



UNIVERSITÀ DI PISA

**Corso di LS in Ingegneria Informatica**

# **PROGETTO DI CONTROLLO DIGITALE**

Giovanni Accongiagioco 302114

Daniele Giannetti 302413

Marco Camilli 300885

**A.A. 2008/2009**

## Sommario

Sezione 1 . Descrizione del problema.....	2
Sezione 2 . Discretizzazione del sistema Accoppiato .....	8
Sezione 3 . Controllo con assegnamento dei poli .....	11
Sezione 4 . Controllo ad orizzonte sfuggente.....	18
Sezione 5 . Controllo Linear Quadratic Regulator (LQR).....	25
Sezione 6 . Disaccoppiamento del sistema .....	34
Sezione 7 . Discretizzazione del sistema disaccoppiato.....	38
Sezione 8 . Controllo algebrico .....	41
Sezione 9 . Controllo proporzionale .....	55
Sezione 10. Eliminazione dei flussi non consentiti .....	59
Sezione 11. Studio della sensibilità al disturbo .....	63
Appendice A. Script Matlab.....	77
Appendice B. Modelli Simulink .....	83
Appendice C. Guida alla GUI .....	87

## SEZIONE 1: DESCRIZIONE DEL PROBLEMA

Si consideri il doppio serbatoio illustrato in Figura 1 nel quale  $f_1$  e  $f_4$  (flusso di ricircolo) indicano dei flussi in ingresso manipolabili (variabili di controllo), mentre il flusso  $f_3$  viene considerato un disturbo.

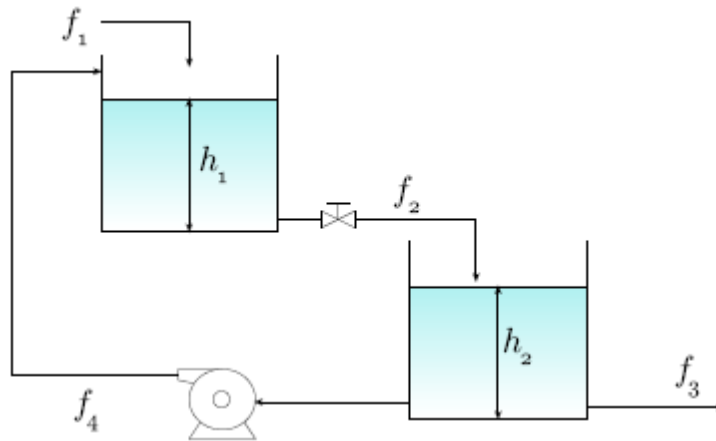


Figura 1: Sistema Doppio Serbatoio

L'obiettivo del sistema di controllo consiste nel mantenere i livelli  $h_1$  e  $h_2$  vicini a dei livelli forniti come riferimento a fronte del disturbo  $f_3$  e ipotizzando di avere delle misure dei livelli  $h_1$  e  $h_2$ .

Nelle ipotesi di fluido incomprimibile e di serbatoi a pressione atmosferica, il flusso  $f_2$  dipende unicamente dal livello del liquido nel primo serbatoio. Si ipotizza di trascurare la dinamica della pompa di ricircolo; indicando con  $A_1$  e  $A_2$  le aree delle basi dei due serbatoi cilindrici e con  $\rho$  la densità del liquido, il bilanciamento delle masse da luogo alle equazioni seguenti, che sono state ricavate uguagliando l'insieme dei flussi entranti e uscenti in ogni serbatoio con la derivata temporale del volume dei liquidi nei serbatoi.

$$A_1 \dot{h}_1(t) = f_1(t) + f_4(t) - f_2(t) \quad (1)$$

$$A_2 \dot{h}_2(t) = f_2(t) - f_4(t) - f_3(t) \quad (2)$$

Consideriamo adesso  $f_2$ .

Si suppone che la valvola sia tale che il flusso  $f_2$  è dato dalla seguente relazione:

$$f_2(t) = k\sqrt{h_1(t)}$$

con  $k$  parametro della valvola.

Ossia il flusso attraverso la valvola dipende esclusivamente dalla altezza del liquido nel primo serbatoio; questa dipendenza è non lineare, si decide allora di linearizzare attorno ad un ipotetico punto di equilibrio dato in simboli da:

$$\begin{cases} h_1(t) = h_{1,0} \\ h_2(t) = h_{2,0} \\ f_1(t) = f_{1,0} \\ f_2(t) = f_{2,0} \\ f_3(t) = f_{3,0} \\ f_4(t) = f_{4,0} \end{cases}$$

Definiamo inoltre le seguenti deviazioni:

$$\begin{cases} \tilde{h}_1(t) = h_1(t) - h_{1,0} \\ \tilde{h}_2(t) = h_2(t) - h_{2,0} \\ \tilde{f}_1(t) = f_1(t) - f_{1,0} \\ \tilde{f}_2(t) = f_2(t) - f_{2,0} \\ \tilde{f}_3(t) = f_3(t) - f_{3,0} \\ \tilde{f}_4(t) = f_4(t) - f_{4,0} \end{cases}$$

Sviluppando in serie di Taylor la relazione imposta dalla valvola possiamo trovare la sua linearizzazione attorno al punto di equilibrio, sapendo infatti che

$$\sqrt{x} = \sqrt{x_0} + \frac{1}{2\sqrt{x_0}}(x - x_0) + \dots$$

Si può allora linearizzare  $\sqrt{h_1(t)}$  come segue

$$\sqrt{h_1(t)} \cong \sqrt{h_{1,0}} + \frac{1}{2\sqrt{h_{1,0}}}\tilde{h}_1(t)$$

E quindi

$$\begin{aligned} f_2(t) &= k \left[ \sqrt{h_{1,0}} + \frac{1}{2\sqrt{h_{1,0}}}\tilde{h}_1(t) \right] = f_{2,0} + \frac{k}{2\sqrt{h_{1,0}}}\tilde{h}_1(t) \\ \tilde{f}_2(t) &= \frac{k}{2\sqrt{h_{1,0}}}\tilde{h}_1(t) = \frac{f_{2,0}}{2h_{1,0}}\tilde{h}_1(t) \end{aligned}$$

Dove l'ultima uguaglianza si è ottenuta moltiplicando sopra e sotto per  $\sqrt{h_{1,0}}$ .

Riprendendo adesso la (1) e la (2), si può osservare che  $\dot{h}_1(t) = \dot{\tilde{h}}_1(t)$  e dunque si può scrivere

$$\begin{aligned} A_1 \dot{\tilde{h}}_1(t) &= \tilde{f}_1(t) + \tilde{f}_4(t) - \tilde{f}_2(t) + f_{1,0} + f_{4,0} - f_{2,0} = \\ &= \tilde{f}_1(t) + \tilde{f}_4(t) - \frac{f_{2,0}}{2h_{1,0}}\tilde{h}_1(t) + f_{1,0} + f_{4,0} - f_{2,0} = \tilde{f}_1(t) + \tilde{f}_4(t) - \frac{f_{2,0}}{2h_{1,0}}\tilde{h}_1(t) \end{aligned}$$

Dove l'ultimo passaggio deriva dal fatto che siccome abbiamo supposto di essere in condizione di equilibrio allora necessariamente  $f_{1,0} + f_{4,0} - f_{2,0} = 0$ , altrimenti il livello del liquido nel primo serbatoio cambierebbe.

Riapplicando quanto detto sopra anche alla (2) si ottiene il seguente sistema di equazioni differenziali

$$\begin{cases} \dot{\tilde{h}}_1(t) = -\frac{f_{2,0}}{2A_1 h_{1,0}}\tilde{h}_1(t) + \frac{\tilde{f}_1(t)}{A_1} + \frac{\tilde{f}_4(t)}{A_1} \\ \dot{\tilde{h}}_2(t) = \frac{f_{2,0}}{2A_2 h_{1,0}}\tilde{h}_1(t) - \frac{\tilde{f}_4(t)}{A_2} - \frac{\tilde{f}_3(t)}{A_2} \end{cases}$$

$\tilde{f}_3(t)$  rappresenta appunto il disturbo, il valore di  $f_{3,0}$  è fondamentale per trovare l'equilibrio, ma uno scostamento di  $f_3(t)$  da  $f_{3,0}$  è un qualcosa di inaspettato.

In seguito verrà condotta una analisi della sensibilità dei sistemi controllati ottenuti rispetto a questo disturbo, tuttavia per la ricerca dei controllori si ipotizza in prima istanza che  $\tilde{f}_3(t) = 0$ .

Detto questo allora possiamo riscrivere il sistema in forma matriciale

$$\begin{pmatrix} \dot{\tilde{h}}_1(t) \\ \dot{\tilde{h}}_2(t) \end{pmatrix} = \begin{pmatrix} -\frac{f_{2,0}}{2A_1h_{1,0}} & 0 \\ \frac{f_{2,0}}{2A_2h_{1,0}} & 0 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_2(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{A_1} & \frac{1}{A_1} \\ 0 & -\frac{1}{A_2} \end{pmatrix} \begin{pmatrix} \tilde{f}_1(t) \\ \tilde{f}_4(t) \end{pmatrix}$$

Questa equazione matriciale descrive la dinamica del sistema tempo-continuo MIMO in esame, dove le  $\tilde{h}_1(t)$  e  $\tilde{h}_2(t)$  sono le nostre variabili di stato, mentre gli ingressi manipolabili del sistema sono  $\tilde{f}_1(t), \tilde{f}_4(t)$ .

Dal momento che le altezze sono misurabili, il sistema è completamente accessibile nelle sue variabili di stato. Si scelgono allora come uscite  $y_1(t)$  e  $y_2(t)$  direttamente le variabili di stato del sistema, da cui si ha che:

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_2(t) \end{pmatrix}$$

Notiamo che i punti di equilibrio in questo sistema sono molteplici (infiniti), per semplicità si sceglie come punto di equilibrio (punto vicino al quale il sistema dovrà avere il suo punto di lavoro), pari a:

$$\begin{cases} h_{1,0} = 1 \\ h_{2,0} = 1 \\ f_{1,0} = 1 \\ f_{2,0} = 2f_{1,0} = 2 \\ f_{3,0} = f_{1,0} = 1 \\ f_{4,0} = f_{1,0} = 1 \end{cases}$$

dove infatti sono rispettate le condizioni di equilibrio, date da:

$$\begin{aligned} f_{1,0} + f_{4,0} - f_{2,0} &= 0 \\ f_{2,0} - f_{4,0} - f_{3,0} &= 0 \end{aligned}$$

Notiamo inoltre che  $h_{1,0}$  è stato scelto per semplificare la forma numerica delle matrici sopra, mentre  $h_{2,0}$  è stato scelto in modo del tutto arbitrario.

Supponiamo inoltre che  $A_1 = A_2 = 1$  sempre per semplificare i conti.

Il sistema scritto in forma matriciale diventa allora, con le ipotesi dette:

$$\begin{cases} \begin{pmatrix} \dot{\tilde{h}}_1(t) \\ \dot{\tilde{h}}_2(t) \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_2(t) \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \tilde{f}_1(t) \\ \tilde{f}_4(t) \end{pmatrix} \\ \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_2(t) \end{pmatrix} \end{cases}$$

Prima di passare allo studio di alcune soluzioni di controllo, possiamo ricavare la matrice delle funzioni di trasferimento del sistema come segue:

$$\begin{aligned} G(s) &= C(sI - A)^{-1}B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \left[ s \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} s+1 & 0 \\ -1 & s \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \\ &= \frac{1}{s(s+1)} \begin{pmatrix} s & 0 \\ 1 & s+1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} = \frac{1}{s(s+1)} \begin{pmatrix} s & s \\ 1 & -s \end{pmatrix} = \begin{pmatrix} \frac{1}{s+1} & \frac{1}{s+1} \\ \frac{1}{s(s+1)} & -\frac{1}{s+1} \end{pmatrix} \\ &= \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix} \end{aligned}$$

e quindi si può scrivere (trasformando secondo Laplace)

$$\begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} = \begin{pmatrix} H_1(s) \\ H_2(s) \end{pmatrix} = \begin{pmatrix} \frac{1}{s+1} & \frac{1}{s+1} \\ \frac{1}{s(s+1)} & -\frac{1}{s+1} \end{pmatrix} \begin{pmatrix} F_1(s) \\ F_4(s) \end{pmatrix}$$

Notiamo subito che ogni variabile manipolabile  $F_1(s)$  e  $F_4(s)$  influenza entrambe le variabili controllate  $H_1(s)$  e  $H_2(s)$ ; di conseguenza la matrice delle funzioni di trasferimento  $G(s)$  del processo è piena.

Questo implica che se si hanno due variabili controllate (le altezze  $h_1(t)$  e  $h_2(t)$ ) e due ingressi di controllo (i flussi  $f_1(t)$  e  $f_4(t)$ ), usando due anelli di controllo separati, non è immediato decidere quale ingresso di controllo usare per controllare  $h_1(t)$  e quale per  $h_2(t)$ . Questo problema viene comunemente chiamato input/output pairing

Cerchiamo allora di capire cosa succede tentando di controllare usando due anelli separati per i due ingressi, con lo schema seguente:

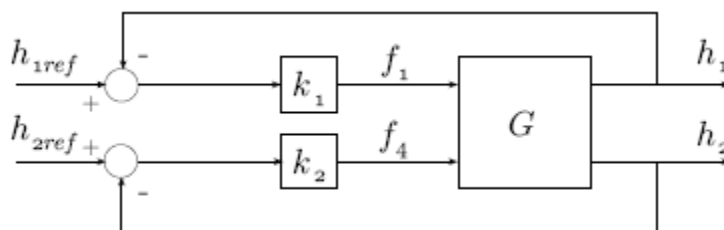


Figura 2: Controllo mediante anelli separati

Qualunque sia il valore scelto per  $k_1$  e  $k_2$  possiamo ricavarci simbolicamente la fdt tra  $H_1(s)$  e  $F_1(s)$ , come segue (considerando ovviamente il riferimento nullo), infatti dalla figura 2 emerge la relazione seguente:

$$F_4(s) = -k_2 H_2(s)$$

Da cui, considerando la matrice delle funzioni di trasferimento:

$$H_2(s) = G_{21}(s)F_1(s) + G_{22}(s)F_4(s) = G_{21}(s)F_1(s) - k_2 G_{22}(s)H_2(s)$$

e quindi:

$$H_2(s) = \frac{G_{21}(s)}{1 + k_2 G_{22}(s)} F_1(s)$$

Quindi possiamo trovare la funzione di trasferimento di cui sopra osservando che:

$$\begin{aligned} H_1(s) &= G_{11}(s)F_1(s) + G_{12}(s)F_4(s) = G_{11}(s)F_1(s) - k_2 G_{12}(s)H_2(s) \\ &= G_{11}(s)F_1(s) - k_2 G_{12}(s) \frac{G_{21}(s)}{1 + k_2 G_{22}(s)} F_1(s) \end{aligned}$$

Sostituiamo allora i componenti della  $G(s)$  con le loro espressioni in  $s$

$$H_1(s) = \frac{1}{s+1} F_1(s) - k_2 \frac{1}{s+1} \frac{\frac{1}{s(s+1)}}{1 - k_2 \frac{1}{s+1}} F_1(s) = \frac{1}{s+1} \left[ 1 - k_2 \frac{1}{s(s+1 - k_2)} \right] F_1(s)$$

Si può allora notare che questa funzione di trasferimento è influenzata molto dal valore di  $k_2$ . Questo fenomeno per il quale il guadagno d'anello di un anello dipende dal guadagno d'anello di un altro anello viene definito **interazione**. La struttura del sistema di controllo a più anelli qui utilizzata non prende in considerazione questa interazione.

Un controllo mediante anelli separati come quello visto è un esempio di controllo decentralizzato, a differenza di uno schema centralizzato nel quale tutte le misure sono a disposizione di tutti i canali di controreazione.

In Figura 3 viene illustrato l'effetto della chiusura di un generico singolo anello di controreazione  $m_2(s) = K(s)y_2(s)$ .

Ad anello chiuso anche l'uscita  $y_1(s)$  risente della controreazione da  $y_2(s)$ .

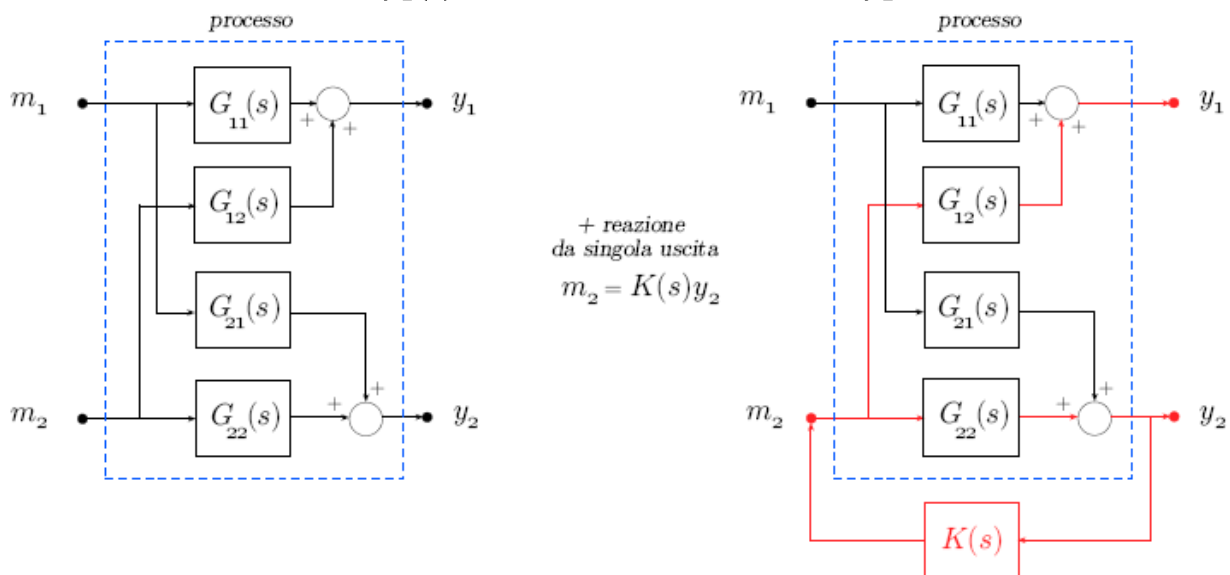


Figura 3: Interazione nel sistema MIMO

A questo punto si capisce come per controllare il sistema vi siano due strade separate:

- Controllare il sistema affetto da interazione (controllo accoppiato)
- Disaccoppiare il sistema, tentando di compensare tale interazione inserendo, in serie con il processo, un compensatore  $K^{pre}(s)$  come illustrato in Figura 4. Se si riesce ad individuare  $K^{pre}(s)$  in modo tale che  $G(s) K^{pre}(s)$  sia diagonale, si raggiunge l'obiettivo desiderato.

