



# Controllo di Sistemi Robotici

- *Implementazione in Matlab/Simulink del filtro di Kalman e controllo LQG*

# Progetto del filtro di Kalman stazionario

➔ Comando Kalman:

```
sys = ss(A, [B G], C, [D zeros(-, -)]);
```

Matrice d'ingresso dei  
disturbi  $\Gamma$

Dimensioni opportune

```
[kest, L_o, P_o] = kalman(sys, W, V, 0);
```

Guadagno del filtro di  
Kalman stazionario  
tempo - continuo

Matrice di covarianza  
dell'errore di stima a  
regime

# Script di inizializzazione

```
%% Sistema tempo continuo

A = [-1 3 0;0 -2 1;0 0 -5];
B = [1 0;0 1;0 1];
C = eye(3);
D = zeros(3,2);
G = eye(3); % matrice d'ingresso dei
disturbi
x0 = [1;-1;0.1];

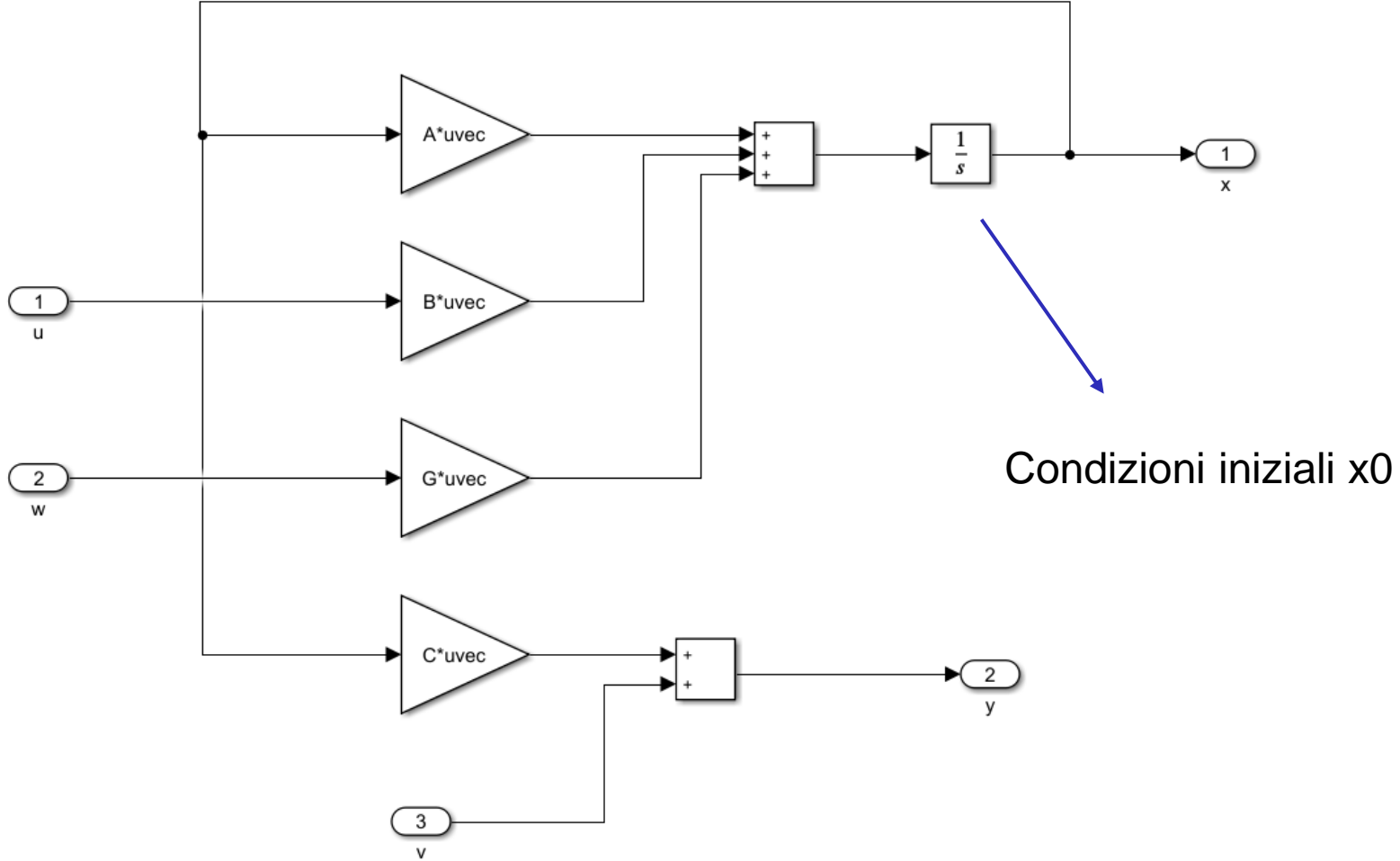
%% Matrici di covarianza del rumore
W = 0.1*eye(3);
V = 0.1*eye(3);
```

