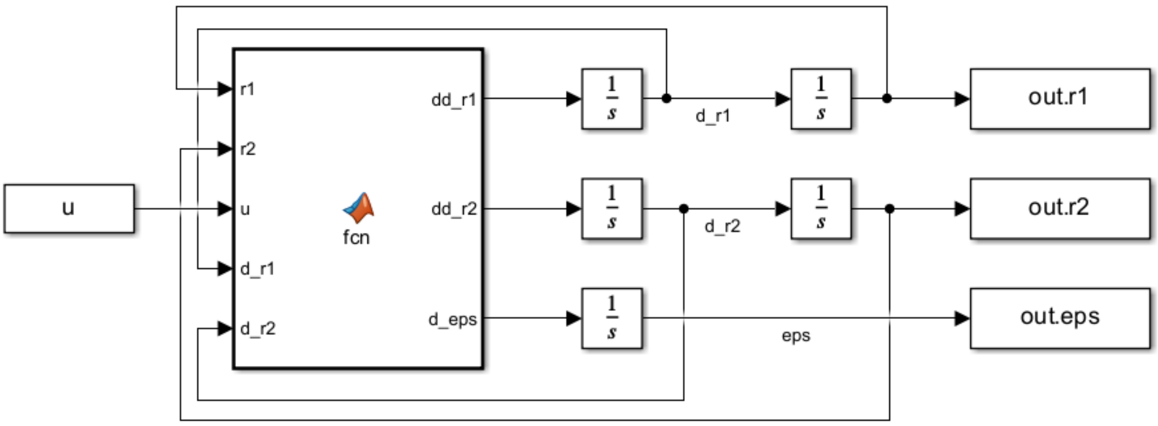
Model Orbita Sateliti

Cerinta 1

```matlab
%% Cerinta 1
r10=1e6*[-3.111566746661099;2.420733547442338;-5.626803092559423];
r20=1e6*[-3.422723421327209;2.662806902186572;-6.189483401815366];
d_r10=1e3*[4.953572247000772;-3.787243278806948;-4.362500902062312];
d_r20=1e3*[5.448929471700850;-4.165967606687643;-4.798750992268544];
eps0=1;
x0=[r10;r20;d_r10;d_r20;eps0];
```

Cerinta 2




```matlab
function [dd_r1, dd_r2, d_eps] = fcn(r1, r2, u, d_r1, d_r2)
dd_r1=g(r1,d_r1)+u;
dd_r2=g(r2,d_r2);
d_eps=norm(r1-r2)/norm(r2);
```

```matlab
function dd_r = g(r, d_r)
G=6.674e-11;
M=5.972e24;
R=6371000;
J2=1.08262668e-3;
w=7.2921e-5;


x=r(1); y=r(2); z=r(3);
norm_r=norm(r);
dd_r=-G*M*r/norm_r^3 ...
    -3/2*J2*G*M*R^2*[x-5*x*z^2; y-5*y*z^2; 3*z-5*z^3]/norm_r^7 ...
    +w^2*[x;y;0] ...
    +2*w*[d_r(2);-d_r(1);0];
end
```

Cerinta 3

$$x(t) = \begin{bmatrix} r_1 \\ r_2 \\ \dot{r}_1 \\ \dot{r}_2 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \rightarrow \dot{x}(t) = \begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \ddot{r}_1 \\ \ddot{r}_2 \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ g(t, x_1, x_3) + u(t) \\ g(t, x_2, x_4) \\ \dfrac{\|x_1 - x_2\|}{\|x_2\|} \end{bmatrix} = f(t, x)$$

Cerinta 4

```
%% Cerinta 4
h=1;
tmax=10000;
t=0:h:tmax;
u=0;
x=[x0 zeros(13,length(t)-1)];

for i=1:length(t)-1
    k1=f(u,x(:,i));
    k2=f(u,x(:,i)+h*k1/2);
    k3=f(u,x(:,i)+h*k2/2);
    k4=f(u,x(:,i)+h*k3);
    x(:,i+1)=x(:,i)+h/6*(k1+2*k2+2*k3+k4);
end
[r1_rk,r2_rk,eps_rk]=deal(x(1:3,:),x(4:6,:),x(13,:));
```

