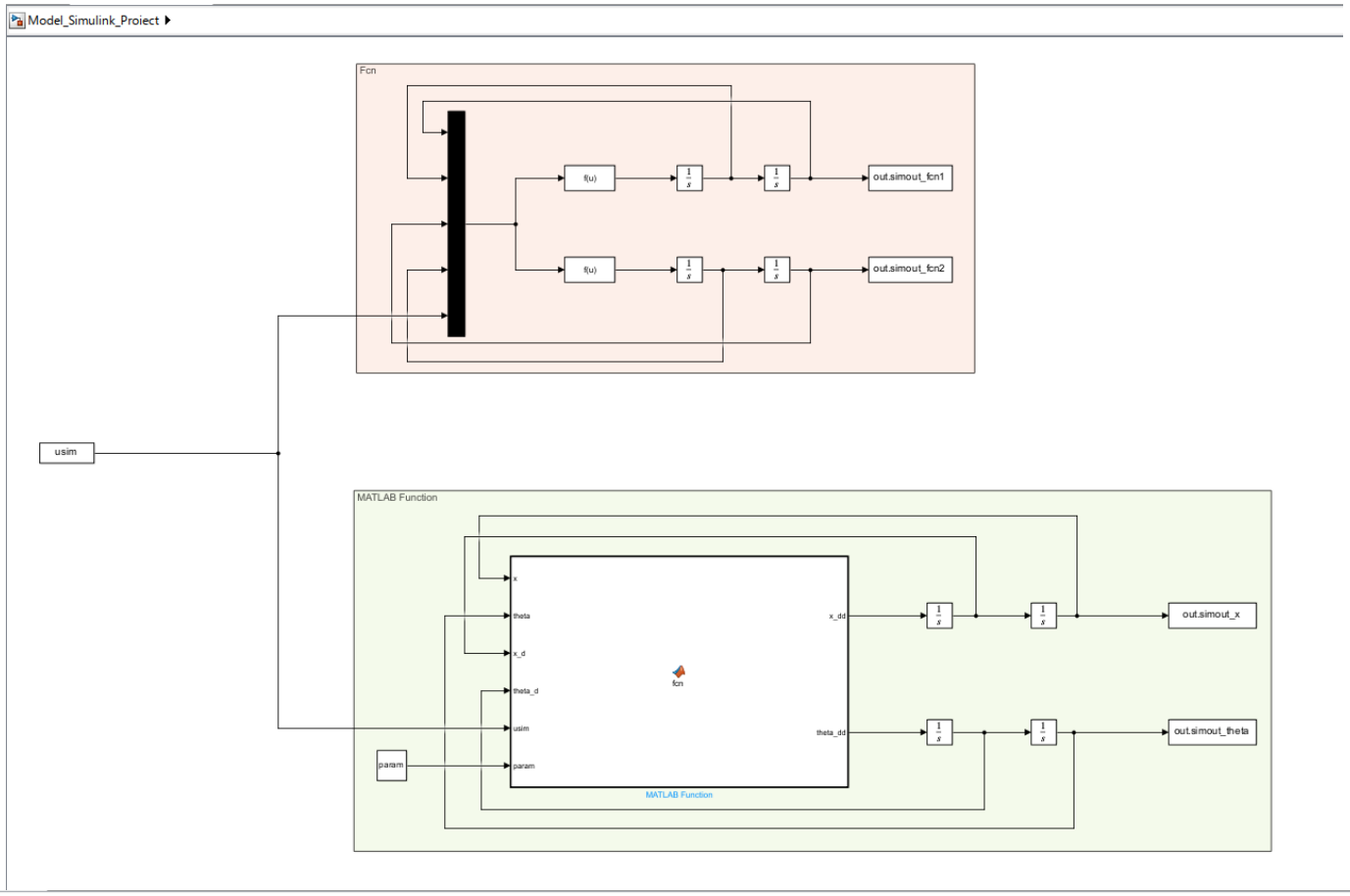
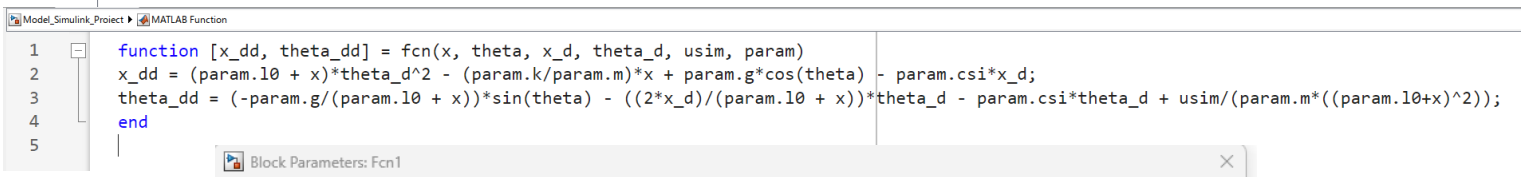
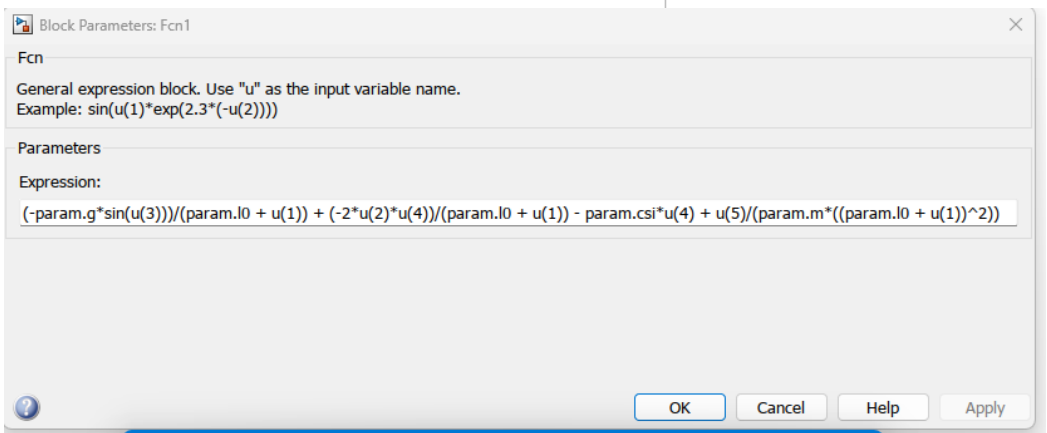
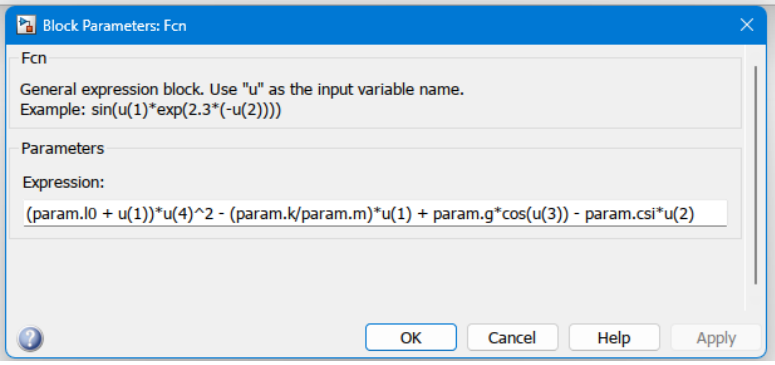
Raport Proiect MS