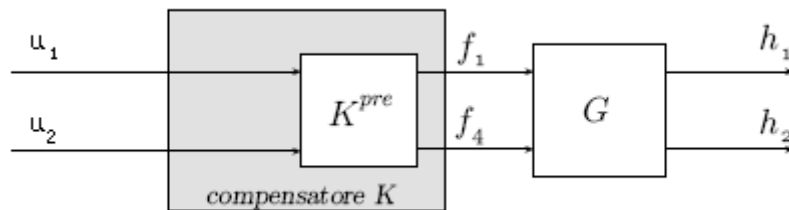


Figura 4: Disaccoppiamento mediante compensatore

In questa trattazione esaminiamo alcune tecniche di controllo digitale per il sistema accoppiato (Sezioni 2,3, 4 e 5), a seguire si mostrerà come sia possibile disaccoppiare il sistema (Sezione 6) e controllarlo digitalmente con alcune tecniche utilizzabili dopo che il sistema è stato disaccoppiato (Sezioni 7, 8 e 9).

Nella Sezione 10 si mostrerà come l'introduzione di un componente non lineare risolva alcuni problemi che si possono verificare nella particolare applicazione del sistema.

Nella Sezione 11 si propone uno studio dell'impatto che ha il disturbo sulla risposta del sistema.

In Appendice si riportano gli script Matlab e i modelli Simulink utilizzati nel corso della trattazione.

## SEZIONE 2: DISCRETIZZAZIONE DEL SISTEMA ACCOPPIATO

---

Lo stato del sistema è espresso mediante le variabili di stato  $h_1(t)$  e  $h_2(t)$ , mentre si considerano ingressi manipolabili i flussi  $f_1(t)$ , in ingresso al primo serbatoio, ed  $f_4(t)$ , di ricircolo.

In uscita si osservano entrambe le variabili di stato. Si noti che le quantità elencate sono in realtà variazioni rispetto allo stato di equilibrio.

Con queste premesse il sistema continuo da controllare risulta esprimibile come:

$$\begin{cases} \dot{h}(t) = A_c h(t) + B_c f(t) \\ y(t) = C h(t) + D f(t) \end{cases}$$

Con le seguenti matrici:

$$A_c = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}; B_c = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}; C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Ottenibili in Matlab mediante i comandi:

```
>> Ac = [-1 0; 1 0]
>> Bc = [1 1; 0 -1]
>> Cc = [1 0; 0 1]
>> Dc = [0 0; 0 0]
```

Si determina a questo punto un modello discreto per il sistema in questione.

Perché un controllore digitale possa controllare il sistema in modo da soddisfare il requisito è fondamentale che il periodo di campionamento sia scelto opportunamente, ossia sia sufficientemente piccolo da andare oltre (possibilmente molto oltre) la banda risultante (espressa in Hz) del sistema in ciclo chiuso. In particolare una buona scelta per il periodo di campionamento è quella che soddisfa:

$$\frac{1}{T} \cong 60 * B_{3db_{cl}}$$

con  $T$  periodo di campionamento, e  $B_{3db_{cl}}$  la banda del sistema a 3dB in ciclo chiuso.

Ricordando allora che per un sistema ad un solo polo sussiste una relazione tra banda e tempo di salita del tipo:

$$B_{3db_{cl}} * T_s = 0.35 \cong \frac{1}{3}$$

con  $T_s$  tempo di salita desiderato, ovvero tempo impiegato dal sistema per andare dal 10% al 90% del valore finale di regime. Allora si arriva a dire che:

$$\begin{aligned} \frac{1}{60 * T} * T_s &\cong \frac{1}{3} \\ T &\cong \frac{3}{60} T_s \cong \frac{1}{20} T_s \end{aligned}$$

Notiamo che abbiamo utilizzato una relazione che è vera solo per sistemi ad un solo polo, ma che comunque costituisce una buona approssimazione per sistemi di ordine superiore.

Una scelta che appare ragionevole è allora quella di scegliere il periodo di campionamento in modo che vi siano circa 20 campioni nel tempo di salita.

Volendo che un sistema in ciclo chiuso risponda con una certa velocità, bisogna imporre che il tempo di salita sia dell'ordine di un certo valore. Nel nostro caso il sistema è meccanico, per cui il tempo di salita non può essere eccessivamente ridotto. In particolare imponiamo che quest'ultimo circa sia pari a 5 secondi.

Da questo si ricava subito che:

$$T \cong \frac{1}{20} T_s = 0.25s$$

A questo punto è possibile discretizzare il sistema applicando le seguenti trasformazioni esatte:

$$A_d = e^{A_c T}; B_d = \int_0^T e^{A_c \tau} d\tau B$$

Indicando con *sample\_time* il periodo di campionamento e dando i seguenti comandi Matlab:

```
>> sample_time = 0.25
>> fcn = @(tau) expm(Ac*tau)
>> Ad = fcn(sample_time)
>> Bd = quadv(fcn,0,sample_time)*Bc
>> Cd=Cc; Dd=Dc
```

Si ottiene il seguente sistema discreto:

$$\begin{cases} h(k+1) = A_d h(k) + B_d f(k) \\ y(k) = C h(k) + D f(k) \end{cases}$$

con le seguenti matrici:

$$A_d = \begin{pmatrix} 0.7788 & 0 \\ 0.2212 & 1 \end{pmatrix}; B_d = \begin{pmatrix} 0.2212 & 0.2212 \\ 0.0288 & -0.2212 \end{pmatrix}; C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

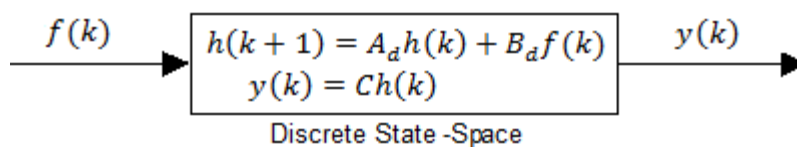


Figura 5: Sistema in variabili di stato

Il sistema in questione risulta instabile nella risposta ad un segnale gradino unitario, difatti la matrice della dinamica ha un autovalore sul bordo della circonferenza di raggio unitario, come verificabile mediante Matlab:

```
>> eig(Ad)
```

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 0.7788 \end{aligned}$$

La risposta può essere verificata mediante simulazione su Simulink.

In uscita, vista la scelta fatta per  $C$  e  $D$ , si ha che  $y(k) = h(k)$ .

Ad esempio imponendo  $f(k) = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ , si ottiene l'andamento riportato in figura 6.

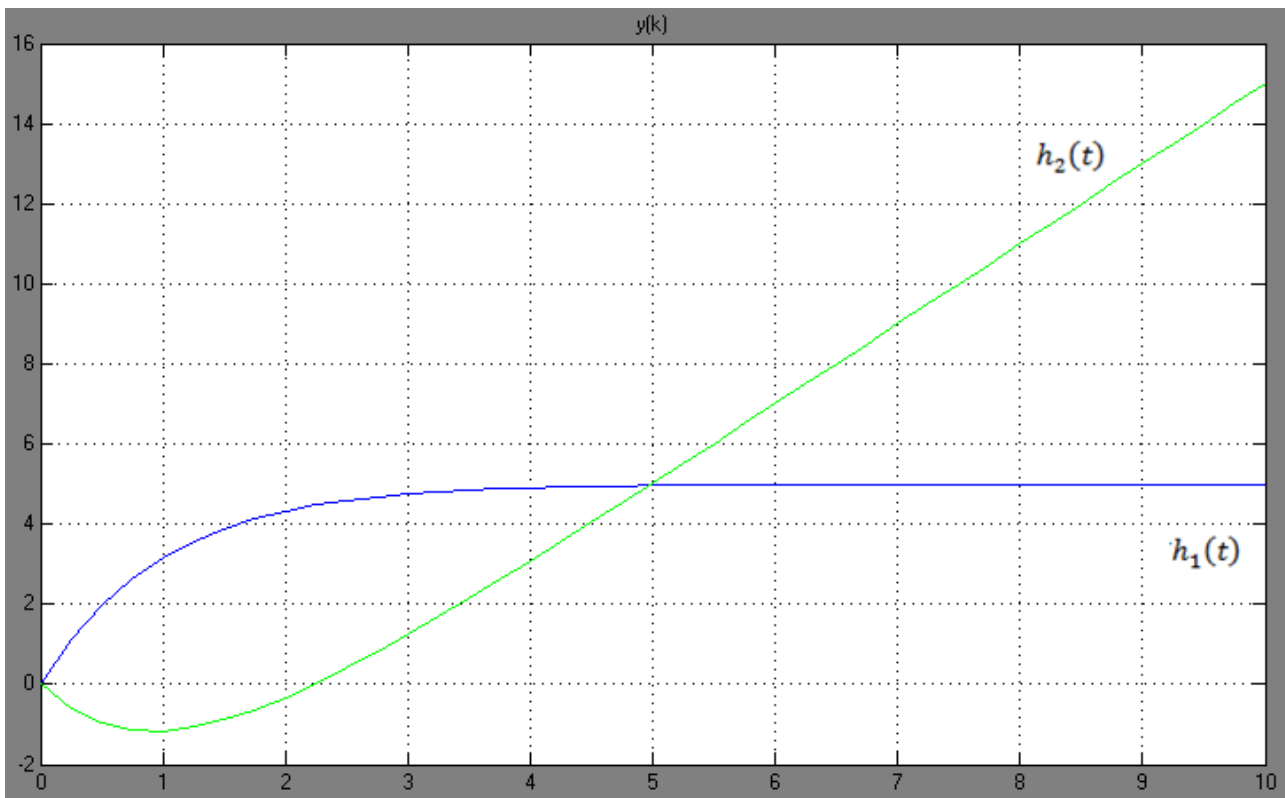
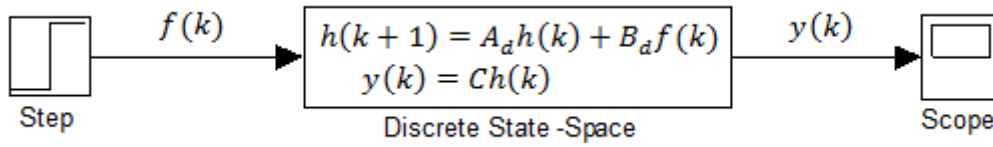


Figura 6: Schema a blocchi e andamento dei livelli

Da cui è possibile verificare che le uscite (ovvero le altezze dei due serbatoi) non raggiungono mai una condizione di stabilità (almeno per quanto riguarda una delle due).

### SEZIONE 3: CONTROLLO CON ASSEGNAMENTO DEI POLI

Si procede adesso al controllo del sistema accoppiato mediante la tecnica di retroazione dello stato.

Tale tecnica può essere applicata sotto le seguenti ipotesi:

Il sistema è completamente osservabile (eventualmente deve essere costruito un osservatore che permetta di ricostruire, a partire dalle uscite, lo stato del sistema)

Il sistema è completamente controllabile.

Nel caso in esame il punto 1 è automaticamente soddisfatto avendo imposto che le uscite coincidono con lo stato del sistema. Viceversa si verifica il punto 2 calcolando il rango della matrice di controllabilità. Il comando:

```
>> rank(ctrb(Ad, Bd))
```

fornisce come valore di uscita 2, per cui il sistema soddisfa anche la completa controllabilità, visto che la coppia  $A_d, B_d$  ha rango massimo.

Si sceglie di piazzare i poli del sistema in ciclo chiuso ricordando i seguenti criteri di progetto:

- Più i poli tendono a zero, maggiore sarà la velocità con cui il sistema raggiunge la stabilità, ma maggiore sarà la spesa sostenuta, inoltre un sistema troppo veloce non rispetta i requisiti per il tempo di salita che abbiamo imposto al momento della scelta del periodo di campionamento.
- Più i poli si avvicinano alla circonferenza unitaria, minore sarà la velocità per il raggiungimento del valore di regime (e la spesa).

Dal momento che il sistema fisico ha certe limitazioni, dovute principalmente ai tempi impiegati da pompe, rubinetti e valvole per gestire i flussi, e dal fatto che questi ultimi non possono crescere in modulo a dismisura, si usa un valore conservativo di 0.9 per il piazzamento dei poli.

Successivamente andrà verificata la coerenza del valore scelto con il tempo di salita di 5 secondi imposto durante la scelta dell'intervallo di campionamento.

Una volta reazionato il sistema, la nuova rappresentazione in variabili di stato risulterà:

$$\begin{cases} h(k+1) = (A_d - B_d K)h(k) + B_d r(k) \\ y(k) = Ch(k) \end{cases}$$

dove  $r(k)$  rappresenta l'ingresso del sistema in ciclo chiuso, ovvero teoricamente i valori di altezza dei serbatoi che si intende seguire, la situazione è mostrata in figura 7.

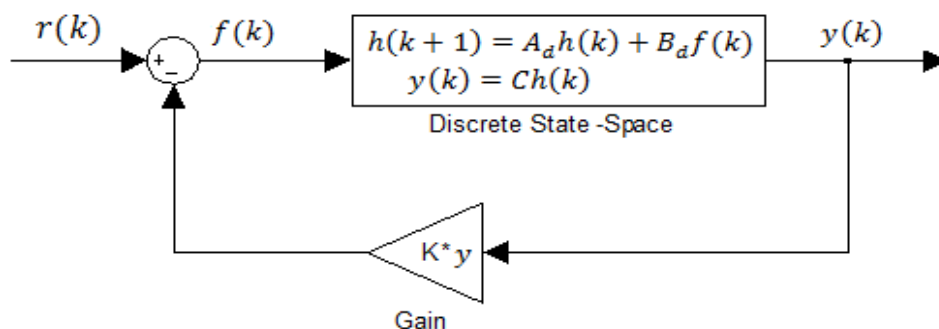


Figura 7: Schema a blocchi del sistema controllato

Il piazzamento dei poli in ciclo chiuso può essere effettuato mediante il seguente comando:

```
>> P=[0.9 0.9]
>> K=place(Ad,Bd,P)
```

che fornisce la seguente matrice di retroazione:

$$K = \begin{pmatrix} 0.4 & 0.4 \\ -0.9479 & -0.4 \end{pmatrix}$$

Mediante la quale, in ciclo chiuso  $A_{cc} = A_d - B_d K$ , è una matrice i cui autovalori si trovano esattamente in:

$$\lambda_1 = 0.9$$

$$\lambda_2 = 0.9$$

Il sistema a questo punto è stato stabilizzato, per cui è possibile verificare la risposta al gradino mediante la solita simulazione su Simulink:

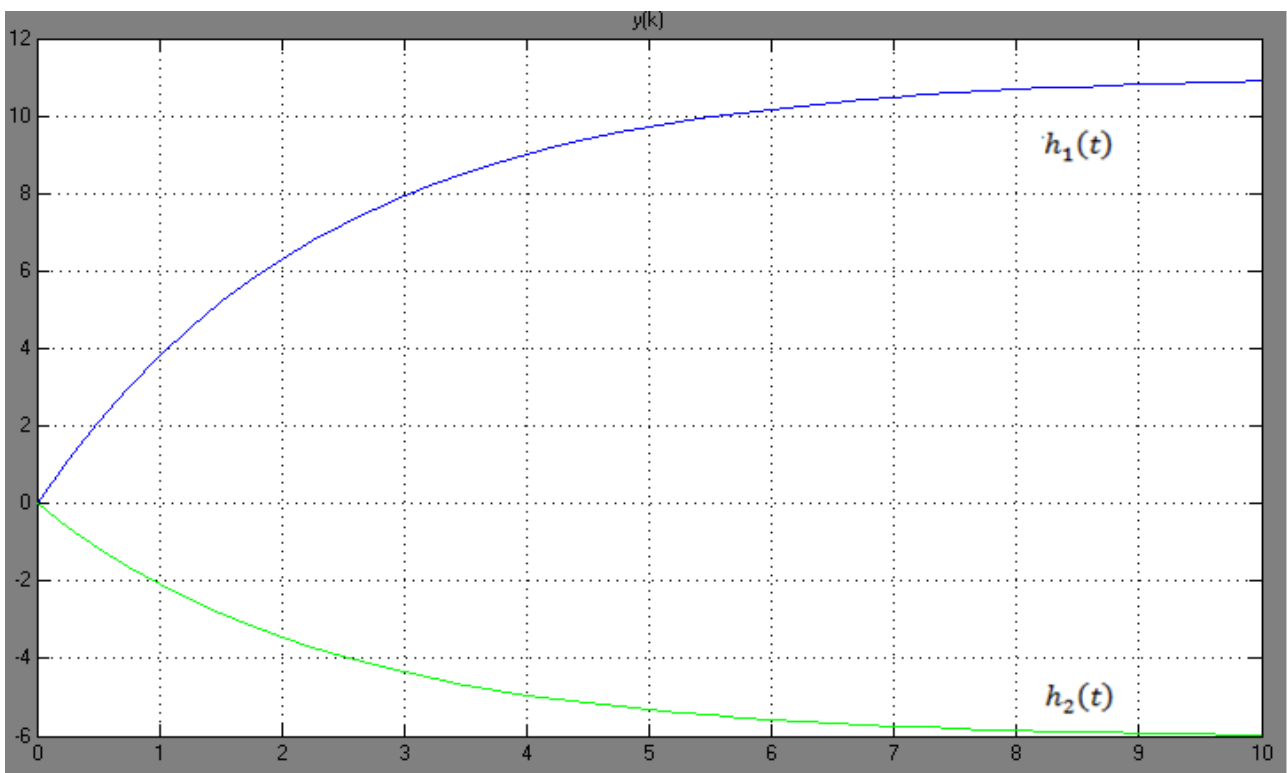
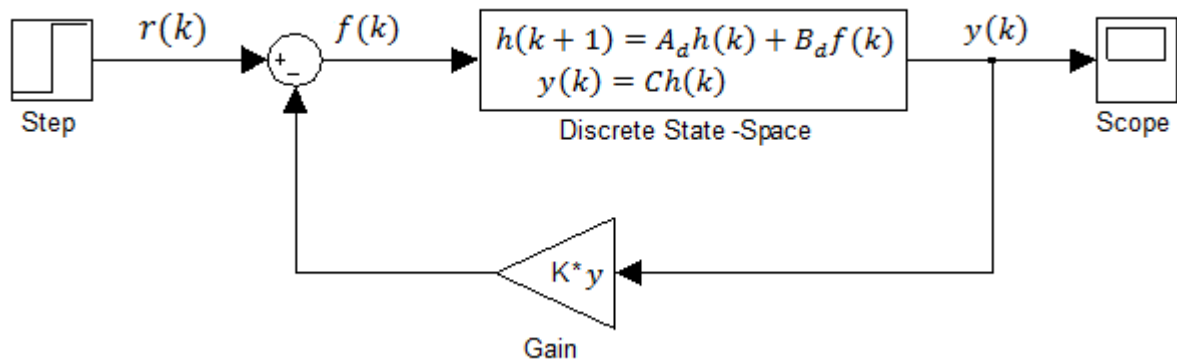


Figura 8: Schema a blocchi e andamento per il sistema controllato

La condizione di stabilità viene raggiunta nel ragionevole tempo di circa 7 secondi.

