Estimation and Control of Dynamica Systems

Stefano Di Lena

2025

Indice

1 Introduzione

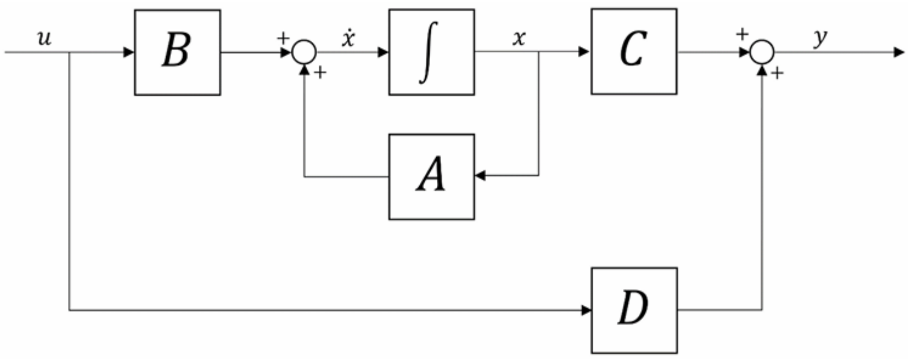
Un sistema dinamico LTI-TC è descritto dalle equazioni:

$$S : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Con $x \in R^n, y \in R^p, u \in R^m$.

Le dimensioni delle matrici sono: $A : n \times n$, $B : n \times m$, $C : p \times n$, $D : p \times m$.

Il sistema può essere rappresentato graficamente dal seguente schema a blocchi:



Se il sistema è strettamente proprio $D = 0$.

Per i sistemi LTI-TD:

$$S : \begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

2 Controllabilità

Un sistema dinamco LTI-TC si dice *controllabile* quando esiste un ingresso u che trasferisce lo stato del sistema da uno stato iniziale $x_0 = x(0)$ ad un qualunque stato $x_f = x(t_f)$ in un tempo $t_f > 0$ finito.

2.1 Gramiano di Controllabilità

É una matrice $n \times n$:

$$\Gamma(t) = \int_0^t e^{A\tau} BB^T e^{A^T\tau} d\tau$$

Il sistema è controllabile se e solo se:

$$\rho(\Gamma(t)) = n \qquad \forall t > 0$$

L'ingresso di controllo u, che trasferisce lo stato del sistema da x_0 ad x_f è:

$$u(t) = B^T e^{A^T(t_f - t)} \Gamma^{-1}(t_f)(x_f - e^{At_f} x_0)$$

Per i sistemi LTI-TD non esiste il gramiano per calcolare l'ingresso in grado di trasferire lo stato da x_0 ad x_f, ma esiste una formula in grado di calcolare $u(k)$ che porta il sistema da x_0 ad x_f in massimo n passi (se il sistema è completamente controllabile). Dato un tempo $t \geq n$, si definisce il vettore contenente il valore di $u(k)$ negli istanti di tempo da 0 a $t-1$:

$$u^{t-1} = \begin{bmatrix} u(t-1 \\ \vdots \\ u(0) \end{bmatrix} = C_t^T (C_t C_t^T)^{-1} x_f$$

con C_t matrice di controllabilità in t passi (di dimensione $n \times mt$).

2.2 Matrice di Kalman di Controllabilità

$$M_c = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

Se $\rho(M_c) = n$ allora il sistema è controllabile. Altrimenti è possibile controllare solo n_c componenti [se $\rho(M_c) = n_c < n$].

2.2.1 Forma di Kalman di Controllabilità

Se si effettua una trasformazione per similitudine sullo stato (recordination), si identifica un nuovo vettore di stato tale che: $x = P_c z \iff z = P_c^{-1} x$.

La matrice di controllabilità diventa $\tilde{M}_c = P_c^{-1} M_c = \begin{bmatrix} M_c \\ 0 \end{bmatrix}$.

Per calcolare P_c: *scegliamo le prime n_c colone indipendenti di M_c e le restanti $n - n_c$ colonne sono scelte in modo da essere indipendenti dalle prime e tra loro.*

Possiamo scrivere scomponendo z le equazioni di stato e di uscita come:

$$\begin{cases} \dot{z}_c = A_c z_c + A_1 z_{n_c} + B_c u \\ \dot{z}_{n_c} = A_{n_c} z_{n_c} \\ y = C_1 z_c + C_2 z_{n_c} \end{cases}$$

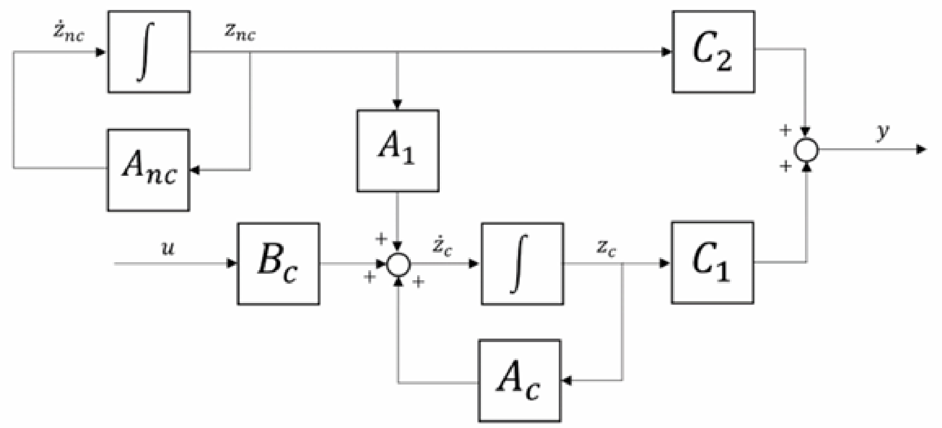
Con:

$$\begin{bmatrix} A_c & A_1 \\ 0 & A_{n_c} \end{bmatrix} = P_c^{-1} A P_c$$

$$\begin{bmatrix} B_c \\ 0 \end{bmatrix} = P_c^{-1} B$$

$$\begin{bmatrix} C_1 & C_2 \end{bmatrix} = C P_c$$

Lo schema a blocchi del sistema è:



L'evoluzione è $z_{n_c} = e^{A_{n_c} t} z_{n_c}(0)$.

3 Retroazione di Stato

Consiste nel sottrarre all'ingresso il vettore di stato, pre-moltiplicato per un opportuna matrice K.

Il nuovo ingresso di controllo è $u = r - Kx$, con $r \in R^m, k \in R^{n \times m}$. L'equazione di stato diventa:

$$\dot{x} = Ax + Bu = Ax + B(r - Kx) = (A - BK)x + Br = A_k x + Br$$

3.1 Forma Compagna

$$\dot{x}_\gamma = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & \cdots & -\alpha_{n-1} \end{bmatrix} x_\gamma + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

Dove $\alpha_0, \cdots, \alpha_{n-1}$ sono i coefficienti del polinomio caratteristico della matrice di stato A_γ.

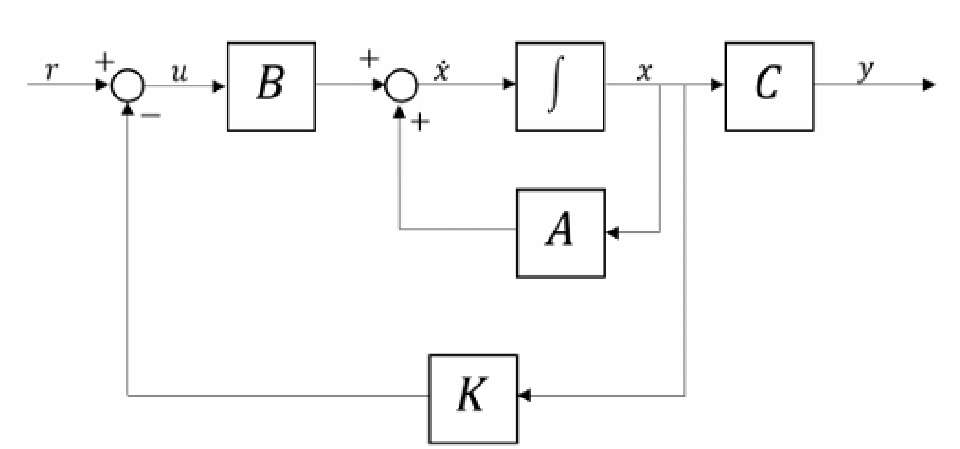
$$p(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_1\lambda + \alpha_0$$

Un sistema in questa forma è controllabile; perciò i sistemi controllabili possono essere trasformati $x = P_\gamma x_\gamma$. Data la matrice di retroazione:

$$K_\gamma = \begin{bmatrix} \bar{\alpha}_0 - \alpha_0 & \bar{\alpha}_1 - \alpha_1 & \cdots & \bar{\alpha}_{n-1} - \alpha_{n-1} \end{bmatrix}$$

Dove $\bar{\alpha}_0, \cdots, \bar{\alpha}_{n-1}$ sono i coefficienti del polinomio caratteristico desiderato. $P = M_c M_c^{-1}$.

$$M_c^{-1} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} & 1 \\ \alpha_2 & \alpha_3 & \cdots & 1 & 0 \\ \vdots & \vdots & \cdots & 0 & 0 \\ \alpha_{n-1} & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$



3.1.1 Stabilizzabilità

Nei sistemi *non controllabili* ha senso fare la retroazione di stato sulla parte controllabile solo se il sistema è *stabilizzabile*, ovvero se per ogni polo instabile è verificata la seguente condizione:

$$\rho(\begin{bmatrix} A - \lambda I & B \end{bmatrix}) = n$$

4 Osservabilità

Un sistema dinamico LTI-TC è detto osservabile se e solo se il suo stato iniziale x_0 può essere determinato a partire dall'uscita y per un tempo finito $t_f > 0$.

