Universitatea Tehnica din Republica Moldova

Facultatea Calculatoare, Informatica si Microelectronica

Departament Inginerie Software si Automatica

Specialitatea Tehnologia Informației

Raport

Curs: Prelucrarea semnalelor

Tema: Sisteme discrete în timp continuu sau discret

A elaborat: Reguș Ruslan Grupa: TI-214

A verificat: Asist. Univ. A. Cazac

Chișinău 2024

**Lucrare de laborator nr. 5**

Tema: Sisteme discrete în timp continuu sau discret .

Scopul lucrării: De a învăţa cum să creăm diferite sisteme în conformitate cu sistemul iniţial.

Programul 5\_1

Prezentarea unui filtru simplu cu M-vîrfuri, numărul cărora se introduce de către utilizator.

|  |  |
| --- | --- |
| % Programul P2\_1  % Simularea filtrului FIR de o lungime arbitrara  % generarea semnalului de intrare  clf; n=0:100;  s1=cos(2\*pi\*0.05\*n); % semnal de o frecventa joasa  s2=cos(2\*pi\*0.47\*n); % semnal de o frecventa inalta  x=s1+s2;  % Realiyarea filtrului FIR  M=input('Lungimea dorita a filtrului = ');  num=ones(1,M);  y=filter(num,1,x)/M;  % Afisarea semnalelor de intare si de isire  subplot(2,2,1);  plot(n,s1);  axis([0, 100, -2, 2]);  xlabel('Timpul n'); ylabel('Amplituda');  title('semnal #1');  subplot(2,2,2);  plot(n,s2);  axis([0, 100, -2, 2]);  xlabel('Timpul n'); ylabel('Amplituda');  title('semnal #2');  subplot(2,2,3);  plot(n,x);  axis([0, 100, -2, 2]);  xlabel('Timpul n'); ylabel('Amplituda');  title('Semnalul de intrare');  subplot(2,2,4);  plot(n,y);  axis([0, 100, -2, 2]);  xlabel('Timpul n'); ylabel('Amplituda');  title('Semnalul de iesire');  axis; | Lungimea dorita a filtrului = 4    Lungimea dorita a filtrului = 50 |

Acest program începe prin curățarea figurii curente și definirea unui vector de timp n cu 101 puncte între 0 și 100. Apoi, se generează două semnale de cosinus cu frecvențe diferite și se adună pentru a crea semnalul de intrare. Se cere utilizatorului să introducă lungimea dorită a filtrului și apoi se creează filtrul FIR utilizând funcția ones. Semnalul de intrare este filtrat folosind funcția filter și se normalizează prin împărțirea la lungimea filtrului.În final, se afișează semnalul de intrare și semnalul de ieșire folosind funcția subplot.

În general, acest program simulează un filtru FIR de o lungime arbitrară și afișează semnalul de intrare și semnalul de ieșire, ceea ce este util pentru înțelegerea cum funcționează aceste filtre.

Programul 5\_2

Programul de mai jos poate fi folosit pentru generarea semnalului de intrare x[n], compus dintr-o secenta sinusoidala.

|  |  |
| --- | --- |
| % generarea semnalului sinusoidal de intrare  clf;  n = 0:200;  x = cos(2 \* pi \* 0.05 \* n);  % Calcularea semnalului de iesire  x1 = [x 0 0]; % x1[n] = x[n+1]  x2 = [0 x 0]; % x2[n] = x[n]  x3 = [0 0 x]; % x3[n] = x[n-1]  y = x2 .\* x2 + x1 .\* x3;  y = y(2:202);  % Vizualizarea semnalului de intrare si a celui de ie?ire  subplot(2, 1, 1);  plot(n, x);  xlabel('Timpul n'); ylabel('Amplituda');  title('semnalul de intrare');  subplot(2, 1, 2);  plot(n, y);  xlabel('Timpul n'); ylabel('Amplituda');  title('Semnalul de iesire'); |  |

Acest cod generează un semnal sinusoidal de intrare și apoi calculează semnalul de ieșire folosind o combinație de semnale întârziate și neîntârziate.