Tuttavia l'ingresso era:  $r(k) = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ , mentre in uscita misuriamo valori di regime abbastanza diversi, in particolare:  $h(\infty) = \begin{pmatrix} 11 \\ -6 \end{pmatrix}$ .

Al fine di ottenere un corretto valore di regime, è necessario imporre alcune condizioni aggiuntive. Supponiamo che l'utente voglia imporre il valore di regime  $r(k)$ , allora dovrà valere, dopo una prima fase transitoria, l'ovvia condizione:

$$y(k) = r(k)$$

Per ottenere questo risultato si inserisce un prefiltro  $F$  come in figura, di cui andremo a calcolare il valore al fine di far valere la condizione di cui sopra.

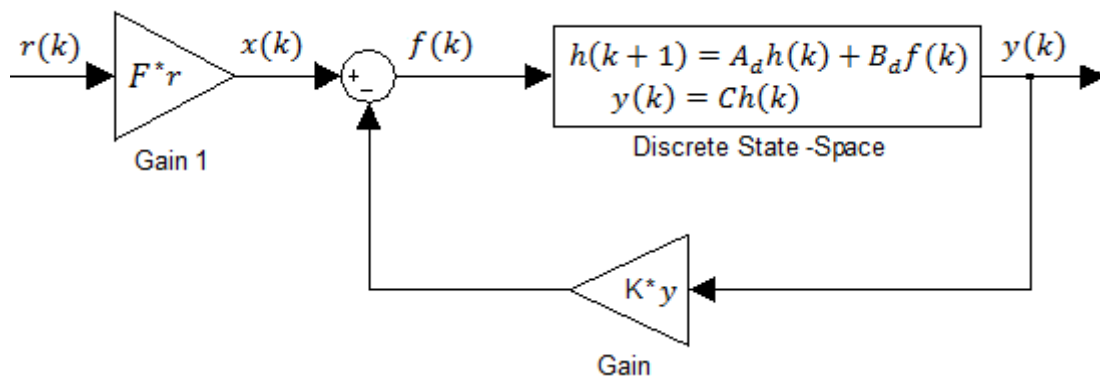


Figura 9: Schema a blocchi per il controllo del valore di regime

Ricordiamo a questo punto che a regime le altezze dei serbatoi non variano, ovvero si verifica che:

$$h(k+1) = h(k)$$

Allora la dinamica a regime in ciclo chiuso è rappresentata dalla relazione:

$$\begin{cases} h(k) = (A_d - B_d K)h(k) + B_d x(k) \\ y(k) = C h(k) \end{cases}$$

Ricordiamo ora che l'uscita deve diventare pari al riferimento imposto dall'utente, inoltre si ha che, per come è fatto il prefiltro,  $x(k) = F r(k)$ , da cui si ricava:

$$\begin{cases} h(k) = (A_d - B_d K)h(k) + B_d F r(k) \\ y(k) = C h(k) = r(k) \end{cases}$$

$$h(k) = (A_d - B_d K)h(k) + B_d F C h(k)$$

$$h(k)(I - (A_d - B_d K)) = B_d F C h(k)$$

$$(I - (A_d - B_d K)) = B_d F C$$

Si ottiene:

$$F = B_d^{-1}(I - (A_d - B_d K))C^{-1}$$

Da cui:

$$F = \begin{pmatrix} 0.4 & 0.4 \\ 0.0521 & -0.4 \end{pmatrix}$$

Per cui è possibile far sì che l'uscita segua il riferimento frazionando un controllore proporzionale di questo tipo fra il riferimento e l'ingresso del sistema in ciclo chiuso  $x(k)$ .

Possiamo ottenere questo risultato scrivendo:

```
>> Acc = Ad-Bd*K
>> F = Bd^(-1)*(I-Acc)*Cd^(-1)
```

Ancora una volta andiamo a verificare i risultati ottenuti mediante l'utilizzo di Simulink.

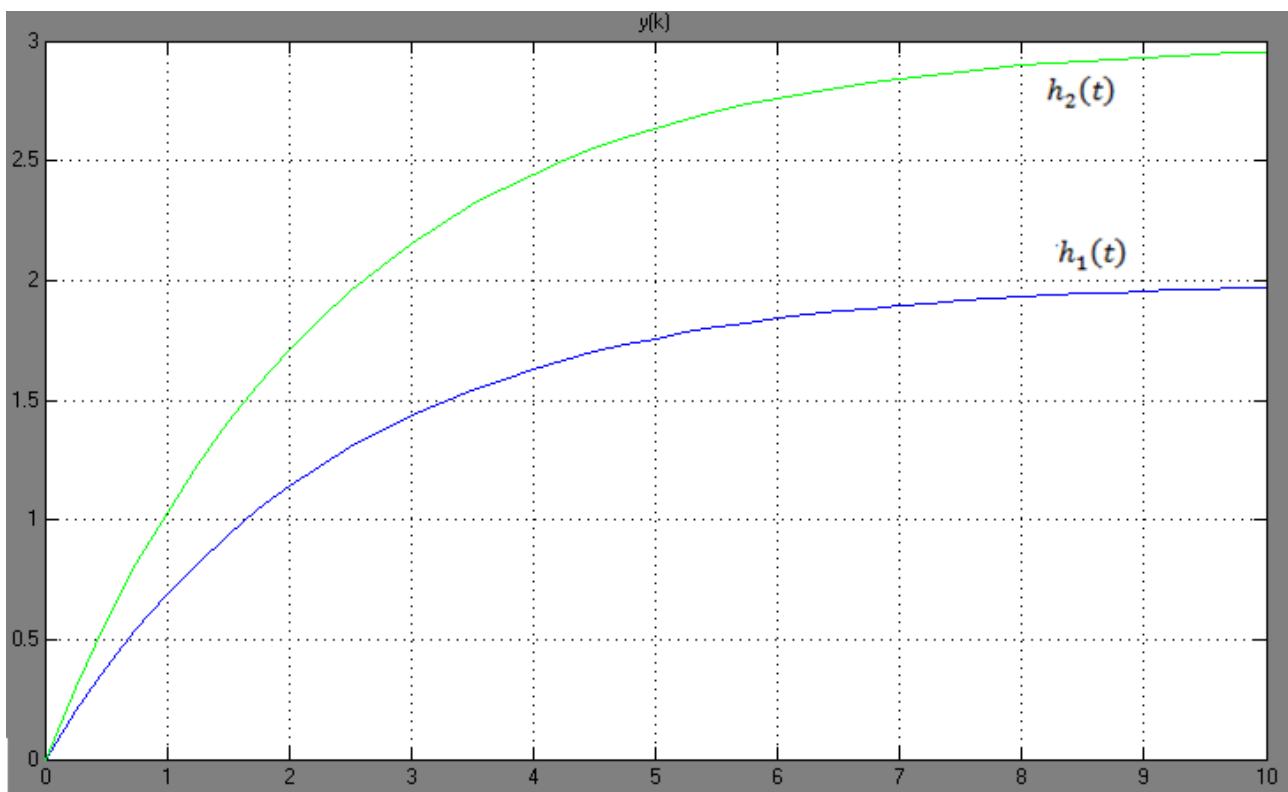
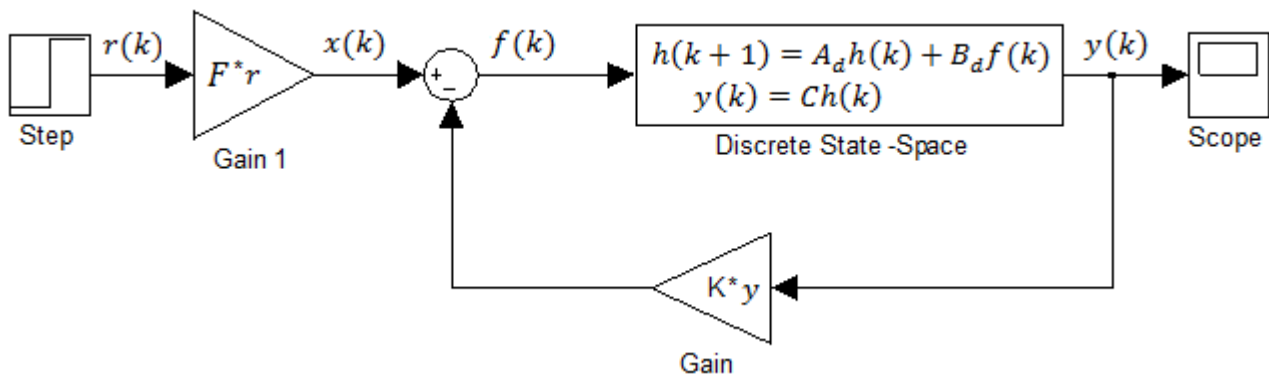


Figura 10: andamento finale per le altezze

In questo caso il sistema risponde correttamente al riferimento dell'utente, avendo imposto  $r(k) = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ . Inoltre possiamo osservare che il tempo di salita è di 5.25 secondi (le uscite raggiungono il 10% del valore di regime dopo 0.25 secondi ed il 90% dopo 5.5 secondi), per cui il piazzamento dei poli è coerente con la scelta effettuata per il periodo di campionamento.

Ora che i requisiti di stabilità e precisione a regime sono stati soddisfatti, verifichiamo sperimentalmente come si comporta il controllo implementato nei confronti del sistema. Per tale motivo, andiamo ad inserire uno "Scope" addizionale nel modello Simulink e verifichiamo l'andamento degli ingressi manipolabili, ovvero dei flussi.

Lo schema e la relativa simulazione risultano:

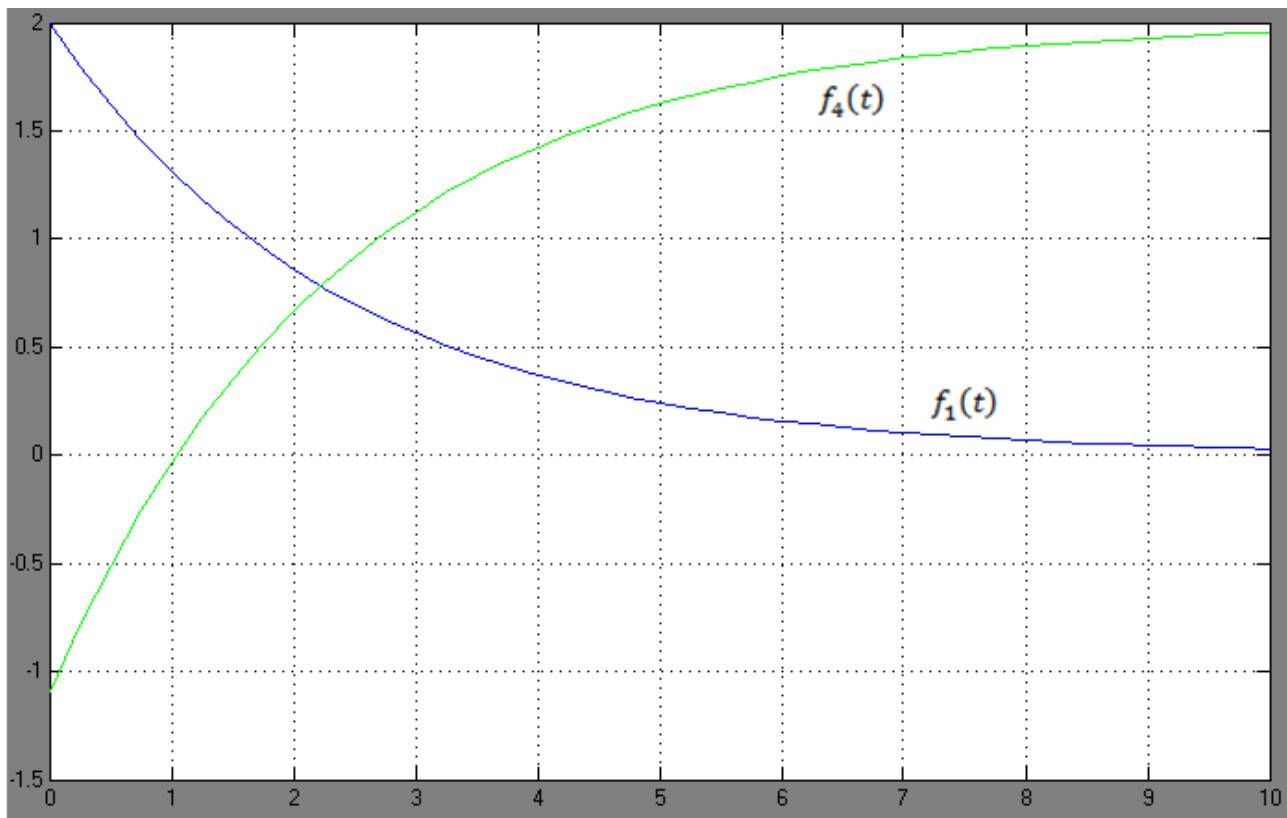
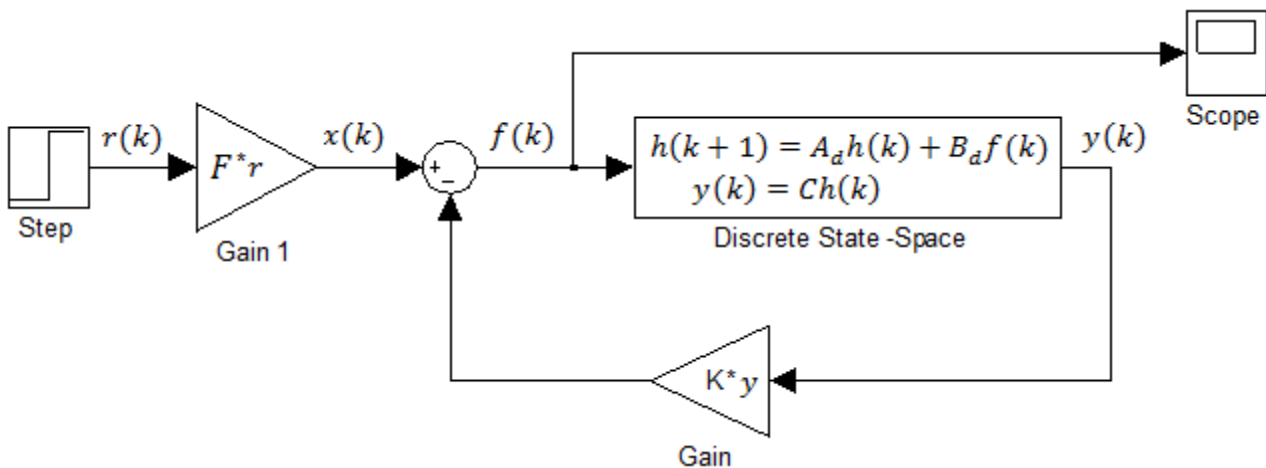


Figura 11: Andamenti dei flussi

La convergenza del sistema è sufficientemente lenta da garantire che i flussi in ingresso non siano eccessivamente elevati (come possiamo notare sono nell'ordine di grandezza dei riferimenti 3 e 2 applicati). Tuttavia notiamo anche che il flusso  $f_4$  inizialmente si attesta su valori negativi. Da un punto di vista fisico questo equivale a dire che inizialmente la pompa non preleva dal serbatoio a monte per scaricare su quello a valle, bensì compie il lavoro opposto.

Notiamo che il valore nel punto di equilibrio per  $f_4$  è 1 ( $f_{4,0}$ ), questo significa che fino a che il valore per  $f_4$  è negativo ma in modulo non superiore ad 1 allora fisicamente la pompa sta comunque pompando un flusso positivo.

