Universitatea Tehnica din Republica Moldova

Facultatea Calculatoare, Informatica si Microelectronica

Departament Inginerie Software si Automatica

Specialitatea Tehnologia Informației

Raport

Curs: Prelucrarea semnalelor

Tema: Estantionarea si cuantizarea semnalelor. Interpolarea semnalelor esantionate.

A elaborat: Reguș Ruslan Grupa: TI-214

A verificat: Asist. Univ. Cazac A.

Chișinău 2024

**Lucrare de laborator nr. 4**

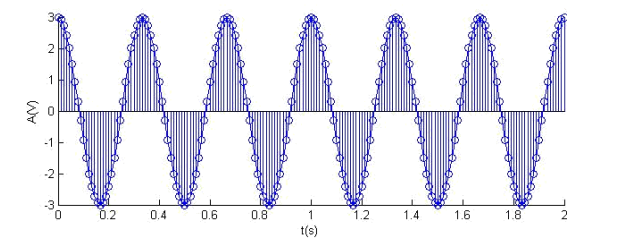
Obiective: înțelegerea conceptelor de eșantionare, cuantizare, de zgomot de cuantizare și interpolare a semnalelor eșantionate.

**Exerciţiu 1 :**

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

|  |
| --- |
| fs = 8000;  N = 1024;  t = linspace(0, (N-1)/fs, N);  x = 0.5\*sin(2\*pi\*50\*t) + 0.2\*sin(2\*pi\*230\*t + pi/3);  stem(t, x);  xlabel('Timp (sec)');  ylabel('Amplitudine (V)');  title('Semnal format din doua sinusoide'); |
|  |

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

****

|  |  |
| --- | --- |
| fs = 90;  t = 0:1/fs:2;  x = 3\*sin(2\*pi\*3\*t-3\*pi/2);  stem(t,x, 'Color','b');  hold on  plot(t,x, 'Color','b');  xlabel('t(s)');  ylabel('A(V)'); |  |

În codul de mai jos, avem următoarele caracteristici ale semnalului sinusoidal x(t):

* Amplitudinea: 3 V
* Frecvența: 3 Hz
* Faza inițială: -3π/2 rad
* Frecvența de eșantionare: 90 Hz
* Durată: 2 secunde

Explicație:

* Frecvența semnalului este determinată de argumentul sin() din expresia x(t), care este 2π3t. Astfel, frecvența semnalului este 3 Hz.
* Amplitudinea semnalului este dată de coeficientul sin() din expresia x(t), care este 3. Amplitudinea este deci 3 V.
* Faza inițială a semnalului este dată de argumentul sin() din expresia x(t), care este -3π/2 rad.
* Frecvența de eșantionare este 90 Hz, deoarece am definit variabila fs ca fiind 90 Hz.
* Durata semnalului este 2 secunde, deoarece am definit variabila t să ia valori între 0 și 2 secunde, cu o rată de eșantionare de 90 Hz, adică 180 eșantioane în total.
* Funcția stem() afișează semnalul discret sub formă de puncte discrete, iar funcția plot() afișează semnalul continuu sub forma unei curbe continue.

3. Fie semnalul:



|  |
| --- |
| fs = 1000;  t = 0:1/fs:1;  x = 10\*sin(2\*pi\*100\*t+pi/2)+20\*sin(2\*pi\*50\*t)-40\*sin(2\*pi\*150\*t-pi/4);  figure  stem(t(1:100),x(1:100));  xlabel('t(s)');  ylabel('A(V)'); %A=70 |
|  |

Cerințe:

• care este frecvența minimă de eșantionare astfel încât să se respecte teorema eșantionării;

Frecvența minimă de eșantionare pentru a respecta teorema eșantionării trebuie să fie de cel puțin dublul frecvenței maxime din semnal. Frecvența maximă din semnal este 150 Hz, deci frecvența minimă de eșantionare trebuie să fie de cel puțin 300 Hz.