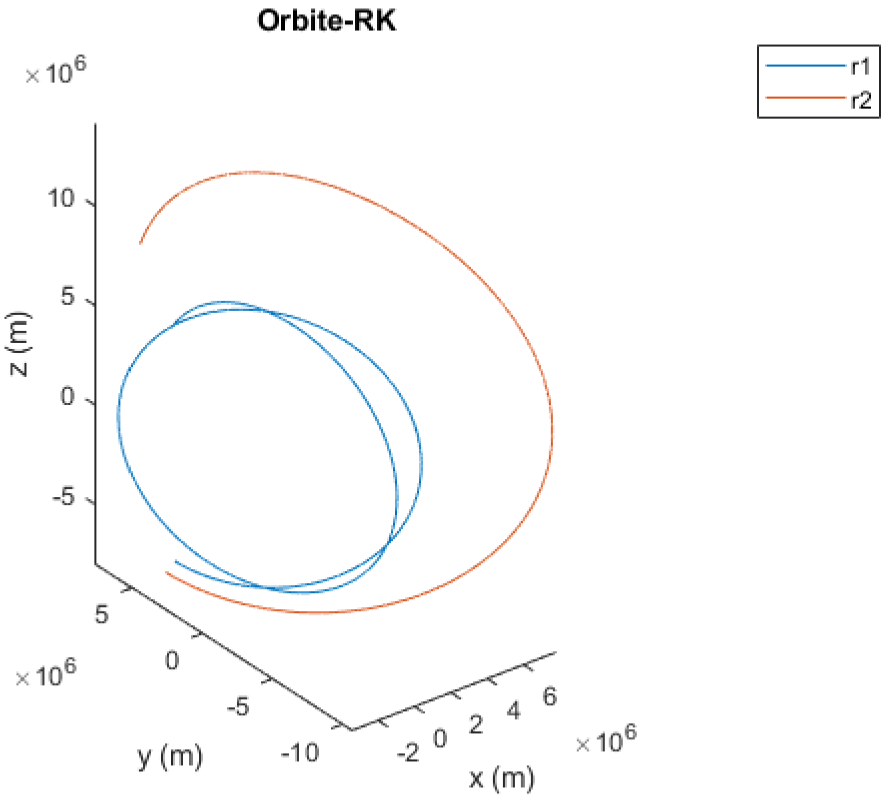
```
function d_x = f(u, x)
x1=x(1:3); x2=x(4:6); x3=x(7:9); x4=x(10:12);
d_x=[x3; x4; g(x1,x3)+u; g(x2,x4); norm(x1-x2)/norm(x2)];
end
```

