

Semnale și sisteme

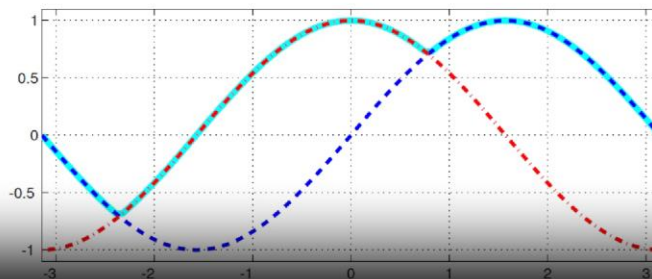
Lucrare de laborator nr. 2

Reprezentarea grafica a semnalelor in Matlab

Reprezentarea semnalelor unidimensionale

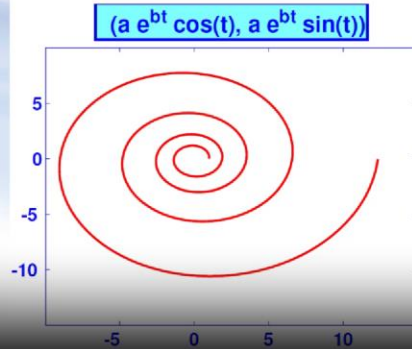
Funcția $plot(X,Y)$ generează un grafic bidimensional în care sunt reprezentate datele din Y în funcție de datele din X . Apelarea funcției cu un singur parametru, $plot(Y)$, va reprezenta datele din Y în funcție de indicii acestora (poziția în vectorul Y). Utilizarea parametrului suplimentar *LineSpec* facilitează modificarea culorii, specificarea tipului de linie, și a unui simbol. Spre exemplu, 'g:*' specifică linie verde punctată și simbol *. Funcția permite de asemenea specificarea altor caracteristici ale liniilor precum *LineWidth*, și ale simbolurilor precum *MarkerEdgeColor*, *MarkerFaceColor* și *MarkerSize*.

```
x = -pi:0.02*pi:pi;  
plot(x, max(sin(x), cos(x)), 'c', 'LineWidth', 5); hold on  
plot(x, sin(x), 'b', x, cos(x), 'r', 'LineWidth', 3); grid  
axis([-pi, pi, -1.1, 1.1]), set(gca, 'FontSize', 12)
```



Reprezentarea semnalelor unidimensionale

```
a=1; b=.1;
t = 0:0.01:8*pi;
x = a*exp(b*t).*cos(t);
y = a*exp(b*t).*sin(t);
figure, plot(x,y, 'r', 'LineWidth', 2), box on
set(gca, 'FontSize', 16, 'FontWeight', 'Bold', ...
    'XColor', [0 0 1], 'YColor', [0 0 1], ...
    'XTick', [-5 0 5 10], 'YTick', [-10 -5 0 5])
title('\fontsize{20}\color{blue}(a e^{bt} cos(t) a e^{bt} sin(t))', ...
    'BackgroundColor', [0.5 1 1], ...
    'EdgeColor', 'b', 'LineWidth', 2);
```



Reprezentarea semnalelor unidimensionale

Functia `fplot(fun,limits)` reprezinta grafic o functie *fun* de forma $y = f(x)$ in intervalul specificat prin *limits*. *fun* poate fi numele unei functii, un identificator de functie sau o expresie reprezentata ca si string care specifica functii de variabila *x*. Functia accepta parametri suplimentari precum *n* (afisare cu un numar minim de $n+1$ puncte si un pas maxim restrictionat la $(1/n) \cdot (x_{max} - x_{min})$), *Linespec*. Implementeaza un control adaptiv al pasului, utilizand un pas mai mic in regiunile cu rata de schimbare mai mare a valorilor functiei.

Functia `subplot(m,n,p)` imparte figura curenta intr-o retea dreptunghiulara de dimensiune $m \times n$ si creeaza axe pentru un grafic la pozitia specificata de *p*. Accesarea unei regiuni a retelei aflata pe pozitia (*l*, *c*) poate fi realizata prin specificarea indicelui linar $p = (l - 1) \cdot m + c$.