4.1 Gramiano di Osservabilità

$$O(t) = \int_0^t e^{A^T\tau} C^T C e^{A\tau} d\tau$$

Il sistema è osservabile se e solo se:

$$\rho(O(t)) = n \quad \forall t > 0$$

Dall'uscita si può ricostruire lo stato iniziale:

$$x(0) = O^{-1}(t_f)\int_0^{t_f} e^{A^T\tau}C^T y(\tau)d\tau$$

Nei sistemi LTI-TD, invece, dato il vattore contenente i valori dell'uscita libera:

$$y_\ell^{t-1} = \begin{bmatrix} y(t-1) \\ \vdots \\ y(0) \end{bmatrix}$$

Il valore dello stato iniziale è dato da:

$$x(0) = (O_t^T O_t)^{-1} O_t^T y_\ell^{t-1}$$

Con O_t matrice di osservabilità in t passi (di dimensione $pt \times n$).

4.2 Matrie di Kalman di Osservabilità

$$M_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

Il sistema è osservabile se e solo se $\rho(M_o) = n$. Se, invece, $\rho(M_o) = n_0 < n$; allora solo n_o componenti dello stato iniziale potranno essere determinate.

4.2.1 Forma di Kalman di Osservabilità

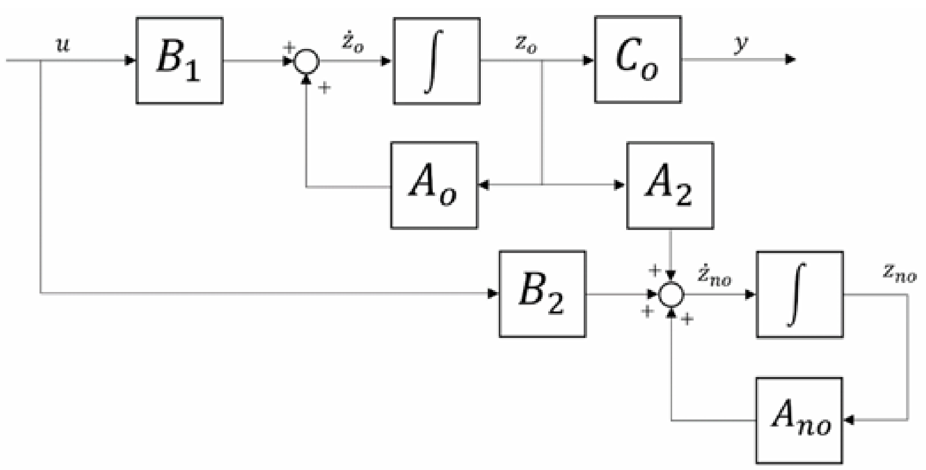
La matrice di trasformazione P_o è scelta in modo tale che *le prime n_o colonne coincidano con le n_o righe indipendenti di M_o trasposte, e le restanti $n - n_o$ colonne sono scelte in modo da essere ortogonali alle prime e linearmente indipendenti tra loro.*

$$x = P_o z$$

Le equazioni di stato e di uscita si trasformano nel seguente modo:

$$\begin{cases} \dot{z}_o = A_o z_o + B_1 u \\ \dot{z}_{n_o} = A_2 z_o + A_{n_o} z_{n_o} + B_2 u \\ y = C_o z_o \end{cases}$$

Lo schema a blocchi che rappresenta il sistema è:



4.2.2 Rivelabilità

Il sistema si dice *rivelabile* se e solo se $\rho(\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}) = n$ per ogni polo instabile.

5 Realizzazione

Consiste nel trovare un sistema dinamico a partire da una f.d.t. assegnata.

Considerando un sistema strettamente proprio (D=0) di ordine n, il grado del numeratore è strettamente minore del grado del denominatore:

$$G(s) = \frac{\beta_{n-1}s^{n-1} + \cdots + \beta_1 s + \beta_o}{s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_o}$$

Le realizzazioni possibili di una f.d.t. sono infinite.

Una realizzazione *minima* è una realizzazione i cui autovalori sono completamente controllabili ed osservabili (il sistema sarà di ordine n). Un esempio di questa realizzazione è ottenibile attraverso la forma compagna:

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & \cdots & -\alpha_{n-1} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} \beta_0 & \beta_1 & \cdots & \beta_{n-1} \end{bmatrix}$$

6 Controllo Ottimo

Dato un generico sistema dinamico, descritto dall'equazione di stato $\dot{x}(t) = f(x(t), u(t), t)$, si vuole che esso assumi un certo comportamento in modo tale da minimizzare un certo obiettivo funzionale:

$$J = S(x(t_f), t_f) + \int_{t_o}^{t_f} f_o(x(t), u(t), t)dt$$

Si definisce la funzione *Hamiltoniana*:

$$H(x, u, \lambda, t) = f_o(x, u, t) + \lambda^T f(x, u, t)$$

Dove $\lambda \in R^n$ è detta variabile di co-stato.

Valgono le seguenti equazioni di stato:

$$\begin{cases} \dot{\lambda} = -(\frac{\partial H}{\partial x})^T \\ \dot{x} = (\frac{\partial H}{\partial \lambda})^T \end{cases}$$

Per minimizzare H rispetto ad u si pone la sua derivata uguale a zero, questa equazione viene detta "di controllo".

Se x_f o t_f sono assegnate possiamo ricavare altre condizioni. Si vuole che:

$$[H + \frac{\partial S}{\partial t}]_{t_f}\delta t_f + [(\frac{\partial S}{\partial x})^T - \lambda]_{t_f}^T \delta x_f = 0$$

Se t_f è assegnato $\delta t_f = 0$; se x_f è assegnato $\delta x_f = 0$. Se invece non sono fissi, deve accadere che:

$$H(t_f) = -\frac{\partial S}{\partial t}|_{t_f} \qquad \lambda(t_f) = (\frac{\partial S}{\partial x})^T|_{t_f}$$

7 Osservatore

Descritto dalle seguenti equazioni:

$$\begin{cases} \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}\hat{u} \\ \hat{y} = \hat{C}\hat{x} \end{cases}$$

Dove $\hat{x}$ è sia lo stato dell'osservatore, che la stima dello stato del sistema da osservare.

Si definisce l'errore sulla stima: $\epsilon(t) = x(t) - \hat{x}(t)$.

Parliamo di *stimatore asintotico* quando: $\lim_{t \to +\infty} |\epsilon(t)|$.

7.1 Osservatore di Luenberger

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K_o(y - \hat{y}) = (A - K_o\hat{C})\hat{x} + Bu + K_o y \\ \hat{y} = \hat{C}\hat{x} \end{cases}$$

Definiamo quindi l'ingresso dell'osservatore $\hat{u} = \begin{bmatrix} u \\ y \end{bmatrix}$. Identifichiamo le matrici:

$$\hat{A} = A - K_o C \qquad \hat{B} = \begin{bmatrix} B & K_o \end{bmatrix}$$

Un osservatore così definito è *asintotico*.

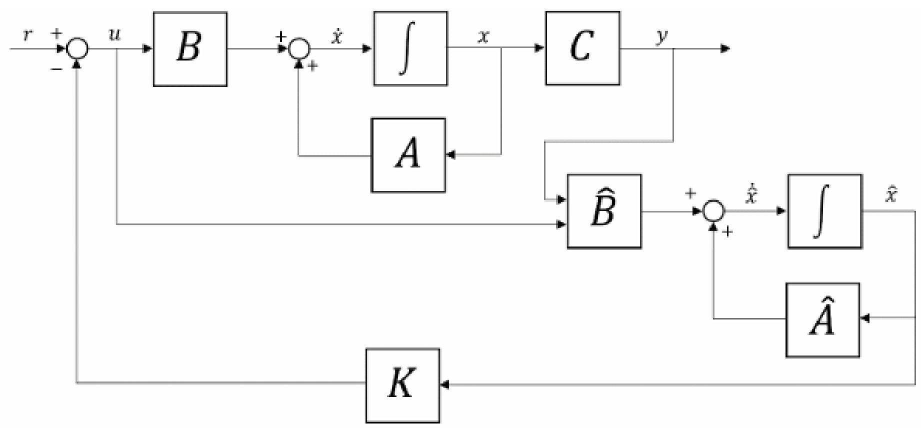
Per determinare K_o dobbiamo effettuare la retroazione di stato sul sistema duale:

$$\begin{cases} \dot{x}_D = A^T x_D + C^T u_D \\ y_D = B^T x_D \end{cases}$$

Quindi $A_D = A^T, B_D = C^T, C_D = B^T$.

La matrice di controllabilità del sistema duale è uguale alla trasposta della matrice di osservabilità del sistema di partenza: $M_{CD} = M_O^T$. Quindi se il sistema originale è completamente osservabile, allora il sistema duale è completamente controllabile e viceversa.

Dopo aver calcolato la retroazione di stato del sistema duale: $K_o = K_D^T$. Lo schema a blocchi:



7.2 Osservatore di Ordine Ridotto

Applicabile se l'uscita è del tipo $y = Cx = \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = C_1 x_1$; con $C_1 \in R^{p \times p}$.

Nel caso in cui sia invertibile, si ottenono le preme p componenti dello stato così: $x_1 = C_1^{-1} y$.

Resta da stimare $\hat{x}_2 = z + Ly$; utilizzando un osservatore di ordine $n - p$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

Lo stimatore avrà uno stato regolato dalla seguente equazione di stato: $\dot{z} = Fz + Gy + Hu$.

Avendo introdotto 4 matrici, abbiamo bisogno di 3 vincoli:

$$\begin{cases} A_{21} + FLC_1 - GC_1 - LC_1 A_{11} = 0 \\ A_{22} - LC_1 A_{12} = 0 \\ B_2 - H - LC_1 B_1 = 0 \end{cases}$$

Possiamo quindi determinare:

$$\begin{cases} F = A_{22} - LC_1 A_{12} \\ G = (A_{21} - LC_1 A_{11})C_1^{-1} + FL \\ H = B_2 - LC_1 B_1 \end{cases}$$

Definiamo il sistema duale al sottosistema 2:

$$\dot{x}_{2D} = A_{22}^T x_{2D} + (C_1 A_{12})^T u$$

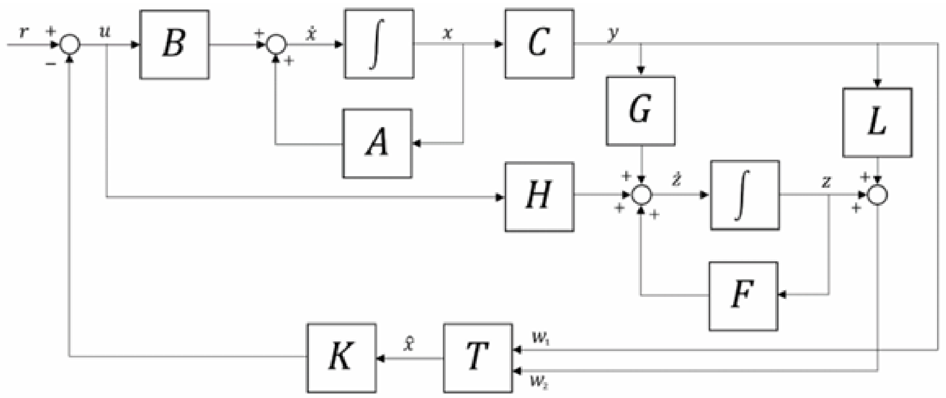
Si effettua la retroazione di stato su questo sistema e si ricava la matrice di retroazione K_D.
Quindi $L = K_D^T$.