Cerința 1: Modelul si expresiile









Am ales valori pentru paramatrii necesari si am creat un obiect de tip bus, din parametrul param, cu scopul de a-l accesa din Workspace si a fi folosit in transferul de date din Simulink. Am incarcat modelul creat.

```
clc, clear, close all

% Definirea Parametrilor
param.m = 5;     % masa obiectului
param.l0 = 1.4;  % lungimea initiala
param.csi = 0.9; % factor amortizare
param.k = 0.75;  % constanta de elasticitate
param.g = 9.81;  % acceleratia gravitationala

% Definire bus
par_bus_info = Simulink.Bus.createObject(param);
par_bus = evalin('base', par_bus_info.busName);

%% Cerinta 1
% Incarcare model, cerintele a) si b)
load_system('Model_Simulink_Proiect.slx')

```

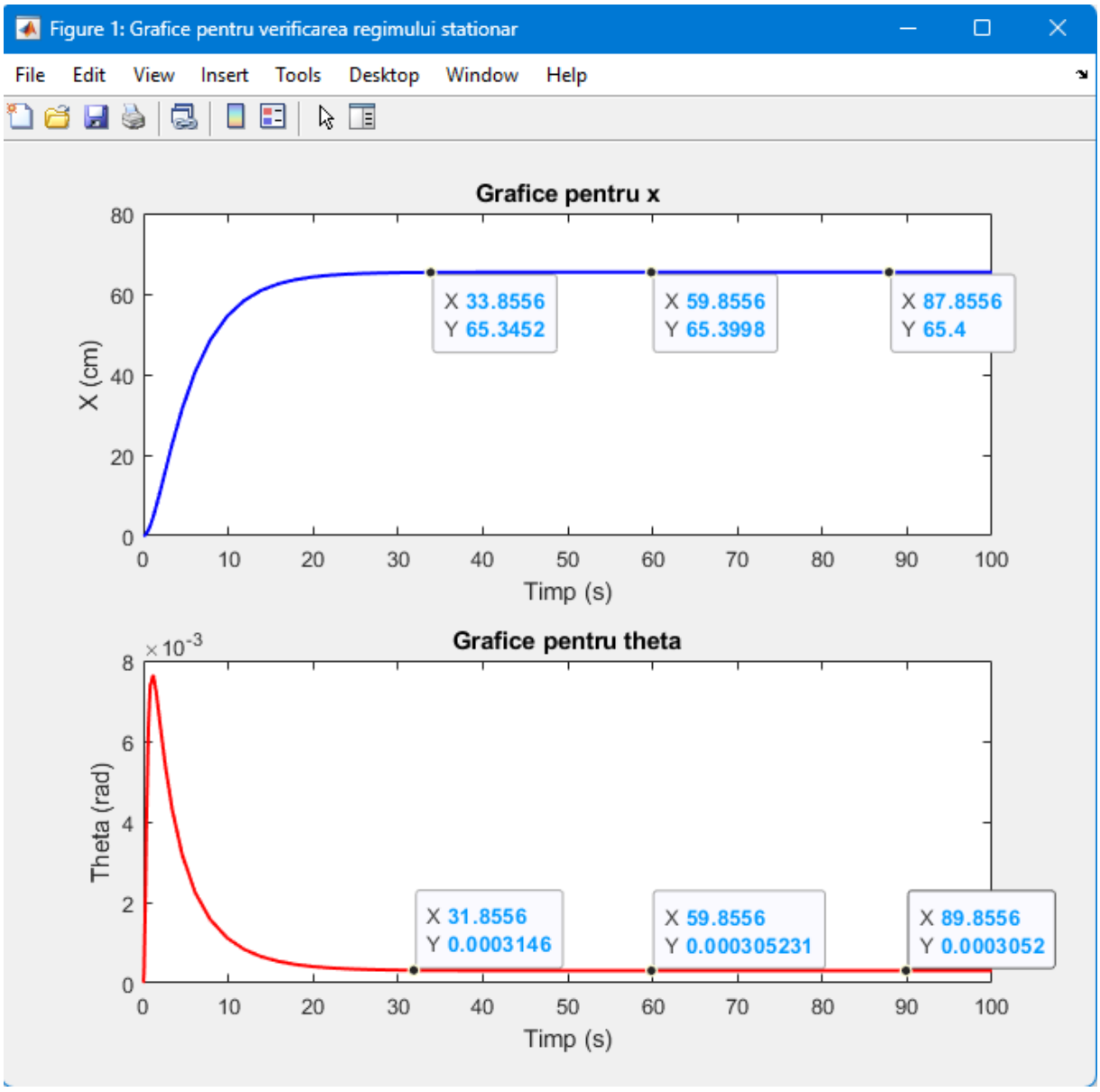
Cerința 2:

```
%% Cerinta 2
% Definirea timpului pentru simulare
T = 100;
set_param('Model_Simulink_Proiect', 'StopTime', num2str(T));

% Generam intrarea