Se generează un semnal sinusoidal de intrare utilizând funcția cosinus și frecvența specificată de 0.05 Hz. Variabila n reprezintă vectorul de timp, care conține eșantioanele semnalului înregistrate la intervale regulate.

Apoi se realizează filtrarea semnalului de intrare prin aplicarea unei combinații de semnale întârziate și neîntârziate. Mai întâi, semnalul de intrare este prelungit cu zero-uri la început și la sfârșit pentru a obține semnalele întârziate x1[n] și x3[n]. Apoi, semnalul de intrare original este prelungit cu zero-uri în mijloc pentru a obține semnalul neîntârziat x2[n].

Semnalul de ieșire este calculat prin înmulțirea semnalelor x1[n] și x3[n], respectiv a semnalelor x2[n] și x2[n], urmată de adunarea celor două produse. Vectorul de ieșire este stocat în variabila y și este translatat la dreapta cu un eșantion pentru a elimina valorile neutilizate din prelungirea cu zero-uri a semnalului de intrare.

Graficele afișate arată semnalul de intrare și semnalul de ieșire ale unui filtru realizat cu o funcție non-liniară, definite prin:

* Semnalul de intrare este un semnal sinusoidal cu o frecvență de 0,05 Hz (perioadă de 20 secunde).
* Semnalul de ieșire este calculat prin înmulțirea element-cu-element a trei variabile auxiliare: x1[n] = x[n+1], x2[n] = x[n], x3[n] = x[n-1]. Aceste variabile auxiliare sunt utilizate pentru a calcula ieșirea y[n] = x2[n] \* x2[n] + x1[n] \* x3[n].
* Primul grafic afișat arată semnalul sinusoidal de intrare în funcție de timp (n).
* Al doilea grafic afișat arată semnalul de ieșire în funcție de timp (n).

Programul 5\_3

În programul acesta este realizat sistemul reprezentat prin următoarea expresie: y[n]-0.4y[n-1]+0.75y[n-2]=2.2403x[n]+2.4908x[n-1]+2.2403x[n-2].

Se generează trei secvenţe de intrare x1[n], x2[n] и x[n]=a\*x1[n]+b\*x2[n]. Se calculează trei secvenţe de ieşire y1[n], y2[n] и y[n], cu reprezentarea lor grafică.

|  |  |
| --- | --- |
| % Programul P2\_3  % Generarea secven?elor de iesire  clf;  n=0:40;  a=2; b=-3;  x1=cos(2\*pi\*0.1\*n);  x2=cos(2\*pi\*0.4\*n);  x=a\*x1+b\*x2;  num=[2.2403 2.4908 2.2403];  den=[1 -0.4 0.75];  ic=[0 0]; % initializarea  y1=filter(num,den,x1,ic); % Calcularea semnalului de iesire y1[n]  y2=filter(num,den,x2,ic); % Calcularea semnalului de iesire y2[n]  y=filter(num,den,x,ic); % Calcularea semnalului de iesire y[n]  yt=a\*y1+b\*y2;  d=y-yt; % calcularea abaterii d[n]  %Vizualizarea semnalelor de iesire si a semnalului de abatere  subplot(3,1,1);  stem(n,y);  ylabel('Amplituda ');  title('Semnalul de iesire, conform semnalului de intrare incarcat : a\cdot x\_{1}[n] + b \cdot x\_{2}[n]');  subplot(3,1,2);  stem(n,yt);  ylabel('Amplituda');  title('semnalul de iesire incarcat : a \cdot y\_{1}[n] + b \cdot y\_{2}[n]');  subplot(3,1,3);  stem(n,d);  xlabel('Timpul n'); ylabel('Amplituda');  title('semnalul diferential '); |  |

Acest cod generează trei semnale de ieșire (y1, y2 și y) și un semnal de abatere (d) pe baza a două semnale de intrare (x1 și x2), a unor coeficienți a și b și a unui filtru FIR definit prin coeficienții num și den.