Se il sistema non è nella forma descritta possiamo effettuare una trasformazione $T^{-1} = \begin{bmatrix} C \\ R \end{bmatrix}$, dove R è una qualsiasi matrice tale da garantire la non singolarità di T. Effettuando la trasformazione $x = Tw$, si ottiene il sistema:

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} u$$

$$y = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = w_1$$

Lo schema a blocchi che rappresenta il sistema è:



8 Matlab & Simulink

Esempio 1

Considerando il seguente sistema lineare dinamico:

$$\begin{cases} \dot{x}_1(t) = 2x_1(t) + x_2(t) + 2x_3(t) \\ \dot{x}_2(t) = 2x_1(t) + 2x_2(t) + u(t) \\ \dot{x}_3(t) = -6x_3(t) + 2u(t) \\ y(t) = 2x_1(t) + x_2(t) + 2x_3(t) \end{cases}$$

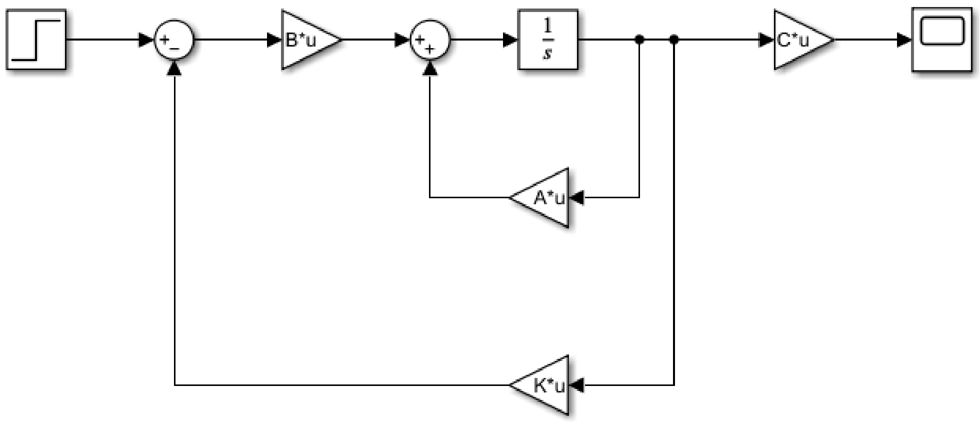
Script Matlab (.m) in cui:

- determinare il sistema in forma di stato;
- analizzare la controllabilità e l'osservabilità (dire se il sistema è completamente controllabile ed osservabile);
- determinare una matrice di retroazione di stato K che imponga gli autovalori [-2, -3, -6];
- determinare il sistema chiuso in retroazione con K in forma di stato e la sua risposta al gradino;
- verificare che il sistema chiuso in retroazione ha gli autovalori richiesti;
- determinare un osservatore di ordine ridotto che imponga gli autovalori [-1±j];
- verificare che il sistema con l'osservatore determinato abbia gli autovalori richiesti.

```
% Sistema lineare dinamico fornito
A = [2, 1, 2; 2, 2, 0; 0, 0, -6];
B = [0; 1; 2];
C = [2, 1, 2];
D = 0;

% Sistema in forma di stato
sys_ol = ss(A,B,C,D);
n = length(A);
% Analizzo la Controllabilita'
Mc = ctrb(A,B);

if (rank(Mc) == n)
    disp("il sistema e' completamente controllabile");
else
    disp("il sistema non e' completamente controllabile");
end
% Analizzo l'Osservabilita'
Mo = obsv(A,C);

if (rank(Mo) == n)
    disp("il sistema e' completamente osservabile");
else
    disp("il sistema non e' completamente osservabile");
end

% Retroazione di stato
p = [-2, -3, -6];
K = place(A,B,p);
% Sistema chiuso in retroazione con K in forma di stato
% e la sua risposta al gradino
A_cl = A-B*K;
sys_cl = ss(A_cl,B,C,D);
step(sys_cl);
% Verifica che il sistema chiuso in retroazione ha gli autovalori
    richiesti
eig_cl =eig(sys_cl);
disp(eig_cl);

% Osservatore di ordine ridotto
po = [-1-1j, -1+1j];

T = [C;0,1,0;0,0,1]; %Trasformazione di stato

A_1 = T*A/T; %con A/T e' equivalente a fare A*inv(T)
B_1 = T*B;
C_1 = C/T;

A_2 = A_1(end-1:end, end-1:end); %Selezione sottomatrice A22
B_2 = (C_1(1)*A_1(1,end-1:end))'; %Calcolo di C1*A12 trasposto
Kd = place(A_2',B_2,po);
L = Kd';

% Verifica che gli autovalori sono quelli richiesti
F = A_2 - L*C_1(1)*A_1(1,2:3);
eig_obs = eig(F);
disp(eig_obs);
```

Simulink:

- schema a blocchi del sistema chiuso in retroazione con K con la possibilità di visualizzare la risposta del sistema ad un gradino di ampiezza 2.



Esempio 2

Considerando il seguente sistema lineare dinamico:

$$\begin{cases} \dot{x}_1(t) = x_1(t) + 4x_2(t) + 4x_3(t) \\ \dot{x}_2(t) = 2x_1(t) + 2x_2(t) \\ \dot{x}_3(t) = -6x_3(t) + u(t) \\ y(t) = 4x_1(t) \end{cases}$$

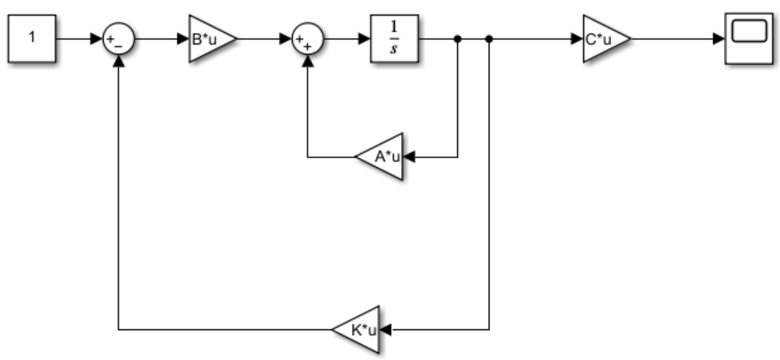
Script Matlab (.m) in cui:

- determinare il sistema in forma di stato;
- determinare la f.d.t. (funzione di trasferimento) del sistema;
- analizzare la controllabilità e l'osservabilità (dire se il sistema è completamente controllabile ed osservabile);
- determinare una matrice di retroazione di stato K che imponga gli autovalori [-1, -2, -3];
- determinare il sistema chiuso in retroazione con K in forma di stato e la sua risposta al gradino;
- verificare che il sistema chiuso in retroazione ha gli autovalori richiesti;
- determinare un osservatore di ordine ridotto che imponga gli autovalori [-5, -30];
- verificare che il sistema con l'osservatore determinato abbia gli autovalori richiesti.

Simulink:

- schema a blocchi del sistema chiuso in retroazione con K, con la possibilità di visualizzare la risposta del sistema ad un riferimento costante di ampiezza 1.

```
% Definizione matrici
A = [1, 4, 4; 2, 2, 0; 0, 0, -6];
B = [0; 0; 1];
C = [4, 0, 0];
D = 0;

% Definizione del sistema nello spazio di stato
sys_ol = ss(A,B,C,D);
[num,den] = ss2tf(A,B,C,D);

% Controllabilita' e Osservabilita'
Mc=ctrb(A,B);
if(rank(Mc)==length(A))
    disp("Il sistema e' completamente controllabile")
else
    disp("Il sistema non e' completamente controllabile")
end

Mo=obsv(A,C);
if(rank(Mo)==length(A))
    disp("Il sistema e' completamente osservabile");
else
    disp("Il sistema non e' completamente osservabile");
end

% Retroazione di stato
p = [-1,-2,-3];
K = place(A,B,p);
sys_cl = ss(A-B*K,B,C,D);
step(sys_cl); %risposta al gradino

eig_cl = eig(A-B*K);
disp(eig_cl);

%Osservatore di ordine ridotto
po = [-5,-30];
kd = place(A(end-1:end,end-1:end)',C(1)*A(1,end-1:end)',po);
L=kd';
eig_L = eig(A(end-1:end,end-1:end)-L*C(1)*A(1,end-1:end));
disp(eig_L);
```



Esempio 3

Considerando il seguente sistema lineare dinamico:

$$\dot{x}(t) = \begin{bmatrix} -100 & 0 & 0 \\ 0 & 0 & 1 \\ -5 & 200 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 500 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x(t)$$

Script Matlab (.m) in cui:

- determinare il sistema in forma di stato;
- verificare la stabilità del sistema;
- verificare la controllabilità e l'osservabilità;
- determinare un osservatore di ordine ridotto ce abbia i seguenti autovalori [-10, -20];
- calcolare le matrici F, G, H;
- verificare che l'osservatore abbia gli autovalori richiesti.