O functie anonima este o functie care permite definirea unei singure expresii si care nu se stocheaza intr-un fisier functie, fiind asociata unei variabile de tip *function handle* (identificator de functie). Un identificator de functie este un tip de date care stocheaza o asociere la o functie si care furnizeaza o modalitate de apel indirect al functiei. Spre exemplu, definirea functiei $f(x) = 2x^2 + 0.2|x| + 20$ se poate realiza cu comanda:

```
f = @(x) (2*x.^2 + 0.2*abs(x) + 20)
```

iar evaluarea functiei pentru anumite valori ale lui *x*:

```
x = -1:0.1:1
result = f(x)
```

Semnale elementare continue

```
% semnal treapta u(t) = (t>=0) ? 1 : 0;
t1 = -2; t2 = 6; tstep = 0.001;
t = t1:tstep:t2;
x = (t>=0);
plot(t, x)
xlabel('t'); ylabel('u(t)'); title('Semnal treapta unitara');
axis([t1-tstep, t2+tstep, -0.1, 1.1])
```

```
x = @(t) (t>=0);
t1 = -2; t2 = 6;
t = [t1, t2];
fplot(x, t)
xlabel('t'); ylabel('u(t)');
axis([t -0.1 1.1])
title('Semnal treapta unitara')
```

```
% semnal puls p(t) = ((t>=0)&&(t<=e)) ? 1/e : 0;
e = 1/100;
t1 = -1; t2 = 5; tstep = 0.005;
t = t1:tstep:t2;
x = (1/e)*((t>0)&&(t<=e));
plot(t, x)
xlabel('t'); ylabel('p-\epsilon(t)');
title('Semnal puls, \epsilon = 1/100')
```

```
e = 1/100;
x = @(t)(1/e)*((t>0)&&(t<=e));
t1 = -1; t2 = 5;
t = [t1, t2];
fplot(x, t, 5000);
set(gca, 'FontSize', 14)
xlabel('t'); ylabel('p-\epsilon(t)');
axis([t1 t2 -0.1 1.1/e])
title('Semnal puls, \epsilon = 1/100')
```

0:07 / 0:21

Semnale elementare continue

```
% semnal sinusoidal x(t) = A sin(2 pi t + phi);
A = 2; T = 1/10; phi = pi/3;
t1 = -0.1; tstep = 0.005; t2 = 0.2;
t = t1:tstep:t2;
xs = A*sin(2*pi/T*t + phi);
plot(t, xs, 'LineWidth', 2); grid;
xlabel('t'); ylabel('x_s(t)');
title('x_s(t)=A sin((2 \pi /T) t + \phi)')
```

```
A = 2; T = 1/10; phi = pi/3;
xs = @(t)(A*sin(2*pi/T*t + phi));
t1 = -0.1; t2 = 0.2;
t = [t1, t2];
fplot(xs, t); grid
xlabel('t'); ylabel('x_s(t)');
title('x_s(t)= A sin((2 \pi /T) t + \phi)')
```

```
% semnal exponential
a = 0.2; b = -0.5;
t1 = 0; tstep = 0.5; t2 = 8;
t = t1:tstep:t2;
xe = a*exp(b*t);
plot(t, xe); grid
xlabel('t'); ylabel('x_e(t)');
title('x_e(t) = a e^{-bt}')
```

```
a = 0.2; b = -0.5;
xe = @(t)(a*exp(b*t));
t1 = 0; t2 = 8;
t = [t1, t2];
fplot(xe, t); grid
xlabel('t'); ylabel('x_e(t)');
title('x_e(t) = a e^{-bt}')
```