Graficele afișate prezintă semnalul de ieșire y, semnalul de ieșire estimat yt și semnalul de abatere d în funcție de timp.

* Primul grafic arată semnalul de ieșire y ca o funcție a timpului
* Al doilea grafic arată semnalul de ieșire estimat yt ca o funcție a timpului
* Al treilea grafic arată semnalul de abatere d ca o funcție a timpului

Toate graficele sunt reprezentate utilizând funcția stem din Matlab, care afișează o diagramă a semnalului discretizat, cu valori afișate ca bare verticale.

Programul 5\_4

Pentru această expresie se crează sistemul ce o simulează:

y[n]-0.4y[n-1]+0.75y[n-2]=2.2403x[n]+2.4908x[n-1]+2.2403x[n-2].

Două secvenţe de intrare diferite x[n] şi x[n-D]. Se calculează şi se vizualizează corespunzător doua secvenzede ieşire şi abaterea y1[n] – y2[n+D].

|  |  |
| --- | --- |
| % Programul P2\_4  % Generarea secventelor de intrare  clf; n=0:40; D=10; a=3.0; b=-2;  x=a\*cos(2\*pi\*0.1\*n)+b\*cos(2\*pi\*0.4\*n);  xd=[zeros(1,D) x];  num=[2.2403 2.4908 2.2403]; den=[1 -0.4 0.75];  ic=[0 0]; % setarea conditiilor initiale  %Calcularea semnalului de iesire y[n]  y=filter(num,den,x,ic);  % Calcularea semnalului de iesire yd[n]  yd=filter(num,den,xd,ic);  % Calcularea semnalului deabatere d[n]  d=y-yd(1+D:41+D);  %Afisarea graficelor semnalelor de iesire  subplot(3,1,1);  stem(n,y);  ylabel('Amplituda');  title('Semnalul de iesire y[n]'); grid;  subplot(3,1,2);  stem(n,yd(1:41));  ylabel('Amplituda');  title(['iesirea,la intirzierea semnalului la intrarex[n',num2str(D),']']); grid;  subplot(3,1,3);  stem(n,d);  xlabel('Timpul n'); ylabel('Amplituda');  title('semnalul de abatere'); grid; |  |

Acest cod generează semnale de intrare x[n] ca o combinație liniară de două semnale sinusoidale de frecvențe diferite. Apoi, filtrează semnalul de intrare și calculează semnalul de ieșire y[n] și semnalul de ieșire yd[n] la care semnalul de intrare este întârziat cu o perioadă de D eșantioane.

În cele din urmă, se calculează semnalul de abatere d[n] ca diferența dintre semnalul de ieșire y[n] și semnalul de ieșire yd[n] întârziat.

Programul 5\_5

Aici se calculează şi se afişează la ecran impulsul răspunsului sistemului, corespunzător expresiei: y[n]-0.4y[n-1]+0.75y[n-2]=2.2403x[n]+2.4908x[n-1]+2.2403x[n-2].

|  |  |
| --- | --- |
| % Programul P2\_5  % calcularea impulsului de raspuns y  clf;  N=40;  num=[2.2403 2.4908 2.2403];  den=[1 -0.4 0.75];  y=impz(num,den,N);  % Reprezentareagrafică a impulsului de raspuns  stem(y);  xlabel('Timpul n'); ylabel('Amplituda');  title('Impulsul de raspuns'); grid; |  |

Acest cod calculează și afișează impulsul de răspuns al unui filtru cu coeficienții specificați în variabilele num și den. Variabila N specifică numărul de eșantioane din impulsul de răspuns și este setată la 40.

Funcția impz este utilizată pentru a calcula impulsul de răspuns și este apoi afișat utilizând funcția stem. Axele sunt etichetate corespunzător și se adaugă o grilă pentru o mai bună vizualizare.