• alegând o frecvență de eșantionare de 10 ori mai mare decât cea determinată la punctul anterior, să se eșantioneze semnalul x(t) și să se reprezinte grafic;

Alegând o frecvență de eșantionare de 10 ori mai mare decât cea determinată la punctul anterior, putem folosi o frecvență de eșantionare de 3000 Hz. Pentru a eșantiona semnalul x(t), putem folosi funcția sin și vectorul de timp t, după cum urmează:

|  |  |
| --- | --- |
| fs = 3000;  t = 0:(1/fs):0.03;  x = 10\*sin(2\*pi\*100\*t+pi/2)+20\*sin(2\*pi\*50\*t)-40\*sin(2\*pi\*150\*t-pi/4);  stem(t,x);  xlabel('t(s)');  ylabel('A(V)'); |  |

• care este frecvența de repetiție a semnalului x(t)?

Frecvența de repetiție a semnalului x(t) este 50 Hz. Aceasta poate fi observată din componentele sinusoidale ale semnalului, care sunt la frecvențele 100 Hz, 50 Hz și 150 Hz. Frecvența fundamentală a semnalului este 50 Hz, care este frecvența la care semnalul se repetă periodic.

În MATLAB, "eșantionarea" (eng. "sampling") se referă la procesul de prelevare a unui număr finit de eșantioane dintr-un semnal continuu în timp sau dintr-un set de date continuu.

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

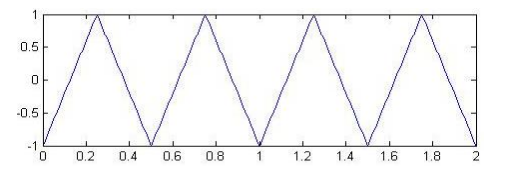
|  |
| --- |
| t = 0:0.001:1;  x = 3\*sin(2\*pi\*5\*t);  subplot(2,1,1);  plot(t,x);  title('Semnal original');  xlabel('t(s)');  ylabel('A(V)'); %A=3 |
|  |

Semnal eșantionat

|  |  |
| --- | --- |
| fs1 = 4;  n1 = 0:1/fs1:1;  xn1 = 3\*sin(2\*pi\*5\*n1);  fs2 = 50;  n2 = 0:1/fs2:1;  xn2 = 3\*sin(2\*pi\*5\*n2);  subplot(2,1,1);  stem(n1,xn1);  title('Eșantionare cu fs1 = 4 Hz');  xlabel('t(s)');  ylabel('A(V)');  subplot(2,1,2);  stem(n2,xn2);  title('Eșantionare cu fs2 = 50 Hz');  xlabel('t(s)');  ylabel('A(V)');  %A=3 |  |

Observăm că semnalul eșantionat cu frecvența fs1 = 4 Hz nu poate fi utilizat pentru a reconstitui semnalul original, deoarece nu respectă teorema eșantionării (frecvența de eșantionare este prea mică). În schimb, semnalul eșantionat cu frecvența fs2 = 50 Hz respectă teorema eșantionării și poate fi utilizat pentru a reconstitui semnalul original prin intermediul interpolării.

5. Să se genereze și să se prezinte în MATLAB, în două ferestre separate, semnalul din figura următoare și semnalul discretizat.



Indiciu: pornind de la figură, trebuie să se identifice toți parametrii semnalului triunghiular (amplitudine, frecvență, frecvență de eșantionare, durată). Frecvența fs se alege astfel încât să fie 30 de eșantioane într-o perioadă.

|  |  |
| --- | --- |
| t=0:0.01:2;  x=1\*sawtooth(2\*pi\*2\*t,1/2);  subplot(2,1,1)  plot(t, x, 'Color', [0.5 0 0.5]);  xlabel('t(s)');  ylabel('A(V)');  fs=2\*30;  t=0:1/fs:2;  x=1\*sawtooth(2\*pi\*2\*t,1/2);  subplot(2,1,2)  stem(t, x, 'Color', [0.5 0 0.5]);  xlabel('t(s)');  ylabel('A(V)'); |  |

**Exerciţiu 2 :**

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

Durata semnalului este de 2 minute, adică 120 secunde. Frecvența de eșantionare este de 4 kHz, ceea ce înseamnă că sunt prelevate 4.000 de eșantioane în fiecare secundă.