t = linspace(0,T,1000);
st = double(t>=0);
usim = timeseries(st,t);

% Simulam modelul
out = sim('Model_Simulink_Proiect');

% Afisez graficele pentru a verifca daca ajung in regim stationar
figure('Name', 'Grafice pentru verificarea regimului stationar');
subplot(2,1,1);
plot(out.simout_x.time, out.simout_x.data, 'b', 'LineWidth', 1.5);
title('Grafice pentru x')
xlabel('Timp (s)');
ylabel('X (cm)');
hold on
subplot(2,1,2);
plot(out.simout_theta.time, out.simout_theta.data, 'r', 'LineWidth', 1.5);
title('Grafice pentru theta')
xlabel('Timp (s)');
ylabel('Theta (rad)');

```

Am definit orizontul de timp ales: [0,100] secunde.

Am generat semnalul de intrare, asigurandu-ma ca sistemul ajunge la regim stationar.

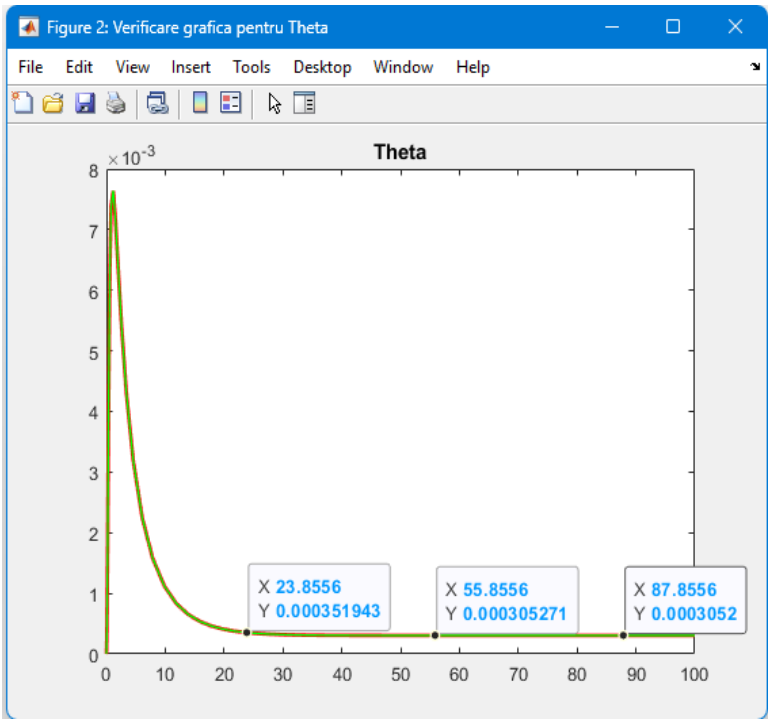
Am afisat 2 grafice, pentru x si theta, unde se poate vedea ca acestea ajung cu bine la un regim stationar. Pentru x, valoarea este 65.4, iar pentru theta ~ 0.000305.