Graficul afișat prezintă amplitudinea impulsului de răspuns pe axa verticală și timpul pe axa orizontală. Impulsul de răspuns reprezintă răspunsul unui sistem la un impuls Dirac, ceea ce indică modul în care sistemul va răspunde la semnale de intrare de tip impuls.

Programul 5\_6

Programul acesta realizează sistemul de 4 conditii: y[n]+1.6y[n-1]+2.28y[n-2]+1.325y[n3]+0.68y[n-4] = 0.06x[n]-0.19x[n-1]+0.27x[n-2]-0.26x[n-3]+0.12x[n-4] şi Sisteme cascade :

Etapa 1: y1[n]+0.9y1[n-1]+0.8y1[n-2]=0.3x[n]-0.3x[n-1]+0.4x[n-2];

Etapa 2: y2[n]+0.7y2[n-1]+0.85y2[n-2]=0.2y1[n]-0.5y1[n-1]+0.3y1[n-2];

|  |  |
| --- | --- |
| % Realizarea cascadelor  clf;  x = [1 zeros(1, 40)]; % Generarea semnalului de intrare  n = 0:40;  % Coeficientii sistemului de gradul 4  den = [1 1.6 2.28 1.325 0.68];  num = [0.06 -0.19 0.27 -0.26 0.12];  % calcularea semnalului de ieasire asistemului de gradul 4 y=filter(num,den,x);  % Coefocoentii sistemelor de gradul 2  num1 = [0.3 -0.2 0.4]; den1 = [1 0.9 0.8];  num2 = [0.2 -0.5 0.3]; den2 = [1 0.7 0.85];  % Semnalul de iesire y1[n] a primei etape a cascadei  y1 = filter(num1, den1, x);  % Semnalul deiesire y2[n] etapei a doua a cascadei  y2 = filter(num2, den2, y1);  % abaterea dintre y[n] si y2[n]  d = y1 - y2;  % Graficele semnalelor de iesire si a abaterii  subplot(3, 1, 1);  stem(n, y1);  ylabel('Amplituda');  title('afisarea realiz?rii de gradul 4'); grid;  subplot(3, 1, 2);  stem(n, y2);  ylabel('Amplituda');  title('Afisarea realizarii cascada'); grid;  subplot(3, 1, 3);  stem(n, d);  xlabel('Timpul n'); ylabel('Amplituda ');  title('semnalul abaterii'); grid; |  |

Acest program implementează o cascada de doua filtre, unul de gradul 4 și unul de gradul 2. Semnalul de intrare este generat ca un impuls unitar. Coeficienții sistemului de gradul 4 sunt specificați prin vectorul den și num, iar coeficienții sistemului de gradul 2 sunt specificați prin vectorii den1 și num1 și den2 și num2. Semnalele de ieșire ale fiecărui filtru sunt calculate cu funcția filter și sunt stocate în variabilele y1 și y2. Abaterea dintre semnalul de ieșire al primului filtru și cel de-al doilea filtru este stocată în variabila d.

In MATLAB, funcția de filtru este folosită pentru a filtra o secvență de date folosind un filtru digital. Este de obicei folosit pentru a elimina zgomotul sau frecvențele nedorite dintr-un semnal sau pentru a netezi un semnal. Funcția de filtru are trei argumente:

b: coeficienții numărătorului filtrului

a: coeficienții numitorului filtrului

x: datele de intrare care trebuie filtrate

Programul 5\_7

Operaţiunea de convoluţie se realizează în MATLAB cu comanda conv. Pentru aceasta se folosesc două secvenţe finite şi de aceeaşi lungime.

|  |  |
| --- | --- |
| % Programul P2\_7  clf;  h=[3 2 1 -2 1 0 -4 0 3]; % raspuns de impuls  x=[1 -2 3 -4 3 2 1]; % secventa de intrare  y=conv(h,x);  n=0:14;  subplot(2,1,1);  stem(n,y);  xlabel('Timpul n'); ylabel('Amplituda');  title('Semnalul de iesire,obtinut in urma convolutiei'); grid;  x1=[x zeros(1,8)];  y1=filter(h,1,x1);  subplot(2,1,2);  stem(n,y1);  xlabel('Timpul n'); ylabel('Amplituda');  title('Semnalul de iesire obtinut in urma filtrarii'); grid; |  |

Acest cod implementează două metode diferite pentru calcularea semnalului de ieșire al unui sistem liniar, invariant în timp (LTI), reprezentat prin vectorul de răspuns la impuls (h) și semnalul de intrare (x).

