

```

% SCRIPT DI INIZIALIZZAZIONE PARAMETRI PER AEROSTATO AD ARIA CALDA

if(~exist('init_done'))
V=[];
m=[];
theta_t=[];
f = [];

rho_a = 1.2928;
c_a = 1008;
g=9.80665;

init_done = 1;
end

while isempty(m)
    m = input('Inserire le prime due cifre (esclusi zeri) del proprio
numero di matricola:\n\n');
end

text=sprintf('\nMassa dell''aerostato: %3.0f kg\n',m); disp(text);

while isempty(V)
    V = input('Inserire l''ultima cifra del proprio numero di
matricola (se 0 -> 5):\n\n');
end

V = V*m*2;
C = V*rho_a*c_a;

text=sprintf('\nVolume del pallone dell''aerostato: %3.0f m^3\n',V);
disp(text);

text=sprintf('\nCapacita'' termica dell''aerostato: %6.0f J/°C\n',C);
disp(text);

while isempty(theta_t)
    theta_t = input('Inserire la penultima cifra del proprio numero
di matricola (se 0 -> 4):\n\n');
end

theta_t = theta_t/50;

text=sprintf('\nResistenza termica tra aerostato e aria esterna:
%1.3f °C/W\n',theta_t); disp(text);

while isempty(f)

```

```

    f = input('Inserire le ultime due cifre (esclusi zeri) del
proprio numero di matricola:\n\n');
end

f = f / 10;

text=sprintf('\nCoefficiente di attrito viscoso aerodinamico: %1.2f N
s / m\n',f); disp(text);

%Temperatura ambiente (20 °C, riportati in °K)
Ta = 20+273.15;
%SET-POINT:
x1_d = Ta/(1-m/(V*rho_a));
x2_d = 120;
x3_d = 0;
%CONDIZIONI INIZIALI:
x1_0 = x1_d;
x2_0 = 100;
x3_0 = 0;

text=sprintf('\nCondizioni iniziali: temperatura interna (x1_0) =
%1.2f °K, quota (x2_0) = %1.2f m, velocita'' ascensionale(x3_0) =
%1.2f m/s\n',x1_0,x2_0,x3_0);
disp(text);

text=sprintf('\nCondizioni desiderate (set-point): temperatura
interna (x1_d) = %1.2f °K, quota (x2_d) = %1.2f m, velocita''
ascensionale(x3_d) = %1.2f m/s\n',x1_d,x2_d,x3_d);
disp(text);

clear text;