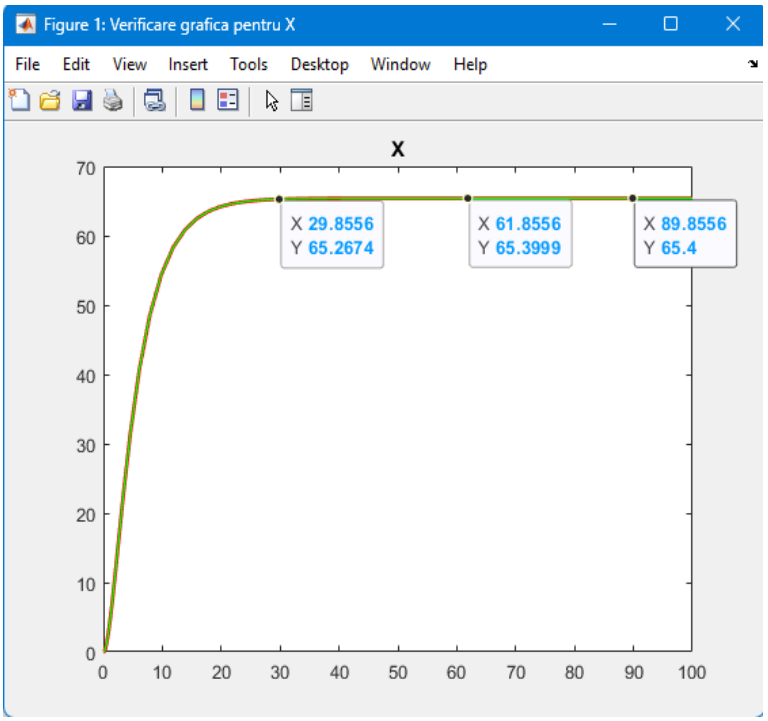


Cerința 3:

Am selectat datele importante, in variabilele simout_mat si simout_fcn si am simulat, mai apoi afisat pe grafice modelul implementat cu Matlab Fucntion cat si cel cu blocul fcn, acestea fiind identice.

```
%% Cerinta 3
% Simulam modelul
out = sim('Model_Simulink_Proiect');

simout_mat1 = out.simout_x.data;
simout_fcn1 = out.simout_fcn1.data;

simout_mat2 = out.simout_theta.data;
simout_fcn2 = out.simout_fcn2.data;

% Verificare grafic pentru X
figure('Name', 'Verificare grafica pentru X');
plot(out.simout_x.time, simout_mat1, 'r', 'LineWidth', 2);
title('X')
hold on;
plot(out.simout_fcn1.time, simout_fcn1, 'g', 'LineWidth', 1);

% Verificare grafic pentru Theta
figure('Name', 'Verificare grafica pentru Theta');
plot(out.simout_theta.time, simout_mat2, 'r', 'LineWidth', 2);
title('Theta')
hold on;
plot(out.simout_fcn2.time, simout_fcn2, 'g', 'LineWidth', 1);

% Se poate observa ca in ambele cazuri, cele 2 grafice sunt identice,
% au aceleasi valori

```





Cerința 4:

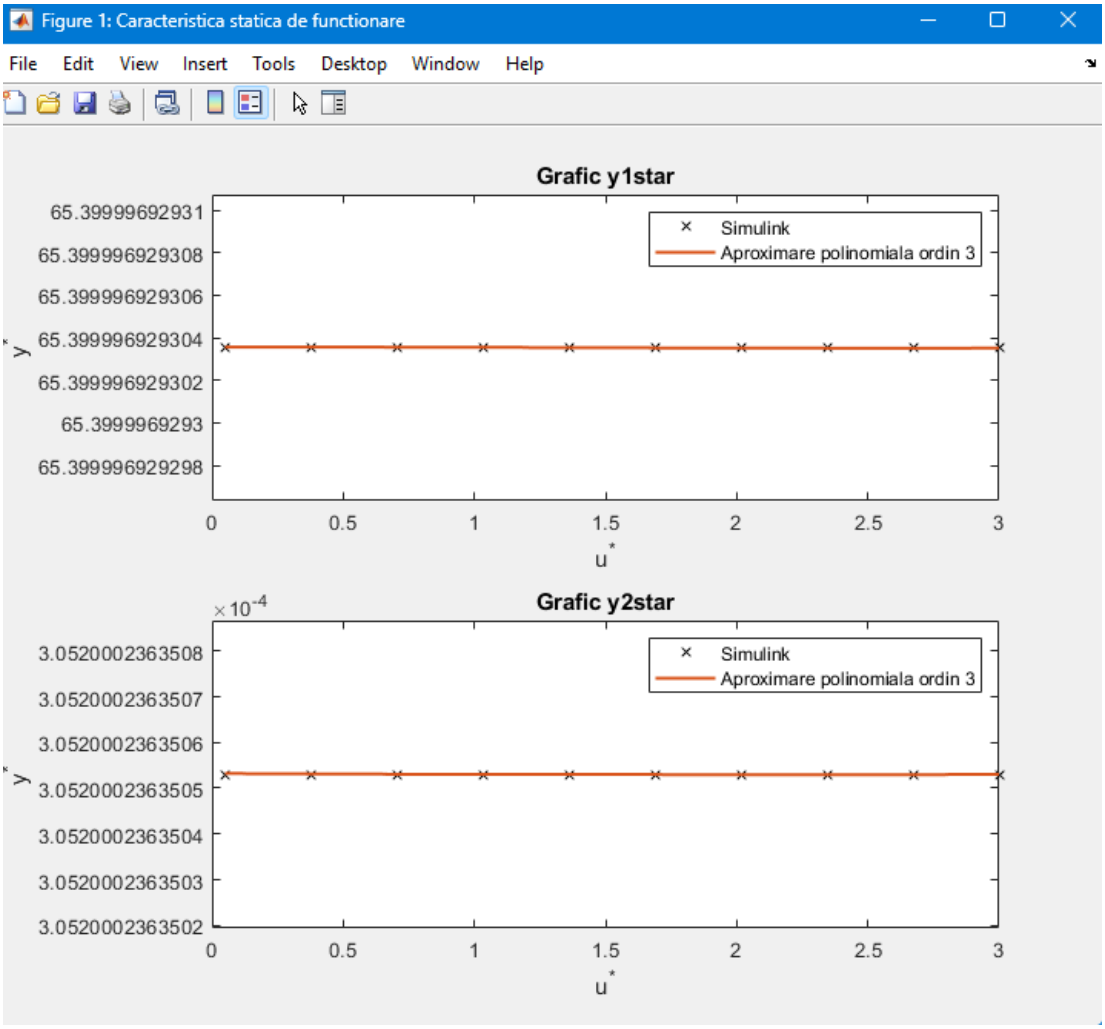
Eroarea rezultata este nula, ceea ce inseamna ca raspunsurile obtinute de ambele blocuri este identic.

```
72  %% Cerinta 4
73  % Calcularea norma 2
74  err1 = norm(simout_mat1 - simout_fcn1);
75  err2 = norm(simout_mat2 - simout_fcn2);
76  disp('Eroarea dintre simout_mat1 si simout_fcn1');
77  disp(err1);
78  disp('Eroarea dintre simout_mat2 si simout_fcn2');
79  disp(err2);
80
81  % Asteptam sa obtin valori mici pentru err1 si err2, ceea ce s-a intamplat
82  % Aceste valori le-am obtinut deoarece graficele celor 2 aproape coincid
83
```

Command Window

```
Eroarea dintre simout_mat1 si simout_fcn1
     0

Eroarea dintre simout_mat2 si simout_fcn2
   2.1833e-18
```

Cerința 5:



Am generat cele 10 intrari, si am simulat modelul pentru fiecare dintre acestea, retinand valorile de regim stationar. Am calculat aproximarea polinomiala de ordin 3 si am afisat pe grafic. Se poate observa ca aproximarea este buna, toate valorile reale se afla pe dreapta calculata.

```
%% Cerinta 5
% Cerinta a)
k = linspace(0.05, 3, 10);
ustar = zeros(10, 1);
y1star = zeros(10, 1);
y2star = zeros(10, 1);

% Cerinta b)
for i = 1:10
    in = k(i) .* double(t>=0);
    usim = timeseries(in,t);

    out = sim('Model_Simulink_Proiect');
    ustar(i) = k(i);
    y1star(i) = simout_mat1(end);
    y2star(i) = simout_mat2(end);
end
disp('Ustar');
disp(ustar);
disp('Y1star');
disp(y1star);
disp('Y2star');
disp(y2star);

% Cerinta c)
% Aproximare polinomiala ordin 3
p1_ord3 = polyfit(ustar, y1star, 3);
p2_ord3 = polyfit(ustar, y2star, 3);

ustar1 = linspace(ustar(1), ustar(end), 100);
y1starr_ord3 = polyval(p1_ord3, ustar1);
y2starr_ord3 = polyval(p2_ord3, ustar1);

% Cerinta d)
figure('Name', 'Caracteristica statica de functionare');
subplot(2,1,1);
plot(ustar, y1star, 'xk');
hold on
plot(ustar1, y1starr_ord3, 'LineWidth', 1.5);
xlabel('u^*'), ylabel('y^*');
legend('Simulink', 'Aproximare polinomiala ordin 3');
title('Grafic y1star');

subplot(2,1,2);
plot(ustar, y2star, 'xk');
hold on
plot(ustar1, y2starr_ord3, 'LineWidth', 1.5);
xlabel('u^*'), ylabel('y^*');
legend('Simulink', 'Aproximare polinomiala ordin 3');
title('Grafic y2star');
```