0:09 / 0:21

Semnale elementare discrete

```
% secventa treapta unitara u[n] = (n>=0) ? 1:0;
n1 = -5; n2 = 10;
n = n1:n2;
x = (n>=0);
stem(n, x);
xlabel('n'); ylabel('u[n]');
title('Secventa treapta unitara')
axis([n1 n2 -0.1 1.1]);

n1 = -5; n2 = 10; n = n1:n2;
x = @(n) (n>=0);
stem(n, x(n));
xlabel('n'); ylabel('u[n]');
title('Secventa treapta unitara')
```

```
% semnal sinusoidal discret
A = 2; Ns = 10; phi = pi/3;
n1 = -5; n2 = 10; n = n1:n2;
xs = A*sin(2*pi*(1/Ns)*n + phi);
stem(n, xs)
xlabel('n'); ylabel('x_s[n]')
title('x_s[n] = A sin(2\pi (1/N_s) n + \phi)')

A = 2; Ns = 10; phi = pi/3;
n1 = -5; n2 = 10; n = n1:n2;
xs = @(n) (A*sin(2*pi*(1/Ns)*n + phi));
stem(n, xs(n))
xlabel('n'); ylabel('x_s[n]')
title('x_s[n] = A sin(2\pi (1/N_s) n + \phi)')
```

```
% secventa impuls delta[n] = (n==0) ? 1:0;
n1 = -5; n2 = 10; n = n1:n2;
x = (n==0);
stem(n, x);
xlabel('n'); ylabel('\delta_n');
title('Secventa impuls unitar')
axis([n1 n2 -0.1 1.1]);

n1 = -5; n2 = 10; n = n1:n2;
x = @(n) (n==0);
stem(n, x(n));
xlabel('n'); ylabel('\delta_n');
title('Secventa impuls unitar')
```

```
% semnal exponential discret
a = 0.2; b = 0.5;
n1 = 0; n2 = 10; n = n1:n2;
xe = a*b.^n;
stem(n, xe)
xlabel('n'); ylabel('x_e[n]'); title('x_e[n] = a b^n')

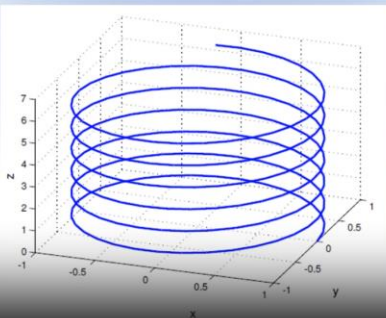
a = 0.2; b = 0.5;
n1 = 0; n2 = 10; n = n1:n2;
xe = @(n) (a*b.^n);
stem(n, xe(n))
xlabel('n'); ylabel('x_e[n]'); title('x_e[n] = a b^n')
```

0:11 / 0:21

Reprezentarea semnalelor bidimensionale

Funcția $plot3(X, Y, Z)$ generează un grafic 3D al unui set de puncte cu coordonate specificate prin matricile X , Y și Z . În mod similar funcției $plot$, permite particularizarea proprietăților liniilor.

```
t = 0:0.01*pi:2*pi;
x = cos(2*pi*t); y = sin(2*pi*t); z = t;
plot3(x,y,z, 'LineWidth', 2); grid; view(20,30)
xlabel('x'), ylabel('y'), zlabel('z'),
```

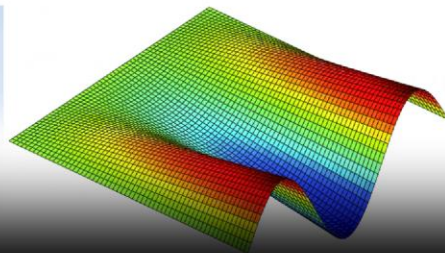


0:19 / 0:21

Reprezentarea semnalelor bidimensionale

Funcția *surf*(X,Y,Z) permite reprezentarea datelor ca și suprafețe definite de coordonatele z specificate în nodurile unei rețele de discretizare din planul $x-y$. Algoritmul de umbrire al suprafeței și harta de culoare utilizate pentru afișare pot fi modificate cu ajutorul funcțiilor *shading*, respectiv *colormap*. Generarea unei astfel de rețele de puncte (grid) în planul $x-y$ este facilitată de funcția *meshgrid*.

```
[t,p]=meshgrid(0:.02*pi:3*pi,0:.02*pi:3*pi);  
figure, surf(t,p,t/2.*sin(p)), xlabel('t'), ylabel('p'), title('t/2 sin(p)')  
view(40,60)
```