```

```
% Calcoliamo la linearizzazione approssimata in  $x = [x1\_d, x2\_d, x3\_d]$   
e con
```

```
u_d = (Ta*m)/((V*rho_a - m)*theta_t)
```

```
[A_balloon, B_balloon, C_balloon, D_balloon] =  
linmod('balloon_model', [x1_d, x2_d, x3_d], u_d);
```

```
Q_balloon = [1 0 0; 0 10 0; 0 0 1];  
R_balloon = 0.1;
```

```
[K, S, E] = lqr(A_balloon, B_balloon, Q_balloon, R_balloon)
```

```

%%%
%%% Progetto filtro di Kalman per il problema LQG del balloon
%%%

%%% Filtro di Kalman con 4 ingressi e 3 uscite

%%%
%%% Si fa riferimento al filtro di Kalman per il modello
%%%
%%%      .
%%%      x = Ax + Bu + Gw           {State equation}
%%%      y = Cx + Du + Hw + v     {Measurements}
%%%

G_balloon = B_balloon;

%%% Verificare la controllabilita' della coppia (A_balloon,Gballoon)
nc = rank(ctrb(A_balloon,G_balloon))

%%% Verificare l'osservabilita' della coppia (A_balloon,C_balloon)
no = rank(observ(A_balloon,C_balloon))

Wballoon = 1e-6;                               % = E{ww'},
Vballoon = diag([1e-7 1e-7 1e-7]); % = E{vv'}

%[Pkf,Ekf,Kkft] =
care(A_balloon',C_balloon',G_balloon*Wballoon*G_balloon',Vballoon); %
CARE duale!

%Kkf = Kkft';

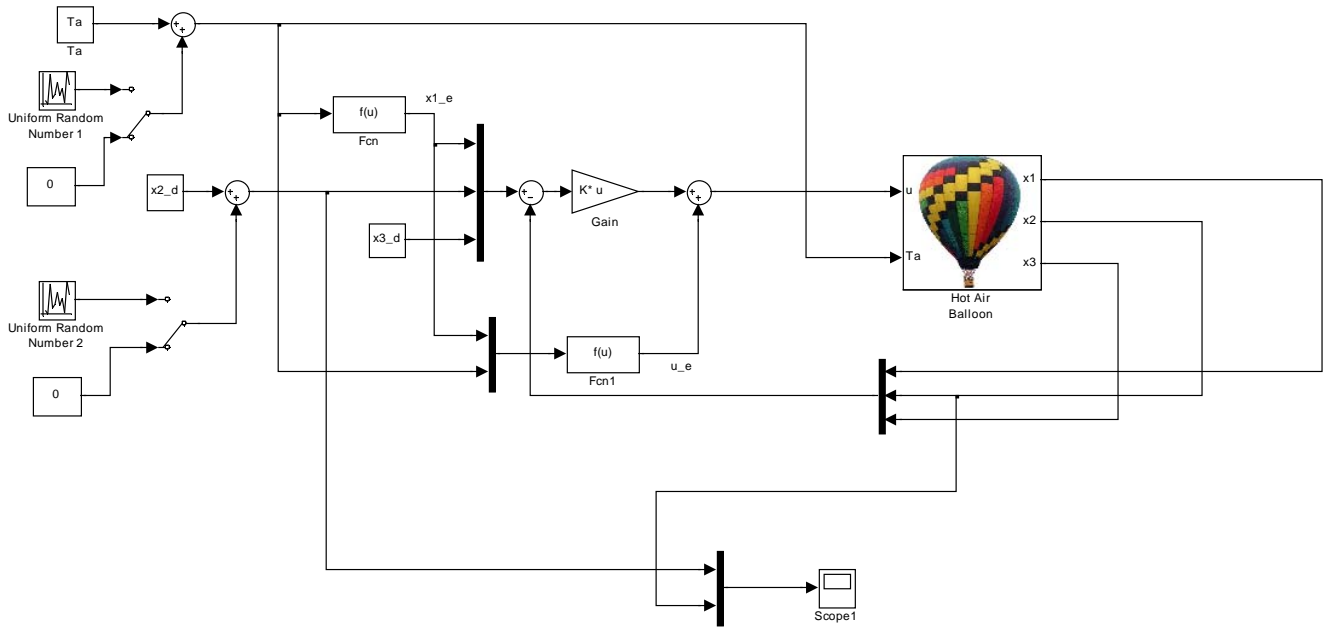
[Kkf,Pkf,Ekf] =...
lqr(A_balloon',C_balloon',G_balloon*Wballoon*G_balloon',Vballoon)

%%% Matrici del filtro di Kalman: E' un osservatore dello stato!

Akf = A_balloon - Kkf * C_balloon;
Bkf = [B_balloon Kkf];
Ckf = eye(3);
Dkf = zeros(3,4);

return