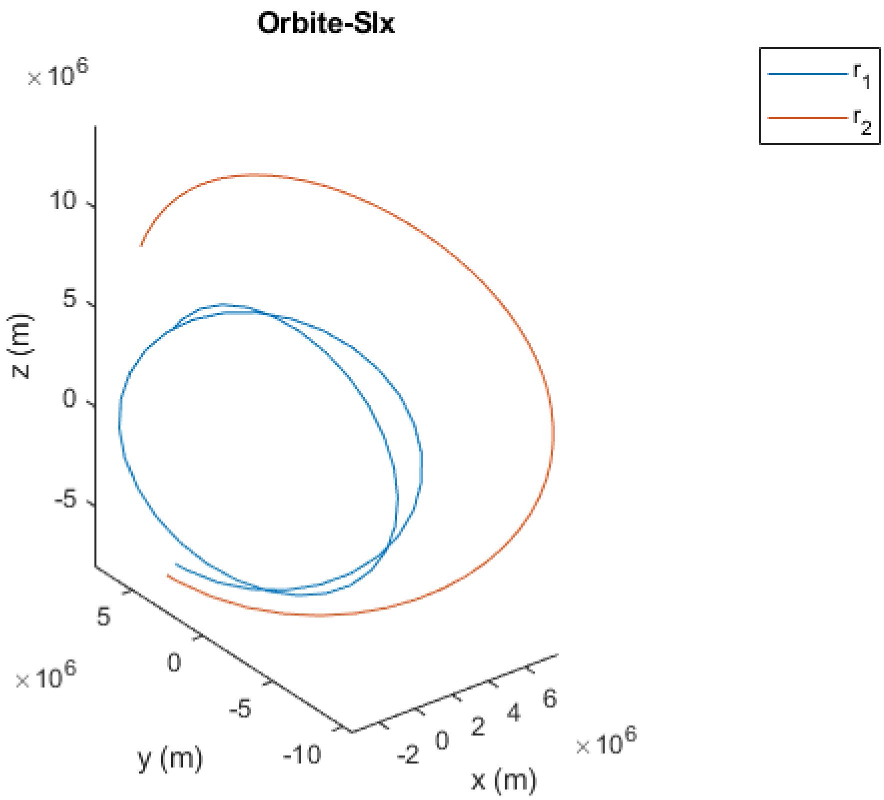
Cerinta 5

```
%% Cerinta 5
u=timeseries(zeros(size(t)),t);
load_system('Model_sim')
set_param('Model_sim','StopTime',num2str(tmax))
out=sim('Model_sim');
[r1_slx,r2_slx,eps_slx]=deal(squeeze(out.r1.Data),squeeze(out.r2.Data),squeeze(out.eps.Data));

figure
plot3(r1_rk(1,:),r1_rk(2,:),r1_rk(3,:))
hold on
plot3(r2_rk(1,:),r2_rk(2,:),r2_rk(3,:))
axis equal
xlabel("x (m)")
ylabel("y (m)")
zlabel("z (m)")
title("Orbite-RK")
legend("r1","r2")

figure
plot3(r1_slx(1,:),r1_slx(2,:),r1_slx(3,:))
hold on
plot3(r2_slx(1,:),r2_slx(2,:),r2_slx(3,:))
axis equal
xlabel("x (m)")
ylabel("y (m)")
zlabel("z (m)")
title("Orbite-Slx")
legend("r_1","r_2")
```





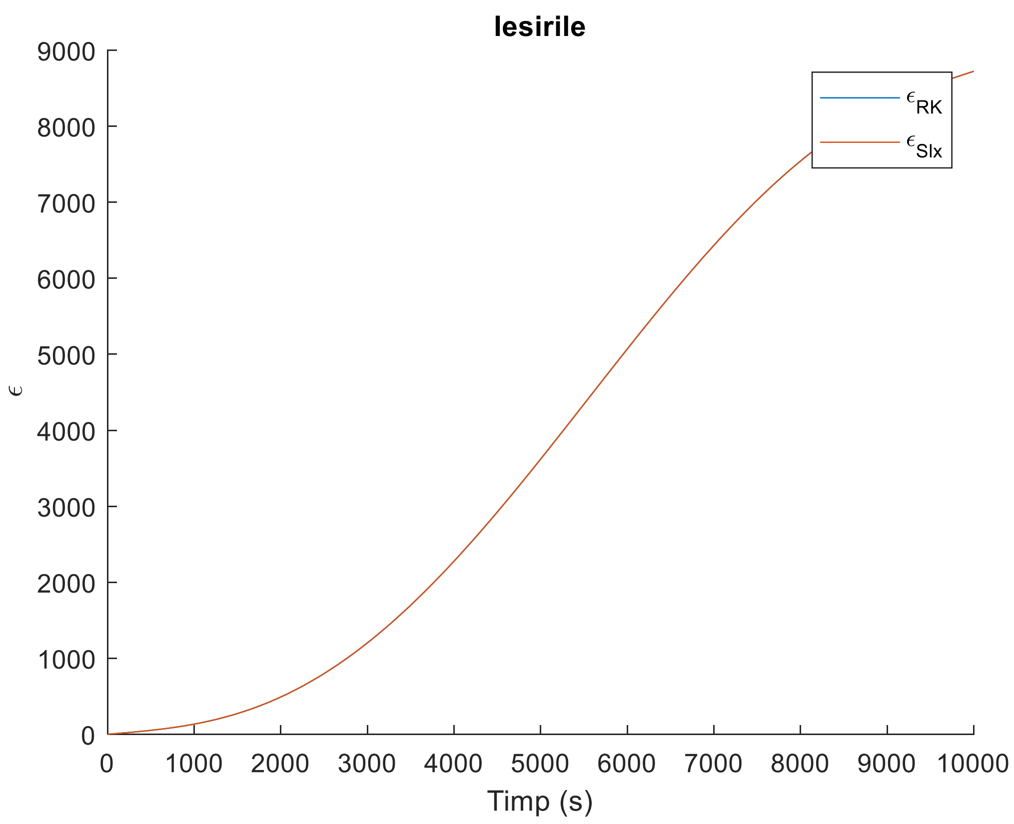
Comparand grafic rezultatele metodei Runge-Kutta cu cele din Simulink, observam ca sunt asemanatoare, insa orbita rezultata din Simulink nu este la fel de smooth deoarece foloseste un pas adaptiv, fata de RK care foloseste un pas constant destul de mic.