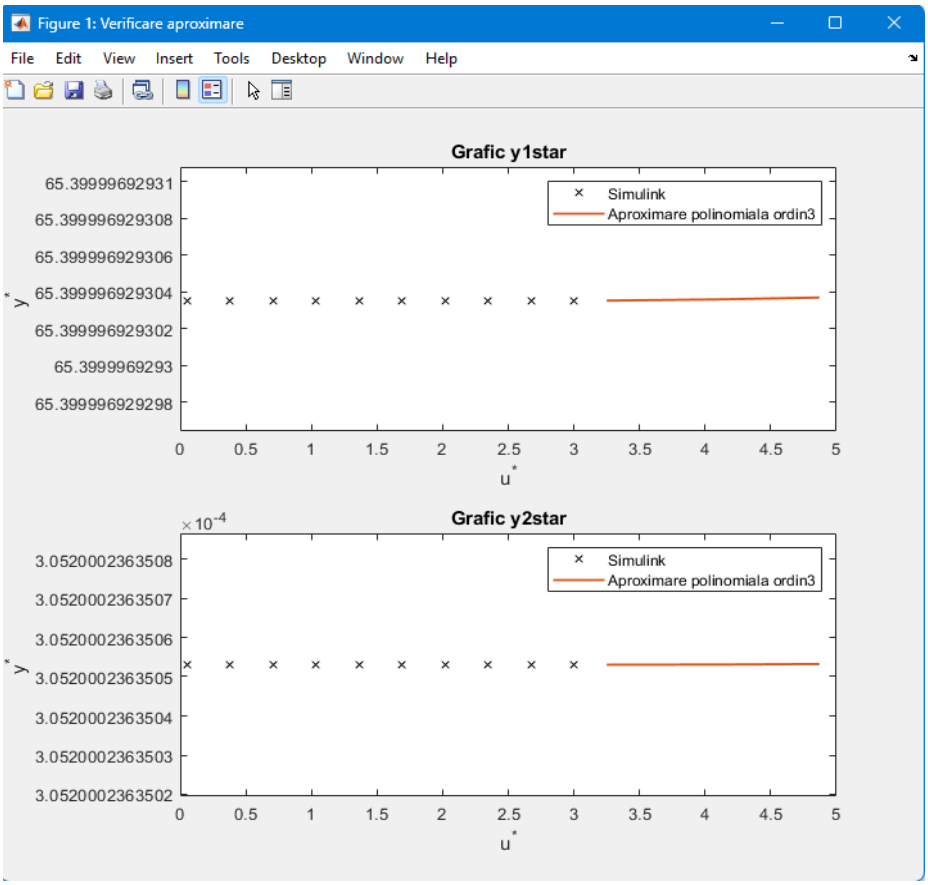
Command Window

```
Ustar
    0.0500
    0.3778
    0.7056
    1.0333
    1.3611
    1.6889
    2.0167
    2.3444
    2.6722
    3.0000

Y1star
   65.4000
   65.4000
   65.4000
   65.4000
   65.4000
   65.4000
   65.4000
   65.4000
   65.4000
   65.4000

Y2star
   1.0e-03 *

    0.3052
    0.3052
    0.3052
    0.3052
    0.3052
    0.3052
    0.3052
    0.3052
    0.3052
    0.3052
```