```



**Note**

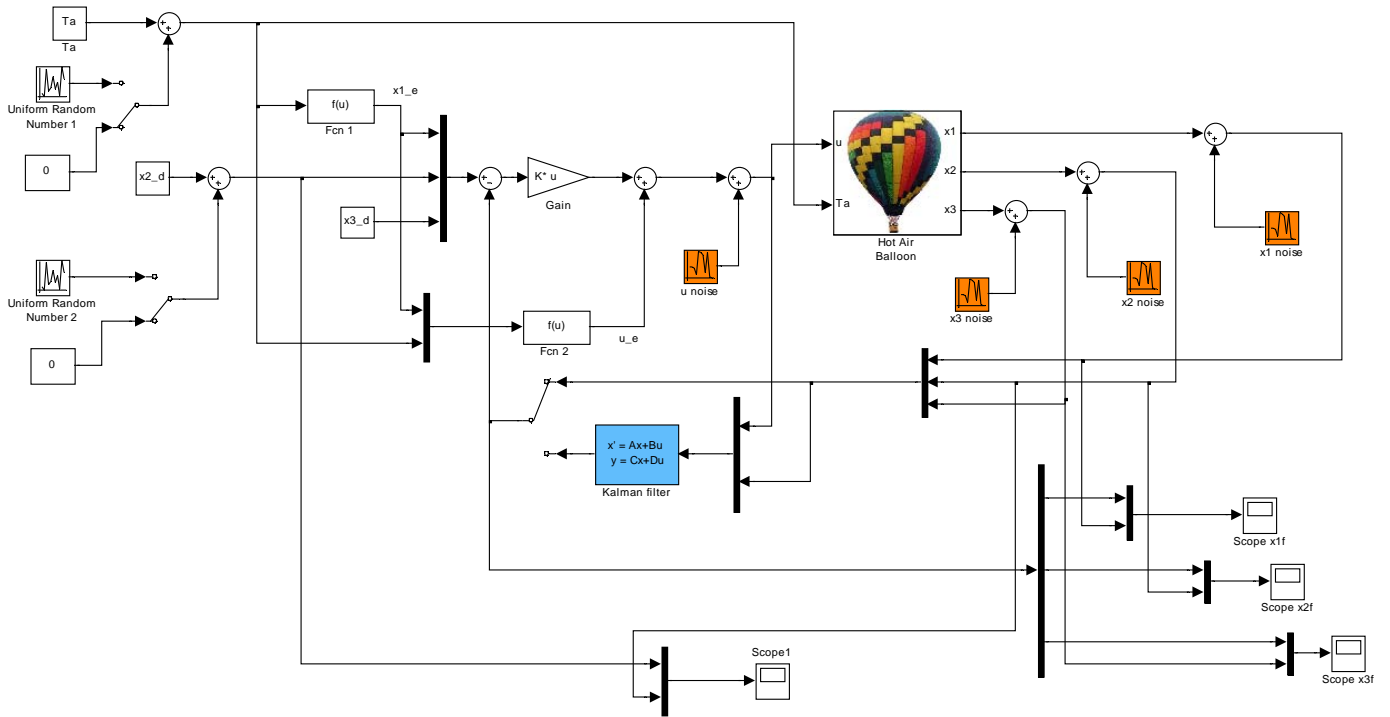
Simulation time: 50000

Fnc:  $u(1)/(1-m/(V*\rho_a))$

Fnc1:  $(u(1)-u(2))/\theta_{t_t}$

Uniform random number 1: [-1, 1], sample time 200

Uniform random number 2:  $[-x2\_d*0.1, x2\_d*0.1]$ , sample time 4000



**Note**

Fcn 1:  $u[1]/(1-m/(V*\rho_a))$

Fcn 2:  $(u[1]-u[2])/theta_t$

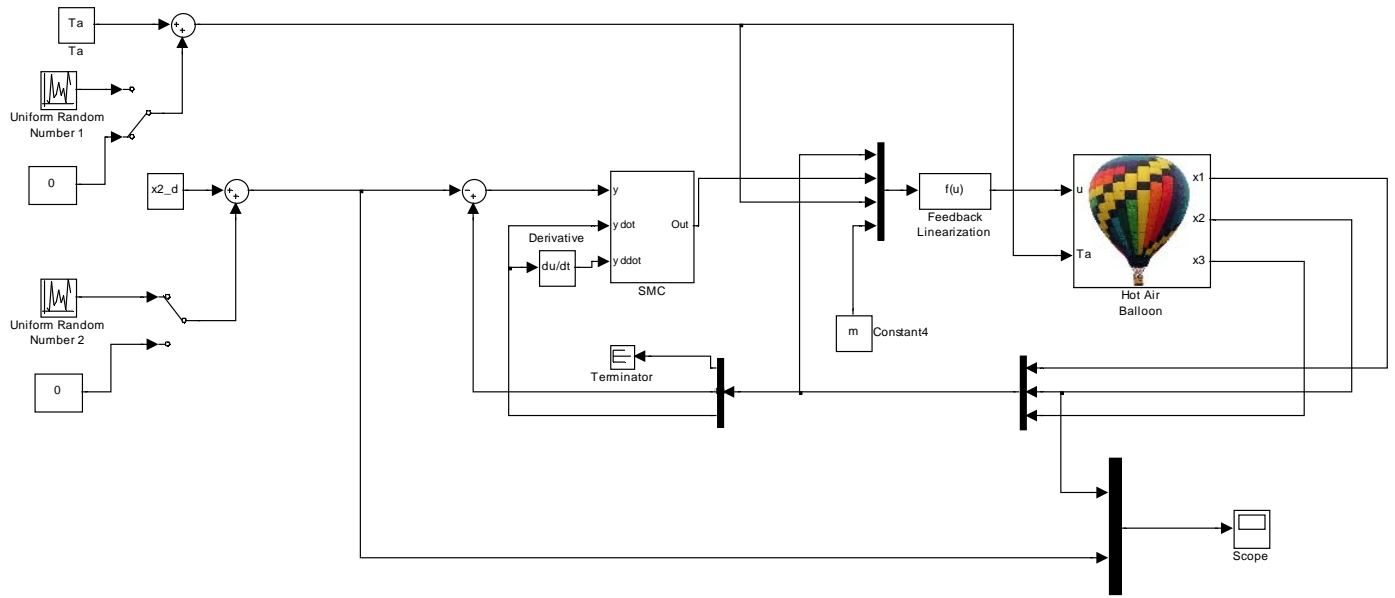
Simulation time 50000

U noise: variance  $1e-6$

X1 noise: variance  $1e-7$

X2 noise: variance  $1e-7$