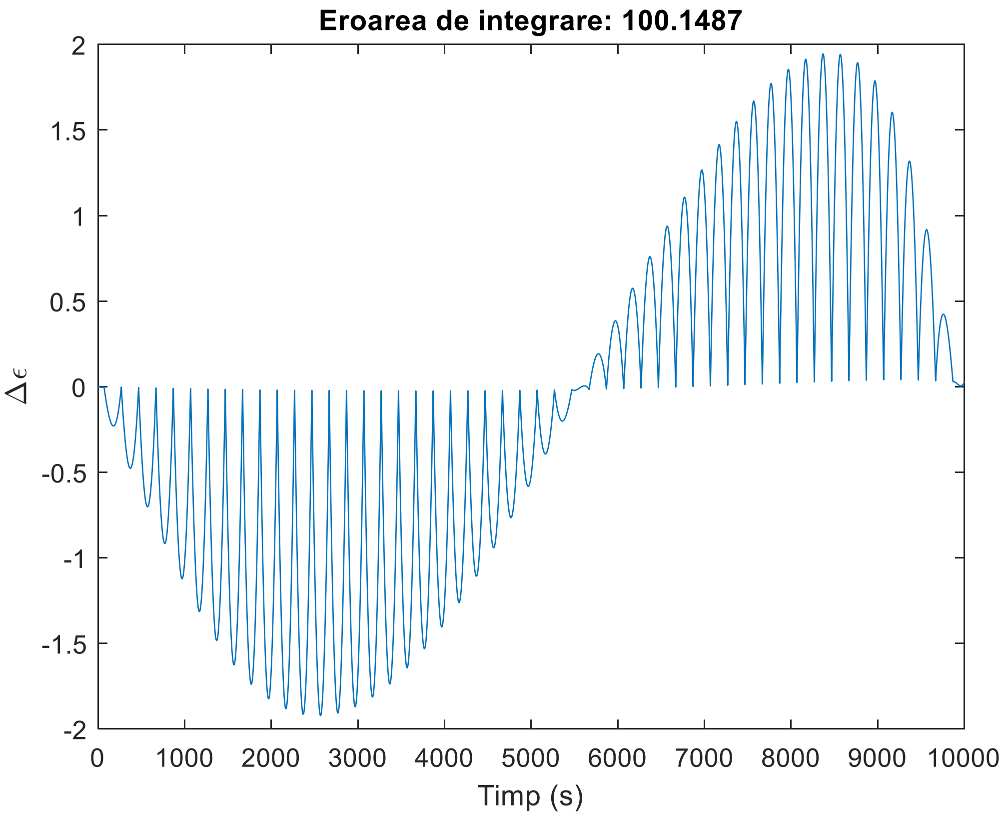
Cerinta 6

```
%% Cerinta 6
eps_slx_int=interp1(out.tout,eps_slx,t);

figure
hold on
plot(t,eps_rk)
plot(t,eps_slx_int)
xlabel("Timp (s)")
ylabel("\epsilon")
title("Iesirile")
legend("\epsilon_{RK}","\epsilon_{Slx}")

figure
plot(t,eps_rk-eps_slx_int)
xlabel("Timp (s)")
ylabel("\Delta\epsilon")
title("Eroarea de integrare: "+string(norm(eps_rk-eps_slx_int)))
```