Simulink:

- schema a blocchi del sistema con l'osservatore di ordine minimo, evidenziando le variabili in gioco.

```
% Matrici del sistema
A = [-100, 0, 0; 0, 0, 1; -5, 200, 0];
B = [500; 0; 1];
C = [0, 1, 0];
D = 0;

% Sistema in forma di stato
sys = ss(A,B,C,D);

% stabilita' del sistema
autoalori = eig(sys);
stable = isstable(sys);
if(stable == true)
    disp("il sistema e' stabile")
else
    disp("il sistema e' instabile")
end

% controllabilita' ed osservabilita'
if (rank(ctrb(A,B)) == length(A))
    disp("il sistema e' completamente controllabile")
else
    disp("il sistema non e' completamente controllabile")
end

if (rank(obsv(A,C)) == length(A))
    disp("il sistema e' completamente osservabile")
else
    disp("il sistema non e' completamente osservabile")
end
```

```
% osservatore di ordine ridotto
po = [-10, -20];

Tinv = [C; 1,0,0; 0,0,1];

Abar = Tinv*A/Tinv;
Bbar = Tinv*B;
Cbar = C/Tinv;

A11 = Abar(1,1);
A12 = Abar(1,2:3);
A21 = Abar(2:3,1);
A22 = Abar(2:3,2:3);
B1 = Bbar(1);
B2 = Bbar(2:3);
C1 = Cbar(1);

Ad = A22';
Bd = (C1*A12)';

kd = place(Ad, Bd, po);
L = kd';

F = A22 - L*C1*A12;
G = (A21 - L*C1*A11)/C1 + F*L;
H = B2 - L*C1*B1;

% verifica gli autovalori degli osservatori
autovalori_obs = eig(F);
```

Esempio 4

A partire dal modello lineare costituito dalle seguenti equazioni di stato:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -\frac{mg}{M}x_3(t) + \frac{1}{M}u(t) \\ \dot{x}_3(t) = x_4(t) \\ \dot{x}_4(t) = \frac{(M+m)g}{ML}x_3(t) - \frac{1}{ML}u(t) \end{cases}$$

con $M = 1.7; m = 0.7; L = 0.7; g = 9.81$.

Considerando che le uscite del sistema sono gli stati (x_1, x_3), si realizzi uno script Matlab che consenta di:

- definire i parametri caratteristici del sistema e definire il sistema in forma di stato;
- determinare gli autovalori del sistema;
- verificare controllabilità ed osservabilità;
- applicare una retroazione di stato K che minimizzi il funzionale di costo

$$J = \int_0^{+\infty} (x^T(t)Qx(t) + u^T(t)Ru(t))dt$$

  con R fissato in modo da ridurre l'azione di controllo e Q scelta opportunamente;
- definire il sistema in anello chiuso in retroazione con K;
- visualizzare gli stati e le uscite del sistema in due figure diverse (inserire legenda);

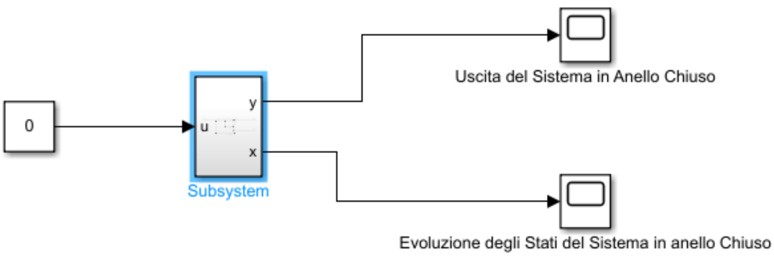
- considerando lo stato iniziale $\bar{x} = [-0.5, -0.3, 0.3, 0.5]^T$ calcolare la risposta libera del sistema.

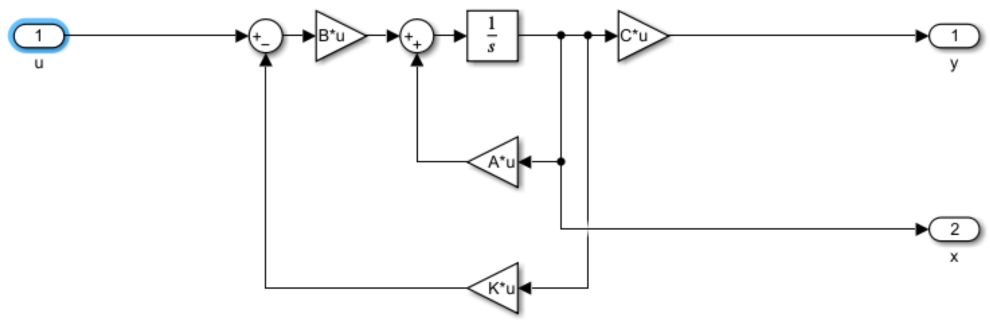
Si analizzi il modello in Simulink (creando un subsystem), verificando l'efficacia del controllore in retroazione di stato progettato ($u(t) = -Ku(t)$).

```
%% Definisco i parametri caratteristici:
M = 1.7;     % Massa carrello [kg]
m = 0.7;     % Massa pendolo [kg]
L = 0.7;     % Lunghezza asta pendolo [m]
g = 9.81;    % Accelerazione di gravita' [m/s^2]

%% Matrici del sistema lineare basate sulle equazioni fornite
% x = [x1; x2; x3; x4] = [posizione carrello; velocita' carrello; angolo pendolo; velocita' angolare pendolo]
A = [0, 1, 0, 0;
     0, 0, -m*g/M, 1/M;
     0, 0, 0, 1;
     0, 0, (M+m)*g/(M*L), -1/(M*L)];

B = [0; 1/M; 0; -1/(M*L)];

C = [1, 0, 0, 0; % Uscita 1: posizione carrello (x1)
     0, 0, 1, 0]; % Uscita 2: angolo pendolo (x3)

D = [0; 0];

% Definizione del sistema in forma di stato
sys = ss(A, B, C, D);

% Determinare gli autovalori del sistema
autovalori_anello_aperto = eig(A);

% Controllabilita' ed Osservabilita'
if(rank(ctrb(A,B)) == length(A))
    disp("il sistema e' completamente controllabile");
else
    disp("il sistema non e' completamente controllabile");
end

if(rank(obsv(A,C)) == length(A))
    disp("il sistema e' completamente osservabile");
else
    disp("il sistema non e' completamente osservabile");
end

%% Retroazione di stato con controllore LQR
% Scelta delle matrici di peso Q e R

% Q penalizza lo stato. Q = C'*C penalizza le uscite (x1 e x3) in modo quadratico.
Q = C'*C; % Penalizza (x1)^2 e (x3)^2 con peso 1.


% R penalizza l'ingresso di controllo (u).
R = 10; % Scelta R=10 per "ridurre l'azione di controllo" rispetto ad R=1. Un valore di R maggiore penalizza l'energia di controllo, tipicamente risultando in un controllo di minore ampiezza.

```

```
% Calcolare K, S e gli autovalori del sistema in anello chiuso usando lqr
[K, S, autovalori_anello_chiuso_LQR] = lqr(A, B, Q, R);

P = autovalori_anello_chiuso_LQR;


%% Definizione sistema in anello chiuso retroazionato con K
sys_cl = ss(A-B*K,B,C,D);

%% Visualizzare gli stati e le uscite del sistema e calcolare la risposta libera
x0 = [-0.5; -0.3; 0.3; 0.5]; % Stato iniziale fornito: [x1; x2; x3; x4]

% Intervallo di tempo per la simulazione della risposta libera
T_sim = 15; % Aumentato il tempo di simulazione per osservare la convergenza
t = 0:0.01:T_sim; % Vettore dei tempi

% Calcolare la risposta libera del sistema in anello chiuso dallo stato iniziale x0
% initial(sys, x0, t) calcola la risposta libera del sistema 'sys' dallo stato iniziale 'x0'
% agli istanti di tempo specificati nel vettore 't'.
[Y, T, X] = initial(sys_cl, x0, t); % Y = C*X
figure(1);
plot(T,X);
legend('x1', 'x2', 'x3', 'x4');
xlabel('Tempo [s]');
ylabel('Stati');
title('Evoluzione degli stati del sistema in anello chiuso');
grid on;

figure(2);
plot(T,Y);
legend('y1=x1', 'y2=x3');
xlabel('Tempo [s]');
ylabel('Uscite');
title('Uscite del sistema in anello chiuso');
grid on;
```

Sia l'analisi teorica (autovalori in anello chiuso) che le simulazioni in Matlab e Simulink dimostrano che il controllore LQR progettato è efficace nello stabilizzare asintoticamente il pendolo inverso linearizzato a partire da condizioni iniziali non nulle, riportando e mantenendo lo stato del sistema al suo punto di equilibrio.





Esempio 5

Si consideri un sistema lineare tempo discreto:

$$\begin{cases} x(n+1) = Ax(n) + B(u(n) + w(n)) \\ y(n) = Cx(n) \end{cases}$$

$$A = \begin{bmatrix} 1.14 & -0.49 & 0.12 \\ 1.1 & 0 & 0 \\ 0 & 1.1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} -0.39 \\ 0.49 \\ 0.53 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

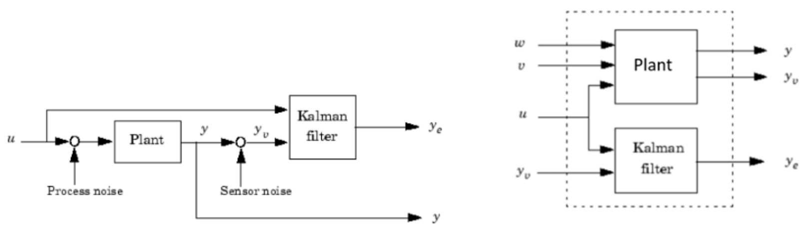
Per tale sistema e dati gli schemi (a) e (b), si realizzi uno script Matlab in grado di:

- definire il sistema, plant in (a), con le sie matrici in forma di stato;
- determinare il filtro di Kalman (a) in forma di stato sapendo che Q = R = 1;
- realizzare il modello in forma di stato rappresentato nel diagramma a blocchi (b);
- simulare il comportamento del filtro;
- visualizzare il confronto tra uscita reale ed uscita filtrata e tra l'errore di misura e l'errore di stima;
- calcolare e confrontare la covarianza dell'errore di misura e stima.