Allora per risolvere il problema si può pensare di riposizionare i poli facendo in modo che il valore di regime per  $h_2$  venga raggiunto più lentamente

$$\begin{aligned}\lambda_1 &= 0.9 \\ \lambda_2 &= 0.95\end{aligned}$$

Con questa modifica ci si avvicina alla instabilità, tuttavia il problema risulta risolto, infatti si arriva a:

$$K = \begin{pmatrix} 0.4 & 0.2 \\ -0.9479 & -0.2 \end{pmatrix}$$

$$F = \begin{pmatrix} 0.4 & 0.2 \\ 0.0521 & -0.2 \end{pmatrix}$$

Con gli andamenti per  $h_1$  e  $h_2$  mostrati in figura 12, e i corrispondenti andamenti dei flussi riportati in figura 13.

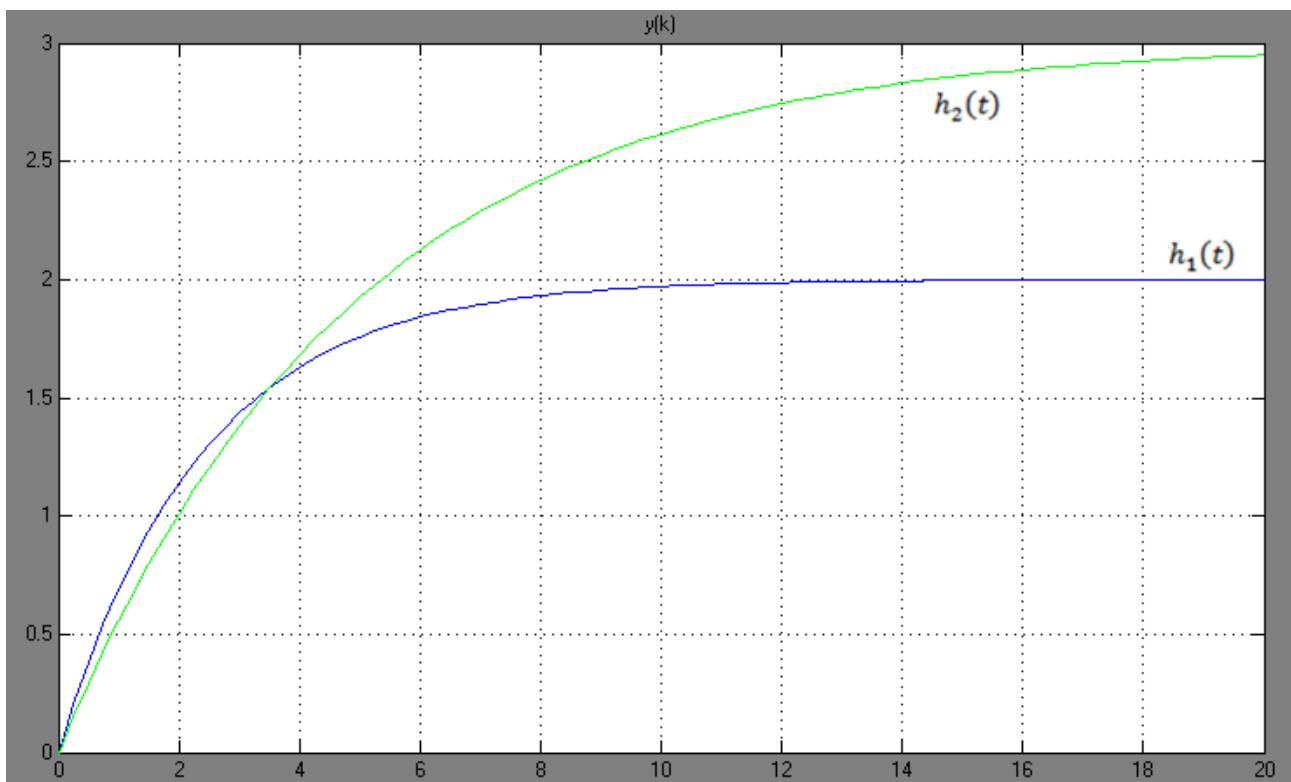


Figura 12: Nuovo andamento dei livelli

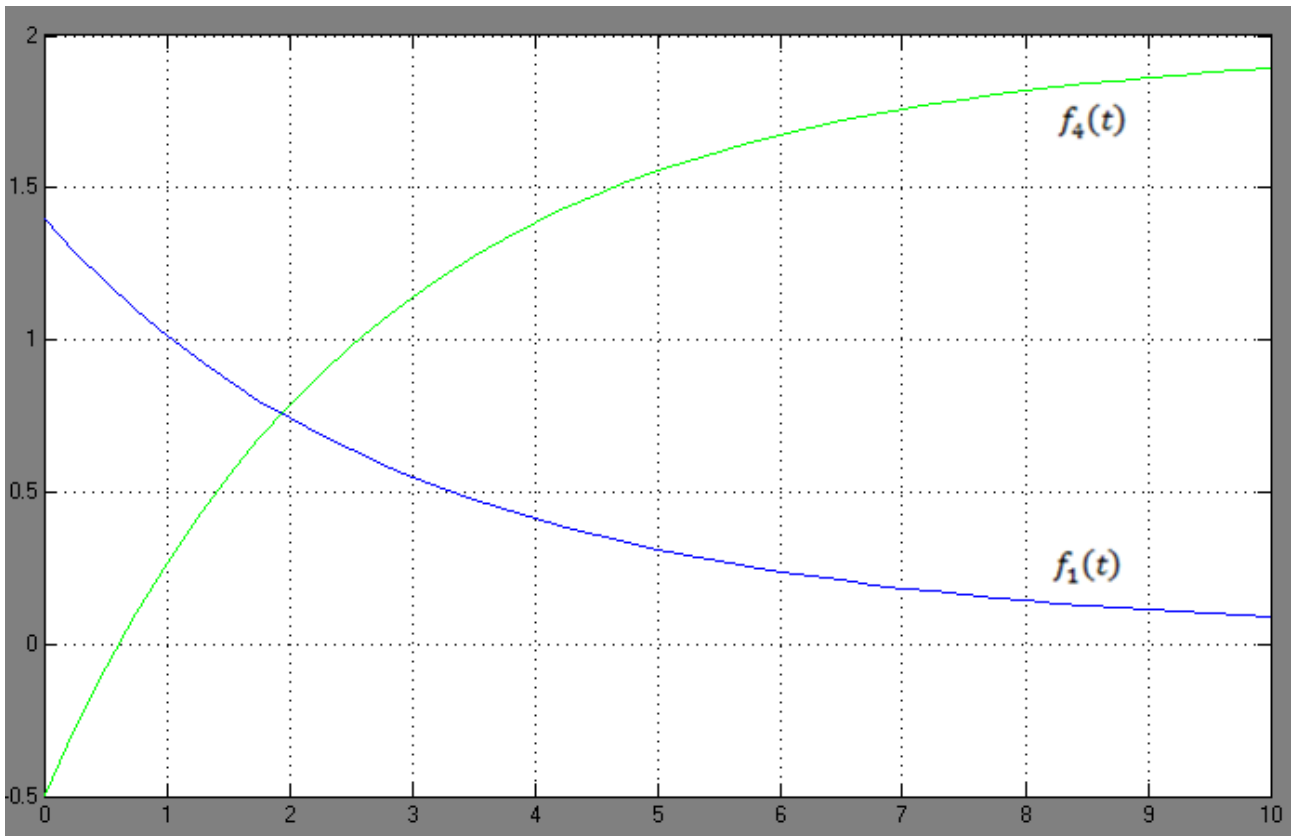


Figura 13: Nuovo andamento dei flussi

E come si nota il massimo valore negativo per  $f_4$  è in modulo pari a 0.5.

Si sottolinea comunque che questa soluzione è stata adattata ai particolari valori scelti per il riferimento, in realtà il problema si può presentare anche in altri casi e va ogni volta affrontato in modo diverso.

Quindi prima di realizzare un controllo in retroazione con la tecnica vista si raccomanda di fare accurate simulazioni per l'andamento dei flussi nello spazio dei possibili riferimenti utilizzati.

In sezione 10 vedremo come sia possibile spostarsi verso una soluzione generale al problema dei flussi negativi.

## SEZIONE 4: CONTROLLO AD ORIZZONTE SFUGGENTE

---

Prima di applicare questo controllo al sistema in esame, diamo una breve presentazione sulla teoria per il suo funzionamento.

Il controllo ad orizzonte sfuggente consiste nell'imporre che il riferimento venga raggiunto in un numero finito di passi  $N$ , si calcola poi la forma che dovrebbero avere gli ingressi ottimi in questi  $N$  passi e si fa il primo passo, il nuovo stato viene poi utilizzato come stato iniziale per una seconda iterazione del procedimento precedente, utilizzando lo stesso  $N$ .

Per capire il funzionamento del controllo ad orizzonte sfuggente partiamo dalla semplice relazione che si può ottenere dalla espressione del sistema in variabili di stato (già discretizzato):

$$\begin{cases} h(k+1) = A_d h(k) + B_d f(k) \\ y(k) = C h(k) + D f(k) \end{cases}$$

Da cui si deriva che  $h(N)$  (ossia lo stato alla iterazione  $N$ ) è dato da:

$$h(N) = A_d^N h_0 + (B_d \quad A_d B_d \quad \dots \quad A_d^{N-1} B_d) \begin{pmatrix} f(N-1) \\ \vdots \\ f(0) \end{pmatrix}$$

che si può scrivere in modo semplificato come:

$$h_d = A_d^N h_0 + \mathbf{C}_N \tilde{f}$$

Con:

$$\mathbf{C}_N = (B_d \quad A_d B_d \quad \dots \quad A_d^{N-1} B_d)$$

$$\tilde{f} = \begin{pmatrix} f(N-1) \\ \vdots \\ f(0) \end{pmatrix}$$

$$h_d = h(N) \text{ [valore desiderato al passo } N\text{-esimo]}$$

A questo punto possiamo isolare la  $\tilde{f}$  considerando che la  $\mathbf{C}_N$  ha dimensione  $[n \times Nm]$ , con  $n$  numero di variabili di stato e  $m$  numero degli ingressi.

Per quanto detto sopra allora siccome abbiamo  $Nm \geq n$  allora la pseudoinversa di  $\mathbf{C}_N$  si può calcolare come segue:

$$\mathbf{C}_N^+ = \mathbf{C}_N^T (\mathbf{C}_N \mathbf{C}_N^T)^{-1}$$

Ne deriva allora che:

$$\tilde{f}_{opt} = \mathbf{C}_N^T (\mathbf{C}_N \mathbf{C}_N^T)^{-1} (h_d - A_d^N h_0)$$

Dove il pedice "opt" sta a significare che la pseudoinversione calcola solo una delle possibili soluzioni, che rappresenta quella a norma minima. Notiamo inoltre che  $(\mathbf{C}_N \mathbf{C}_N^T) = G_N$ , ossia il gramiano di controllabilità in  $N$  passi.

Allora infine si arriva a:

$$\widetilde{f}_{opt} = \begin{pmatrix} B^T \\ \vdots \\ B^T A^{T^{N-1}} \end{pmatrix} G_N^{-1} (h_d - A_d^N h_0)$$

Si può dunque esprimere il k-esimo elemento di  $\widetilde{f}_{opt}$  in funzione della iterazione k, come segue:

$$\widetilde{f}_{opt}(k) = B^T A^{T^{N-1-k}} G_N^{-1} (h_d - A_d^N h_0)$$

Notiamo allora che in pratica bisognerebbe calcolare il valore opportuno per l'ingresso per ognuno degli N passi e poi applicarli in sequenza per ottenere il valore di regime desiderato proprio in N passi.

Consideriamo adesso di voler semplicemente stabilizzare il sistema (ossia controllare il sistema in modo che in evoluzione libera, con  $r(k) = 0$ , lo stato tenda ad annullarsi) il che comporta che per qualunque  $h_0$  vorremmo che venisse raggiunto il valore desiderato  $h_d = [0 \ 0]^T$ .

La tecnica del controllo ad orizzonte sfuggente (Receding Horizon Control, RHC) si basa sullo evitare di fare il calcolo dell'ingresso per ognuno degli N passi e invece utilizzare sempre il primo passo.

Nel caso in esame abbiamo allora che alla prima iterazione verrà applicato l'ingresso:

$$\widetilde{f}_{opt}(0) = -B^T A^{T^{N-1}} G_N^{-1} A_d^N h_0 = -K h_0$$

E alla seconda iterazione, al posto di applicare quella che sarebbe  $\widetilde{f}_{opt}(1)$ , si applica invece una nuova  $\widetilde{f}_{opt}(0)$  calcolata (teoricamente) con una nuova iterazione come sopra, però a partire da uno stato iniziale dato da:

$$h(1) = A_d h_0 - B_d K h_0$$

e avremo la nuova  $\widetilde{f}_{opt}(0) = -K h(1)$ .

Questo procedimento fa sì che il controllore venga molto semplice e abbia la stessa struttura di quello retroazionato con l'assegnamento dei poli, infatti in pratica si va sempre ad imporre:

$$f(k) = -K h(k)$$

E quindi si arriva alla usuale forma per il controllo retroazionato:

$$\begin{cases} h(k+1) = A_d h(k) - B_d K h(k) = (A_d - B_d K) h(k) \\ y(k) = C h(k) \end{cases}$$

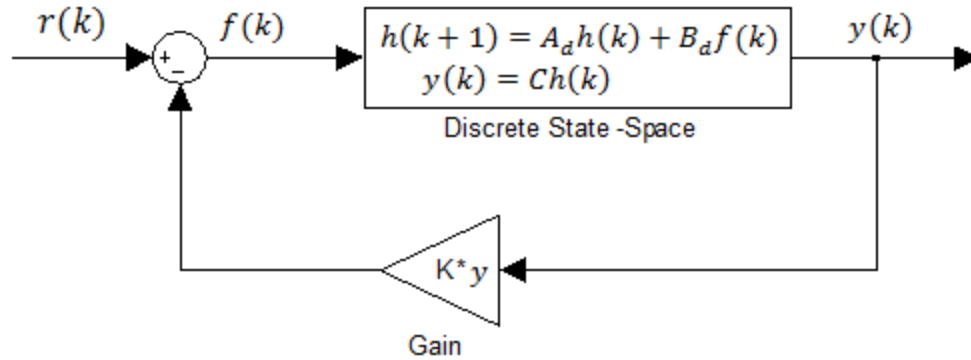


Figura 14: schema a blocchi per il controllo ad orizzonte sfuggente

Il controllore viene allora calcolato come:

$$K = B_d^T A_d^{T^{N-1}} G_N^{-1} A_d^N$$

Con  $N$  numero di passi da eseguire, e  $G_N$  gramiano di controllabilità in  $N$  passi:

$$G_N = B_d B_d^T + A_d B_d B_d^T A_d^T + \dots + A_d^{N-1} B_d B_d^T A_d^{T^{N-1}}$$

Allora il controllore può essere calcolato in Matlab come:

```
>> G = 0
>> for i = 1:N
>>     G = G + Ad^(i-1) * Bd * Bd' * Ad'^(i-1)
>> end
>> K = Bd' * Ad'^(N-1) * G^(-1) * Ad^N
```

Anche in questo caso, mediante ragionamenti analoghi ai precedenti, si nota che il corretto valore di regime non viene raggiunto, pertanto si inserisce il solito prefiltro  $F$ , calcolabile con la medesima formula vista in sezione 3:

$$F = B_d^{-1} (I - (A_d - B_d K)) C^{-1}$$

Il risultato ottenuto varierà al variare del numero  $N$  di passi.

In particolare maggiore sarà  $N$ , più bassi saranno gli ingressi, mentre al diminuire di  $N$  aumenterà la velocità di risposta del sistema.

Andiamo a verificare il risultato ottenuto mediante simulazione su Simulink.

Lo schema a blocchi utilizzato sarà il medesimo della sezione 3, mentre varieranno le matrici  $K$  ed  $F$ , la situazione è rappresentata in figura 15.