Observam din nou ca cele doua metode dau rezultate asemanatoare, eroarea de integrare la fiecare moment de timp (si norma acesteia) fiind destul de mici, relativ la valorile iesirilor. In plus, eroarea oscileaza periodic cu amplitudine mica, indicand o concordanta buna intre ele.

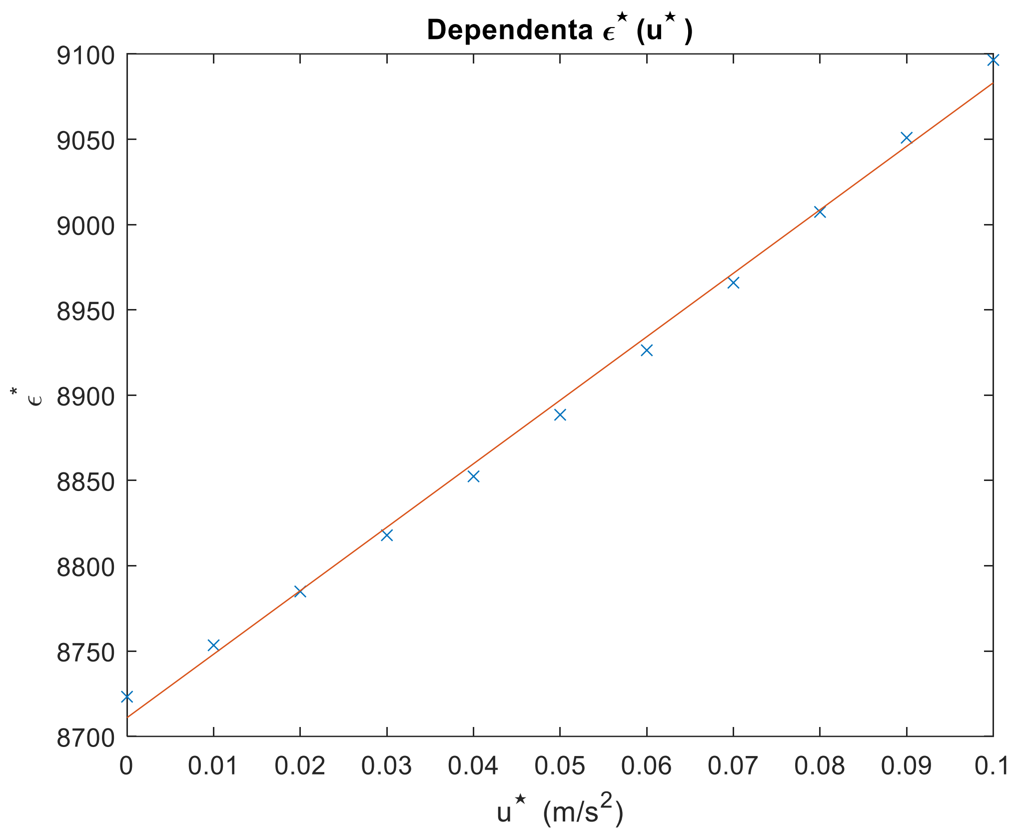
Cerinta 7/8

```
%% Cerinta 7
i=1;
for k=0:10:100
    u=timeseries(k*1e-3.*double(t>=0),t);
    out=sim('Model_sim');
    eps_star(i)=out.eps.Data(end);
    ustar(i)=k*1e-3;
    i=i+1;
end

figure
plot(ustar,eps_star,'x');
xlabel("u^* (m/s^2)");
ylabel("\epsilon^*")
title("Dependenta \epsilon^*(u^*)")
```

```
%% Cerinta 8
p=polyfit(ustar,eps_star,1);
ustar_grid=ustar(1):0.01:ustar(end);
eps_int=polyval(p,ustar_grid);

hold on;
plot(ustar_grid,eps_int);
```



Dependenta intre iesirea stationara (eroarea relativa totala intre cei doi sateliti) si semnalul exogen constant este aproximativ liniara, indicand o crestere proportionala cu semnalul.

