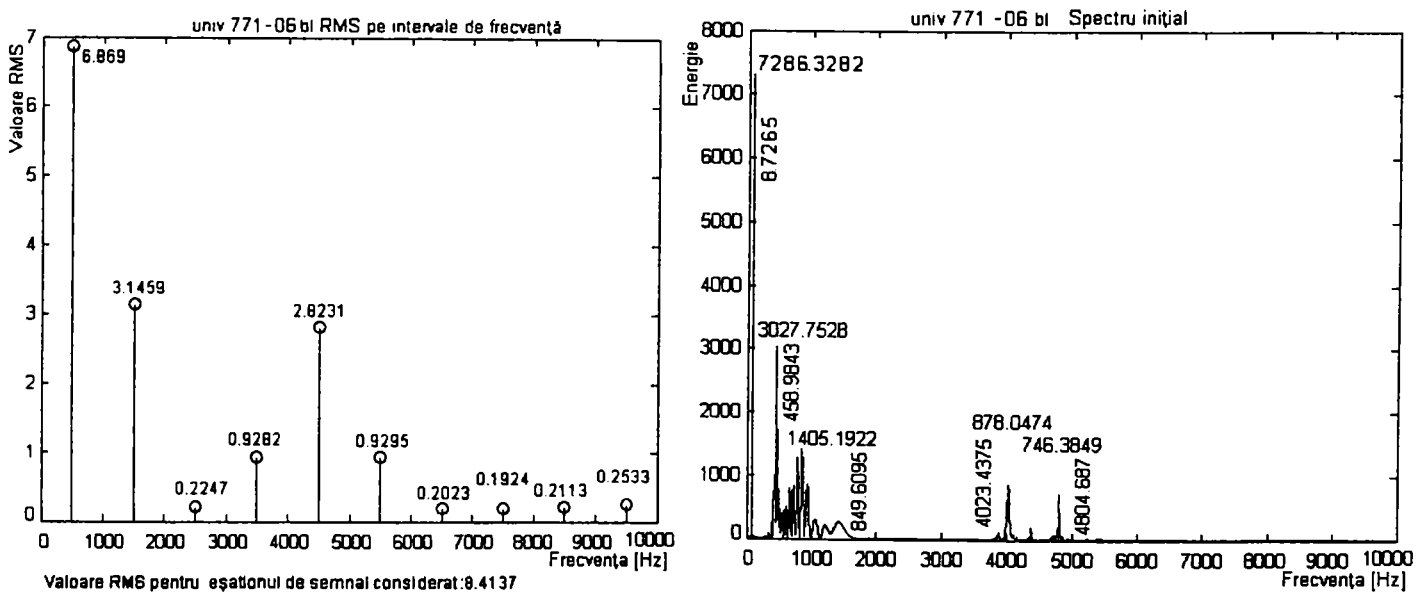


Fig. A.43.3.4.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=771 şi v_s=6.

