

MODELARE ȘI SIMULARE

LABORATOR NR. 10 – SISTEME STOCASTICE (2), MATLAB (7), SIMULINK (9)

10.1 Rulați urmatorul live script Matlab. Observați funcțiile Matlab pentru calcul statistic și generare numere aleatoare.

```
% clear screen
clc
% sterge toate variabilele din workspace, elibereaza memorie
clear all
% sterge toate figurile
close all

% instructiunile tic si toc
tic; roots([2 6 8 9 6 1]), toc;

% instructiunea now
% Current date and time as serial date number
t=now()
% floor - Round toward negative infinity
% perechea lui floor este ceil - Round toward positive infinity
t2=floor(t)
% data si ora curenta
d = datetime(t,'ConvertFrom','datenum')
% doar data, nu si ora
d2 = datetime(t2,'ConvertFrom','datenum')
% atentie
% MATLAB® Online™ returns the current date and time in Coordinated Universal
Time (UTC) rather than local time.

% elemente de statistica in Matlab
x=[2 6 8 9 6 1];
sum(x)
mean(x)
% varianta
var(x)
% deviatia standard
sqrt(var(x))
% sau cu functia std
std(x)
```

```

% combinari de n luate cate k, 6 luate cate 3 in exemplul de mai jos
nchoosek(6,3)

% initializam generatorul de numere aleatoare, generatorul folosit este cel
implicit
% The factory default is the Mersenne Twister generator with seed 0.
% a Mersenne prime is a prime number that is one less than a power of two
% The exponents n which give Mersenne primes are 2, 3, 5, 7, 13, 17, 19,
31, ...
% and the resulting Mersenne primes are 3, 7, 31, 127, 8191, 131071, 524287,
2147483647, ..
% if a pseudorandom number generator is later reinitialized with the same seed
% it will produce the same sequence of numbers.
% Random seeds are often generated from the state of the computer system (such
as the time)
% vezi functia now de mai sus
% Starting in R2023b, you can set the default algorithm and seed from the
MATLAB Preferences window.
% afisam parametrii impliciti ai generatorului de numere aleatoare
S=rng
% modificam seed
Seed=1; rng(Seed);
% afisam parametrii (modificati) ai generatorului de numere aleatoare
S=rng

% generam matricea A_1 de numere aleatoare uniform distribuite
A_1=rand(5,3)
% suma elementelor din A_1 (pe coloane)
sum(A_1)
% media pe fiecare coloana din cele 3
mean(A_1)
% media pe fiecare coloana din cele 3
mean(A_1,1)
% media pe fiecare linie din cele 5
mean(A_1,2)
% media tuturor elementelor lui A_1
mean(A_1,'all')

```

% generam 20000 de numere aleatoare uniform distribuite

The uniform density:

$$f_X(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{elsewhere} \end{cases},$$

where a and b are constants with $b > a$.

```
x=rand(20000,1);
```

```
% generam o histograma
```

```
% Histograms are a type of bar plot that group data into bins.
```

```
hist(x)
```

```
% cate sunt mai mari sau egale cu 0.5?
```

```
y=x(find(x>=0.5))
```

```
% Set output display format
```

```
format long
```

```
% afisam media valorilor x, cel mai mic x, cel mai mare x si procentul de
```

```
% valori mai mari sau egale cu 0.5
```

```
[mean(x), min(x), max(x), length(y)/length(x)*100]
```

% functia teoretica de densitate de probabilitate pentru distributia normala este