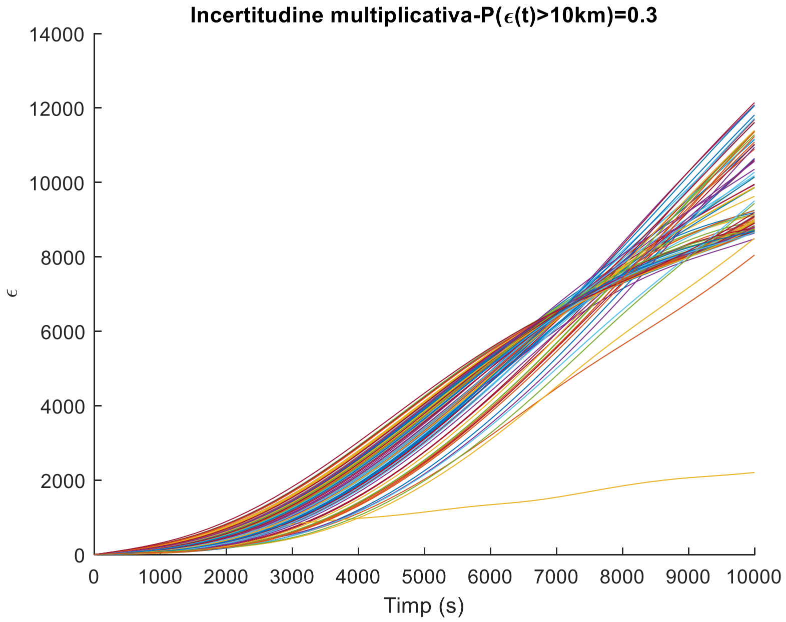
Cerinta 9/10

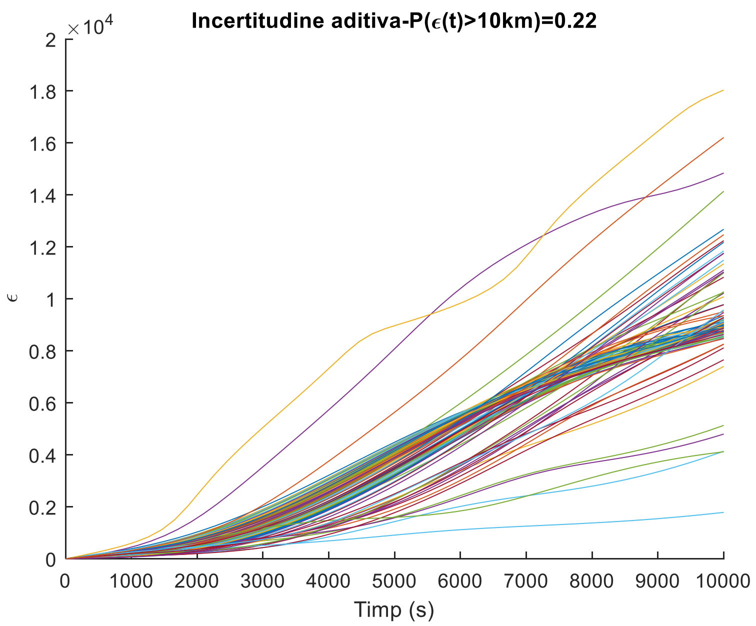
```
%% Cerinta 9
alfa=randn(3,100)/10;
n=0;

figure
hold on;
for i=1:size(alfa,2)
    x0=[r10;(1+alfa(:,i)).*r20;d_r10;d_r20;eps0];
    out=sim("Model_sim");
    plot(out.tout,out.eps.Data)
    n=n+any(out.eps.Data>1e4);
end
xlabel("Timp (s)")
ylabel("\epsilon")
title("Incertitudine multiplicativa-P(\epsilon(t)>10km)="+string(n/size(alfa,2)))
```

```
%% Cerinta 10
alfa=1e6*randn(3,100);
n=0;

figure
hold on;
for i=1:size(alfa,2)
    x0=[r10;alfa(:,i)+r20;d_r10;d_r20;eps0];
    out=sim("Model_sim");
    plot(out.tout,out.eps.Data)
    n=n+any(out.eps.Data>1e4);
end
xlabel("Timp (s)")
ylabel("\epsilon")
title("Incertitudine aditiva-P(\epsilon(t)>10km)="+string(n/size(alfa,2)))
```