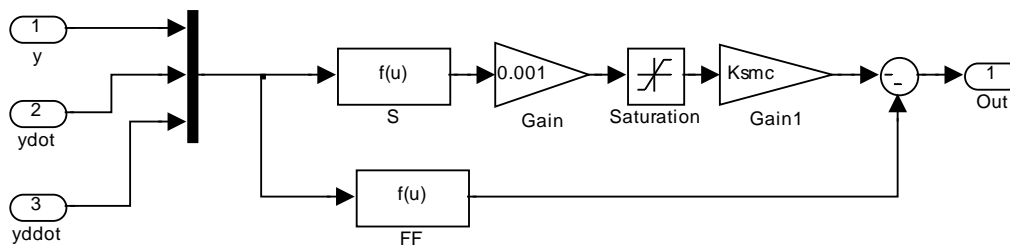
x3 noise: variance  $1e-7$



## Feedback linearization

$$F_{cn} = u(4) * (C * u(6) * u(1)^2) / (g * V * u(5) * rho\_a) - (C * f * u(1)) / u(6) + (C * f * u(1)^2) / (u(6) * u(5)) - u(5) / theta\_t + u(1) / theta\_t - (C * f * u(1)^2) / (V * u(5) * rho\_a) - (C * f^2 + u(1)^2 * u(3)) / (g * u(6) * V * u(5) * rho\_a)$$

## Sliding mode



$$S = u[3] + 2 * \lambda * u[2] + \lambda^2 * u[1]$$

$$FF = 2 * \lambda * u[3] + \lambda^2 * u[2]$$