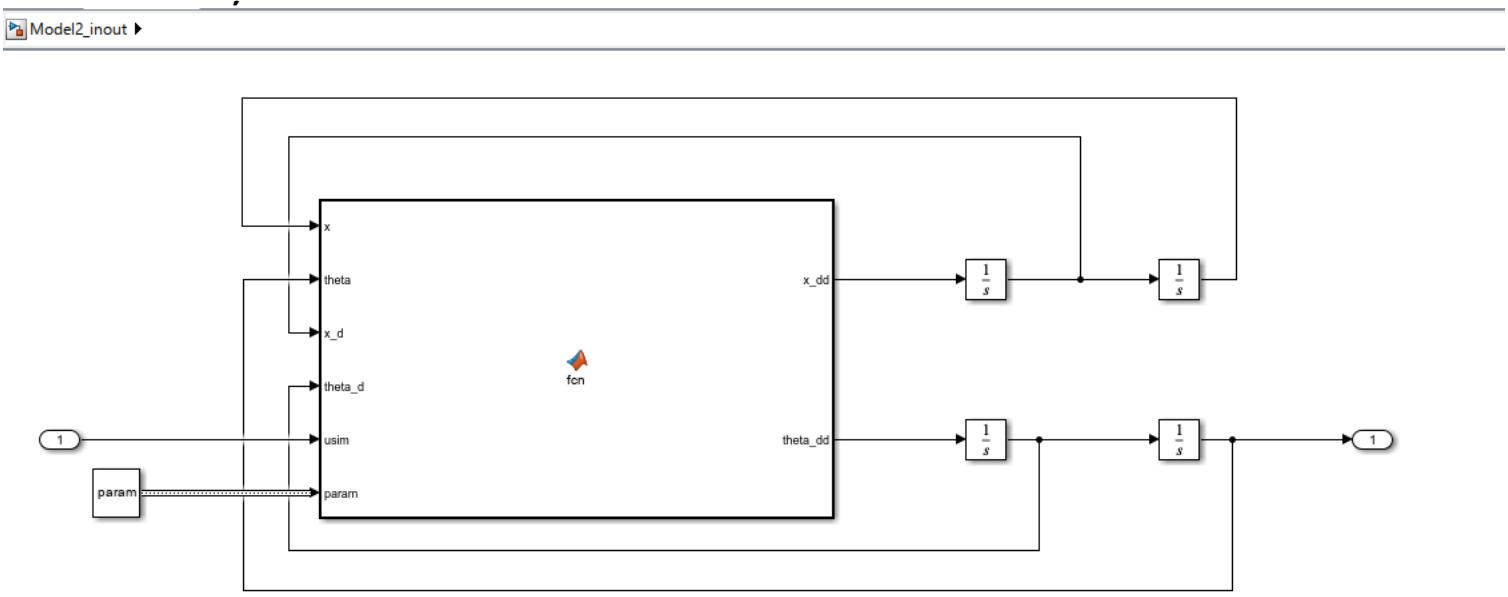
Cerința 6:

Am ales valori pentru alfa, beta, gamma si am creat si afisat aproximarea raspunsurilor viitoare.

```
%% Cerinta 6
% Alegerea valorilor
alfa = 3.25;
beta = 4.14;
gamma = 4.87;

% Crearea noilor variabile pentru a verifica aproximarea
ustar6 = [alfa, beta, gamma];
y1star6 = polyval(p1_ord3, ustar6);
y2star6 = polyval(p2_ord3, ustar6);

figure('Name', 'Verificare aproximare');
subplot(2,1,1);
plot(ustar, y1star, 'xk');
hold on
plot(ustar6, y1star6, 'LineWidth', 1.4);
xlabel('u^*'), ylabel('y^*');
legend('Simulink', 'Aproximare polinomiala ordin3');
title('Grafic y1star');

subplot(2,1,2);
plot(ustar, y2star, 'xk');
hold on
plot(ustar6, y2star6, 'LineWidth', 1.4);
xlabel('u^*'), ylabel('y^*');
legend('Simulink', 'Aproximare polinomiala ordin3');
title('Grafic y2star');

```



Cerința 7:



Cerința 8:

```
166  %% Cerinta 8
167  %Valoarea intrarii pentru care se determina Punctul Static de Functionare
168  u0 = ustar(1);
169  %Determinam Punctul Static de Functionare
170  [xstarr, ustarr, ystarr, ~] = trim("Model2_inout", [], u0, [], [], 1, []);
171
172  err = norm(abs(ustarr - u0));
173  disp('Eroarea dintre ustarr si u0:');
174  disp(err);
175  % Nu exista diferente intre cele doua valori,
176  % asa cum asteptam, in functia trim am setat ca intrarea sa fie fixata
177
```

```
Eroarea dintre ustarr si u0:
     0
```

Cerința 9:

```
178  %% Cerinta 9
179  %Liniarizarea in Punctul Static de Functionare anterior
180  [A_lin, B_lin, C_lin, D_lin] = linmod("Model2_inout", xstarr, ustarr);
181
```

Command Window

```
A_lin =

         0         0         0    1.0000
   -0.0001   -0.9000   -0.1500         0
         0    1.0000         0         0
   -0.1469         0   -0.0000   -0.9000


B_lin =

   1.0e-04 *

         0
         0
         0
    0.4482


C_lin =

     1     0     0     0


D_lin =

     0
```