Considerando il sistema lineare tempo continuo descritto dalle matrici:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.02 & 0.04 \\ 0 & -6334 & -50 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1667 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Si ripetano le operazioni per applicare il filtro di Kalman anche a tale sistema.



(a) (b)

Commentare i risultati in entrambi i casi, evidenziando l'eventuale azione positiva del filtro di Kalman.

N.B.: impostare un tempo di campionamento pari a 100s, con passo 1; imporre come ingresso una sinusoide di frequenza pari a 0.5 e fase nulla; determinare il ruore di processo e di misura con la funzione `randn()` moltiplicata per la radice quadrata di Q ed R rispettivamente.

```
%% Definizione matrici
A= [1.14,-0.49,0.12; 1.1,0,0; 0,1.1,0];
B= [-0.39;0.49;0.53];
C= [1,0,0];
D= 0;

% Definizione del sistema nello spazio di stato
sys_ol = ss(A,B,C,D);

% Generazione filtro di Kalman in forma di stato
Plant = ss(A,[B B],C,D,-1,'inputname',{'u','w'},'outputname',{'y'});
Q = 1;
R = 1;
[kalmf,L,P,M] = kalman(Plant,Q,R); % L e' la matrice di guadagno di Kalman, P e' la soluzione di Riccati, M e' il guadagno sull'uscita per il caso con rumore incrociato
kalmf=kalmf(1,:);

a=A; b=[B B 0*B]; c=[C;C]; d=[0 0 0;0 0 1];
plant = ss(a,b,c,d,-1,'inputname',{'u''w''v'},'outputname',{'y''yv'});
sys = parallel(P,kalmf,1,1,[],[]); % collega P e kalmf in parallelo sullo stesso ingresso e prendi in uscita tutte le uscite di entrambi
SimModel = feedback(sys,1,4,2,1); % sistema, guadagno statico, indice dell'input, indice dell'output, segno della retroazione
SimModel = SimModel([1 3],[1 2 3]); % selezione le uscite 1 e 3 e gli ingressi da 1,2 e 3

disp(SimModel.InputName); % w,v e u
disp(SimModel.OutputName); % y e yv

% Simulazione del comportamento del filtro
t=[0:100]';
u=sin(t/5);
n=length(t);
rng default;
w = sqrt(Q)*randn(n,1); % rumore bianco di processo
v = sqrt(R)*randn(n,1); % rumore bianco di misura
[out,T,x] = lsim(SimModel,[w,v,u]);
y = out(:,1); %Risposta vera
ye = out(:,2); %Risposta filtrata
yv = y+v; %Risposta misurata

errore_di_misura = y-yv;
errore_di_stima = y-ye;

figure(1);
subplot(211),plot(t,y,'--',t,ye,'-'),legend('Uscita reale','Uscita filtrata'),xlabel('No. of samples'),ylabel('Output'),title('Kalman filter response');
subplot(212),plot(t,errore_di_misura,'-.',t,errore_di_stima,'-'),
```

```
    legend('Errore di misura','Errore di stima'),xlabel('No. of
    samples'), ylabel('Error');

% Calcolo covarianza errore di stima e covarianza errore di misura
MeasErrCov = sum(errore_di_misura.*errore_di_misura)/length(
    errore_di_misura);

EstErrCov = sum(errore_di_stima.*errore_di_stima)/length(
    errore_di_stima);


%% Applicazione del filtro progettato ad un altro sistema
A2 = [0,1,0;0,-0.02,0.04;0,-6334,-50];
B2 = [0;0;1667];
C2 = [1,0,0];
D2 = 0;

sys_ol_2 = ss(A2,B2,C2,D2);

Plant2 = ss(A2,[B2 B2],C2,D2,-1,'inputname',{'u','w'},'outputname'
    ,{'y'});
Q2 = 1;
R2 = 1;
[kalmf2,L2,P2,M2] = kalman(Plant,Q2,R2); %L e' la matrice di
    guadagno di Kalman, P e' la soluzione di Riccati, M e' il
    guadagno sull'uscita per il caso con rumore incrociato
kalmf2=kalmf2(1,:);

a2=A2; b2=[B2 B2 0*B2]; c2=[C2;C2]; d2=[0 0 0;0 0 1];
plant2 = ss(a2,b2,c2,d2,-1,'inputname',{'u''w''v'},'outputname',{'y
    ''yv'});
sys2 = parallel(P2,kalmf2,1,1,[],[]); % collega P e kalmf in
    parallelo sullo stesso ingresso e prendi in uscita tutte le
    uscite di entrambi
SimModel2 = feedback(sys2,1,4,2,1); % sistema, guadagno statico,
    indice dell'input, indice dell'output, segno della retroazione
SimModel2 = SimModel2([1 3],[1 2 3]); % Selezione le uscite 1 e 3 e
     gli ingressi da 1,2 e 3

% Debug
disp(SimModel2.InputName); % w,v e u
disp(SimModel2.OutputName); % y e yv

% Simulazione del comportamento del filtro
t = [0:100]';
u2 = sin(t/5);
n2 = length(t);
rng default;
w2 = sqrt(Q2)*randn(n2,1); % rumore bianco di processo
v2 = sqrt(R2)*randn(n2,1); % rumore bianco di misura
[out2,T2,x2] = lsim(SimModel2,[w2,v2,u2]);
y2 = out2(:,1); % Risposta vera
ye2 = out2(:,2); % Risposta filtrata
yv2 = y2+v2; %Risposta misurata

errore_di_misura2 = y2-yv2;
errore_di_stima2 = y2-ye2;

```

```
figure(2);
subplot(211),plot(t,y2,'--',t,ye2,'-'),legend('Uscita reale','
    Uscita filtrata'),xlabel('No. of samples'),ylabel('Output'),
    title('Kalman filter response');
subplot(212),plot(t,errore_di_misura2,'-.',t,errore_di_stima2,'-'),
    legend('Errore di misura','Errore di stima'),xlabel('No. of
    samples'), ylabel('Error');

% Calcolo covarianza errore di stima e covarianza errore di misura
MeasErrCov2 = sum(errore_di_misura2.*errore_di_misura2)/length(
    errore_di_misura2);

EstErrCov2 = sum(errore_di_stima2.*errore_di_stima2)/length(
    errore_di_stima2);
```

Esempio 6

Considerando il modello SISO LTI di un motore in c.c. pilotato in armatura, che muove un carico calettato sull'albero motore. Le equazioni che descrivono il sistema con l'ipotesi che la coppia di carico sia trascurabile sono le seguenti:

$$v_a = R_a(t)i_a(t) + L_a\frac{di_a(t)}{dt} + e(t), \qquad e(t) = k_v\frac{d\theta(t)}{dt}, \qquad c_m(t) = J\frac{d^2\theta(t)}{dt} + f\frac{d\theta}{dt},$$

$$c_m(t) = k_m i_a(t), \qquad \omega(t) = \frac{d\theta(t)}{dt}, \qquad v_a(t) = G_a v(t)$$

Tale sistema è descritto in forma di stato dalle seguenti matrici:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -f/J & k_m/J \\ 0 & -k_v/L_a & -R_a/L_a \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ G_a/L_a \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad D = 0$$

con $R_a = 0.06\Omega, L_a = 0.0006H, f = 1Nms, J = 100Nms^2, k_m = 3.5, k_v = 3.5$; parametri noti del sistema. Si sceglie il vettore di stato $x(t) = [\theta(t), \omega(t), i_a(t)]^T$, costituito nell'ordine della posizione del carico, dalla velocità angolare del carico e dalla corrente di armatura. Inoltre si sceglie l'ingresso del sistema uguale alla tensione di alimentazione, $u(t) = v(t)$, e l'uscita pari alla posizione angolare, $y(t) = \theta(t)$. Si suppone infine che il parametro G_a abbia un valore iniziale pari ad 1.

Realizzare lo script Matlab in grado di:

- definire il sistema con i suoi parametri caratteristici e le matrici della forma di stato;
- determinare il polinomio caratteristico in anello aperto;
- determinare guadagno, zeri e poli del sistema;
- visualizzare la risposta all'impulso e al gradino unitario;
- verificare la completa controllabilità del sistema.

Determinare inoltre la matrice di retroazione di stato in due differenti casi (K_1, K_2):

1. il controllore deve imporre che il sistema in anello chiuso abbia i seguenti poli $S_1 = -10, S_{2,3} = \sqrt{3}(-1 \pm j)$;
2. i poli devono essere $S_1 = -20, S_2 = -8, S_3 = -1$.

Bisogna:

- fissare il polinomio caratteristico desiderato in anello chiuso;
- scegliere il guadagno di amplificazione G_a, in modo da imporre a zero l'errore di posizione in anello chiuso;

- definire il sistema in anello chiuso;
- determinare la funzione di trasferimento in anello chiuso, guadagno, zeri e poli.

Visualizzare e commentare la risposta al gradino unitario in anello chiuso confrontando i due casi.

