Universitatea Tehnica din Republica Moldova

Facultatea Calculatoare, Informatica si Microelectronica

Departament Inginerie Software si Automatica

Specialitatea Tehnologia Informației

Raport

Curs: Prelucrarea semnalelor

Tema: Formarea semnalelor elementare in sistemul Matlab

A elaborat: Reguș Ruslan Grupa: TI-214

A verificat: Asist. Univ. Cazac A.

Chișinău 2024

**Lucrare de laborator nr. 2**

**Obiective:** studierea posibilităților sistemului MATLAB în modelarea diferitor forme de semnale în vederea cercetării particularităților lor de bază.

1. Să se genereze şi să se reprezinte următoarele secvenţe.

|  |
| --- |
| x1[n]=0.8 δ[n], -15≤n≤15 |
| n=-15:15;  y=zeros(size(n));  y(16)=0.8;  figure(1);  stem(n,y)  O secvență de eșantioane de la -15 la 15 și apoi atribuie valorile zero la fiecare eșantion în variabila y utilizând funcția zeros(size(n)).  Mai apoi, se atribuie valoarea de 0.8 elementului 16 al lui y, ceea ce face ca elementul de pe poziția 1 a axei OY să fie de 0.8 atunci când se va afișa graficul.În cele din urmă, se creează o figură cu numărul 1 și se afișează graficul utilizând funcția stem(), care desenează un grafic de tipul "tulpinilor" pentru datele furnizate. În acest caz, stem() va afișa un punct marcat cu o linie verticală la poziția 16 și la înălțimea 0.8 pe axa OY, iar toate celelalte puncte de pe axa OY vor fi situate la zero. |

|  |
| --- |
| x2[n]=0.9 δ [n-5], 1≤n≤20 |
| n=1:20;  y=zeros(size(n-5));  y(5)=0.9;  figure; stem(n,y);  xlabel('n'); ylabel('y(n)') |

|  |
| --- |
| x3[n]=1.5 δ [n-333], 300≤n≤350 |
| n=300:350;  y=zeros(size (n-333));  y(34)= 1.5;  figure; stem(n,y);  xlabel('n'); ylabel('y(n)') |

|  |
| --- |
| x4[n]=4.9 δ [n+7], -10≤n≤0 |
| n=-10:0;  y=zeros(size(n+7));  y(4)=4.9;  figure;  stem(n,y);  xlabel('n');  ylabel('y(n)'); |
| x5[t]=4 u[t], -10≤t≤10 |
| t=-10:10;  U=ones(size(t));  U(t<0)=0;  U=U\*4;  plot(t,U);  xlabel('t,s');ylabel('Amplitudine');  title('Treapta unitara'); |

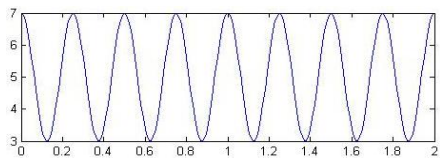
|  |
| --- |
| x6[n]=1.4 u[n-7], -5≤n≤20 |
| n=-5:20;  U=ones(size(n-7));  U((n-7)<0)=0;  U=U\*1.4;  plot(t,U);  xlabel('n,s');ylabel('Amplitudine');  title('Treapta unitara'); |
| x7[n]=2.3 u[n+5], -15≤n≤10 |
| n=-15:10;  U=ones(size(n+5));  U((n+5)<0)=0;  U=U\*2.3;  plot(n,U);  xlabel('n,s');ylabel('Amplitudine');  title('Treapta unitara'); |

Să se construiască un tren de impulsuri unitare periodice.