# Script di inizializzazione

```
%% progetto filtro stazionario
sys = ss(A, [B G], C, [D zeros(3,3)]);
[kest, L_o, P_o] = kalman(sys, W, V, 0);
x0s = [0;0;0];

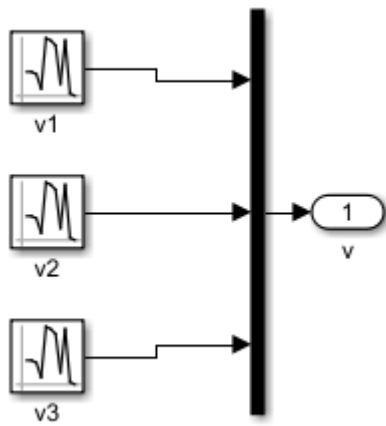
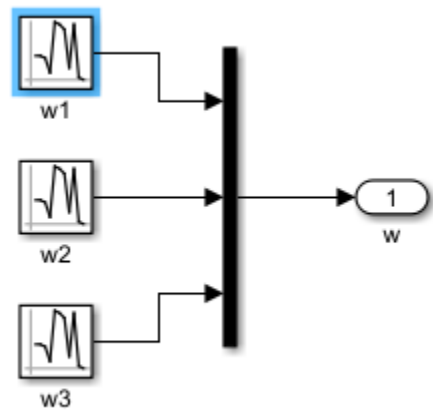
%% Osservatore Luenberger (per confronto)
p = [-5; -6; -7];
L_l = place(A', C', p)';
```

# Modello simulink del sistema in ambiente stocastico



# Rumore di processo e di misura

Random number



Block Parameters: w1

Random Number

Output a normally (Gaussian) distributed random signal. Output is repeatable for a given seed.

Parameters

Mean: 0

Variance: W(1,1)