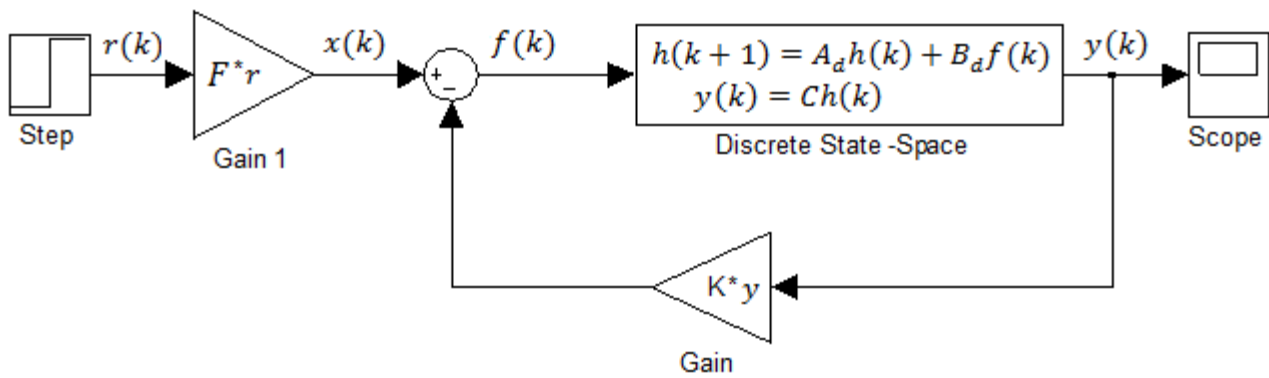


Figura 15: nuovo schema a blocchi per il controllo ad orizzonte sfuggente

Supponendo di effettuare la manovra in  $N = 3$  passi, simulando si ottiene:

$$K = \begin{pmatrix} 1.4294 & 1.6256 \\ -0.3088 & -0.94 \end{pmatrix}; F = \begin{pmatrix} 1.4294 & 1.6256 \\ 0.6912 & -0.94 \end{pmatrix}$$

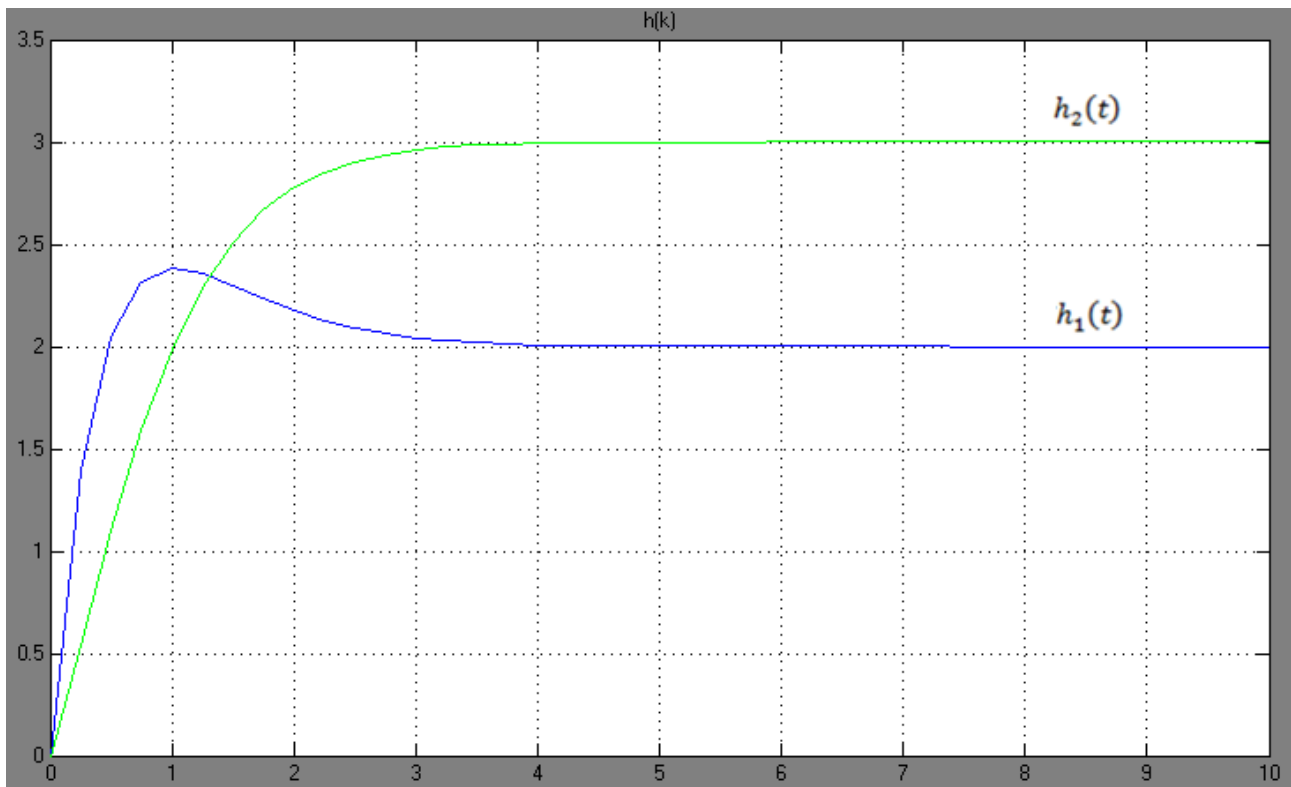


Figura 16: Altezze dei serbatoi, RHC, N=3

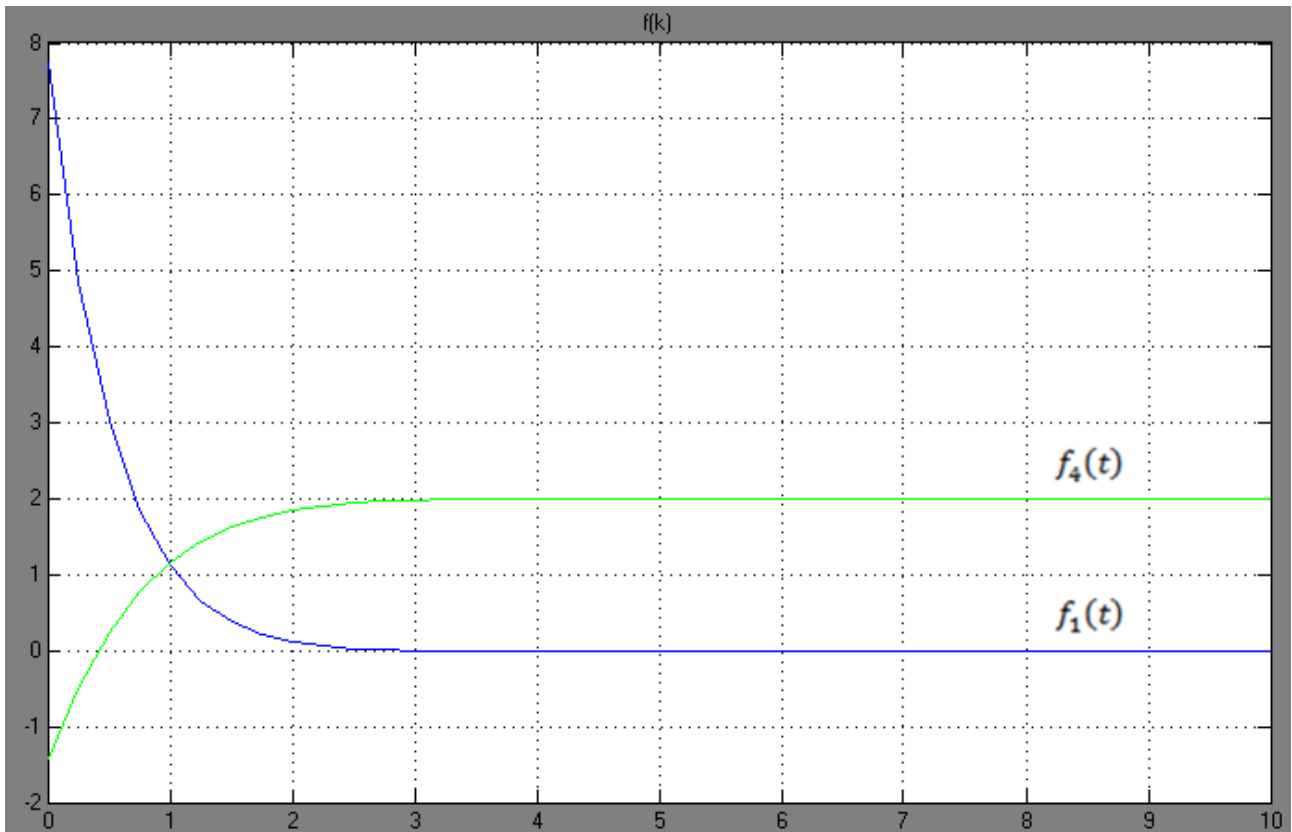


Figura 17: Flussi di ingresso, RHC, N=3

Possiamo osservare che la condizione di regime viene raggiunta in un tempo troppo breve (rispetto al tempo di salita imposto in fase di discretizzazione del sistema). Una delle uscite è inoltre soggetta a sovralongazione.

Per quanto riguarda gli ingressi possiamo notare che il fatto di utilizzare un ridotto numero di passi per la manovra rende gli ingressi elevati in modulo nel transitorio. Inoltre si può osservare che il flusso di ricircolo, come già accadeva in altri casi, è negativo nella fase iniziale.

Se andiamo ad imporre un numero maggiore di passi, ci aspettiamo che la situazione migliori, in particolare per quel che riguarda i flussi in ingresso per effettuare la manovra.

Ponendo  $N = 10$  passi, si ottengono le seguenti matrici di controllo:

$$K = \begin{pmatrix} 0.5199 & 0.5545 \\ -0.0408 & -0.0526 \end{pmatrix}; F = \begin{pmatrix} 0.5199 & 0.5545 \\ 0.9592 & -0.0526 \end{pmatrix}$$

Verifichiamo anzitutto l'andamento degli ingressi, riportato in figura 18.

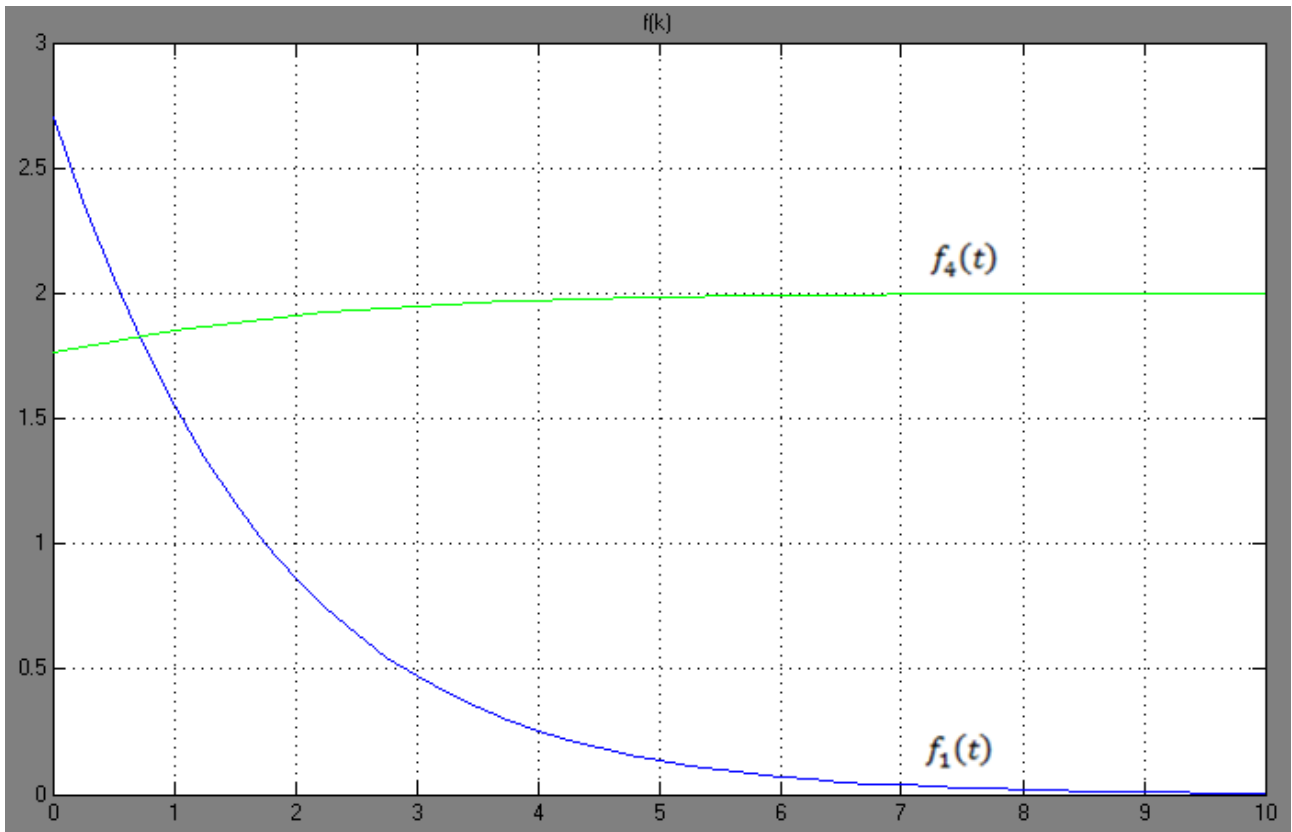


Figura 18: Flussi di ingresso, RHC, N=10

Come possiamo notare, stavolta gli ingressi non assumono valori negativi, inoltre hanno un valore massimo abbastanza contenuto. Ciò si deve al fatto che con  $N = 10$  la manovra viene “dilazionata” nel tempo, rispetto all’andamento “ripido” del caso precedente. Si verifica a questo punto l’andamento delle uscite, sempre mediante la medesima simulazione:

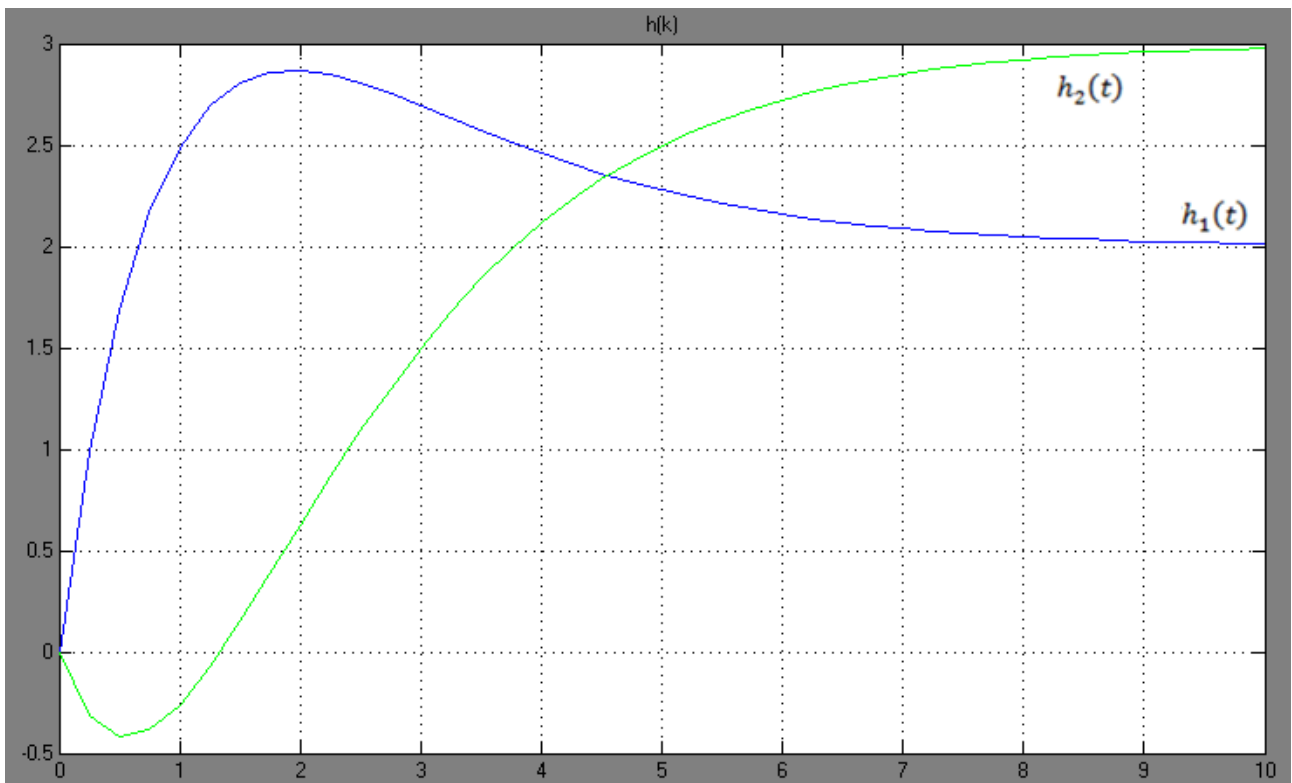


Figura 19: Altezze dei serbatoi, RHC, N=10

Conseguenza diretta dell'aver dilazionato la manovra è che anche le uscite seguono un andamento più "smooth": il valore di regime viene raggiunto più lentamente.

Tuttavia è possibile osservare diversi fenomeni poco piacevoli:

- La sovraelongazione della prima uscita ha avuto un notevole incremento, e nonostante il valore di regime venga raggiunto in modo non troppo rapido, quest'ultima fa sì che il tempo di salita sia comunque di molto inferiore a quello da noi imposto per la progettazione in fase di discretizzazione.
- La seconda uscita inizia a presentare fenomeni di sottoelongazione, che potrebbero (per determinati riferimenti) portare il serbatoio al di sotto dell'altezza minima consentita (ossia  $-1$ , visto che  $h_1(k)$  e  $h_2(k)$  rappresentano una variazione rispetto al punto di equilibrio con  $h_{1,0} = h_{2,0} = 1$ ).

Si potrebbe a questo punto passare ad esaminare casi con  $N > 10$ , tuttavia l'unico fenomeno che si osserva è che i comportamenti sopra descritti si accentuano sempre di più all'aumentare di  $N$ .

## SEZIONE 5: CONTROLLO LINEAR QUADRATIC REGULATOR (LQR)

---

La tecnica di Riccati per il controllo ottimo consente di trovare il controllo retroazionato che minimizza un determinato indice di costo che risulta particolarmente significativo.

In particolare, nella sua forma generale, l'indice di costo  $J$  da minimizzare viene scelto come segue:

$$J = \frac{1}{2} \sum_{k=0}^N [h^T(k)Qh(k) + f^T(k)Rf(k)]$$

con  $h(k)$  le variabili di stato e  $f(k)$  i flussi da noi imposti.

$R$  e  $Q$  sono matrici dei pesi che sono definite positive e tipicamente (come nel nostro caso) vengono scelte simmetriche (ossia  $Q = Q^T$ ,  $Q^{-1} = Q^{-T}$ ,  $R = R^T$  e  $R^{-1} = R^{-T}$ ).

Quello che inoltre deve sempre valere per il nostro sistema è che:

$$h(k+1) = A_d h(k) + B_d f(k)$$

Volendo allora trovare il controllore tale che  $J$  risulta minimo, per rimuovere il vincolo dovuto alla evoluzione del sistema (che renderebbe il problema estremamente complesso) si può pensare di ricavare il controllore che minimizza:

$$J' = \max_{\lambda(k+1)} \left\{ \sum_{k=0}^{\infty} \left[ \frac{1}{2} h^T(k)Qh(k) + \frac{1}{2} f^T(k)Rf(k) + \lambda^T(k+1)(-h(k+1) + A_d h(k) + B_d f(k)) \right] \right\}$$

Dove  $\lambda^T(k+1)$  è appunto il vettore dei moltiplicatori di Lagrange.

Cercando di minimizzare l'equazione dove  $\lambda(k+1)$  è scelto per massimizzare il costo allora implicitamente si sta imponendo che:

$$h(k+1) - A_d h(k) + B_d f(k) = 0 \quad \forall k$$

perché altrimenti i  $\lambda$  sarebbero capaci solo col terzo termine di portare il costo a valori arbitrariamente grandi.