Martin Mihnea Matei-334AB

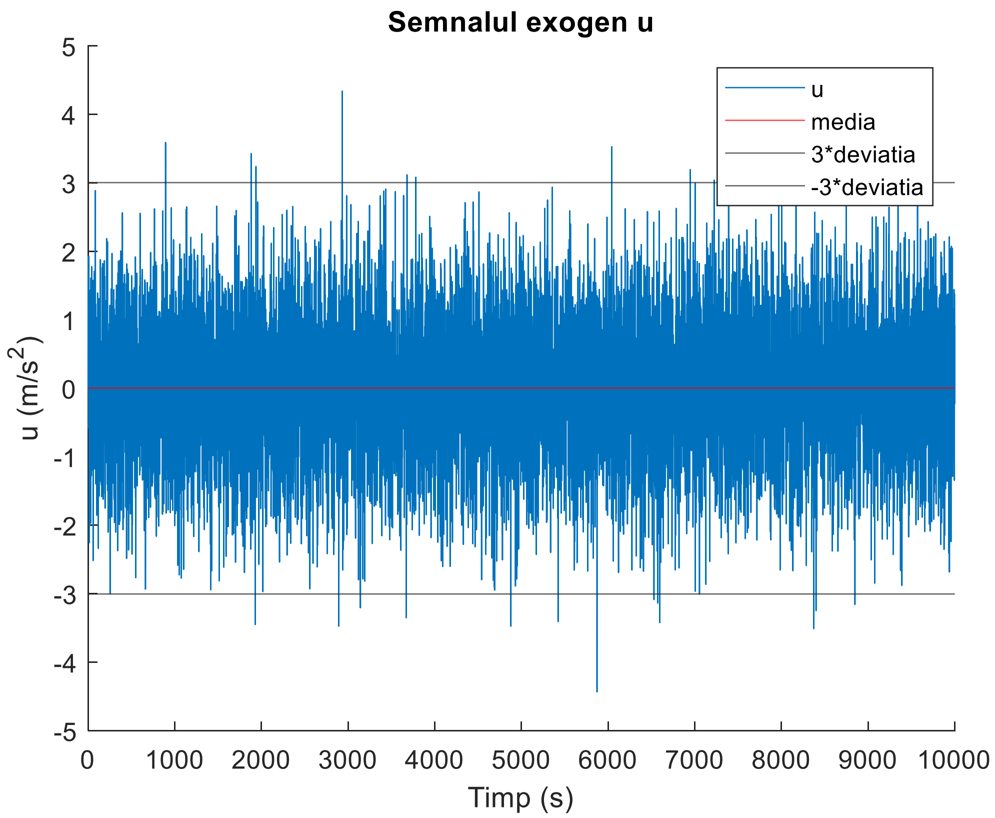
O incertitudine multiplicativa (proportionala) mentine iesirea intr-o anumita zona, pe cand o incertitudine aditiva de aceeasi scala imprastie mai mult iesirea. Acest lucru se observa si prin calculul probabilitatii ca distanta relativa totala intre cei doi sateliti (iesirea) sa fie mai mare de 10km: in cazul celei multiplicative avem o certitudine mai mare decat in cazul celei aditive.