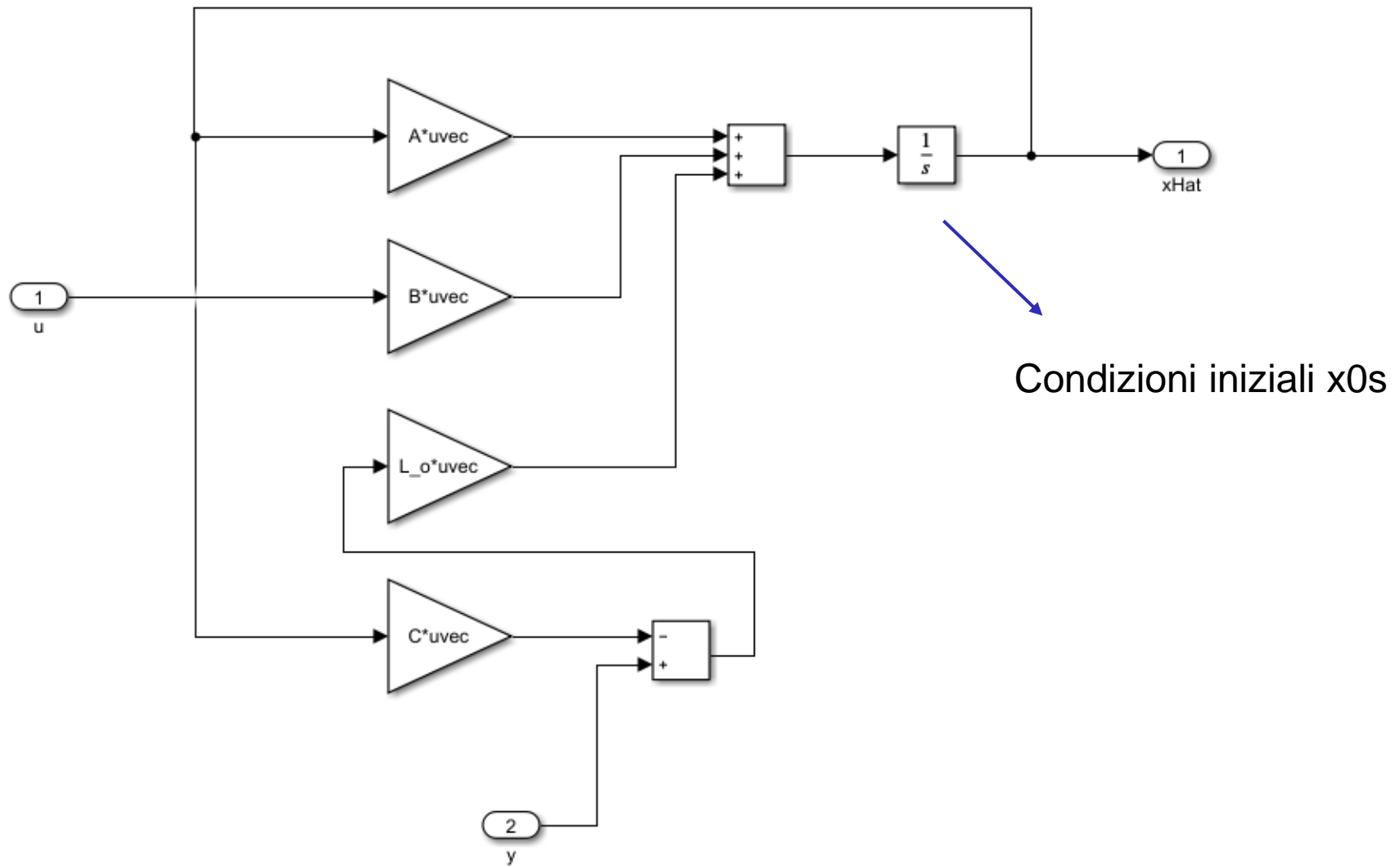
Seed: 0

Sample time: 0.01

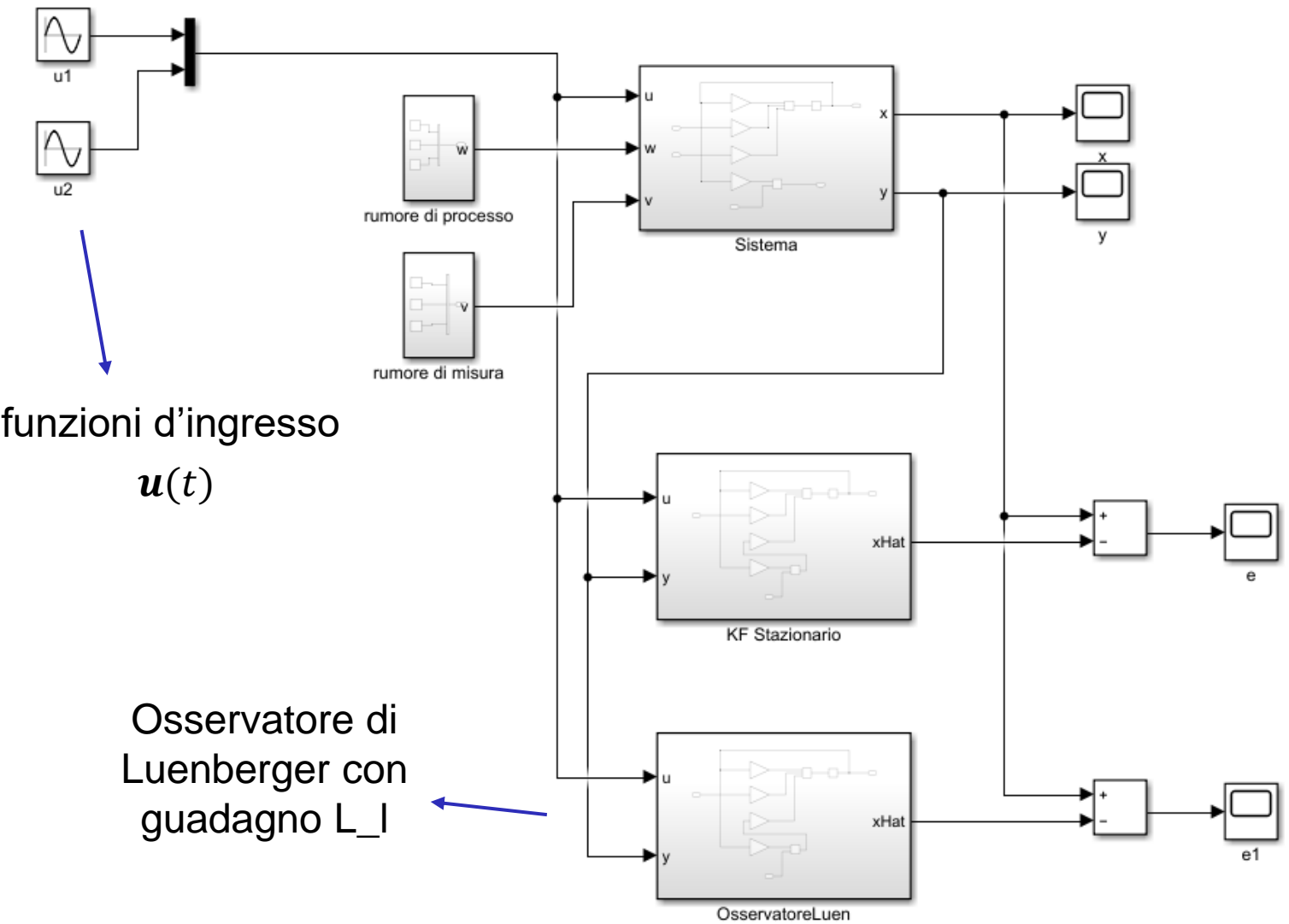