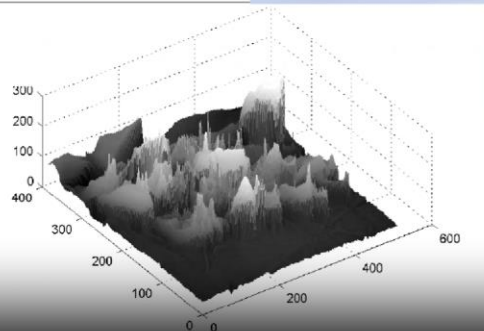
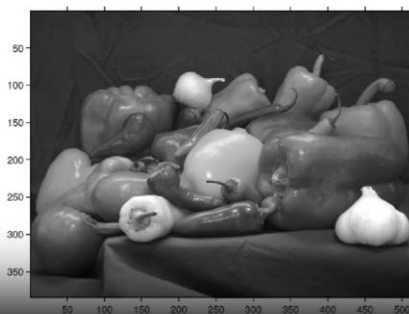


Funcția *imshow* facilitează afișarea unei imagini specificate fie prin calea spre fișierul în care e stocată imaginea, fie prin variabila în care a fost încărcată anterior imaginea.

Importarea unei imagini în Matlab este facilitată de funcția *imread*.

Un semnal bidimensional poate fi reprezentat fie tridimensional - sub forma unei suprafețe, fie în plan - prin asocierea unei nuanțe care să codifice valoarea acestuia.

```
I = imread('peppers.png'); I=rgb2gray(I);  
figure, imshow(I)  
[x,y] = meshgrid(1:size(I,2), 1:size(I,1));  
figure, surf(x,y,double(I)), shading flat, colormap gray
```



Exercitii

1. Se consideră semnalele:

$$x_1(t) = \begin{cases} -t + 2, & t \in [0, 4] \\ -t - 2, & t \in [-4, 0) \\ 0, & \text{în rest} \end{cases} \quad x_2[n] = \begin{cases} -n/2 + 1, & n \in (0, 4] \\ n/2 + 1, & n \in [-4, 0] \\ 0, & \text{în rest} \end{cases}$$

- a. Să se schițeze următoarele semnale: $x_1(t-1)$, $x_1(t+1)$, $x_1(2t)$, $x_1(t/2)$, $x_2[n-1]$, $x_2[n+1]$, $x_2[2n]$, $x_2[n/2]$.
- b. Să se reprezinte în Matlab semnalele $x_1(t)$, $x_2[n]$ precum și cele definite la subpunctul a., respectând următoarele instrucțiuni: i) se va utiliza comanda *subplot* pentru a reprezenta semnalele continue în grafice diferite pe aceeași figură. Axa X se va eticheta cu 't', iar axa Y cu denumirea semnalului (ex. $x_1(t-1)$) ii) se va utiliza comanda *subplot* pentru a reprezenta semnalele discrete în grafice diferite pe aceeași figură. Axa X se va eticheta cu 'n', iar axa Y cu denumirea semnalului (ex. $x_2[n-1]$).

2. Reprezentați în Matlab următoarele semnale:

- a. $x[n] = 4 \cos(\pi n)$
b. $x[n] = 2 \sin(3n)$
c. $x[n] = 0.9^n \cos(\frac{\pi n}{10})$
d. $x(t) = 4 \sin(5\pi t - \pi/4)$
e. $x(t) = \cos(4t) + 2 \sin(8t)$
f. $x(t) = 3 \cos(4t) + \sin(\pi t)$