L'indice  $k+1$  per  $\lambda^T$  è stato scelto arbitrariamente per semplificare la trattazione a seguire.

Allora si cerca il minimo rispetto a  $h(k)$ ,  $f(k)$ , del massimo rispetto a  $\lambda(k+1)$ .

Si sta cercando di minimizzare una sommatoria, esplicitando alcuni termini abbiamo:

$$\begin{aligned} J' = & \frac{1}{2} h^T(0)Qh(0) + \frac{1}{2} f^T(0)Rf(0) + \lambda^T(1)(-h(1) + A_d h(0) + B_d f(0)) + \\ & \frac{1}{2} h^T(1)Qh(1) + \frac{1}{2} f^T(1)Rf(1) + \lambda^T(2)(-h(2) + A_d h(1) + B_d f(1)) + \\ & \dots\dots\dots + \\ & \frac{1}{2} h^T(k-1)Qh(k-1) + \frac{1}{2} f^T(k-1)Rf(k-1) + \lambda^T(k)(-h(k) + A_d h(k-1) + B_d f(k-1)) + \\ & \frac{1}{2} h^T(k)Qh(k) + \frac{1}{2} f^T(k)Rf(k) + \lambda^T(k+1)(-h(k+1) + A_d h(k) + B_d f(k)) + \\ & \dots\dots\dots \end{aligned}$$

Allora derivando rispetto a  $h(k)$ ,  $f(k)$  e  $\lambda(k+1)$  e imponendo il risultato pari a zero otteniamo:

$$\begin{cases} \frac{\partial J'}{\partial f(k)} = f^T(k)R + \lambda^T(k+1)B_d = 0 \\ \frac{\partial J'}{\partial h(k)} = h^T(k)Q - \lambda^T(k) + \lambda^T(k+1)A_d = 0 \\ \frac{\partial J'}{\partial \lambda(k+1)} = -h(k+1) + A_d h(k) + B_d f(k) = 0 \end{cases}$$

E dalla seconda:

$$\lambda(k+1) = -A_d^{-T} Q^T h(k) + A_d^{-T} \lambda(k) = -A_d^{-T} Q h(k) + A_d^{-T} \lambda(k)$$

Dalla prima invece abbiamo che:

$$f(k) = -R^{-T} B_d^T \lambda(k+1) = -R^{-1} B_d^T \lambda(k+1)$$

E quindi dalla terza:

$$\begin{aligned} h(k+1) &= A_d h(k) + B_d f(k) = A_d h(k) - B_d R^{-1} B_d^T \lambda(k+1) \\ &= A_d h(k) - B_d R^{-1} B_d^T [-A_d^{-T} Q h(k) + A_d^{-T} \lambda(k)] \end{aligned}$$

Da cui:

$$h(k+1) = (A_d + B_d R^{-1} B_d^T A_d^{-T} Q) h(k) - B_d R^{-1} B_d^T A_d^{-T} \lambda(k)$$

Mettendo i due risultati sottoforma di sistema otteniamo:

$$\begin{cases} h(k+1) = (A_d + B_d R^{-1} B_d^T A_d^{-T} Q) h(k) - B_d R^{-1} B_d^T A_d^{-T} \lambda(k) \\ \lambda(k+1) = -A_d^{-T} Q h(k) + A_d^{-T} \lambda(k) \end{cases}$$

O, in forma matriciale:

$$\begin{bmatrix} h(k+1) \\ \lambda(k+1) \end{bmatrix} = \begin{bmatrix} A_d + B_d R^{-1} B_d^T A_d^{-T} Q & -B_d R^{-1} B_d^T A_d^{-T} \\ -A_d^{-T} Q & A_d^{-T} \end{bmatrix} \begin{bmatrix} h(k) \\ \lambda(k) \end{bmatrix}$$

Questa prende il nome di equazione Hamiltoniana, dove la matrice:

$$H_c = \begin{pmatrix} A_d + B_d R^{-1} B_d^T A_d^{-T} Q & -B_d R^{-1} B_d^T A_d^{-T} \\ -A_d^{-T} Q & A_d^{-T} \end{pmatrix}$$

è la matrice Hamiltoniana.

Questa equazione fornisce la soluzione ottima per il sistema.

Cerchiamo adesso di arrivare al controllore che rende il comportamento del sistema quello desiderato.

$H_c$  è una matrice con  $2n$  autovalori (dove  $n$  al solito è la dimensione del sistema).

I  $2n$  autovalori sono tali che se  $z$  è autovalore allora anche  $z^{-1}$  lo è e quindi sicuramente si avranno  $n$  autovalori stabili ed  $n$  instabili, chiamiamo allora:

- $E^{-1}$  matrice diagonale degli  $n$  autovalori stabili
- $E$  matrice diagonale degli  $n$  autovalori instabili
- $\begin{pmatrix} X \\ \Lambda \end{pmatrix}$  generica forma per gli autovettori relativi agli  $n$  autovalori stabili
- $\begin{pmatrix} X_0 \\ \Lambda_0 \end{pmatrix}$  generica forma per gli autovettori relativi agli  $n$  autovalori instabili

Possiamo trasformare l'equazione Hamiltoniana come segue

$$T^{-1} \begin{pmatrix} h(k+1) \\ \lambda(k+1) \end{pmatrix} = T^{-1} H_c T T^{-1} \begin{pmatrix} h(k) \\ \lambda(k) \end{pmatrix}$$

Con:

$$T = \begin{pmatrix} X & X_0 \\ \Lambda & \Lambda_0 \end{pmatrix}$$

Tale trasformazione di similitudine è tale da diagonalizzare  $H_c$ , si ha allora che:

$$T^{-1} H_c T = \begin{pmatrix} E^{-1} & 0 \\ 0 & E \end{pmatrix}$$

Chiamiamo inoltre:

- $T^{-1} \begin{pmatrix} h(k+1) \\ \lambda(k+1) \end{pmatrix} = \begin{pmatrix} h^*(k+1) \\ \lambda^*(k+1) \end{pmatrix}$
- $T^{-1} \begin{pmatrix} h(k) \\ \lambda(k) \end{pmatrix} = \begin{pmatrix} h^*(k) \\ \lambda^*(k) \end{pmatrix}$

Dunque possiamo scrivere:

$$\begin{pmatrix} h^*(k+1) \\ \lambda^*(k+1) \end{pmatrix} = \begin{pmatrix} E^{-1} & 0 \\ 0 & E \end{pmatrix} \begin{pmatrix} h^*(k) \\ \lambda^*(k) \end{pmatrix}$$

Da cui, semplicemente:

$$\begin{cases} h^*(k+1) = E^{-1} h^*(k) \\ \lambda^*(k+1) = E \lambda^*(k) \end{cases}$$

che possono esser visti come due sistemi in evoluzione libera, quindi:

$$\begin{cases} h^*(N) = (E^{-1})^N h^*(0) \\ \lambda^*(N) = E^N \lambda^*(0) \end{cases}$$

Ricordiamo adesso che:

$$\begin{pmatrix} h(k) \\ \lambda(k) \end{pmatrix} = \begin{pmatrix} X & X_0 \\ \Lambda & \Lambda_0 \end{pmatrix} \begin{pmatrix} h^*(k) \\ \lambda^*(k) \end{pmatrix}$$

da cui:

$$h(k) = Xh^*(k) + X_0\lambda^*(k)$$

Quindi se  $\lambda^*(k)$  avesse valore arbitrariamente grande allora significherebbe che lo stato  $h(k)$  tende a divergere, questo ci dice che in realtà  $\lambda^*(k)$  non può assumere valore infinito.

Quanto detto sopra significa che, siccome la  $E$  è tale da causare una evoluzione instabile (autovalori instabili), l'unico modo in cui si ottiene  $\lambda^*(N)$  finito  $\forall N$  è con  $\lambda^*(0) = 0$ , il che comporta anche che  $\lambda^*(k) = 0 \forall k$ .

Ma allora è possibile scrivere:

$$\begin{pmatrix} h(k) \\ \lambda(k) \end{pmatrix} = T \begin{pmatrix} h^*(k) \\ \lambda^*(k) \end{pmatrix} = \begin{pmatrix} X & X_0 \\ \Lambda & \Lambda_0 \end{pmatrix} \begin{pmatrix} h^*(k) \\ \lambda^*(k) \end{pmatrix} = \begin{pmatrix} Xh^*(k) \\ \Lambda h^*(k) \end{pmatrix}$$

da cui:

$$\begin{cases} h(k) = Xh^*(k) \xrightarrow{\text{invertendo}} h^*(k) = X^{-1}h(k) \\ \lambda(k) = \Lambda h^*(k) \end{cases}$$

e quindi:

$$\lambda(k) = \Lambda X^{-1}h(k)$$

Questo risultato, considerato assieme ai risultati ottenuti in precedenza e riportati in seguito:

- $Rf(k) = -B_d^T \lambda(k+1)$
- $h(k+1) = A_d h(k) + B_d f(k)$

porta finalmente a ricavare il controllore:

$$\begin{aligned} Rf(k) &= -B_d^T \lambda(k+1) = -B_d^T \Lambda X^{-1} h(k+1) = -B_d^T \Lambda X^{-1} (A_d h(k) + B_d f(k)) \\ &= -B_d^T \Lambda X^{-1} A_d h(k) - B_d^T \Lambda X^{-1} B_d f(k) \end{aligned}$$

EQuindi isolando la  $f(k)$  otteniamo:

$$(R + B_d^T \Lambda X^{-1} B_d) f(k) = -B_d^T \Lambda X^{-1} A_d h(k)$$

e finalmente:

$$f(k) = -(R + B_d^T \Lambda X^{-1} B_d)^{-1} B_d^T \Lambda X^{-1} A_d h(k) = -Kh(k)$$

Il controllore reazionato cercato è dunque il seguente:

$$K = (R + B_d^T \Lambda X^{-1} B_d)^{-1} B_d^T \Lambda X^{-1} A_d$$

Il sistema controllato avrà la usuale forma:

$$\begin{cases} h(k+1) = A_d h(k) - B_d K h(k) = (A_d - B_d K) h(k) \\ y(k) = C h(k) \end{cases}$$

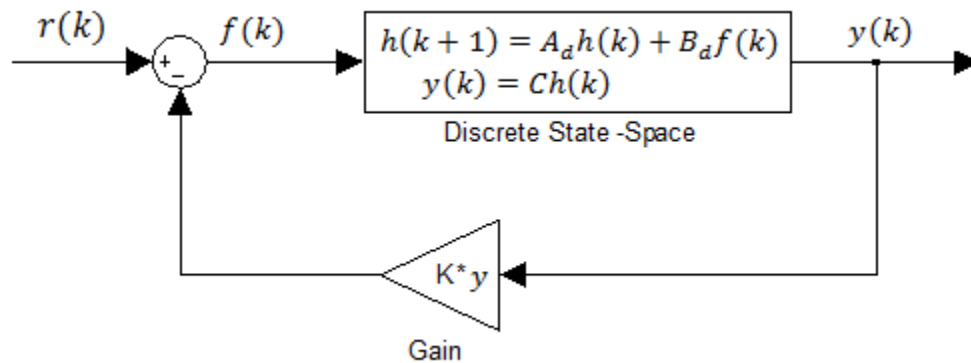


Figura 20: schema a blocchi per il controllo LQR

Al solito però bisogna notare che abbiamo semplicemente stabilizzato il sistema, assumendo riferimento nullo ( $r(k) = 0$ ).

Si progetta a questo punto un controllore ottimo che minimizzi l'indice  $J$  (Linear Quadratic Regulator, LQR) per il sistema "Doppio Serbatoio".

Anzitutto va calcolata la matrice Hamiltoniana:

$$H_c = \begin{pmatrix} A_d + B_d R^{-1} B_d^T A_d^{-T} Q & -B_d R^{-1} B_d^T A_d^{-T} \\ -A_d^{-T} Q & A_d^{-T} \end{pmatrix}$$

La Matrice  $H_c$  ha dimensione  $2n$ , con  $n$  dimensione della matrice  $A_d$ .

Di questi si determinano gli autovettori  $\begin{pmatrix} X \\ \Lambda \end{pmatrix}$  relativi ad i suoi  $n$  autovalori stabili.

Infine il controllore può essere determinato mediante la relazione:

$$K = (R + B_d^T \Lambda X^{-1} B_d)^{-1} B_d^T \Lambda X^{-1} A_d$$

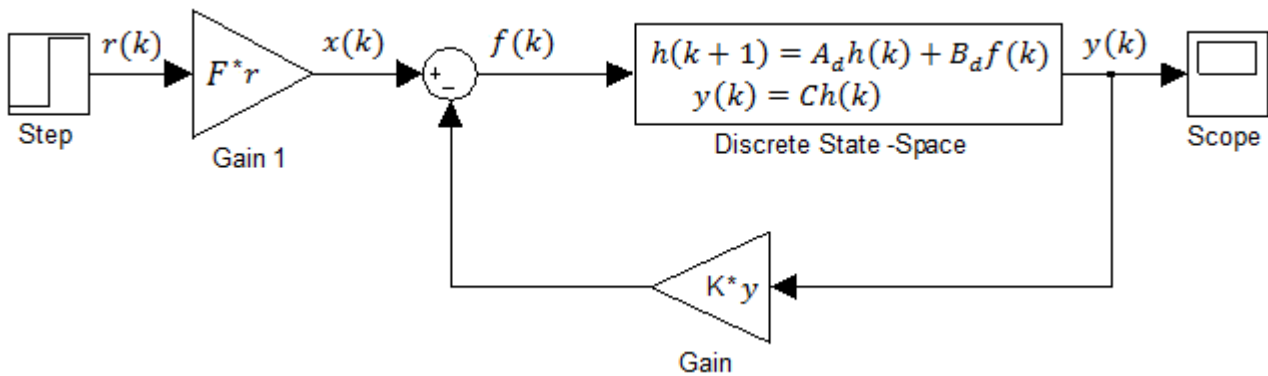
In Matlab la procedura sopra descritta può essere eseguita nel modo seguente.

```
>> Hc = [ Ad+Bd*R^(-1)*Bd'*Ad'^(-1)* Q , -Bd*R^(-1)* Bd'*Ad'^(-1);
>>          - Ad'^(-1) * Q , Ad'^(-1) ]; %Hamiltoniana
>> % Autovalori ed Autovettori
>> [auv,D] = eig(Hc);
>> j=1;
>> V=zeros(size(D,1),size(D,1)/2);
>> for i=1:size(D)
>>     if abs(D(i,i)) < 1
>>         % Autovettori Associati Ad Autovalori stabili
>>         V(:,j) = auv(:,i);
>>         j=j+1;
>>     end
>> end
>> X = V(1:size(V,1)/2,:);
>> L = V(size(V,1)/2+1:size(V,1),:);
>> % Controllore
>> K = ( R + Bd' * L * X^(-1) * Bd )^(-1) * Bd' * L * X^(-1) * Ad;
```

Anche in questo caso, mediante ragionamenti analoghi ai precedenti, si nota che il corretto valore di regime non viene raggiunto, pertanto si inserisce il solito prefiltro  $F$ , calcolabile con la medesima formula dei casi precedenti:

$$F = B_d^{-1}(I - (A_d - B_d K))C^{-1}$$

Ovviamente la procedura varia al variare di  $Q$ , che pesa lo stato, ed  $R$ , che pesa gli ingressi. Nel nostro caso lo stato è rappresentato dalle altezze dei serbatoi, mentre gli ingressi sono i flussi. Andiamo a verificare il risultato ottenuto mediante simulazione su Simulink. Lo schema a blocchi del sistema sarà il medesimo utilizzato per i controlli precedenti, mentre varieranno le matrici  $K$  ed  $F$ .