Interpret vector parameters as 1-D

OK Cancel Help Apply

# Modello simulink del filtro di Kalman stazionario

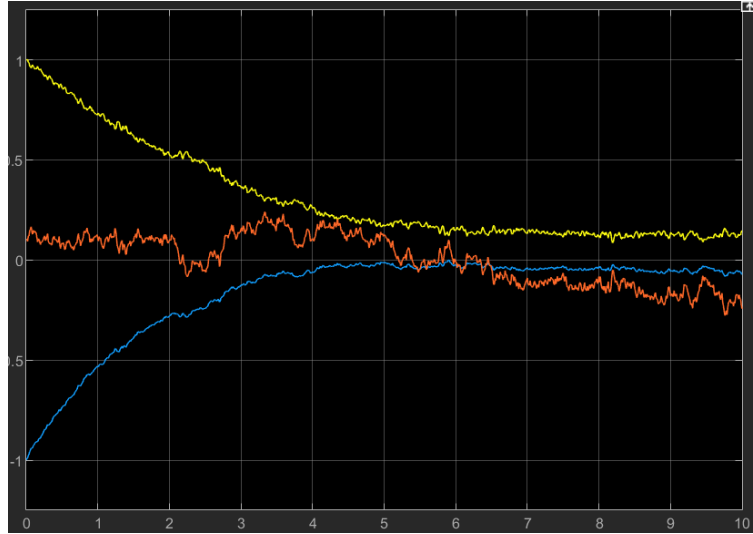
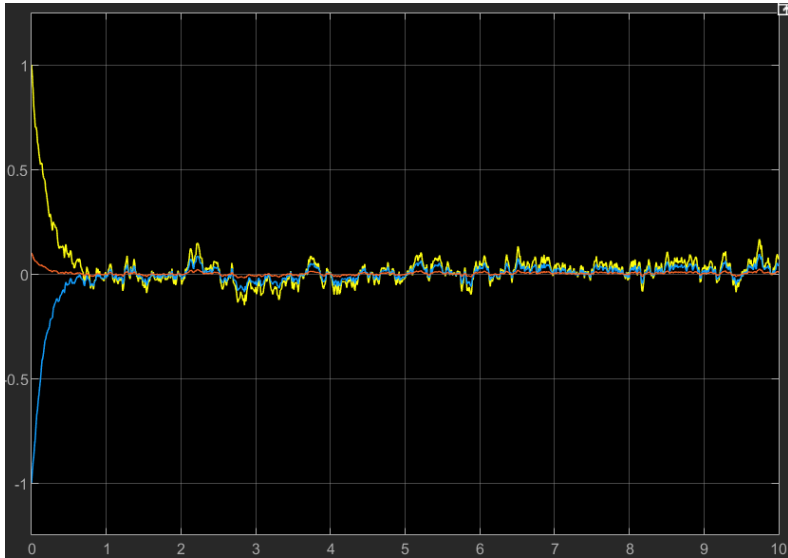