În primul rând, se calculează semnalul de ieșire prin convoluție, folosind funcția MATLAB conv(), care efectuează operația de convoluție directă a semnalelor de intrare și de ieșire. Rezultatul convoluției este stocat în vectorul y și apoi afișat pe un grafic, împreună cu etichetele și titlul aferente.

În al doilea rând, semnalul de ieșire este calculat prin filtrare, folosind funcția MATLAB filter(). Această metodă este echivalentă cu convoluția în cazul sistemelor LTI. Intrarea x este prelungită cu zero-uri pentru a asigura că semnalul de ieșire are aceeași lungime cu cel obținut prin convoluție. Rezultatul filtrării este stocat în vectorul y1 și afișat pe un grafic, împreună cu etichetele și titlul aferente.

Graficele afișate compară semnalul de ieșire obținut prin cele două metode de calcul, evidențiind astfel echivalența lor.

Funcția "conv" din Matlab realizează o convoluție între două semnale. Convoluția este o operație matematică care combină două semnale de intrare pentru a produce un al treilea semnal, care poate fi văzut ca o măsură a gradului de suprapunere între semnalele de intrare. În cazul semnalelor continue, convoluția poate fi calculată prin integrarea produsului semnalelor la fiecare moment de timp, iar în cazul semnalelor discrete, convoluția este calculată prin înmulțirea și adunarea valorilor semnalelor la fiecare moment de timp. Funcția "conv" din Matlab este utilizată pentru a efectua convoluția între semnalele discrete.

Programul 5\_8

Programul calculează suma valorilor absolute ale raspunsului inpuls

|  |  |
| --- | --- |
| % Programul P2\_8  % Testarea stabilitatii bazat pe suma valorilor absolute ale elementelor raspuns impuls  clf;  num=[1 -0.8]; den=[1 1.5 0.9];  N=200;  h=impz(num,den,N+1);  parsum=0;  for k=1:N+1;  parsum=parsum+abs(h(k));  if abs(h(k))<10^(-6), break, end  end  % Afisarea r?spuns impuls  n=0:N;  stem(n,h);  xlabel('Timpul n'); ylabel('Amplituda');  disp('Valoarea='); disp(abs(h(k))); %Afisarea valorilor |  |

impz în MATLAB este o funcție folosită pentru a calcula și a reprezenta grafic răspunsul la impuls al unui sistem liniar invariant în timp (LTI). Acesta ia ca intrare coeficienții funcției de transfer a sistemului și returnează coeficienții de răspuns la impuls ca ieșire.

Acest program testează stabilitatea sistemului reprezentat de funcția de transfer dată, bazându-se pe suma valorilor absolute ale elementelor din impulsul de răspuns.

În primul bloc de cod, se definesc coeficienții funcției de transfer și se calculează impulsul de răspuns cu ajutorul funcției impz, folosind un număr N de eșantioane.

În următorul bloc de cod, se initializează o variabilă parsum cu valoarea 0, care va fi utilizată pentru a calcula suma valorilor absolute ale elementelor impulsului de răspuns. Se parcurge impulsul de răspuns și, pentru fiecare element, se adaugă valoarea absolută la suma parsum. De asemenea, se verifică dacă valoarea absolută a elementului curent este mai mică decât o anumită valoare foarte mică (10^(-6) în acest caz). Dacă da, se iese din bucla for, ceea ce indică faptul că elementele rămase au valori foarte apropiate de 0 și, prin urmare, suma lor nu ar avea un impact semnificativ asupra stabilității sistemului.