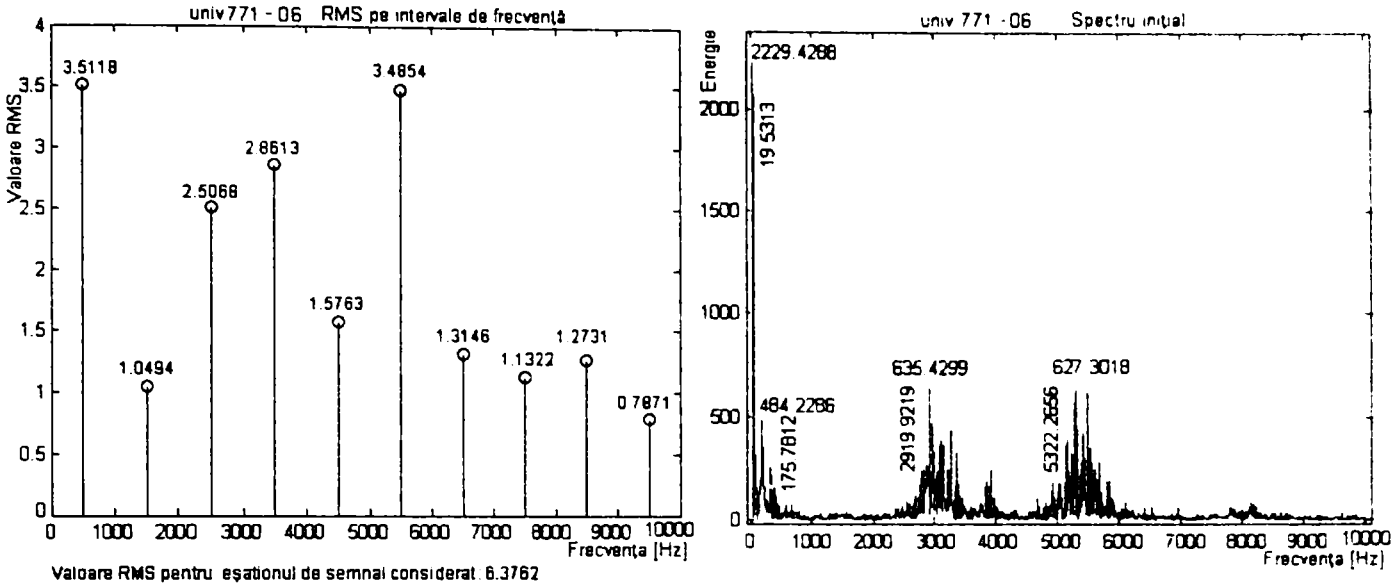


Fig. A.43.3.4.2. Diagrama RMS şi spectrul de frecvenţă la găurirea vibropercutantă: n=771 şi v_s=6.

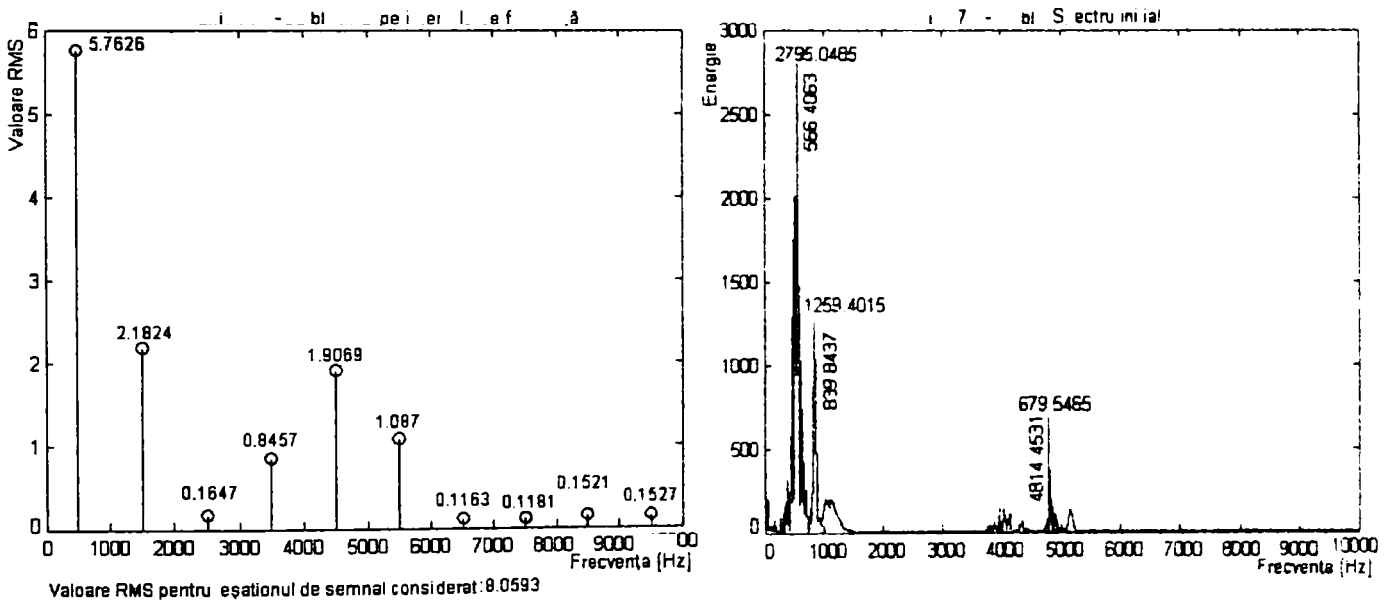


Fig. A.43.3.5.1. Diagrama RMS şi spectrul de frecvenţă la găurirea obişnuită: n=771 şi v_s=8.