# Modello simulink completo



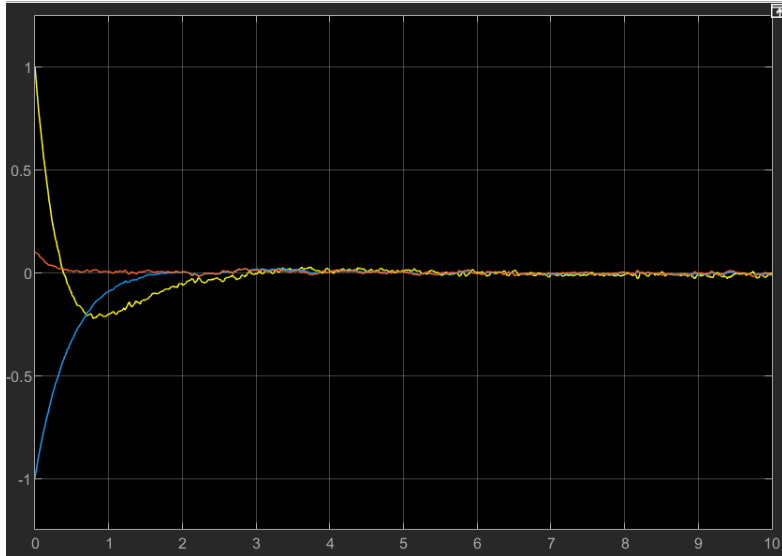


# Risultati, errore di stima



Osservatore Luenberger,  $p = [-5, -6, -7]$

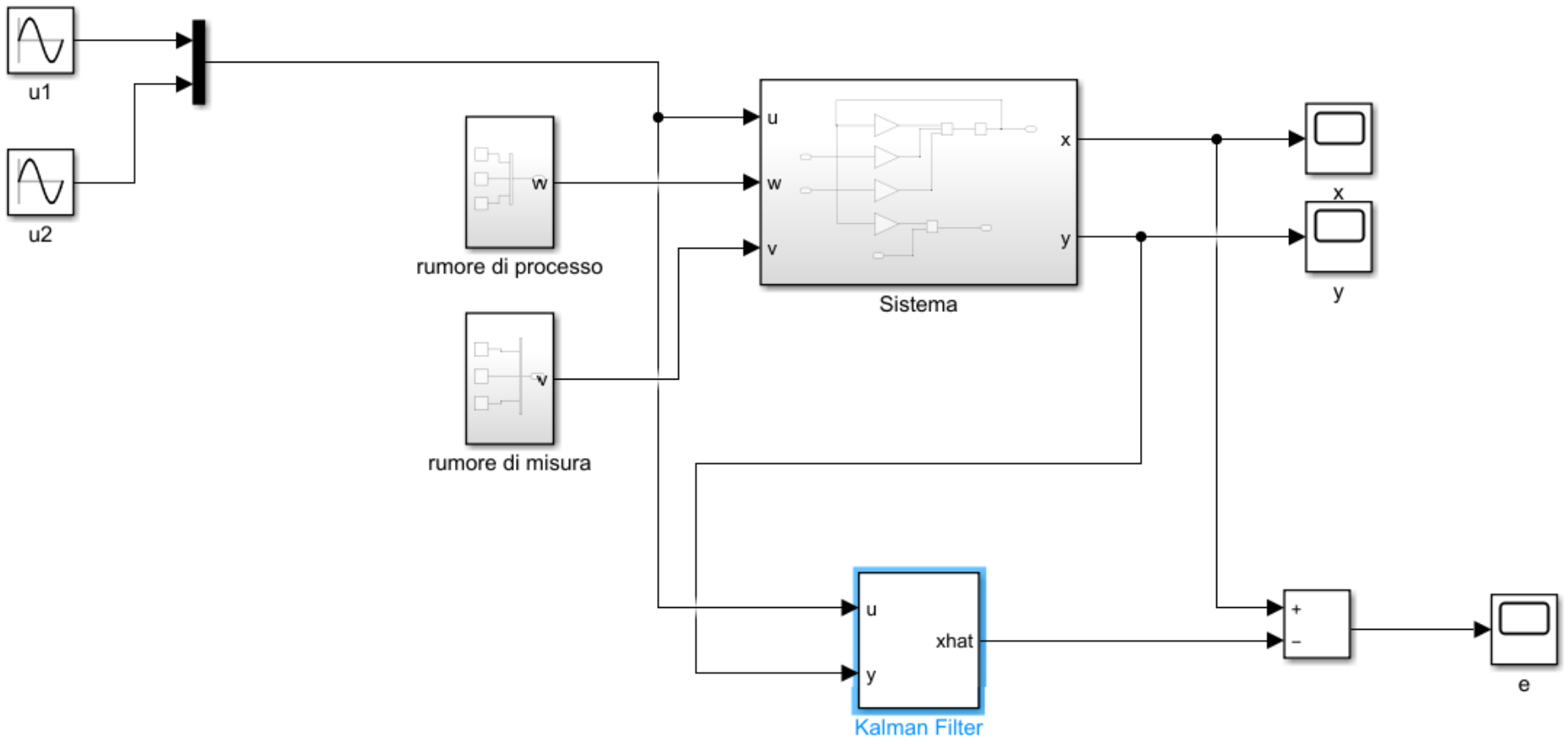
Osservatore Luenberger,  $p = [-0.3, -0.6, -0.7]$



Filtro di Kalman



# Alternativa, blocco Kalman filter



Blocco Simulink 'Kalman Filter'

# Alternativa, blocco Kalman filter (configurazione)

Block Parameters: Kalman Filter

Kalman Filter

Estimate the states of a discrete-time or continuous-time linear system. Time-varying systems are supported.