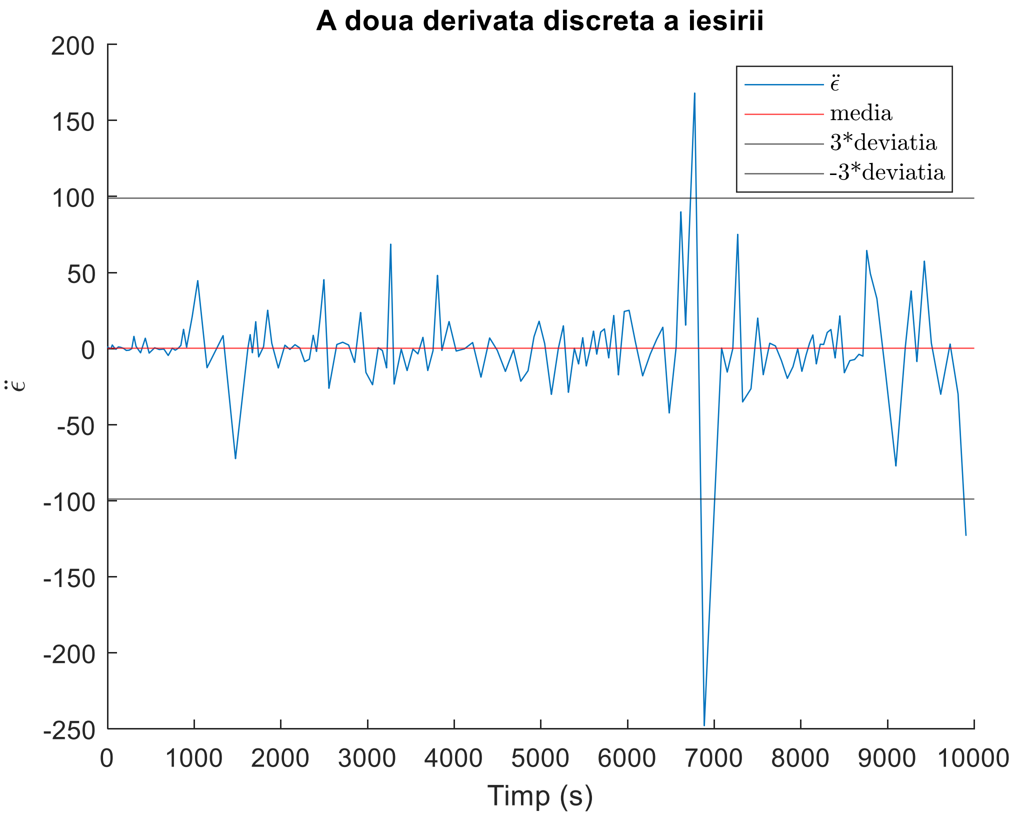
Cerinta 11

```
%% Cerinta 11
u=timeseries(randn(1,length(t)),t);
out=sim("Model_sim");

figure
hold on
plot(t,squeeze(u.Data))
xlabel("Timp (s)")
ylabel("u (m/s^2)")
yline(mean(u),'r')
yline(3*std(u))
yline(-3*std(u))
title("Semnalul exogen u")
legend("u","media","3*deviatia","-3*deviatia")

figure
hold on
dd_eps=diff(diff(out.eps.Data));
plot(out.tout(1:end-2),dd_eps)
xlabel("Timp (s)")
ylabel("$\ddot{\epsilon}$",'Interpreter','latex')
yline(mean(dd_eps),'r')
yline(3*std(dd_eps))
yline(-3*std(dd_eps))
title("A doua derivata discreta a iesirii")
legend("$\ddot{\epsilon}$","media","3*deviatia","-3*deviatia",'Interpreter','latex')
```





Observam ca a doua derivata discreta a iesirii are tot un comportament aleator de distributie normala ca semnalul exogen, avand tot media 0, dar deviatia mai mare.