Numărul total de eșantioane prelevate este:

120 secunde × 4.000 eșantioane/secundă = 480.000 de eșantioane.

Dacă fiecare eșantion este stocat pe 2 octeți, atunci dimensiunea totală a memoriei necesare este:

480.000 de eșantioane × 2 octeți/eșantion = 960.000 de octeți sau 960 KB.

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

|  |  |
| --- | --- |
| n=1:0.01:40;  x=17./n;  x1 = round(x);  x2 = floor(x);  y=-17./n;  y1 = round(y);  y2 = floor(y);  figure  subplot(231);  plot(n,x);  title('semnal original 17/n');  grid  subplot(232);  plot(n,x1);  title('round');  grid  subplot(233);  plot(n,x2);  title('floor');  grid  subplot(234);  plot(n,y);  title('semnal original -17/n');  grid  subplot(235);  plot(n,y1);  title('round');  grid  subplot(236);  plot(n,y2);  title('floor');  grid |  |

|  |  |
| --- | --- |
| x1eror=x1-x;  x2eror=x2-x;  y1eror=y1-y;  y2eror=y2-y;  figure  subplot(221);  plot(n,x1eror);  title('round');  grid  subplot(222);  plot(n,x2eror);  title('floor');  grid  subplot(223);  plot(n,y1eror);  title('round');  grid  subplot(224);  plot(n,y2eror);  title('floor');  grid |  |

Observăm că semnalele cuantizate folosind funcția round sunt mai apropiate de semnalele originale decât cele cuantizate folosind funcția floor. Zgomotul de cuantizare este reprezentat în ultimul subplot și observăm că acesta are amplitudine mică în jurul valorii de zero, dar crește la valorile extreme ale semnalelor, ceea ce poate duce la distorsiuni în procesul de redare a semnalului.

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

|  |  |
| --- | --- |
| nr=15;  z = 0.2\*randn(1,nr);  levels = [-1, -3/4, -1/2, -1/4, 0, 1/4, 1/2, 3/4, 1];  figure;  subplot(2,1,1);  plot(z);  title('Semnalul alb gaussian');  xlabel('Esantioane');  ylabel('Valoare');  subplot(2,1,2);  quantized\_z = zeros(size(z));  for i = 1:length(z)  if z(i) <= levels(1)  quantized\_z(i) = levels(1);  elseif z(i) > levels(end)  quantized\_z(i) = levels(end);  else  for j = 1:length(levels)-1  if (z(i) > levels(j)) && (z(i) <= levels(j+1))  quantized\_z(i) = levels(j+1);  break;  end  end  end  end    plot(quantized\_z);  title('Semnalul alb gaussian cuantizat');  xlabel('Esantioane');  ylabel('Valoare'); |  |

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

|  |  |
| --- | --- |
| fs = 4000;  N = 30;  t = linspace(0, (N-1)/fs, N);  x = 0.5\*sin(2\*pi\*50\*t) + 0.2\*sin(2\*pi\*230\*t+pi/3);    x8 = round(x \* (2^7-1)) / (2^7-1);  q8 = x8 - x;    subplot(3,1,1)  plot(t,x)  title('Semnalul original')  xlabel('t(s)')  ylabel('A(V)')  subplot(3,1,2)  stairs(t,x8)  title('Semnalul cuantizat pe 8 biti')  xlabel('t(s)')  ylabel('A(V)')  subplot(3,1,3)  plot(t,q8)  title('Zgomotul de cuantizare pe 8 biti')  xlabel('t(s)')  ylabel('A(V)') |  |