Filter Settings

Time domain: Continuous-Time

Model Parameters Options

System Model

Model source: Individual A, B, C, D matrices

A: A B: B

C: C D: D

Initial Estimates

Source: Dialog

Initial states  $x(0)$ :  $x0s$

Noise Characteristics

Use G and H matrices (default  $G=I$  and  $H=0$ )

Q: W  Time-invariant Q

R: V  Time-invariant R

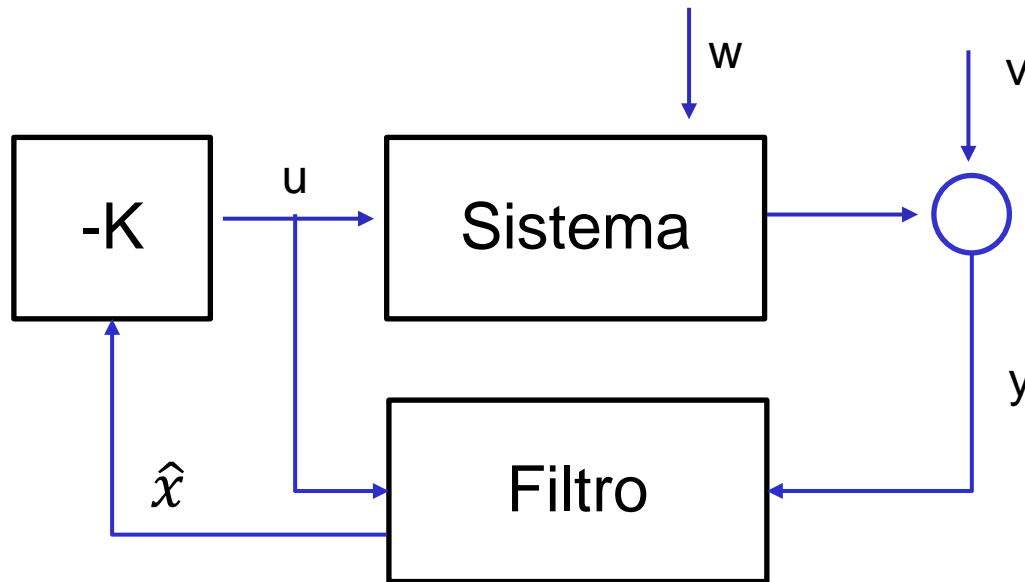
N: 0  Time-invariant N

OK Cancel Help Apply



# Regolatore LQG (Lineare Quadratico Gaussiano)

- ➔ Si vuole regolare il sistema nell'origine utilizzando la tecnica LQ
- ➔ Non si ha a disposizione lo stato effettivo del sistema, ma solo le misure su di esso, la conoscenza dell'ingresso di controllo e del modello (stocastico) del sistema
- ➔ Si utilizza il filtro di Kalman stazionario per generare la stima ottima dello stato e retroazionarla secondo la legge  $u = -K\hat{x}$ , con  $K$  guadagno ottimo del regolatore LQ



# Regolatore LQG, script di inizializzazione

```
%% Sistema tempo continuo
```

```
A = [-1 3 0;0 -2 1;0 0 -5];
```

```
B = [1 0;0 1;0 1];
```

```
C = eye(3);
```

```
D = zeros(3,2);
```

```
G = eye(3); % matrice d'ingresso dei disturbi
```

```
x0 = [1;-1;0.1];
```

```
%% Matrici di covarianza del rumore
```

```
W = 0.1*eye(3);
```

```
V = 0.1*eye(3);
```

# Regolatore LQG, script di inizializzazione

```
%% progetto filtro stazionario
sys = ss(A, [B G], C, [D zeros(3,3)]);
[kest, L_o, P_o] = kalman(sys, W, V, 0);
x0s = [0;0;0];
```

```
%% Controllo LQ
```

```
Q = eye(3);
```