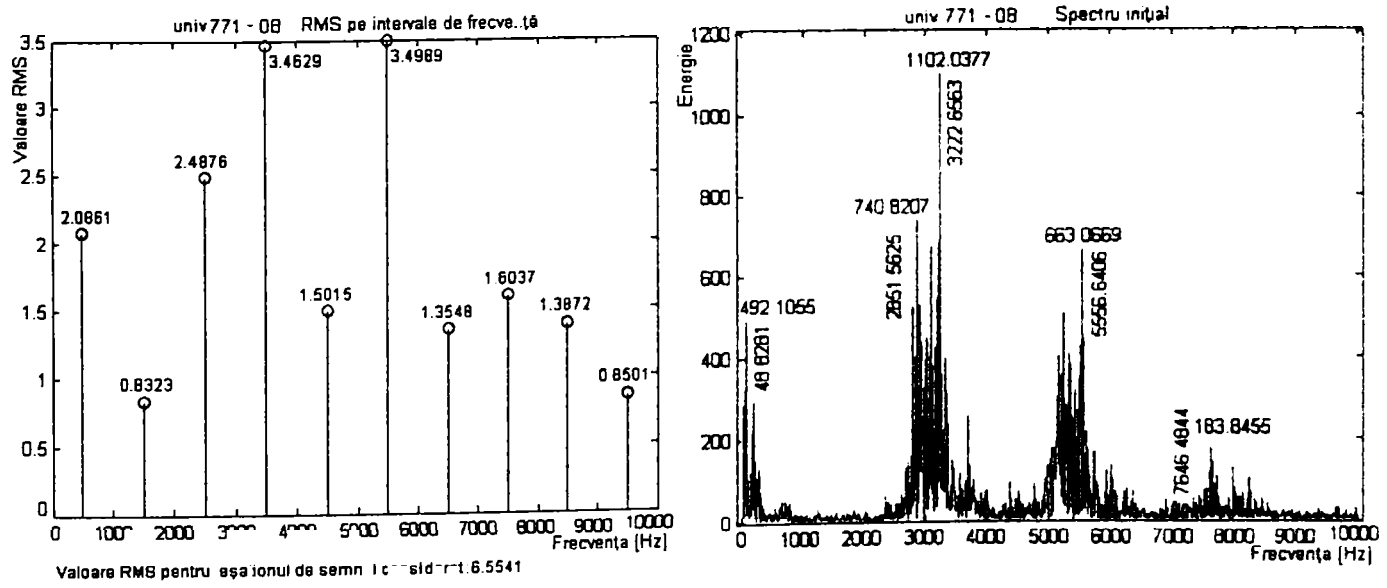


Fig. A.43.3.5.2. Diagrama RMS şi spectrul de frecvenţă la găurirea vibropercutantă: n=771 şi v_s=8.

Notă: n – reprezintă turaţia arborelui maşinii-unelte, iar v_s – viteza de avans al burghiului.

Anexa 44.