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

% in cazul cu media 0 varianta 1, ea devine

$$p(x) = e^{-x^2/2} / \sqrt{2\pi}$$

```
% generam 10000 de numere aleatoare din distributia normala
```

```
x = randn(10000,1);
```

```
hist(x)
```

```
% generam matricea A_2 de numere aleatoare normal distribuite
```

```
% cu media 0 varianta 1 si deviatia standard 1
```

```
A_2=randn(5,3)
```

```
% media pe fiecare coloana
```

```
mean(A_2)
```

```
% generam matricea A_3 de numere intregi aleatoare intre -3 si 3
A_3=randi([-3 3],5,3)
```

```
% generam N numere aleatoare in intervalul [a,b]
% cu formula r=a+(b-a).*rand(N,1) - uniform distribuite U(0,1)
% sau r=a+(b-a).*randn(N,1) - normal distribuite
a=2; b=5; N=7;
r=a+(b-a).*rand(N,1)
% daca vrem numere intregi in acelasi interval
r=floor(a+(b-a).*rand(N,1))
% sau asa
r=randi([2 7],N,1)
% randperm genereaza valori unice
randperm(5)
% randperm(n,k) genereaza k valori unice intre 1 si n
randperm(10, 4)
```

```
% generam o variabila Gaussiană cu media  $\mu$  si deviatia standard  $\sigma$ 
sigma=0.18178; mu=0.3265;
X= sigma*randn(10,1)+mu
% functia de densitate de probabilitate normala
```

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

```
x=[-5:0.1:5];
% miu = 0 si sigma = 1
y=normpdf(x,0,1);
plot(x,y)
grid on
```

```
% cdf, cummulative distribution function
%  $\mu = 0.7$  si  $\sigma = 0.5$ 
pdf=makedist('Normal',0.7,0.5)
```

```
z=cdf(pdf,x)
```

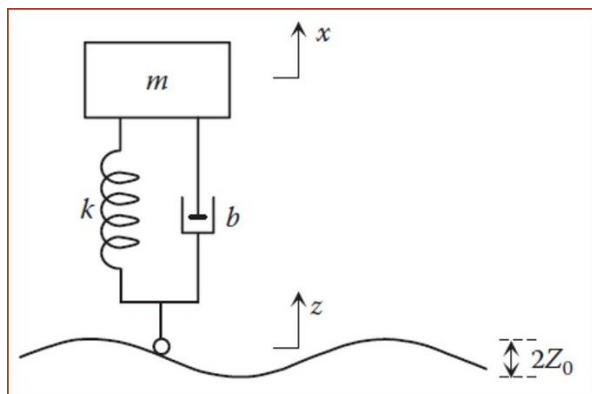
```
plot(x,z)
grid on
```

```
% click pe grafic la x=0.5 si observam ca z=0.344578
% adica
```

$$\int_{-\infty}^{0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx = 0.344578$$

10.2 Pentru suspensia de automobil de mai jos, construiți un model Simulink pentru a simula dinamica sistemului la o intrare z aleatoare.

$$m = 3000 \text{ kg}, b = 2000 \text{ N} \frac{\text{s}}{\text{m}}, k = 50 \frac{\text{kN}}{\text{m}}.$$



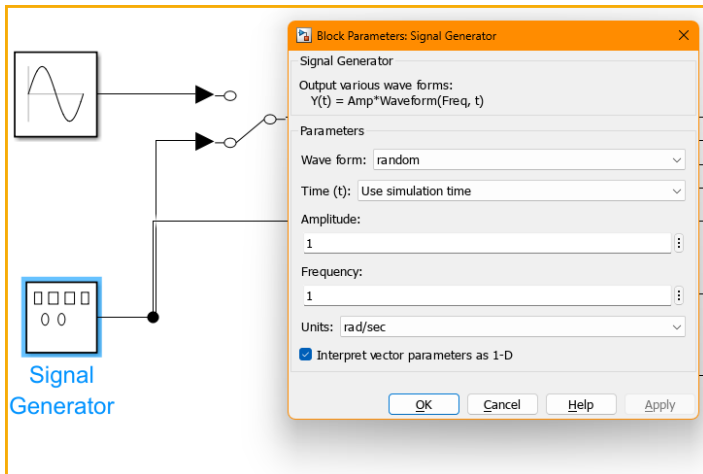
1. Determinăm funcția de transfer (în condiții nule)