x1[n]=0.8 δ[n], -15≤n≤15

|  |
| --- |
| P=1; M=30;  d=[0.8;zeros(P-1,1)]; % generarea impulsului generator, de lungime P  y=d.\*ones(1,M);  tren=y(:); % generarea trenului de impulsuri de lungime P\*M  t=-15:M\*P-16;  stem(t, tren);  xlabel('t');  ylabel('Amplitudine');  title('Tren de impulsuri unitate periodice');  axis([-16 16 0 1.2]);  grid  Această secvență de cod MATLAB creează un tren de impulsuri unitate periodice, de perioadă P și lungime M. Variabilele P și M sunt definite la începutul secvenței.Mai apoi, se creează un vector coloană d, care conține un impuls de amplitudine 0.8 în prima sa poziție și zero în restul pozițiilor. Acest vector va fi utilizat pentru a construi trenul de impulsuri.Variabila y este construită prin extinderea vectorului d pe o matrice de dimensiune PxM, astfel încât fiecare impuls de amplitudine 0.8 să fie separat de celelalte prin P-1 valori nule. Această matrice este apoi transformată într-un vector coloană tren prin aplicarea operatorului colon (:) care concatenează toate elementele matricei.Variabila t este definită ca un vector care conține toate eșantioanele trenului de impulsuri. Acesta începe de la -15 și se termină la M\*P-16, deoarece lungimea trenului de impulsuri este M, iar fiecare impuls este separat de P-1 valori nule. De asemenea, sunt setate limitele axei y, pentru a afișa impulsurile în întregime și pentru a exclude eventualele valori care ar depăși amplitudinea maximă a impulsurilor.În cele din urmă, se afișează graficul utilizând funcția stem(), care desenează un grafic de tipul "tulpinilor" pentru datele furnizate. În acest caz, stem() va afișa un impuls marcat cu o linie verticală la fiecare perioadă, iar toate celelalte puncte de pe axa OY vor fi situate la zero. De asemenea, se adaugă etichete pentru axele x și y, cu ajutorul funcțiilor xlabel() și ylabel(). Se adaugă un titlu graficului cu ajutorul funcției title() și se afișează o grilă pentru o mai bună vizualizare a impulsurilor. |

1. Să se genereze în MATLAB următorul semnal sinusoidal.



|  |
| --- |
| t=0:0.001:2; % intervalul de timp  A=2; % amplitudinea  f=1/(2/8); % frecvența  p=5; % o valoare de amplitudine constantă adăugată la semnalul sinusoidal  x=A\*cos(2\*pi\*f\*t)+p; % o funcție sinusoidală  plot(t,x); |

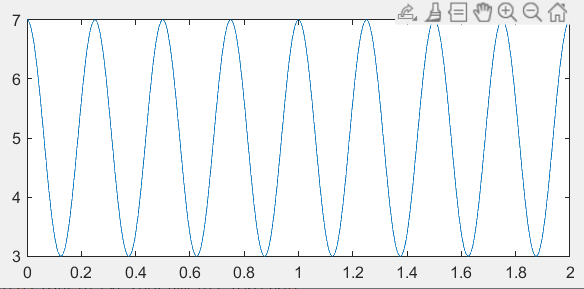
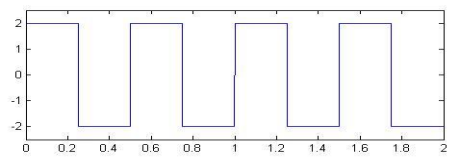


Figura 4 – Execuția semnalului sinusoidale

1. Să se genereze în MATLAB următorul semnal dreptunghiular.



|  |
| --- |
| t=0:0.001:2;  d=[0.125:0.5:1.625];  y= 4\*pulstran(t,d, 'rectpuls', 0.2)-2;  plot(t, y), grid |

d- secvență de puncte de pe axa timpului în care semnalul este ne-zero.

"**pulstran**" este utilizată pentru a genera semnalul dreptunghiular pulsatoriu.

„**rectpuls**” specifică tipul de impuls (în acest caz triunghiular), iar valoarea 0.2 specifică lățimea impulsului triunghiular

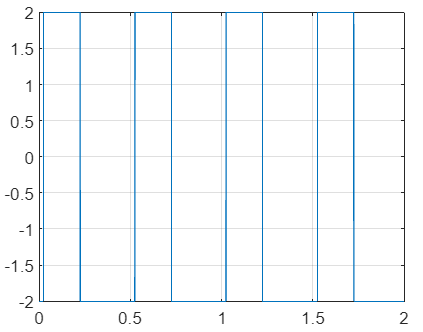
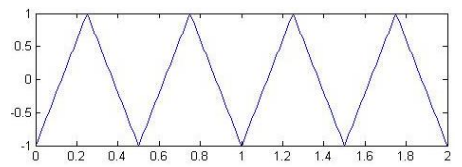


Figura 5 – Execuția semnalului dreptunghic

1. Să se genereze în MATLAB următorul semnal triunghiular.



|  |
| --- |
| t = 0:0.01:2;  d = 0.25:0.5:1.75;  y = 2\*pulstran(t,d,'tripuls',0.5)-1;  figure(1);  plot(t,y); grid; |

d- secvență de puncte de pe axa timpului în care semnalul este ne-zero.

y- o funcție triunghiulară pulsată cu o lățime de 0.5 unități de timp și scalată la amplitudinea 2

"**pulstran**" este utilizată pentru a genera semnalul triunghiular pulsatoriu.