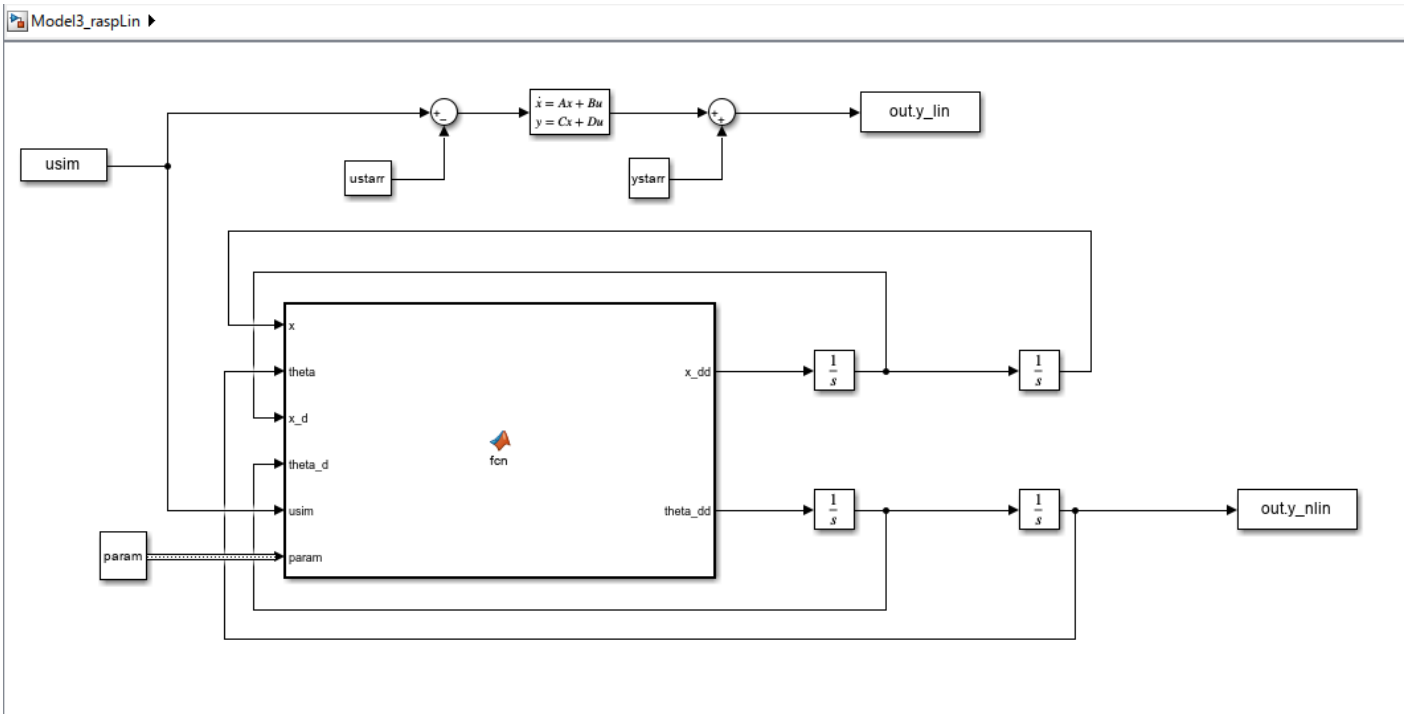
Cerința 10:

```
182   %% Cerinta 10
183   % Sistemul liniarizat este stabil deoarece toate valorile proprii ale lui
184   % A_lin au partea reala negativa
185   vp = eig(A_lin);
186   disp('Valorile proprrii A_lin:');
187   disp(vp);
188
```

```
Valorile proprrii A_lin:
   -0.2141
   -0.2209
   -0.6859
   -0.6791
```

Cerința 11:

```
189   %% Cerinta 11
190   load_system('Model3_raspLin')
191   usim = timeseries(st, t);
192   set_param('Model3_raspLin', 'StopTime', num2str(T))
193   out = sim('Model3_raspLin');
194   y_lin = out.y_lin;
195
```



Cerința 12:

```
196   %% Cerinta 12
197   y_nlin = out.y_nlin;
198   
199   y_lin_reshaped = reshape(y_lin.data, [1,66]);
200   y_nlin_reshaped = reshape(y_nlin.data, [1,66]);
201   
202   err = norm(y_lin_reshaped - y_nlin_reshaped, 'inf');
203   disp('Eroarea dintre iesirea liniara si iesirea neliniara');
204   disp(err);
205
```

```
Eroarea dintre iesirea liniara si iesirea neliniara
    0.0076
```

Cerința 13:

```
206   %% Cerinta 13
207   [b, a] = ss2tf(A_lin, B_lin, C_lin, D_lin);
208   H_lin = tf(b, a);
209   Te = 0.09;
210   H_disc = c2d(H_lin, Te, 'tustin')
211   
212   % Am ales aproximarea Tustin deoarece sistemul este stabil, aam demonstrat
213   % la cerinta 10. Aproximarea Tustin plaseaza polii stabili ai sistemului
214   % continuu in polii stabili ai sistemului discret.
215   % Astfel, sistemul discretizat va fi si el stabil.
216
```

```
H_disc =

  8.72e-08 z^4 + 6.888e-09 z^3 - 1.674e-07 z^2 - 6.685e-09 z + 8.042e-08
  ----------------------------------------------------------------------
             z^4 - 3.842 z^3 + 5.535 z^2 - 3.543 z + 0.8504
```