În ultima parte a programului, se afișează impulsul de răspuns cu ajutorul funcției stem și se afișează valoarea elementului care a condus la ieșirea din bucla for.

Un sistem liniar invariant în timp (SLIT) este un tip de sistem în teoria sistemelor care îndeplinește două proprietăți importante: liniaritate și invarianța în timp.

Liniaritatea înseamnă că dacă un sistem răspunde la o anumită intrare cu un anumit răspuns, atunci dacă intrarea este scalată (înmulțită cu o constantă), răspunsul sistemului va fi de asemenea scalat în același mod.

Invarianța în timp înseamnă că răspunsul sistemului la o anumită intrare va fi același, indiferent de momentul în care intrarea este aplicată la sistem.

Programul 5\_9

Aici se prezintă două sisteme: y[n] = 0.5x[n]+0.27x[n-1]+0.77x[n-2] şi y[n] = 0.45x[n] +0.5x[n-1]+0.45x[n-2]+0.53y[n-1]-0.46y[n-2].

|  |  |
| --- | --- |
| % Programul P2\_9  % Generarea semnalului deintare  clf;  n=0:299;  x1=cos(2\*pi\*10\*n/256);  x2=cos(2\*pi\*100\*n/256);  x=x1+x2;  % Calcularea secven?elor de iesire  num1=[0.5 0.27 0.77];  y1=filter(num1,1,x); % Iesirea sistemului #1  den2=[1 -0.53 0.46];  num2=[0.45 0.5 0.45];  y2=filter(num2,den2,x); % Iesirea sistemului #2  % Graficele secventelor de iesire  subplot(2,1,1);  plot(n,y1); axis([0 300 -2 2]);  ylabel('Amplituda');  title('semnalul de iesire #1'); grid;  subplot(2,1,2);  plot(n,y2); axis([0 300 -2 2]);  xlabel('Timpul n'); ylabel('Amplituda');  title('semnalul de iesire #2'); grid; |  |

Acest program generează două semnale de intrare și calculează semnalele de ieșire a două sisteme filtru diferite. Semnalul de intrare este suma a două semnale cosinusoidale, una cu frecvență de 10 Hz și cealaltă cu frecvență de 100 Hz.

Primul sistem filtru are coeficienții [0.5 0.27 0.77] și se obține semnalul de ieșire y1 prin filtrarea semnalului de intrare cu acești coeficienți, folosind funcția filter.

Al doilea sistem filtru are coeficienții [0.45 0.5 0.45] și [1 -0.53 0.46] și se obține semnalul de ieșire y2 prin filtrarea semnalului de intrare cu acești coeficienți, folosind din nou funcția filter.

**Concluzii:**

În concluzie, aceasta lucrare de laborator ne-a oferit o perspectivă amplă asupra modului în care putem crea diferite tipuri de sisteme în conformitate cu cerințele inițiale. Am învățat despre diferențele dintre sistemele discrete și cele continue, precum și avantajele și dezavantajele fiecărui tip de sistem. De asemenea, am studiat diverse tehnici de proiectare a sistemelor, inclusiv teoria semnalelor, transformatele Fourier și Laplace și metode de analiză a sistemelor în domeniul timpului și frecvenței.

Înțelegerea noțiunilor cheie precum sistem, semnal, frecvență și transformate ne-a permis să creăm și să analizăm diferite tipuri de sisteme, precum filtre digitale, amplificatoare și controlere. Am învățat despre modul în care putem proiecta aceste sisteme pentru a îndeplini anumite cerințe, cum ar fi atenuarea zgomotului sau creșterea amplificării semnalului.

În concluzie, lucrarea ne-a oferit o înțelegere solidă a teoriei sistemelor discrete și a modului în care putem aplica această teorie pentru a crea sisteme care să îndeplinească cerințele noastre. Această cunoaștere ne va fi utilă într-o varietate de domenii, cum ar fi ingineria electronică, ingineria electrică, ingineria de control și multe altele.