$$\ddot{x} = \frac{1}{m}(kz + b\dot{z} - kx - b\dot{x})$$

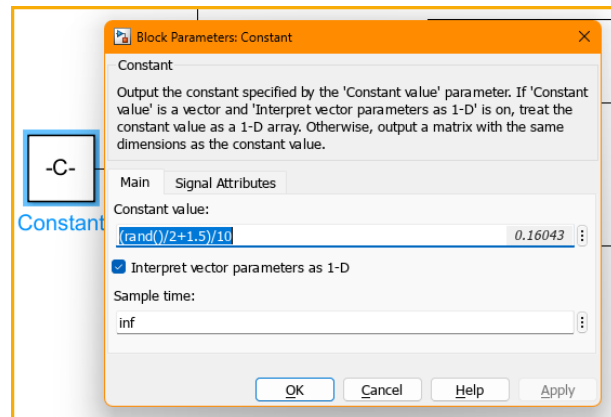
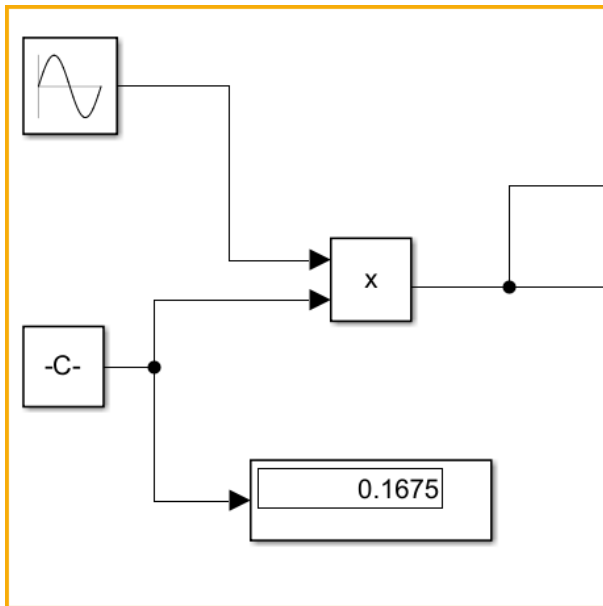
$$\frac{X(s)}{Z(s)} = \frac{bs + k}{ms^2 + bs + k}$$

2. Simulam dinamica sistemului pentru o intrare sinusoidala si o intrare aleatoare generata cu blocul Signal generator. Comutarea intre cele doua intrari se face cu un Manual switch.

$$z(t) = Z_0 \sin(\omega t), Z_0 = 0.01 \text{ m}, \omega = \frac{3.5 \text{ rad}}{\text{s}}$$



3. Simulam dinamica sistemului pentru o intrare sinusoidala cu amplitudine aleatoare intre 0.15 si 0.2, generata dintr-o distributie uniforma. Restul parametrilor sinusoidei raman neschimbati.



4. Intrarea va fi generata de un bloc 'Uniform random number' cu seed=0. Apoi vom face Seed=now().

5. Intrarea va fi generata de un bloc 'Random number' cu seed=0. Apoi vom face Seed=now().

6. Intrarea va fi aleatoare dintr-o distributie uniforma, dar incarcata dintr-un fisier .mat generat in prealabil.

```
function inputData = random_input(filename, low_limit, up_limit, sampleTime,
numSteps)
time = sampleTime*[0:(numSteps-1)];
size(time)
len = numSteps;
data = low_limit+(up_limit-low_limit).*rand(len,1);
size(data)
time=time';
inputData = timeseries(data,time);
save(filename,"inputData","-v7.3");
end
```

```
inputData = random_input('Lab_101_random.mat', 1, 2, 0.01, 1001)
```

7. Intrarea va fi aleatoare dintr-o distributie normala, dar generata de un script Matlab inainte de rularea diagramei Simulink.

```
inputDataStructure.time=[0:0.01:10]';  
inputDataStructure.signals(1).values=sin(inputDataStructure.time)+0.2*randn(size(inputDataStructure.time));  
inputDataStructure.signals(1).dimensions=1;
```