„**tripuls**” specifică tipul de impuls (în acest caz triunghiular), iar valoarea 0.5 specifică lățimea impulsului triunghiular

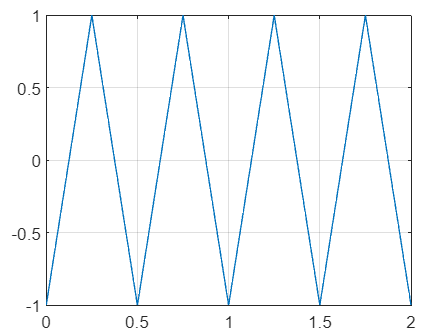


Figura 6 – Execuția semnalului triunghiular

1. Să se creeze în intervalul 0s≤t≤256s o oscilaţie armonică cu amplitudinea unitate, perioada T=50 sec și faza inițială 𝜋/3.

|  |
| --- |
| t=0:256;  A=1; % amplitudinea  p=50; % perioada  f=pi/3; % faza initiala  x=A\*sin(2\*pi\*f\*t+p); % functia oscilatiei armonice  plot(t,x); % afisarea graficului |

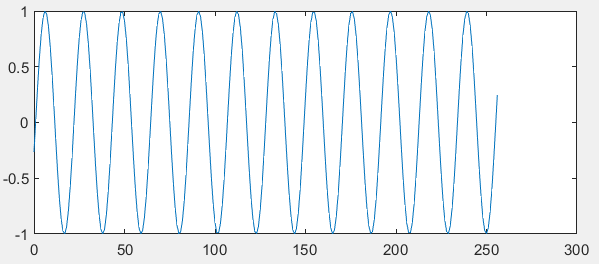


Figura 7 – Execuția oscilaţiei armonice

1. Să se creeze în intervalul 0s≤t≤ls următoarele semnale exponenţiale:

**a) 5exp(-6t);**

**b) exp(5t),**

folosind frecvenţa rezoluției temporale fd=1000Hz.

|  |
| --- |
| t=0:0.001:1;  a=5\*exp(-6\*t); %functie exponentiala descrescatoare  b=exp(5\*t); %functie exponentiala crescatoare  figure(1);  plot(t,a);  figure(2);  plot(t,b); |

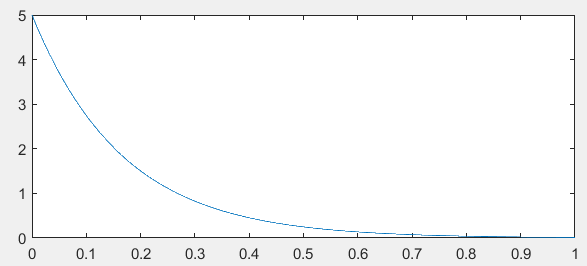


Figura 8 – Execuția semnalului exponențial **5exp(-6t)**

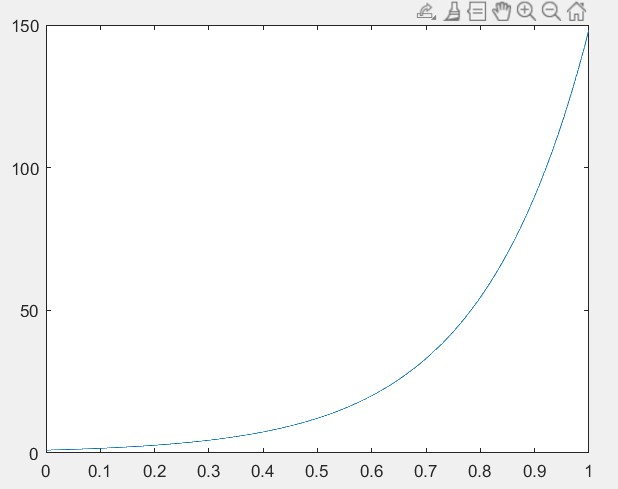


Figura 9 – Execuția semnalului exponențial **exp(5t),**

1. Să se creeze în intervalul -10s≤t≤10s un impuls exponenţial descris de următoarea expresie x(t)=Brt , unde B=l, r=0.8.