Program „ RMS_fft3.m"sub mediu Matlab pentru calculul şi trasarea diagramelor RMS a spectrelor de frecvență

```
clear all ; clc;
% Deschidere fisier
[filename,pathname] = uigetfile('*.*','Acceleratii',300,100);
if filename ~=0
   fid = fopen(strcat(pathname,filename),'r');
   [x1,count1] = fread(fid,inf,'int16');
   fclose('all');
   % Transformare in volti
   x1 = (x1).*(10/4096);
   % Se scade valoarea offset
   x1 = x1 - mean(x1);
   % Constanta traductorului KD 35
   kd35 = 0.0171;
   % Calibrare
   x1 = x1./kd35;
   y1 = x1;
      % Specificare parametrii
      Fs = 20000;
      n_fft = 2^13;
      n_timp0 = input('Introduceti timp de start (secunda) : ');
      n_fft   = input('Interval de calcul FFT : ');
      n_timp1 = n_timp0*Fs;
n_timp2 = n_timp1 + n_fft - 1;
int_timp = (n_timp0:1/20000:n_timp2/20000);
% Afisarea grafica a semnalului
figure; plot(int_timp,y1(n_timp1:n_timp2)); title('Semnal S1');
xlabel('Timp [s]'); ylabel('Acceleratie nx9,81 [m/s^2]');
% Calcul RMS semnal initial
y1_rms = sqrt(sum(y1.^2)/length(y1));
% Afisare parametrii RMS semnal initial
str_0 = strcat('Semnal initial    : ');
str_0rms = strcat('RMS          : ',num2str(y1_rms),';');
str_0afis =str_0rms;
disp(str_0afis);
% Transformata Fourier a semnalului
Y1 = fft(y1(n_timp1:n_timp2),n_fft);
% Spectrul de putere a semnalului
Pyy1 = Y1.*conj(Y1) / n_fft;
% Afisarea grafica a spectrului semnalului
f = Fs*(0:n_fft/2-1)/n_fft; figure; plot(f,Pyy1(1:n_fft/2)); title('Spectru initial');
xlabel('Frecventa [Hz]'); ylabel('Energie');
% Specificare numar de intervale de frecventa
nr_int = 10;
n_lat = floor(n_fft/2/nr_int);
% Calcul fft pe intervale de frecventa si reconstruire semnal filtrat
for k=1:nr_int
   % Stergerea frecventelor ce nu apartin intervalului curent
```

```
        Y2 = Y1;
        Y2(1:(k-1)*n_lat) = 0;
        Y2(k*n_lat+1:end)  = 0;
        % Pregatire pentru afisare interval frecvente
        str_interval = strcat('Interval frecvente : ',...
        num2str((k-1)*Fs/2/nr_int),'-',num2str(k*Fs/2/nr_int),':');
        % Spectrul de putere
        Pyy2(k,:) = (abs(Y2.*conj(Y2)) / n_fft)';
        % Reconstruirea semnalului pe intervale
        iy2(k,:) = (real(ifft(Y2)))'*2;
        figure; plot(int_timp,iy2(k,:));
        title(strcat('Semnal reconstruit. ',str_interval));
        %%%%%%%%%%%%%%%%%%%%%%%% Calcul RMS
        iy2_rms = sqrt(sum(iy2(k,:).^2)/length(iy2(k,:)));
        iy2_rms1(k) = iy2_rms;
        %%%%%%%%%%%%%%%%%% Afisare RMS pe intervale
        str_rms = strcat('RMS         : ',num2str(iy2_rms),';');
        str_afis =str_rms;
        disp(str_afis);
    end
    %%%%%%%%%%%%%%%%%%%% Diagrama RMS pe intervale de frecventa
    int_frecv = (0.5:1:nr_int)*Fs/20;
    figure; stem(int_frecv,iy2_rms1,'k');
    title(strcat('RMS pe intervale de frecventa'));
    xlabel('Frecventa [Hz]'); ylabel('Valoare RMS');
  end
end
```

Anexa 45.