|  |  |
| --- | --- |
| fs = 4000;  N = 30;  t = linspace(0, (N-1)/fs, N);  x = 0.5\*sin(2\*pi\*50\*t) + 0.2\*sin(2\*pi\*230\*t+pi/3);  x16 = round(x \* (2^15-1)) / (2^15-1);  q16 = x16 - x;  subplot(3,1,1)  plot(t,x)  title('Semnalul original')  xlabel('t(s)')  ylabel('A(V)')  subplot(3,1,2)  stairs(t,x16)  title('Semnalul cuantizat pe 16 biti')  xlabel('t(s)')  ylabel('A(V)')  subplot(3,1,3)  plot(t,q16)  title('Zgomotul de cuantizare pe 16 biti')  xlabel('t(s)')  ylabel('A(V)') |  |

**Exerciţiu 3 :**

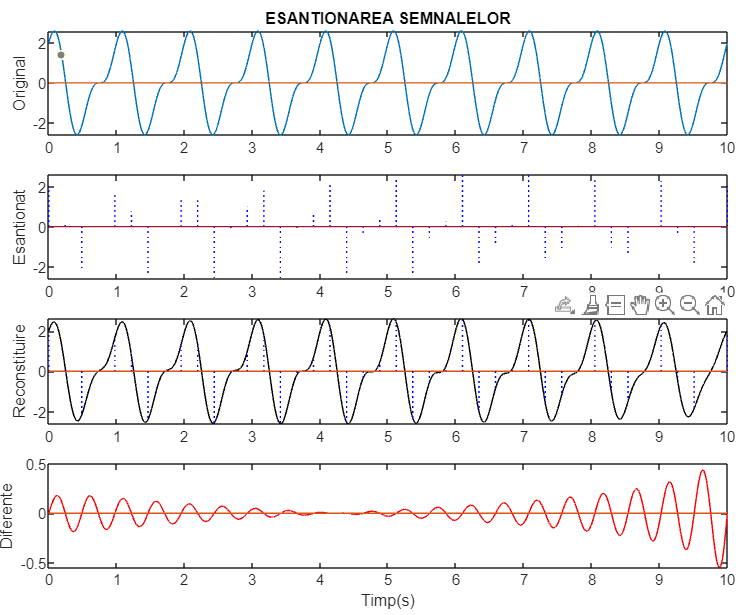
Chestiuni de studiu în baza Exemplului 4.7.

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

b) Executarea de calcule cu valorile diferite pentru frecvenţele 1 f şi 2 f , eventual pentru amplitudini diferite ale sinusoidelor componente, utilizând pentru fiecare caz trei valori ale frecvenţei de eşantionare: mai mică, egală şi mai mare (de 10 ori) ca frecvenţa Nyquist.