Supponendo di utilizzare delle semplici matrici identità per  $Q$  ed  $R$  si ottiene:

$$K = \begin{pmatrix} 0.7778 & 0.7076 \\ 0.0010 & -0.5784 \end{pmatrix}; F = \begin{pmatrix} 0.7778 & 0.7076 \\ 1.0010 & -0.5784 \end{pmatrix}$$

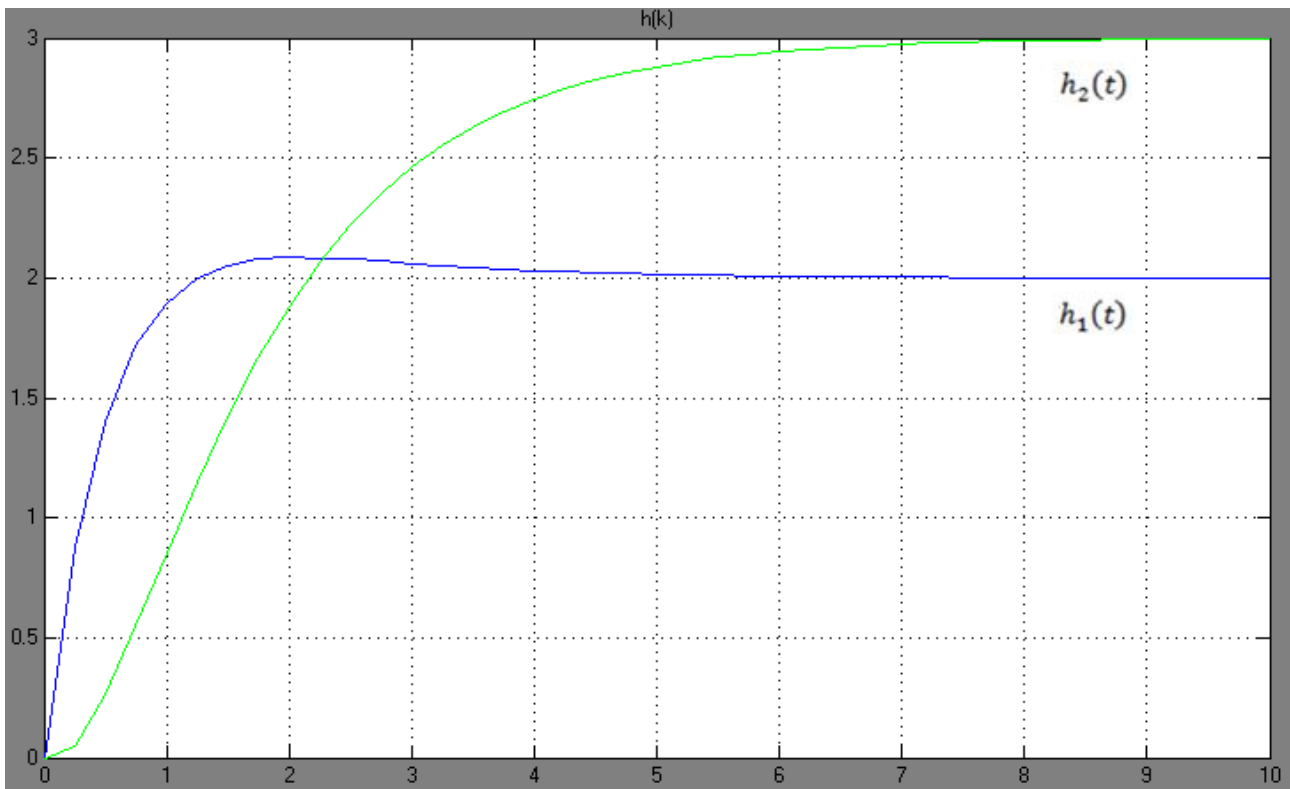


Figura 21: Altezze dei serbatoi, LQR,  $Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$   $R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

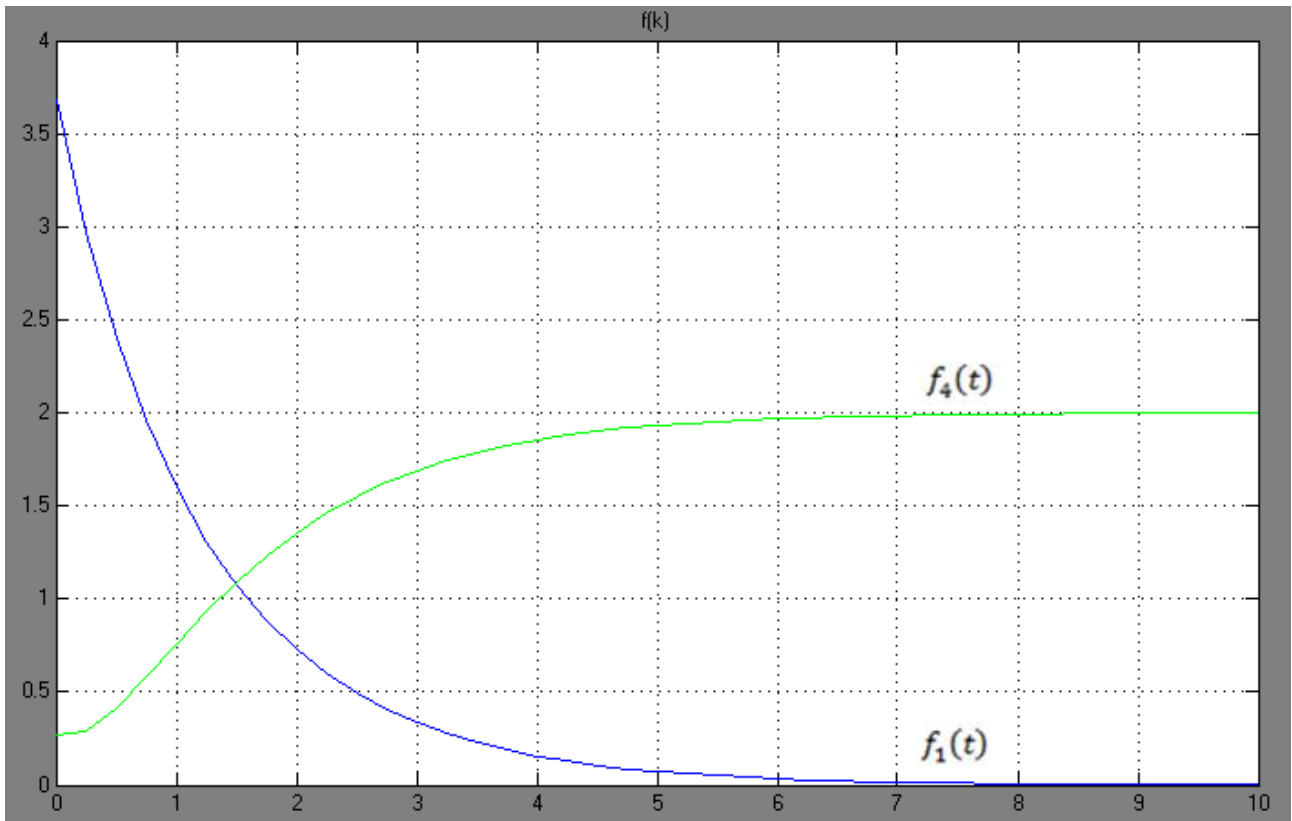


Figura 22: Flussi di ingresso, LQR,  $Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Il risultato ottenuto non è a pienamente soddisfacente, tuttavia è un buon inizio.

Possiamo osservare i seguenti comportamenti:

- Le uscite raggiungono molto rapidamente il valore di regime, più di quanto sia lecito secondo i criteri sul tempo di salita imposti in fase di discretizzazione
- Il flusso d'ingresso  $f_1(k)$  assume valori più alti del necessario nella fase iniziale, se confrontato con altri flussi ottenuti mediante altri regolatori
- In questo caso  $f_4(k)$  non ci da problemi, in quanto viene ad assumere sempre valori superiori a  $-1$ .

Possiamo cogliere due piccioni con una fava semplicemente andando ad agire sul costo di  $f_1(k)$ , in quanto:

- Aumentando il costo di  $f_1(k)$ , questo diminuirà in modulo, quindi il picco che assume inizialmente tenderà a ridursi
- Riducendosi  $f_1(k)$ , il primo serbatoio tenderà a riempirsi più lentamente, e per forza di cose lo stesso accadrà al secondo, che riceve un flusso dipendente dall'altezza del primo

Basandosi sulle considerazioni fisiche sopra analizzate, andiamo ad applicarle aumentando a 7 il costo di  $f_1(k)$ , ovvero utilizzando le seguenti matrici di peso:

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; R = \begin{pmatrix} 7 & 0 \\ 0 & 1 \end{pmatrix}$$

Da cui si ottengono le seguenti matrici di controllo:

$$K = \begin{pmatrix} 0.3056 & 0.2919 \\ 0.0295 & -0.5521 \end{pmatrix}; F = \begin{pmatrix} 0.3056 & 0.2919 \\ 1.0295 & -0.5521 \end{pmatrix}$$

Andiamo anzitutto a verificare che il peso abbia avuto effetto verificando i nuovi flussi di ingresso all' impianto. Simulando si ottiene il risultato in figura 23.

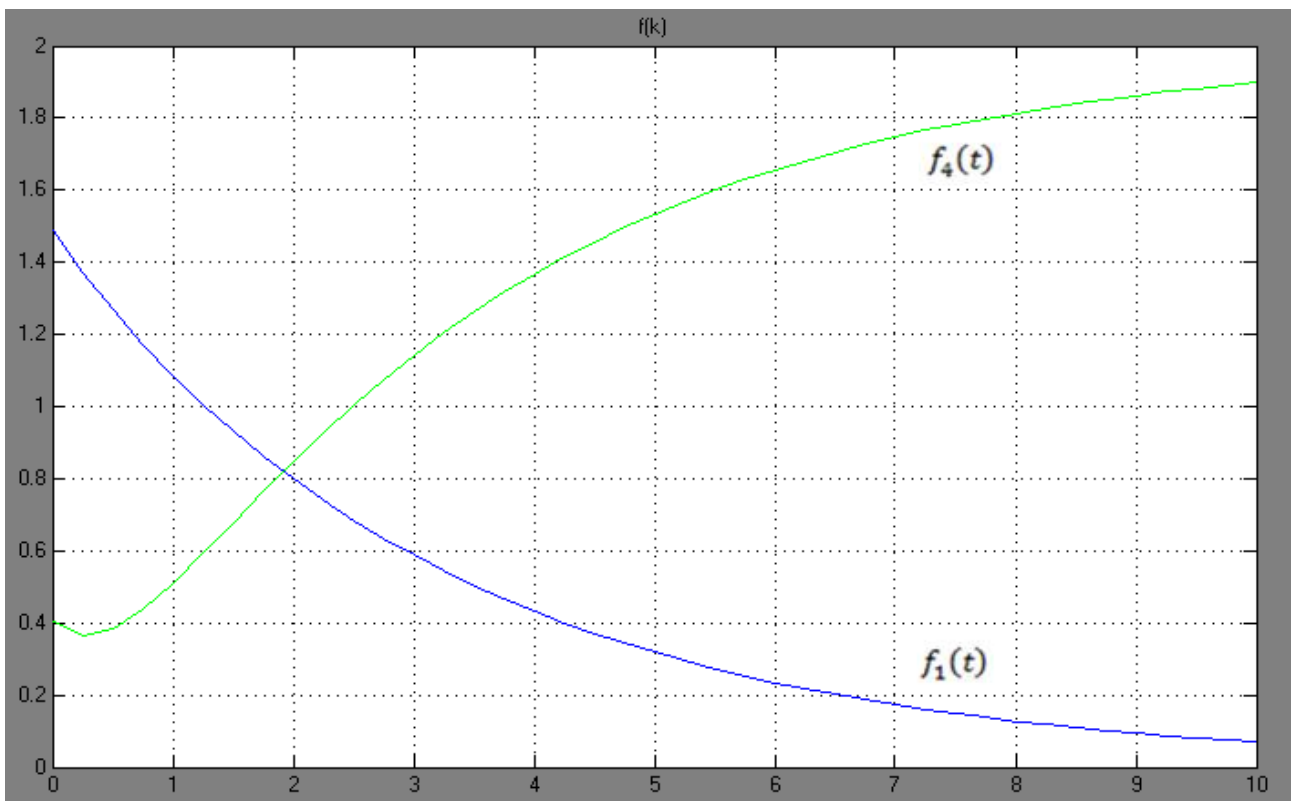


Figura 23: Flussi di ingresso, LQR, R pesato

Che è proprio il comportamento che ci aspettavamo. Verifichiamo a questo punto che anche per le uscite si raggiunga l'andamento desiderato:

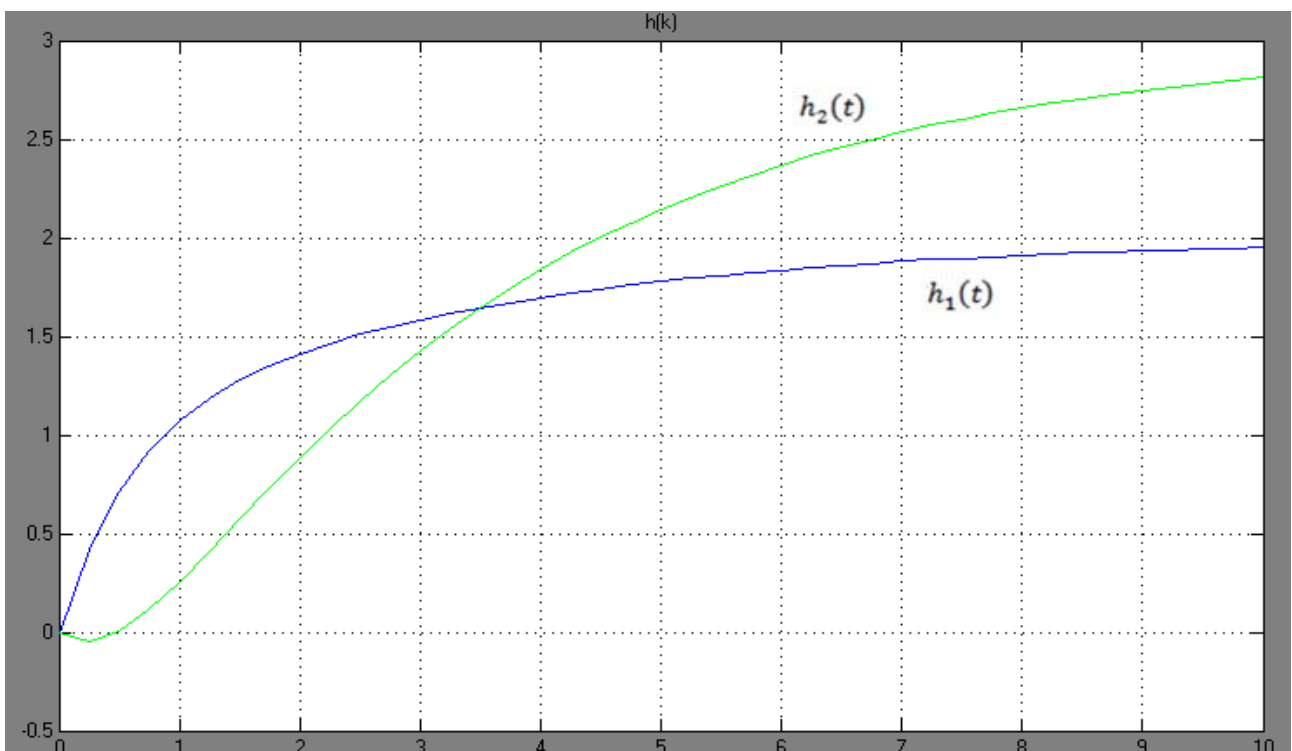


Figura 24: Altezze dei serbatoi, LQR, R pesato

Anche in questo caso la verifica ha esito positivo, infatti le velocità con cui salgono le altezze dei due serbatoi è rallentata notevolmente. In particolare il primo ha un tempo di salita di circa 4.5 secondi, mentre il secondo di 7.5 secondi, comunque in linea con i parametri progettuali.

In ogni caso semplicemente variando questo parametro nella matrice  $R$  è possibile aumentare o ridurre i tempi di salita delle altezze dei serbatoi; la soluzione riportata appare un buon compromesso.

## SEZIONE 6: DISACCOUPIOAMENTO DEL SISTEMA

Come è stato accennato nella sezione 1, per semplificare il controllo del sistema si può anteporre all'impianto un compensatore  $K^{pre}(s)$ , come in figura 25.

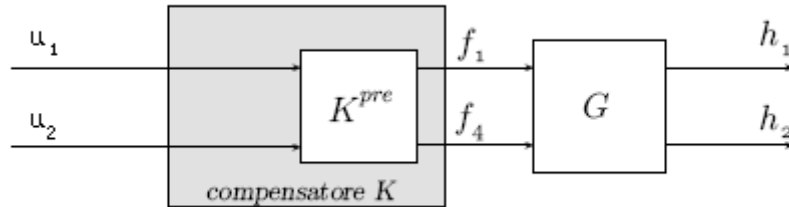


Figura 25: disaccoppiamento mediante compensatore

Il compensatore è un componente che viene collegato direttamente all'impianto (analogico), e quindi non viene realizzato digitalmente.

Il sistema da controllare sarà invece dato da:

$$P(s) = G(s)K^{pre}(s)$$

Ed è su questo sistema che si dovrà operare la discretizzazione per il successivo controllo.

L'obiettivo del compensatore (disaccoppiatore) è rendere la matrice delle funzioni di trasferimento  $P(s)$  diagonale, in modo tale che il sistema complessivo che prende in ingresso le  $u(t)$  (controllate) e restituisce in uscita le  $h(t)$  risulti divisibile in due sistemi SISO (Single Input Single Output) completamente indipendenti tra loro.

Per effettuare tale operazione si utilizza la tecnica del disaccoppiamento in avanti: si impone che all'interno del  $K^{pre}(s)$  non si abbiano retroazioni, pertanto la struttura interna del  $K^{pre}(s)$  è quella riportata in figura 26.

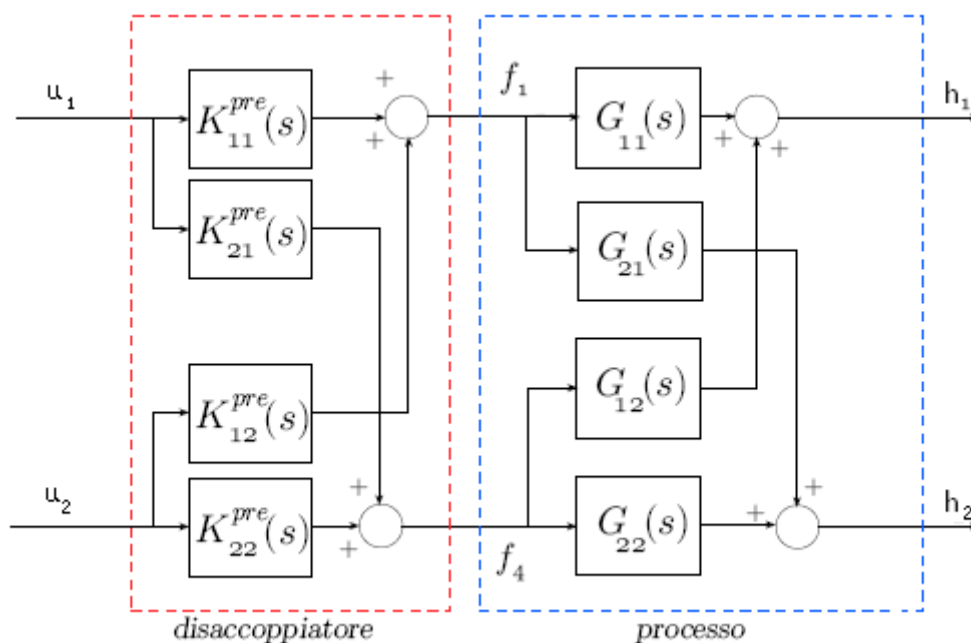


Figura 26: disaccoppiatore in avanti

Quindi si cerca una  $K^{pre}(s)$  tale la  $P(s)$  abbia una struttura diagonale del tipo:

$$P(s) = \begin{pmatrix} P_1(s) & 0 \\ 0 & P_2(s) \end{pmatrix}$$

In questo modo si ottiene, in riferimento alla figura 25:

$$\begin{cases} H_1(s) = P_1(s)U_1(s) \\ H_2(s) = P_2(s)U_2(s) \end{cases}$$

$P_1(s)$  e  $P_2(s)$  potranno allora essere analizzati separatamente e per ognuno di essi si prevederà un opportuno controllore, ottenendo la situazione seguente:

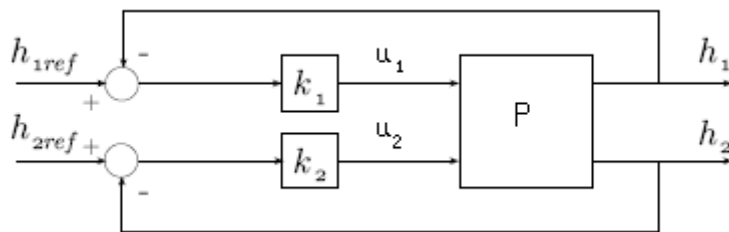


Figura 27: controllo mediante anelli separati

Stavolta notiamo infatti che non si hanno problemi di interazione tra i controlli sui due anelli, a differenza di quanto visto nella sezione 1.