|  |
| --- |
| t=-10:10;  r = 0.8; % coeficient de creștere exponențiala  b = 1; % defineste amlitudinea  x = b\*r.^t; % funcție exponențială discret  plot(t,x);  Factorul de creștere exponențială este mai mic decât 1, ceea ce înseamnă că valoarea funcției va scădea exponențial pe măsură ce t crește. |

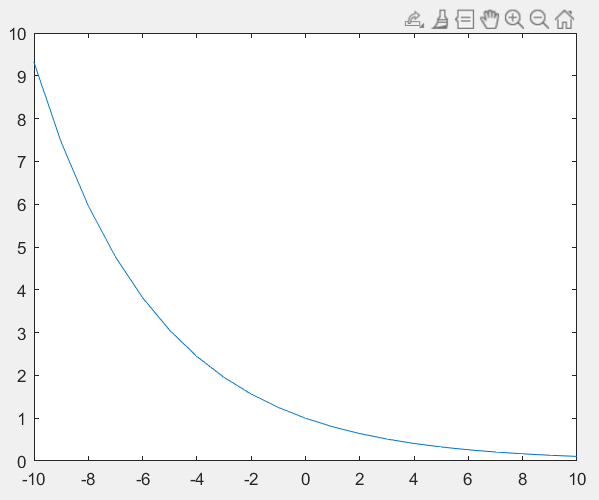


Figura 10 – Execuția impulsului exponențial

1. Să se creeze în intervalul -10s≤t≤10s un semnal sinusoidal dat de expresia:



|  |
| --- |
| t = -10:0.01:10;  x = 2\*sin(2\*pi\*t/12); // functia semnalului sinusoidal  figure(1);  plot(t,x), grid; |

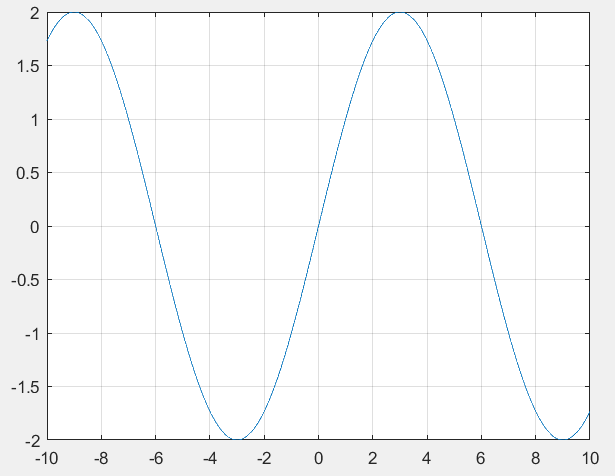


Figura 11 – Execuția semnalului sinusoidal

1. Să se creeze un semnal sinusoidal atenuat, pe baza înmulţirii exponentei atenuatoare formate în punctul 7 şi a semnalului sinusoidal creat în punctul 8, ambele obținute pentru intervalul - 10s≤t≤10s.

|  |
| --- |
| t=-10:0.01:10;  r = 0.8; b = 1;  x = b\*r.^t;  x1 = 2\*sin(2\*pi\*t/12);  x2=x1.\*x;  plot(t,x2); |

Creează o variabilă t care conține o secvență de eșantioane de la -10 la 10. Mai apoi, se creează o variabilă U cu ajutorul funcției ones(), care creează un vector de dimensiune egală cu lungimea vectorului t și umple toate elementele cu 1.Apoi, se setează toate valorile din U la 0, pentru toate eșantioanele negative din t, utilizând operatorul < și asignând valorile zero elementelor corespunzătoare din U. Acest lucru se realizează pentru a crea o treaptă unitară, care este zero pentru valori negative și 1 pentru valorile pozitive sau nule ale timpului.Mai apoi, variabila U este multiplicată cu 4, ceea ce mărește amplitudinea treptei unitare.În cele din urmă, se afișează graficul cu ajutorul funcției plot(), care desenează o linie continuă a timpului t pe axa x și o linie a amplitudinii U pe axa y. Se adaugă etichete pentru axele x și y, cu ajutorul funcțiilor xlabel() și ylabel(). De asemenea, se adaugă un titlu graficului cu ajutorul funcției title().