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

|  |
| --- |
| % optiune de reprezentare (rep=0) grafic unic, (rep>0) grafice separate  tmax=10; % timpul maxim de reprezentare  fi=pi/2; % faza componentei secundare  f1=1; % frecventa componentei primare  f2=2; % frecventa componentei secundare  % pasul utilizat la reprezentarea grafica  pasmic=0.001;  fes=4.1; % frecventa de esantionare  rep=1;  tes=1/fes; % perioada esantioanelor  a1=2;  a2=1;  t=0:pasmic:tmax;  % pregatirea graficului 1 (semnal original)  y=a1\*cos(2\*pi\*f1\*t)+a2\*cos(2\*pi\*f2\*t-fi);  t1=0:tes:tmax;  n=round(tmax/tes)+1;  % pregatirea graficului 2 (esantionarea)  y1=a1\*cos(2\*pi\*f1\*t1)+a2\*cos(2\*pi\*f2\*t1-fi);  % pregatirea graficului 3 (reconstituire)  y2=y1(1)\*sinc(t/tes);  for k=1:(n-1)  y2=y2+y1(k+1)\*sinc(t/tes-k);  end  if rep>0  subplot(4,1,1)  plot(t,y) % trasarea graficului 1 (semnal)  hold on  ylabel('Original')  title('ESANTIONAREA SEMNALELOR')  plot([0 tmax],[0 0])  for i=1:n  subplot(4,1,2)  % trasarea graficului 2 (esantionarea)  plot([t1(i) t1(i)],[0 y1(i)],'b:')  hold on  ylabel('Esantionat')  plot([0 tmax],[0 0])  end  subplot(4,1,3)  % trasarea graficului 3 (reconstituire)  plot(t,y2,'k')  hold on  plot([0 tmax],[0 0])  for i=1:n  plot([t1(i) t1(i)],[0 y1(i)],'b:')  end  ylabel('Reconstituire')  subplot(4,1,4)  % trasarea graficului 4 (diferente)  plot(t,y-y2,'r')  hold on  plot([0 tmax],[0 0])  ylabel('Diferente')  xlabel('Timp(s)')  else  plot(t,y) % trasarea graficului 1 (semnal)  hold on  plot([0 tmax],[0 0])  for i=1:n  % trasarea graficului 3 (reconstituire)  plot([t1(i) t1(i)],[0 y1(i)],'b:')  end  plot(t,y2,'k')  % trasarea graficului 4 (diferente)  plot(t,y-y2,'r')  plot([0 tmax],[0 0])  title('ESANTIONAREA SEMNALELOR')  ylabel('Semnal(albastru)/Reconstituire(negru)/Diferente(rosu)')  xlabel('Timp(s)')  end |



Exemplul 4.7 prezintă procesul de esantionare și reconstituire a unui semnal compus din două componente sinusoidale de frecvențe diferite. Scriptul permite reprezentarea grafică a semnalului original, a semnalului esantionat, a semnalului reconstituit și a diferenței dintre semnalul original și cel reconstituit. Opțiunea de reprezentare grafică poate fi setată să fie fie într-un singur grafic, fie în grafice separate.

a) Pentru execuția calculelor cu valorile recomandate în antetul script-ului, se obțin graficele pentru semnalul original, semnalul esantionat, semnalul reconstituit și diferența dintre semnalul original și cel reconstituit, afișate fie într-un singur grafic, fie în grafice separate.

b) Pentru a executa calcule cu alte valori ale frecvențelor și amplitudinilor componente, se poate modifica valorile parametrilor f1, f2, a1 și a2 și se pot alege trei valori ale frecvenței de esantionare, una mai mică, una egală și una mai mare decât frecvența Nyquist. Acest lucru va permite observarea efectului pe care îl are frecvența de esantionare asupra calității semnalului reconstituit.

|  |  |
| --- | --- |
| f1=2;  f2=4;  fes=5.7; |  |
| f1=2;  f2=4;  fes=10.7; |  |

c) Pentru a aprecia vizual sau cantitativ calitatea semnalului reconstituit, se poate compara semnalul original cu cel reconstituit și se poate evalua gradul de erori între ele, precum și modul în care semnalul reconstituit se apropie de semnalul original în zona de început și de final a intervalului de timp observat.

**Concluzii:**

In urma acestui laborator am înțeles conceptele de eșantionare, cuantizare și de zgomotului de cuantizare.Am reprezentați grafic eșantioanele ale unui semnal si am calculate numarul de esantioane dupa formula. Am inteles ca cu cat sunt utilizate mai multe esantioane cu atat emnalul discretizat se aseamana cu cel original. Am invatat ca frecvența minimă de eșantionare pentru a respecta teorema eșantionării trebuie să fie de cel puțin dublul frecvenței maxime din semnal. Daca frecvența maximă din semnal este 150 Hz, deci frecvența minimă de eșantionare trebuie să fie de cel puțin 300 Hz.