Realizzare il modello del motore cc in Simulink, chiuso in retroazione prima con K_1 e poi con K_2 confrontando e commentando i risultati rispetto a quanto ottenuto in Matlab.

```
%  Definire il sistema con i suoi parametri caratteristici e le
    matrici della forma di stato
Ra = 0.06;       % Resistenza di armatura [Ohm]
La = 0.0006;     % Induttanza di armatura [H]
f = 1;           % Coefficiente di attrito viscoso [Nms]
J = 100;         % Momento di inerzia [Nms^2]
km = 3.5;        % Coefficiente coppia motrice [Nm/A]
kv = 3.5;        % Coefficiente forza controelettromotrice [Vs/rad]
Ga = 1;          % Guadagno dell'amplificatore di alimentazione

% Matrici del sistema in forma di stato
A = [0, 1, 0;
     0, -f/J, km/J;
     0, -kv/La, -Ra/La];

B = [0;
     0;
     Ga/La];

C = [1, 0, 0];

D = 0; % Il sistema e' strettamente proprio

% Creare l'oggetto sistema in spazio di stato
sys = ss(A, B, C, D);

% Determinare il polinomio caratteristico in anello aperto
% Il polinomio caratteristico e' poly(A)
char_poly = poly(A);

% Le radici del polinomio caratteristico sono gli autovalori della
    matrice A
eigenvalues = eig(A); % Determinare gli autovalori in anello aperto


% Determinare guadagno, zeri e poli del sistema
% Usare zpkdata per ottenere zeri, poli e guadagno
[zeros_sys, poles_sys, gain_sys] = zpkdata(sys, 'v'); % 'v'
    restituisce vettori/scalari
disp('Zeri del sistema:');
disp(zeros_sys);
disp('Poli del sistema:');
disp(poles_sys); % Coincidono con gli autovalori della matrice A
disp('Guadagno del sistema:');
disp(gain_sys);

% Visualizzare la risposta all'impulso e al gradino unitario
figure; % Crea una nuova figura
impulse(sys); % Risposta all'impulso
title('Risposta all''impulso del sistema in anello aperto'); %
    Aggiunge un titolo al grafico
xlabel('Tempo'); % Etichetta l'asse x
```

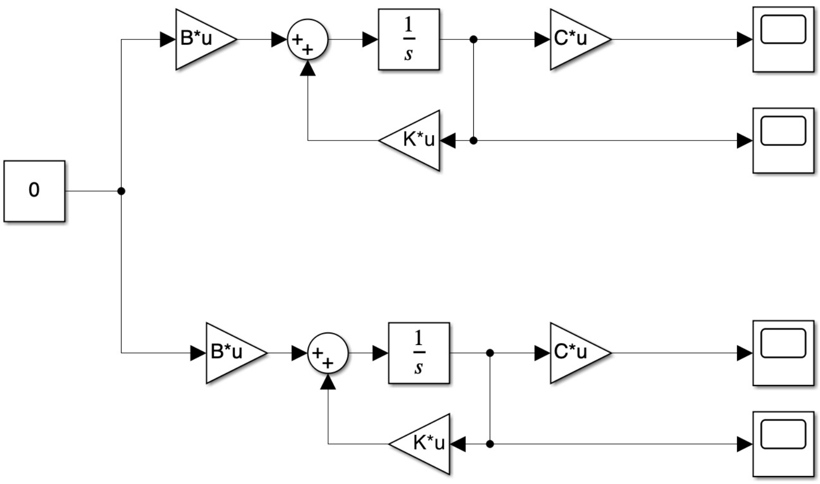
```
ylabel('Uscita (\theta)'); % Etichetta l'asse y
grid on; % Abilita la griglia

figure; % Crea una nuova figura
step(sys); % Risposta al gradino unitario
title('Risposta al gradino unitario del sistema in anello aperto');
xlabel('Tempo');
ylabel('Uscita (\theta)');
grid on;

% Verificare la completa controllabilita' del sistema
Mc = ctrb(A, B);

if (rank(Mc) == length(A))
    disp("Il sistema e' completamente controllabile."); % Il rango massimo indica completa controllabilita'
else
    disp("Il sistema non e' completamente controllabile.");
end

% Definire i poli desiderati per i due casi
% Caso 1: S1 = -10, S2,3 = sqrt(3)(-1 +/- j)
p1 = [-10, sqrt(3)*(-1 + 1j), sqrt(3)*(-1 - 1j)];
K1 = place(A, B, p1);

% Per ottenere errore di posizione nullo per un gradino unitario,
% il riferimento r_sys viene scalato per K1(1) prima di essere applicato
% alla retroazione (r = K1(1) * r_sys).
% Definire il sistema in anello chiuso per il Caso 1
A_cl1 = A - B * K1;
B_cl1 = B * K1(1); % Scalamento del riferimento per errore statico nullo
C_cl1 = C;
D_cl1 = D * K1(1); % D_cl = D * K1(1) ma D=0, quindi D_cl=0

sys_cl1 = ss(A_cl1, B_cl1, C_cl1, D_cl1);

% Determinare FDT, guadagno, zeri e poli del sistema in anello chiuso Caso 1
[zeros_cl1, poles_cl1, gain_cl1] = zpkdata(sys_cl1, 'v');
disp('Zeri del sistema in anello chiuso Caso 1:');
disp(zeros_cl1); % Zeri sono calcolati automaticamente
disp('Poli del sistema in anello chiuso Caso 1 (dovrebbero coincidere con p1):');
disp(poles_cl1); % Verifica che place abbia funzionato correttamente
disp('Guadagno del sistema in anello chiuso Caso 1:');
disp(gain_cl1); % Questo e' il guadagno della ZPK form, non il DC gain
% Verificare il guadagno statico (DC gain) del sistema in anello chiuso Caso 1
dc_gain1 = dcgain(sys_cl1);
disp(['Guadagno statico (DC gain) Caso 1: ', num2str(dc_gain1)]); % Dovrebbe essere 1


% Caso 2: S1 = -20, S2 = -8, S3 = -1
```

```
p2 = [-20, -8, -1];
K2 = place(A, B, p2);

% Definire il sistema in anello chiuso per il Caso 2
A_cl2 = A - B * K2;
B_cl2 = B * K2(1); % Scalamento del riferimento per errore statico nullo
C_cl2 = C;
D_cl2 = D * K2(1); % D_cl = D * K2(1) ma D=0, quindi D_cl=0

sys_cl2 = ss(A_cl2, B_cl2, C_cl2, D_cl2);
disp('Sistema in anello chiuso Caso 2:');
disp(sys_cl2);

% Determinare FDT, guadagno, zeri e poli del sistema in anello chiuso Caso 2
[zeros_cl2, poles_cl2, gain_cl2] = zpkdata(sys_cl2, 'v');
disp('Zeri del sistema in anello chiuso Caso 2:');
disp(zeros_cl2);
disp('Poli del sistema in anello chiuso Caso 2 (dovrebbero coincidere con p2):');
disp(poles_cl2); % Verifica che place abbia funzionato correttamente
disp('Guadagno del sistema in anello chiuso Caso 2:');
disp(gain_cl2); % Questo e' il guadagno della ZPK form, non il DC gain
% Verificare il guadagno statico (DC gain) del sistema in anello chiuso Caso 2
dc_gain2 = dcgain(sys_cl2);
disp(['Guadagno statico (DC gain) Caso 2: ', num2str(dc_gain2)]); % Dovrebbe essere 1

% Visualizzare le risposte al gradino unitario
figure; % Crea una nuova figura
step(sys_cl1, sys_cl2); % Risposte al gradino unitario per entrambi i sistemi
title('Risposta al gradino unitario del sistema in anello chiuso');
xlabel('Tempo [s]');
ylabel('Posizione angolare \theta [rad]');
legend('Caso 1: Poli = [-10, -1.73+/-1.73j]', 'Caso 2: Poli = [-20, -8, -1]');
grid on;
```

Analisi dei poli e delle risposte al gradino: Il Caso 1 ha un polo reale dominante a -10 e una coppia di poli complessi coniugati a -1.73 +/- 1.73j. La parte reale dei poli complessi (-1.73) determina la velocità di convergenza (esponenziale decrescente exp(-1.73t)), mentre la parte immaginaria (1.73) determina la frequenza di oscillazione. Ci si aspetta una risposta con una certa oscillazione e un tempo di assestamento dominato dalla parte reale più vicina all'asse immaginario (-1.73). Il polo a -10 è molto più veloce e quindi meno dominante.

Il Caso 2 ha tutti poli reali e distinti a -20, -8, -1. Il polo più vicino all'asse immaginario e' a -1. Questo polo dominerà la risposta transitoria, determinando un tempo di assestamento più lento rispetto al Caso 1, e una risposta senza oscillazioni (essendo i poli reali). I poli a -20 e -8 sono molto più veloci e il loro effetto transitorio si esaurisce prima.

Confronto atteso: Ci si aspetta che la risposta del Caso 1 presenti una sovraelongazione e un'oscillazione a causa dei poli complessi, con un tempo di assestamento dettato dal polo -1.73. Ci si aspetta che la risposta del Caso 2 non presenti oscillazioni, ma un tempo di assestamento più lungo, dettato dal polo -1. Il guadagno statico unitario per entrambi i casi dovrebbe garantire che la posizione angolare raggiunga il valore di 1 radiante a regime per un ingresso a gradino unitario.



Esempio 7

A partire dal sistema MISO non lineare TI, che modella un sistema idraulico con tre serbatoi comunicanti, se ne consideri la linearizzazione nell'intorno di una soluzione di equilibrio (h_0, q_0, y_0) con $q_0 = [q_{10}, q_{30}]^T$ ingresso di equilibrio (con valori costanti delle portate in ingresso), $h_0 = [h_{10}, h_{20}, h_{30}]^T$ unico stato di equilibrio ottenuto risolvendo l'equazione vettoriale:
$\dot{h} = f(h_1, h_2, h_3, q_{10}, q_{30}) = 0$

$$\begin{cases} h_{10} = h_{20} + \frac{1}{2g}(\frac{q_{10}}{c})^2 \\ h_{20} = h_{30} + \frac{1}{2g}(\frac{q_{10}}{c})^2 \\ h_{30} = H + \frac{1}{2g}(\frac{q_{10}+q_{30}}{c})^2 \\ y_0 = h_{30} \end{cases}$$

Considerando che il sistema linearizzato risulta descritto dalle seguenti matrici di stato:

$$A = \begin{bmatrix} -a & a & 0 \\ a & -2a & a \\ 0 & a & -(a+b) \end{bmatrix} \qquad B = \begin{bmatrix} 1/s & 0 \\ 0 & 0 \\ 0 & 1/s \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

con

$$a = \frac{c^2 g}{s g_{10}} \qquad b = \frac{c^2 g}{s(q_{10} + q_{30})}$$

Si realizzi uno script Matlab che:

- definisca il sistema in forma di stato;
- determini gli autovalori n anello aperto;
- verifichi la completa controllabilità ed osservabilità del sistema;
- realizzi una retroazione costante sullo stato K1 che imponga i seguenti autovalori [-0.03, -0.001+0.001j, -0.001-0.001j];
- determini l'osservatore dello stato con gli autovalori [-0.02, -0.2, -0.4];
- determini la risposta libera del sistema con condizione iniziale non nulla, quando il sistema è chiuso in retroazione con K1 e l'osservatore di stato;
- visualizzi gli stati e gli errori di stima sugli stati.