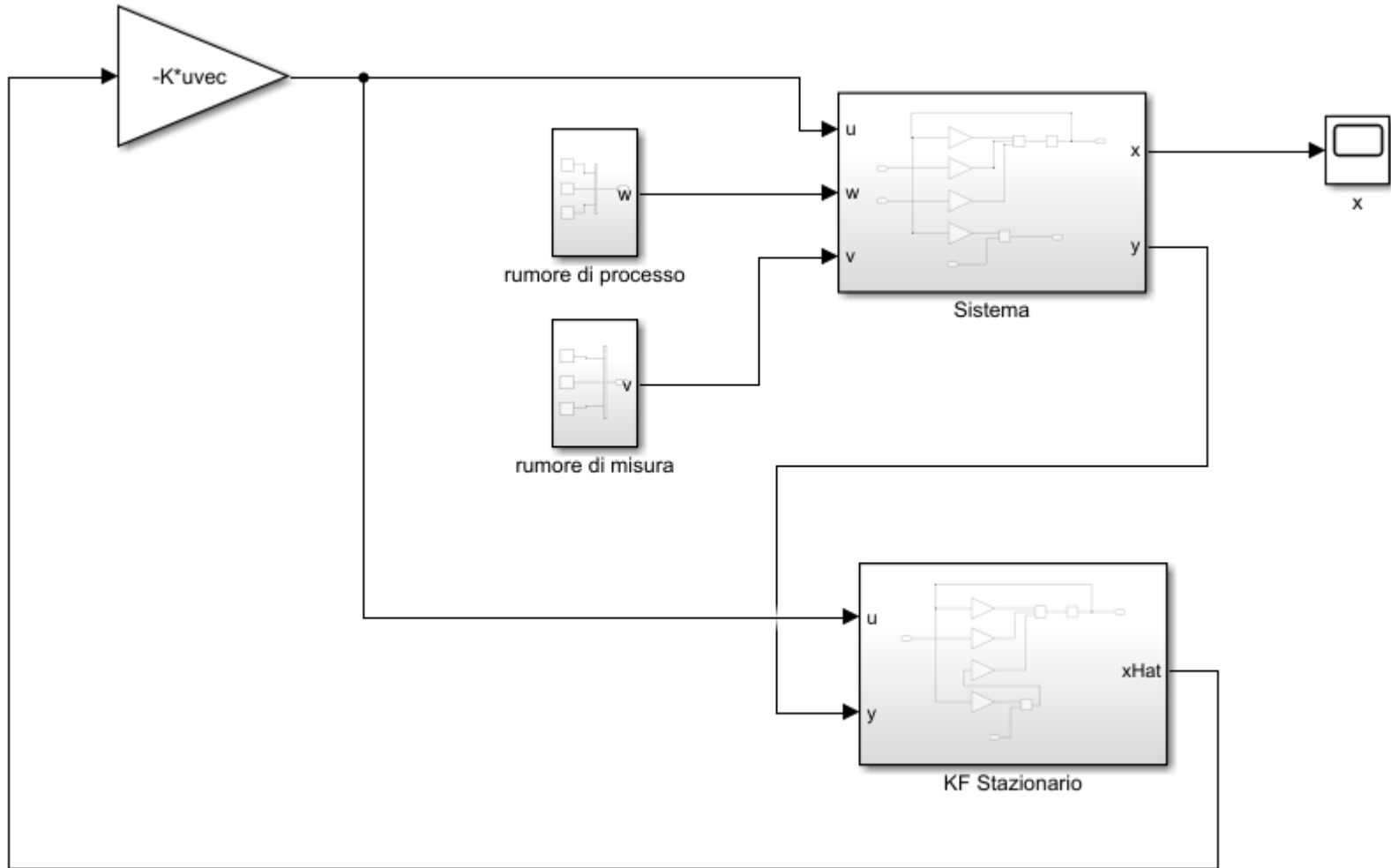
```
R = eye(2);
```

```
sys_det = ss(A, B, C, D);
```

```
[K, S, e] = lqr(sys_det, Q, R, 0);
```

Sistema in ambiente deterministico

# Regolatore LQG, schema di controllo



# Regolatore LQG, risultati

➔ Stato effettivo  $x$