%Program „diag_Ra_3.m" de trasare diagrame rugozități în funcție de turații şi vitezele de avans la aşchierea obisnuită şi vibropercutantă

```
y0n = [1 2 4 6 8]';
x0s = [463 559 619 686 771]';
      %[X,Y] = meshgrid(x0s,y0n);
      %[XI,YI] = meshgrid(463:1:771,1:0.1:8);

 %  Aschiere obisnuita
          %463   559   619  686    771

rug_bl1=  [2.70  6.20  1.40  1.80  4.40
           4.60  3.90  2.20  2.78  4.60
           3.80  3.40  3.80  3.80  4.10
           4.80  2.60  3.90  3.30  4.50
           6.40  5.40  5.20  5.60  4.70]

  %  Aschiere vibropercutanta

            %463   559   619         686   771

  rug_nebl1= [0.90  1.10  1.20  1.10  1.60
              2.40  1.20  0.98  0.86  1.30
              2.60  1.80  0.26  1.30  0.74
              0.88  0.90  0.56  0.54  0.84
              3.80  1.10  3.80  1.70  1.40]
              for i = 1:5
    x  = [463 559 619 686 771];
    xi = (463:1:771);
    yi = interp1(x,rug_bl1(i,:),xi,'spline');
    figure; plot(xi,yi,'r'); hold on
    plot(x,rug_bl1(i,:),'rx');
    yi = interp1(x,rug_nebl1(i,:),xi,'spline');
    hold on; plot(xi,yi,'k');
    plot(x,rug_nebl1(i,:),'kx');
    xlabel('Turatie [rot/min]')
    ylabel('Rugozitate  [microni]')
    title(strcat('Variatia rugozitatii masurate a suprafetelor, avans : ',num2str(y0n(i)),' [mm/min]'));
end
```

Anexa 46.

1. Diagramele de variație ale valorilor măsurate a rugozităților suprafețelor în funcție de regimurile utilizate la așchierea obișnuită și așchierea vibropercutantă

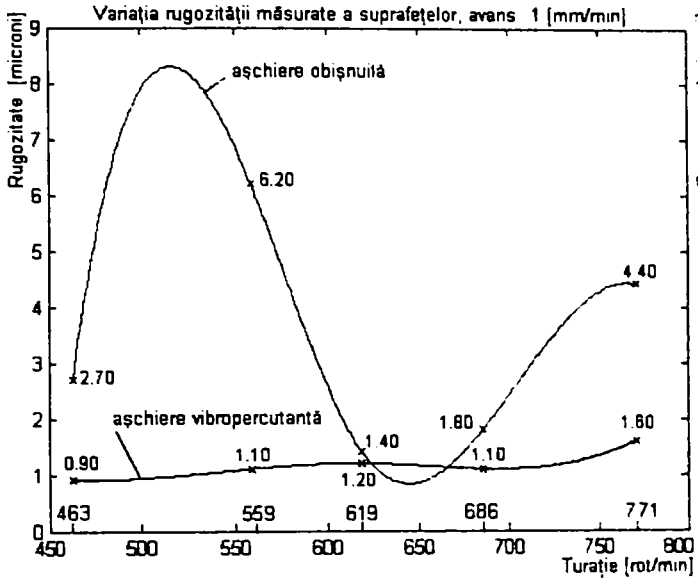


Fig. A.46.1.1.

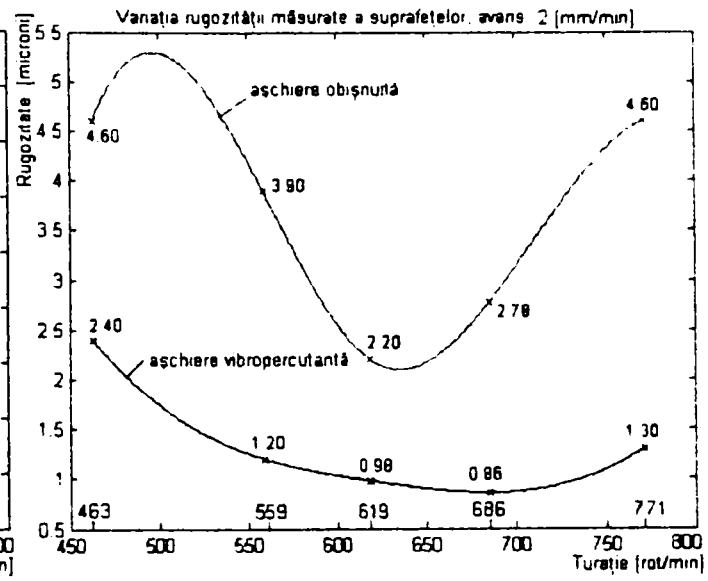


Fig. A.46.1.2.