Si realizzi il modello del sistema linearizzato con la retroazione K1 e l'osservatore in Simulink tale che:

- siano impostate le condizioni iniziali opportune su stati e osservatore degli stati;
- i parametri di simulazione siano configurati opportunamente;
- il sistema idraulico e l'osservatore siano contenuti in due diversi subsystem;
- le variabili d'interesse (uscita, stati, stati stimati, errore di stima) siano memorizzate nel workspace.

Si visualizziono l'uscita, stati ed errori di stima confrontando i risultati con quanto ottenuto in Matlab, commentandoli.

N.B.: per le costanti del sistema si assumono i seguenti valori:
$r = 0.4m, R = 6m, c = \pi r^2, S = \pi R^2, g = 9.81ms^{-2}, H = 4m, q_{10} = 1.2s^{-1}$,
$q_{30} = 0.5m^3s^{-1}$;
$[h_1, h_2, h_3] = [4, 5, 6]; [e_{11}, e_{21}, e_{31}] = [6.2, 9]$; i segnali di riferimento sono nulli.

```
%% Definizione del sistema in forma di stato

% Definizione dei parametri del sistema idraulico
r = 0.4;   % raggio della sezione dei tubi di comunicazione e di uscita [m]
R = 6;     % raggio della sezione dei serbatoi [m]
g = 9.81;  % accelerazione di gravita' [m/s^2]
H = 4;     % altezza del foro di uscita sul terzo serbatoio [m]
q10 = 1.2; % portata di liquido costante in ingresso al primo serbatoio all'equilibrio [m^3/s]
q30 = 0.5; % portata di liquido costante in ingresso al terzo serbatoio all'equilibrio [m^3/s]

% Calcolo delle sezioni dei condotti e dei serbatoi [3, 4]
c = pi * r^2; % sezione circolare dei condotti [m^2]
S = pi * R^2; % sezione circolare dei serbatoi [m^2]
two_g = 2 * g;

% Calcolo delle altezze di equilibrio [h10, h20, h30] basate sulle equazioni di flusso all'equilibrio [5]
% Si assume h10 > h20 > h30 > H, il che implica che le funzioni sgn sono unitarie.
% Le equazioni di equilibrio dei flussi sono (omettendo le dipendenze temporali):
% q1 - q12 = 0  => q10 = q12 = c * sqrt(2*g*(h10-h20))
% q12 - q23 = 0 => q10 = q23 = c * sqrt(2*g*(h20-h30))
% q23 + q3 - qo = 0 => q10 + q30 = qo = c * sqrt(2*g*(h30-H))
% Da queste equazioni si ricavano le differenze di livello all'equilibrio:
delta_h12_eq = (q10/c)^2 / two_g;
delta_h23_eq = (q10/c)^2 / two_g;
delta_h3H_eq = ((q10+q30)/c)^2 / two_g;

% Calcolo delle altezze di equilibrio partendo dall'altezza H:
h30_eq = H + delta_h3H_eq;
h20_eq = h30_eq + delta_h23_eq;
h10_eq = h20_eq + delta_h12_eq;
h_eq = [h10_eq; h20_eq; h30_eq];

disp('Parametri del sistema idraulico:');
fprintf('c = %f m^2\n', c);
fprintf('S = %f m^2\n', S);
```

```
fprintf('H = %f m\n', H);
fprintf('q10 = %f m^3/s\n', q10);
fprintf('q30 = %f m^3/s\n', q30);
disp('Altezze di equilibrio calcolate [h10, h20, h30] (m):');
disp(h_eq');

% Si verifica che le altezze calcolate soddisfino l'ipotesi h10 > h20 > h30 > H
if h10_eq <= h20_eq || h20_eq <= h30_eq || h30_eq <= H
    warning('Le altezze di equilibrio calcolate NON soddisfano l''ipotesi h10 > h20 > h30 > H utilizzata per la linearizzazione con flussi positivi.');
    % Nota: se questa condizione non e' soddisfatta, le radici quadrate nei coefficienti
    % della matrice A diventano immaginarie, indicando un punto di equilibrio non fisico
    % nelle ipotesi di flusso considerate (h1 > h2 > h3 > H).
end

% Calcolo dei coefficienti della matrice A linearizzata
% I coefficienti sono le derivate parziali di f(h,u) = d(delta_h)/dt rispetto agli stati (delta_h)
% valutate nel punto di equilibrio.
% Dal processo di linearizzazione delle equazioni di flusso [6, 7]:
k12 = c * sqrt(two_g) / (2 * S * sqrt(h10_eq - h20_eq)); % Derivata di c*sqrt(2g*(h1-h2))/S wrt h1 o h2 (con segno opposto)
k23 = c * sqrt(two_g) / (2 * S * sqrt(h20_eq - h30_eq)); % Derivata di c*sqrt(2g*(h2-h3))/S wrt h2 o h3 (con segno opposto)
k3H = c * sqrt(two_g) / (2 * S * sqrt(h30_eq - H));     % Derivata di c*sqrt(2g*(h3-H))/S wrt h3

% Definizione delle matrici A, B, C, D del sistema linearizzato in spazio di stato [6]
% Nota: La struttura della matrice A qui utilizzata deriva direttamente dalle derivate
% parziali delle equazioni di flusso linearizzate, come descritto nel sorgente [6],
% sebbene possa presentare piccole differenze rispetto alla matrice A esplicitamente
% riportata nel sorgente [6], possibilmente dovute a variazioni di trascrizione o definizione
% dei parametri 'a' e 'b'.
A = [-k12, k12, 0;
     k12, -(k12 + k23), k23;
     0, k23, -(k23 + k3H)]; % Matrice A derivata dalle derivate parziali

B = [1/S, 0;     % La prima entrata e' delta_q1 / S [6]
     0, 0;     % delta_q1 influisce solo su h1; delta_q3 influisce solo su h3
     0, 1/S]; % La seconda entrata e' delta_q3 / S [6]

C = [0, 0, 1]; % L'uscita e' la variazione del livello del terzo serbatoio delta_h3 [9]
D = [0, 0];    % Matrice di collegamento diretto nulla per sistemi strettamente propri [6, 10]

% Creazione dell'oggetto state-space
```

```
sys = ss(A, B, C, D);
disp('Matrice di stato A:'); disp(A);
disp('Matrice di ingresso B:'); disp(B);
disp('Matrice di uscita C:'); disp(C);
disp('Matrice di collegamento diretto D:'); disp(D);


%%  Determinazione degli autovalori in anello aperto [13, 14]
open_loop_poles = eig(A);
disp('Autovalori del sistema in anello aperto:');
disp(open_loop_poles);


%%  Verifica della completa controllabilita' ed osservabilita' del sistema [11, 13-16]
% Matrice di controllabilita' [14, 16]
Cont = ctrb(A, B);
% Rango della matrice di controllabilita' [14, 16]
rank_Cont = rank(Cont);
n = size(A, 1); % Ordine del sistema (n=3 in questo caso)

disp(['Rango della matrice di controllabilita': ', num2str(rank_Cont)]);
if rank_Cont == n
    disp("Il sistema e' completamente controllabile.");
else
    disp("Il sistema NON e' completamente controllabile.");
end

% Matrice di osservabilita'
Obs = obsv(A, C);
% Rango della matrice di osservabilita'
rank_Obs = rank(Obs);

disp(["Rango della matrice di osservabilita': ", num2str(rank_Obs)]);
if rank_Obs == n
    disp("Il sistema e' completamente osservabile.");
else
    disp("Il sistema NON e' completamente osservabile.");
end

%%  Realizzazione di una retroazione costante sullo stato K1
% Autovalori desiderati per il sistema in anello chiuso con retroazione di stato K1
p_controller = [-0.03, -0.001+0.001j, -0.001-0.001j];
% Utilizzo della funzione 'place' per calcolare la matrice di retroazione K1
% K = place(A, B, autovalori_desiderati)
K1 = place(A, B, p_controller);
disp('Matrice di retroazione K1 (per il controllore):');
disp(K1);

% Verifica degli autovalori del sistema retroazionato (A - B*K1)
A_cl_true = A - B*K1;
cl_poles_true = eig(A_cl_true);
disp('Autovalori del sistema retroazionato con K1 (stato vero) - dovrebbero corrispondere ai desiderati:');
```

```
disp(cl_poles_true); % Dovrebbe corrispondere a p_controller


%% Determinazione dell'osservatore dello stato con guadagno L
% Autovalori desiderati per la dinamica dell'errore di stima (matrice A - L*C)
p_observer = [-0.02, -0.2, -0.4];
% Il guadagno dell'osservatore L per il sistema (A, C) puo' essere calcolato usando il
% principio di dualita'. Il problema di sintesi dell'osservatore per (A, C)
% con autovalori desiderati per (A - LC) e' duale al problema di sintesi di retroazione
% di stato per (A', C') con gli stessi autovalori desiderati per (A' - C'*KD).
% La matrice L e' la trasposta della matrice KD calcolata per il sistema duale.
A_dual = A'; % Matrice di stato del sistema duale
C_dual = C'; % Matrice di ingresso del sistema duale
% Utilizzo della funzione 'place' per calcolare la matrice di retroazione KD per il sistema duale (A_dual, C_dual)
KD = place(A_dual, C_dual, p_observer);
% La matrice di guadagno dell'osservatore L e' la trasposta di KD
L = KD';
disp('Matrice di guadagno dell''osservatore L:');
disp(L);

% Verifica degli autovalori della matrice (A - L*C), che determinano la dinamica dell'errore di stima
A_obs_dynamics = A - L*C;
obs_poles = eig(A_obs_dynamics);
disp('Autovalori della dinamica dell''errore di stima (A - LC) - dovrebbero corrispondere ai desiderati:');
disp(obs_poles); % Dovrebbe corrispondere a p_observer

%% Simulazione della risposta libera del sistema con retroazione dallo stato stimato

% Il sistema complessivo (impianto + osservatore) puo' essere rappresentato da
% un sistema in spazio di stato con 2n variabili di stato.
% Le variabili di stato aumentate sono [x; x_hat], dove x e' lo stato vero
% (variazione rispetto all'equilibrio) e x_hat e' lo stato stimato. n=3, quindi 2n=6.
% L'ingresso di controllo e' u = -K*x_hat (poiche' i segnali di riferimento sono nulli).
% L'uscita misurata e' y = C*x (poiche' il set-point e' 0 per il sistema linearizzato).

% Dinamica dello stato vero: x_dot = A*x + B*u = A*x + B*(-K1*x_hat) = A*x - B*K1*x_hat
% Dinamica dello stato stimato: x_hat_dot = A*x_hat + B*u + L*(y - C*x_hat)
% Sostituendo u e y: x_hat_dot = A*x_hat + B*(-K1*x_hat) + L*(C*x - C*x_hat)
%                          x_hat_dot = (A - B*K1 - L*C)*x_hat + L*C*x