Un **semnal sinusoidal** **atenuat** este un semnal sinusoidal căruia i s-a redus amplitudinea în timp prin trecerea sa printr-un mediu care îl atenuează.

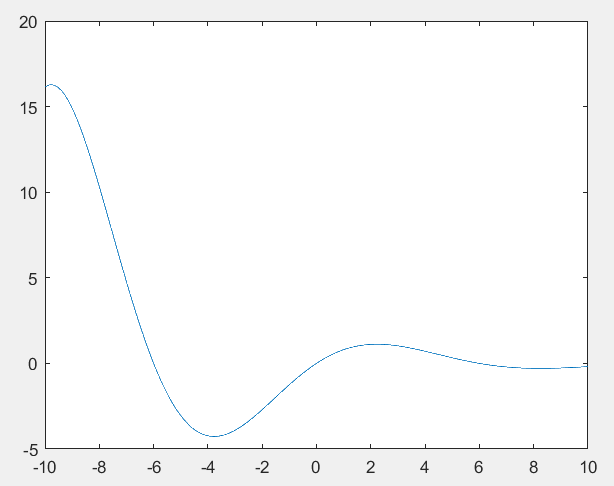


Figura 12 – Execuția semnalului sinusoidal atenuat

1. Să se creeze un impuls dreptunghiular de amplitudine unitate şi durata 1s, amplasat simetric față de originea de coordonate t=0 (-0.5s≤𝜏≤0.5s) descris în intervalul de timp -1s≤t≤1s, utilizând rezoluția temporală t=2ms. Sugestie: impulsul dreptunghiular poate fi creat cu ajutorul diferenţei a două funcţii de tip "treaptă unitară" deplasate în timp cu un interval egal cu durata impulsului.

|  |
| --- |
| t=-1:0.002:1;  x=rectpuls(t,1);  plot(t,x); |

“t” - vector de timp, care variază de la -1 la 1, în pași de 0,002. Aceasta înseamnă că vectorul t conține 1001 de elemente (o durată totală de 2/0,002 + 1 = 1001).

**„rectpuls”** generează un semnal dreptunghiular de durată 1 și amplitudine 1 pentru fiecare interval de timp dat.

**„plot(t,x)”** afișează semnalul dreptunghiular generat în funcția anterioară, în raport cu vectorul de timp t. Acest grafic va arăta semnalul dreptunghiular cu durata de 1 unitate de timp, cu valoarea 1 pe intervalul de la -0,5 la 0,5 și valoarea 0 în restul timpului.

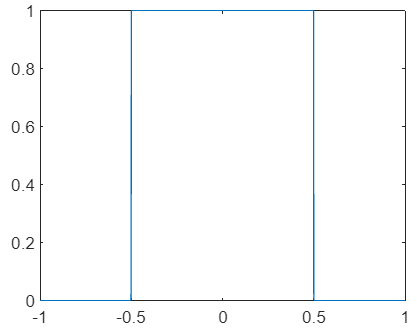


Figura 12 – Execuția impuls dreptunghiular

1. A forma o succesiune periodică de impulsuri dreptunghiulare cu amplitudinea A=±l, viteza unghiulară ω=π/4 și coeficientul de umplere 30% în intervalul -10s≤t≤10s.

|  |
| --- |
| t=-10:0.002:10;  x=square(t\*pi/4, 30);  plot(t,x); |

**„square(t\*pi/4,30)”** pentru a genera o succesiune periodică de impulsuri dreptunghiulare.

„**t\*pi/4”** reprezintă unghiul de fază și determină la ce moment începe semnalul dreptunghiular.

„**plot(t,x)”** afișează semnalul dreptunghiular generat în funcția anterioară, în raport cu vectorul de timp t.

Acest grafic va arăta semnalul dreptunghiular cu o frecvență de 8 Hz și o durată de 30% din perioada semnalului, începând de la un sfert din perioada semnalului.