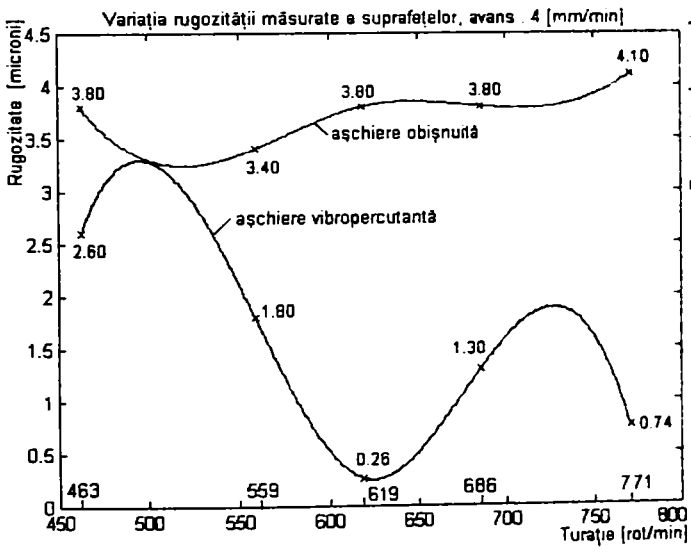


Fig. A.46.1.3.

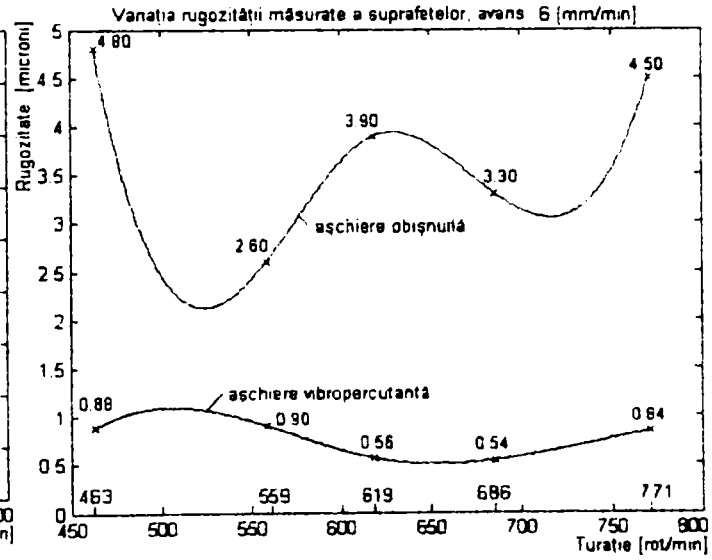


Fig. A.46.1.4

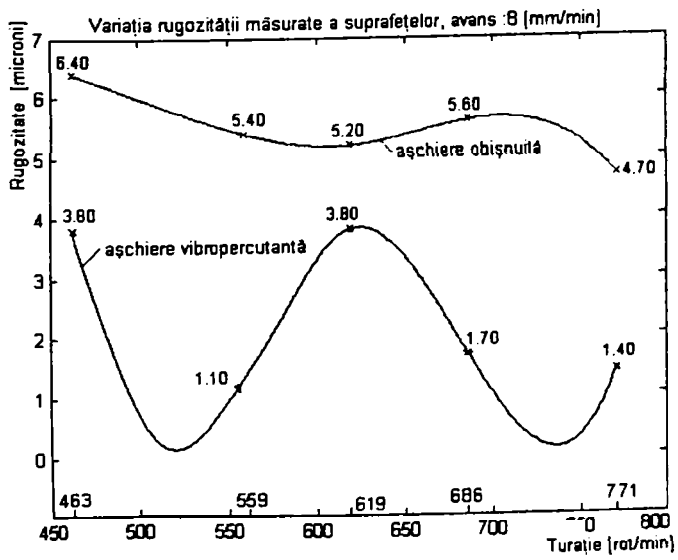


Fig. A.46.1.5.