```

```

% Condizioni iniziali
x0_initial = [4; 5; 6]; % Questa e' la condizione iniziale del vettore di stato VERO x(0)
% L'utente ha specificato l'errore di stima iniziale: [e11, e21, e31] = [32-34].
epsilon0_initial = [6; 2; 9]; % Questa e' la condizione iniziale dell'errore epsilon(0) = x(0) - x_hat(0)

% Calcolo della condizione iniziale dello stato stimato: x_hat(0) = x(0) - epsilon(0)
x_hat0_initial = x0_initial - epsilon0_initial;

% Vettore di stato iniziale combinato [x(0); x_hat(0)]
combined_x0 = [x0_initial; x_hat0_initial];

% Simulazione della risposta libera del sistema combinato
% La risposta libera si ottiene considerando un ingresso nullo e una condizione iniziale non nulla.
% Si utilizza un oggetto state-space per il sistema combinato, con matrici B, C, D nulle
% per simulare la risposta libera dell'intero sistema aumentato.
sys_combined = ss(A_combined, zeros(2*n,1), zeros(1,2*n), 0); % Sistema 2n, ingresso nullo per risposta libera
% Nota: in alternativa, si puo' usare ss(A_combined, [], [], []).

% Vettore dei tempi per la simulazione
T_sim = 0:0.1:1000; % Simulazione per 1000 secondi con passo 0.1s

% Esecuzione della simulazione della risposta libera con condizione iniziale [23, 28]
% [Y, T, X] = initial(sistema_ss, condizione_iniziale, vettore_tempi)
[Y_sim, T_sim, X_sim] = initial(sys_combined, combined_x0, T_sim);

% Estrazione degli stati veri e degli stati stimati dai risultati della simulazione (X_sim)
x_true_sim = X_sim(:, 1:n);        % Le prime n colonne sono gli stati veri
x_hat_sim = X_sim(:, (n+1):end); % Le colonne da n+1 a 2n sono gli stati stimati

% Calcolo dell'errore di stima nel tempo: epsilon(t) = x(t) - x_hat(t)
epsilon_sim = x_true_sim - x_hat_sim;


%%  Visualizzazione degli stati e degli errori di stima

% Visualizzazione dell'evoluzione degli stati veri (variazioni rispetto all'equilibrio)
figure;
plot(T_sim, x_true_sim);
title('Evoluzione delle variazioni di livello (Stati Veri)');
xlabel('Tempo [s]');
ylabel('Variazione livello [m]');
legend({'\delta h_1', '\delta h_2', '\delta h_3'});
grid on;
```

```
% Visualizzazione dell'evoluzione dell'errore di stima sugli stati
figure;
plot(T_sim, epsilon_sim);
title('Evoluzione dell''Errore di Stima sugli Stati');
xlabel('Tempo [s]');
ylabel('Errore [m]');
legend({'\epsilon_1', '\epsilon_2', '\epsilon_3'});
grid on;

% Visualizzazione comparata degli stati veri e stimati per ogni componente
figure;
subplot(3,1,1);
plot(T_sim, x_true_sim(:,1), 'b-', T_sim, x_hat_sim(:,1), 'r--');
ylabel('\delta h_1 [m]');
legend('Vero', 'Stimato');
title('Confronto Stati Veri vs Stimati');
grid on;

subplot(3,1,2);
plot(T_sim, x_true_sim(:,2), 'b-', T_sim, x_hat_sim(:,2), 'r--');
ylabel('\delta h_2 [m]');
legend('Vero', 'Stimato');
grid on;

subplot(3,1,3);
plot(T_sim, x_true_sim(:,3), 'b-', T_sim, x_hat_sim(:,3), 'r--');
xlabel('Tempo [s]');
ylabel('\delta h_3 [m]');
legend('Vero', 'Stimato');
grid on;

% Visualizzazione dei valori finali (a regime) per verificare la convergenza all'equilibrio (0)
disp('Valori finali (a regime, t=1000s) delle variazioni di stato [dh1, dh2, dh3]:');
disp(x_true_sim(end, :));
disp('Valori finali (a regime, t=1000s) degli errori di stima [e1, e2, e3]:');
disp(epsilon_sim(end, :));

% Note finali sul principio di separazione [42]
disp(' ');
disp('Nota: Il principio di separazione degli autovalori afferma che gli autovalori');
disp('del sistema complessivo (impianto + osservatore) sono l''unione degli autovalori');
disp('del sistema retroazionato sullo stato vero (A - BK) e degli autovalori della');
disp('dinamica dell''errore dell''osservatore (A - LC).');
disp('Autovalori del sistema combinato (A_combined):');
disp(eig(A_combined));
disp('Gli autovalori sono l''unione di quelli di (A - BK1) e (A - LC), verificando il principio di separazione.');
```

Esempio 8

A partire dal modello non lineare del pendolo inverso costituito dalle seguenti equazioni di stato:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{mLsinx_3(t)x_4^2(t) - mgsinx_3(t)cosx_3(t) + u(t)}{M + msin^2x_3(t)} \\ \dot{x}_3(t) = x_4(t) \\ \dot{x}_4(t) = \frac{-mLsinx_3(t)cosx_3(t)x_4^2(t) + (M+m)gsinx_3(t) - u(t)cosx_3(t)}{(M + msin^2x_3(t))L} \end{cases}$$

dove $x(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^T = [x(t), \dot{x}(t), \theta(t), \dot{\theta}(t)]^T$ è il vettore di stato e $y(t) = [x(t), \theta(t)] = [x_1(t), x_3(t)]^T$ è il vettore delle uscite. La posizione lineare dell'asta del pendolo è $x(t)$, quella angolare è $\theta(t)$; considerandone la linearizzazione intorno allo stato di equilibrio $\bar{x} = [0, 0, 0, 0]^T$ in corrispondenza dell'ingresso di equilibrio $\bar{u} = 0$ descritta dalle seguenti equazioni:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -\frac{mg}{M}x_3(t) + \frac{1}{M}u(t) \\ \dot{x}_3(t) = x_4(t) \\ \dot{x}_4(t) = \frac{(M+m)g}{ML}x_3(t) - \frac{1}{ML}u(t) \end{cases}$$

con $M = 0.0; m = 0.3; L = 1; g = 9.81$.

Realizzare uno script Matlab che consenta di:

- definire i parametri caratteristici del sistema e definire il sistema in forma di stato;
- determinare gli autovalori del sistema;
- verificare controllabilità ed osservabilità;
- applicare una retroazione di stato K che minimizzi il funzionale di costo

$$J = \int_0^{+\infty} (x^T(t)Qx(t) + u^T(t)Ru(t))dt$$

  con R = 1/50 e Q scelta opportunamente;
- definire il sistema in anello chiuso in retroazione con K;
- visualizzare gli stati e le uscite del sistema in due figure diverse (inserire legenda);

Si analizzi il modello in Simulink (creando un subsystem), verificando l'efficacia del controllore in retroazione di stato progettato ($u(t) = -Ku(t)$), a partire dallo stato iniziale del pendolo pari a $\bar{x} = [0.4, 0, 0.4, 0]$.

- cosa succede con le condizioni iniziali $\bar{x} = [0.4, 0, 0.8, 0]$?

```
% Definizione parametri
M = 0.5;
m = 0.3;
L = 1;
g = 9.81;
% Definizione matrici
A = [0,1,0,0;
     0,0,-(m*g)/M,0;
     0,0,0,1;
     0,0,(M+m)*g/(M*L),0];
B = [0;1/M;0;-1/(M*L)];
C = [1,0,0,0;0,0,1,0];
D = 0;

%Definizione del sistema nello spazio di stato
sys_ol = ss(A,B,C,D);
```

```
autovalori_ol = eig(sys_ol);
%Verifica ctrb e obsv
Mc=ctrb(A,B);
if(rank(Mc)==length(A))
    disp("Il sistema e' completamente controllabile");
else
    disp("Il sistema non e' completamente controllabile");
end
Mo=obsv(A,C);
if(rank(Mo)==length(A))
    disp("Il sistema e' completamente osservabile");
else
    disp("Il sistema non e' completamente osservabile");
end

%LQR
R=1/50;
Q=C'*C;
[K,S,p]=lqr(sys_ol,Q,R);

sys_cl=ss(A-B*K,B,C,D);
X_0=[0.4 0 0.4 0];
[Y,T,X]=initial(sys_cl,X_0);

figure(1);
plot(T,X(:,1),T,X(:,2),T,X(:,3),T,X(:,4)),grid on,legend('Stato X1','Stato X2','Stato X3','Stato X4'), title('Stati del sistema in anello chiuso con K1');
figure(2);
plot(T,Y(:,1),T,Y(:,2)),grid on,legend('Uscita 1','Uscita 2'), title('Uscite del sistema in anello chiuso con K1');
X_0_2=[0.4 0 0.8 0];
[Y1,T1,X1]=initial(sys_cl,X_0_2);
figure(3);
plot(T1,X1(:,1),T1,X1(:,2),T1,X1(:,3),T1,X1(:,4)),grid on,legend('Stato X1','Stato X2','Stato X3','Stato X4'), title('Stati del sistema in anello chiuso con K1 con le nuove condizioni iniziali');
figure(4);
plot(T1,Y1(:,1),T1,Y1(:,2)),grid on,legend('Uscita 1','Uscita 2'), title('Uscite del sistema in anello chiuso con K1');
```