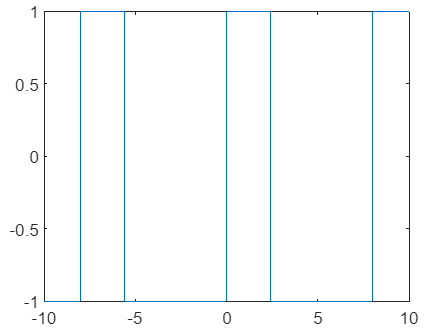
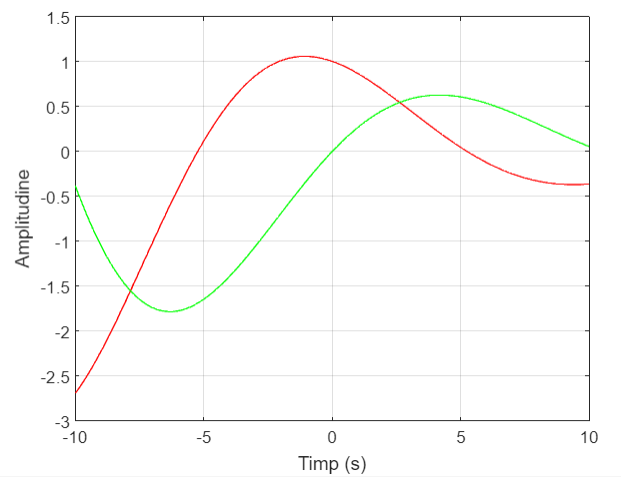


Figura 13 – Execuția unei succesiuni periodice de impulsuri dreptunghiulare

1. Pe intervalul -10s≤t≤10s să se reprezinte grafic în aceeași figură, în sisteme de coordonate diferite un semnal exponenţial complex x(t)=exp((-0.1+j0.3)t)

|  |
| --- |
| t=-10:0.1:10;  x=exp((-0.1+i\*0.3)\*t); % creează un semnal exponențial și o constantă complexă (-0.1 + i\*0.3), înmulțită cu vectorul "t"  plot(t,real(x), 'r'); % va afisa un semnal exponential complex a componentei reale, in culoare rosu  hold % pentru a adăuga ambele grafice la aceeași figură  plot(t, imag(x), 'g'); % va afisa un semnal exponential complex a componentei imaginare, in culoare verde  grid % se adaugă o grilă la figură  xlabel('Timp (s)');  ylabel('Amplitudine'); |



Cu funcţia **plot** se pot reprezenta grafic semnale sau funcţii “continue”, deoarece se unesc cu linie continuă valorile care se reprezintă.

Funcția **plot** este folosită pentru a reprezenta date continue

Funcția **stem** este folosită pentru a reprezenta date discrete/concrete.

**Semnalul dreptunghiular** poate fi afectat de distorsiuni și pierderi de semnal, în funcție de calitatea echipamentelor utilizate și de factorii de zgomot din mediu. În aceste situații, semnalul poate să nu fie identic cu cel dintr-un sistem ideal.

**Semnalul dreptunghiular** este un **semnal analogic**, care poate fi descris prin caracteristici precum amplitudinea, durata, frecvența și ciclul de lucru. Semnalul este generat ca o succesiune de impulsuri dreptunghiulare de aceeași durată și amplitudine, separate de perioade egale.

În schimb, **semnalul digital** este un **semnal discret**, care poate fi reprezentat ca o serie de cifre binare sau de valori discrete. Semnalul digital este utilizat în principal în sistemele electronice și de comunicații pentru a transmite informația între diferite dispozitive digitale.

În ciuda diferențelor lor fundamentale, semnalul dreptunghiular și semnalul digital au o asemănare în ceea ce privește forma lor de undă. Semnalul dreptunghiular poate fi utilizat ca un semnal de bază pentru a genera semnale digitale prin modulare sau transformare.

Astfel, prin aplicarea unei tehnici de modulare, cum ar fi modularea în amplitudine sau modularea în frecvență, semnalul dreptunghiular poate fi transformat într-un semnal digital. În plus, semnalele digitale pot fi reprezentate grafic sub forma unei serii de impulsuri dreptunghiulare, asemănătoare cu semnalul dreptunghiular.

**Concluzie:**

Efectuând această lucrare de laborator am îsușit noțiunile de impuls unitar, funcție impuls Dirac(funcția delta). Am studiat cum se generează semnalele în MATLAB, cum se generează impulsurile unitare, impulsurile unitare dreptunghiulare, impulsurile unitare triunghiulare, oscilații armonice, semnale exponențiale etc