Ricordiamo che la  $G(s)$  ricavata per il nostro impianto aveva la seguente forma:

$$G(s) = \begin{pmatrix} \frac{1}{s+1} & \frac{1}{s+1} \\ \frac{1}{s(s+1)} & -\frac{1}{s+1} \end{pmatrix} = \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix}$$

In Matlab le funzioni di trasferimento possono essere ricavate direttamente dalla rappresentazione già data nello spazio di stato mediante i comandi:

```
>> [NUM1, DEN1] = ss2tf(Ac, Bc, Cc, Dc, 1)
>> [NUM2, DEN2] = ss2tf(Ac, Bc, Cc, Dc, 2)
>> G = minreal(tf({NUM1(1, :), NUM2(1, :)}; NUM1(2, :), NUM2(2, :)};
    {DEN1, DEN2; DEN1, DEN2}))
```

Cercando la  $K^{pre}(s)$ , inizialmente non imponiamo alcun vincolo, quindi:

$$K^{pre}(s) = \begin{pmatrix} K_{11}^{pre}(s) & K_{12}^{pre}(s) \\ K_{21}^{pre}(s) & K_{22}^{pre}(s) \end{pmatrix}$$

Da cui:

$$\begin{aligned} P(s) = G(s)K^{pre}(s) &= \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix} \begin{pmatrix} K_{11}^{pre}(s) & K_{12}^{pre}(s) \\ K_{21}^{pre}(s) & K_{22}^{pre}(s) \end{pmatrix} = \\ &= \begin{pmatrix} G_{11}(s)K_{11}^{pre}(s) + G_{12}(s)K_{21}^{pre}(s) & G_{11}(s)K_{12}^{pre}(s) + G_{12}(s)K_{22}^{pre}(s) \\ G_{21}(s)K_{11}^{pre}(s) + G_{22}(s)K_{21}^{pre}(s) & G_{21}(s)K_{12}^{pre}(s) + G_{22}(s)K_{22}^{pre}(s) \end{pmatrix} \end{aligned}$$

Perché  $P(s)$  sia diagonale deve valere che:

$$\begin{cases} G_{11}(s)K_{12}^{pre}(s) + G_{12}(s)K_{22}^{pre}(s) = 0 \\ G_{21}(s)K_{11}^{pre}(s) + G_{22}(s)K_{21}^{pre}(s) = 0 \end{cases}$$

Ovviamente le soluzioni che soddisfano queste condizioni (visto che abbiamo 2 equazioni in 4 incognite) sono infinite, quindi si possono imporre delle nuove condizioni completamente arbitrarie in modo che sia comunque risolvibile, ottenendo le seguenti:

$$\begin{cases} K_{11}^{pre}(s) = K_{22}^{pre}(s) = 1 \\ K_{12}^{pre}(s) = -\frac{G_{12}(s)}{G_{11}(s)} \\ K_{21}^{pre}(s) = -\frac{G_{21}(s)}{G_{22}(s)} \end{cases}$$

Ricapitolando abbiamo:

$$K^{pre}(s) = \begin{pmatrix} 1 & -\frac{G_{12}(s)}{G_{11}(s)} \\ -\frac{G_{21}(s)}{G_{22}(s)} & 1 \end{pmatrix}$$

che in Matlab può essere scritto come:

```
>> Kpre = minreal([1, -G(1,2)/G(1,1); -G(2,1)/G(2,2), 1])
```

Notiamo che, siccome  $K^{pre}(s)$  non deve essere instabile per evitare problemi nella realizzazione, allora nel sistema in esame non devono esservi zeri a fase non minima o poli instabili, notiamo che per il nostro sistema tempo-continuo abbiamo:

- 2 poli per  $s = 0$  ed  $s = -1$
- Nessuno zero

Quindi le condizioni per il disaccoppiamento vengono rispettate.

Si ottiene allora che:

$$P(s) = \begin{pmatrix} G_{11}(s) - G_{12}(s)\frac{G_{21}(s)}{G_{22}(s)} & 0 \\ 0 & G_{22}(s) - G_{21}(s)\frac{G_{12}(s)}{G_{11}(s)} \end{pmatrix}$$

Ricordando adesso i valori per  $G_{11}(s)$ ,  $G_{12}(s)$ ,  $G_{21}(s)$  e  $G_{22}(s)$  possiamo finalmente calcolare:

$$K^{pre}(s) = \begin{pmatrix} 1 & -\frac{1}{s+1} \\ -\frac{1}{s(s+1)} & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ \frac{1}{s} & 1 \end{pmatrix}$$

$$P(s) = G(s)K^{pre}(s) = \begin{pmatrix} \frac{1}{s+1} - \frac{1}{s+1} \frac{1}{s(s+1)} & 0 \\ 0 & -\frac{1}{s+1} - \frac{1}{s(s+1)} \frac{1}{s+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{s} & 0 \\ 0 & -\frac{1}{s} \end{pmatrix}$$

Quindi nelle analisi a seguire piuttosto che controllare un sistema MIMO (Multiple Input Multiple Output, come è  $G(s)$ ) si controlla il sistema disaccoppiato  $P(s)$  mediante due anelli separati come indicato in figura 27, considerando separatamente due sistemi SISO del tipo:

$$\begin{cases} H_1(s) = P_1(s)U_1(s) = \frac{1}{s}U_1(s) \\ H_2(s) = P_2(s)U_2(s) = -\frac{1}{s}U_2(s) \end{cases}$$

nel dominio di Laplace, mentre nel tempo:

$$\begin{cases} \dot{h}_1(t) = u_1(t) \\ \dot{h}_2(t) = -u_2(t) \end{cases}$$

Possiamo ottenere anche in Matlab l'impianto in forma disaccoppiata come:

```
>> P = minreal(G*Kpre)
```

## SEZIONE 7: DISCRETIZZAZIONE DEL SISTEMA DISACCOPPIATO

Una volta disaccoppiato il sistema come descritto nella sezione 6, prima di vedere come controllarlo, è necessario discretizzare nuovamente il modello.

Per quanto riguarda la scelta del periodo di campionamento si faccia riferimento alla sezione 2, dove la conclusione cui siamo arrivati e che verrà utilizzata anche nel seguito è:

$$T = 0.25s$$

A questo punto piuttosto che trovare il modello discreto per il sistema globale, visto che quest'ultimo è stato disaccoppiato, si possono discretizzare separatamente i due comportamenti indipendenti per le due variabili di stato (e uscite):

$$\begin{cases} H_1(s) = P_1(s)U_1(s) = \frac{1}{s}U_1(s) \\ H_2(s) = P_2(s)U_2(s) = -\frac{1}{s}U_2(s) \end{cases} \Rightarrow \begin{cases} P_1(s) = \frac{1}{s} \\ P_2(s) = -\frac{1}{s} \end{cases}$$

Ricorriamo allora alla tecnica della invarianza al gradino (zero order hold, ZOH) per la discretizzazione esatta di sistemi continui mediante la sola funzione di trasferimento.

Il principio è che, considerando di controllare digitalmente il sistema e imponendo all'ingresso un gradino unitario discreto (usando come convertitore D/A per fornire gli ingressi all'impianto un circuito S&H, Sample & Hold), il gradino (frutto della conversione) visto dal sistema controllato è identico a quello che vedrebbe il sistema se controllato in analogico.

Questo deriva dal fatto che il mantenimento S&H della sequenza discreta "gradino unitario" è proprio il gradino unitario analogico perfetto. La situazione è mostrata in figura 28.

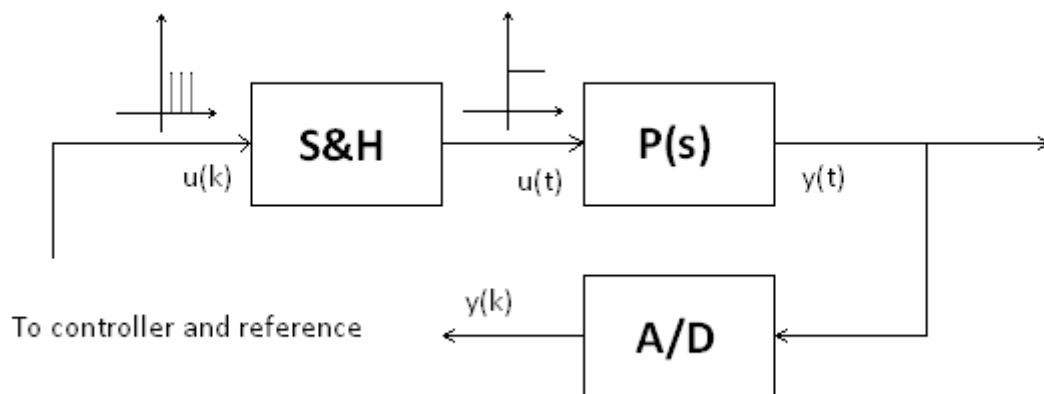


Figura 28: sistema discretizzato

Questo comporta allora che in uscita dall'impianto  $P_1(s)$  (o  $P_2(s)$ ) nel sistema a dati campionati avremo esattamente la stessa uscita che si avrebbe nel caso di controllo analogico.

Quindi la  $y(k)$ , essendo un campionamento della  $y(t)$ , risulterà identica al campionamento della risposta al gradino unitario nel caso continuo.

Dal momento che, esplicitando  $y(k)$ , si ha:

$$y(k) = y(kT)$$

allora possiamo imporre la condizione classica:

$$Y(z) = U(z)P(z)$$

Dove  $P(z)$  rappresenta l'incognita, ossia la forma discreta ricercata per l'impianto, da cui deriva che:

$$P(z) = \frac{Y(z)}{U(z)}$$

Noto allora l'andamento dell'uscita  $y(t)$  nel caso continuo, è sufficiente campionarla col periodo di campionamento  $T$  per trovare la sequenza corrispondente alla  $y(k)$  nel caso discreto (infatti pur dando un ingresso digitale  $u(k)$ , l'uscita non cambia perché il circuito di campionamento e tenuta produce lo stesso gradino unitario di un controllo puramente analogico).

Noto  $y(k)$  è allora sufficiente calcolare  $Y(z)$  per arrivare a  $P(z)$ , che è appunto la forma discreta cercata per il sistema.

Applichiamo adesso il procedimento sopra alle due parti separate e indipendenti tra loro:

- $P_1(s) = \frac{1}{s}$

$$Y(s) = \frac{1}{s}U(s) = \frac{1}{s^2}$$

$$Y(s) \xrightarrow{L^{-1}} y(t) = t$$

$$y(kT) = kT$$

$$Y(z) = \sum_{k=0}^{\infty} kTz^{-k} = T \frac{z^{-1}}{(1-z^{-1})^2}$$

$$P_1(z) = \frac{T \frac{z^{-1}}{(1-z^{-1})^2}}{\frac{1}{1-z^{-1}}} = \frac{Tz^{-1}}{1-z^{-1}} = \frac{T}{z-1}$$

- $P_2(s) = -\frac{1}{s}$

$$Y(s) = -\frac{1}{s}U(s) = -\frac{1}{s^2}$$

$$Y(s) \xrightarrow{L^{-1}} y(t) = -t$$

$$y(kT) = -kT$$

$$Y(z) = \sum_{k=0}^{\infty} -kTz^{-k} = -T \frac{z^{-1}}{(1-z^{-1})^2}$$

$$P_2(z) = \frac{-T \frac{z^{-1}}{(1-z^{-1})^2}}{\frac{1}{1-z^{-1}}} = -\frac{Tz^{-1}}{1-z^{-1}} = -\frac{T}{z-1}$$

I comandi Matlab che si possono dare per calcolare le funzioni di trasferimento di cui sopra sono i seguenti, dove l'impianto  $P$  era stato definito nella sezione precedente.

```
>> [Pd1num, Pd1den] = c2dm(P(1,1).num{1}, P(1,1).den{1}, sample_time)
>> [Pd2num, Pd2den] = c2dm(P(2,2).num{1}, P(2,2).den{1}, sample_time)
>> Pd1 = minreal(tf(Pd1num, Pd1den, sample_time))
>> Pd2 = minreal(tf(Pd2num, Pd2den, sample_time))
```

## SEZIONE 8: CONTROLLO ALGEBRICO

---

Trovate le forme discrete per le due funzioni di trasferimento, possiamo applicare ad ognuna delle due la tecnica del luogo reciproco della radici per conoscere la posizione ottima per i poli del sistema in ciclo chiuso.

Il Reciprocal Root Locus (RRL) è una tecnica che consente di determinare la posizione dei poli in ciclo chiuso che minimizzano l'indice di costo  $J$ , definito come segue:

$$J = \sum_{k=0}^{\infty} (y^2(k) + \rho u^2(k))$$

Dove  $\rho$  è un parametro arbitrario che ci consente di regolare il costo degli ingressi rispetto all'uscita, ovvero se è più importante tenere gli ingressi bassi o alzare la rapidità di risposta del sistema.

Notiamo che le uscite nel nostro caso corrispondono appunto alle altezze ( $y(k) = h(k)$ ) e che gli ingressi non corrispondono ai flussi visto che si sta operando sul sistema disaccoppiato, ma sono proprio le grandezze che abbiamo chiamato  $u(k)$ .

La somiglianza con l'indice  $J$  utilizzato nella trattazione del controllo secondo Riccati è evidente, in tal caso l'indice aveva la seguente forma (sostituendo a  $f(k)$  le  $u(k)$ ):

$$J_R = \frac{1}{2} \sum_{k=0}^N [h^T(k)Qh(k) + u^T(k)Ru(k)]$$

E le soluzioni ottime si potevano ottenere risolvendo:

$$\begin{bmatrix} h(k+1) \\ \lambda(k+1) \end{bmatrix} = \begin{bmatrix} A_d + B_d R^{-1} B_d^T A_d^{-T} Q & -B_d R^{-1} B_d^T A_d^{-T} \\ -A_d^{-T} Q & A_d^{-T} \end{bmatrix} \begin{bmatrix} h(k) \\ \lambda(k) \end{bmatrix}$$

Dove però adesso stiamo considerando un generico sistema nella forma:

$$\begin{cases} h(k+1) = A_d h(k) + B_d f(k) \\ y(k) = C h(k) \end{cases}$$

Con  $C = 1$  ed una unica variabile di stato  $h(k)$ .

Notiamo che tale equazione rappresenta anche la soluzione ottima per l'indice di costo:

$$J_R' = 2J_R = \sum_{k=0}^N [h^T(k)Qh(k) + u^T(k)Ru(k)]$$

L'indice  $J$  è allora semplicemente un caso di  $J_R'$  con sistema di tipo SISO e:

$$\begin{cases} Q = C^T C = 1 \\ R = \rho \end{cases}$$

Ne deriva allora che la soluzione ottima è sempre data dall'equazione Hamiltoniana, che riscritta con le sostituzioni di cui sopra assume la forma seguente:

$$\begin{bmatrix} h(k+1) \\ \lambda(k+1) \end{bmatrix} = \begin{bmatrix} A_d + B_d \frac{1}{\rho} B_d^T A_d^{-T} & -B_d \frac{1}{\rho} B_d^T A_d^{-T} \\ -A_d^{-T} & A_d^{-T} \end{bmatrix} \begin{bmatrix} h(k) \\ \lambda(k) \end{bmatrix}$$

Quindi, Z-trasformando l'equazione sopra, si ottiene:

$$\begin{bmatrix} zH(z) \\ z\lambda(z) \end{bmatrix} = \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} \begin{bmatrix} H(z) \\ \lambda(z) \end{bmatrix} = zI \begin{bmatrix} H(z) \\ \lambda(z) \end{bmatrix} = \begin{bmatrix} A_d + B_d \frac{1}{\rho} B_d^T A_d^{-T} & -B_d \frac{1}{\rho} B_d^T A_d^{-T} \\ -A_d^{-T} & A_d^{-T} \end{bmatrix} \begin{bmatrix} H(z) \\ \lambda(z) \end{bmatrix}$$

La quale definisce la posizione ottima delle  $z$ , e si può riscrivere come:

$$\left\{ zI - \begin{bmatrix} A_d + B_d \frac{1}{\rho} B_d^T A_d^{-T} & -B_d \frac{1}{\rho} B_d^T A_d^{-T} \\ -A_d^{-T} & A_d^{-T} \end{bmatrix} \right\} \begin{bmatrix} H(z) \\ \lambda(z) \end{bmatrix} = (zI - H_c) \begin{bmatrix} H(z) \\ \lambda(z) \end{bmatrix} = 0$$

Affinché l'equazione sopra ammetta una soluzione diversa da  $\begin{bmatrix} H(z) \\ \lambda(z) \end{bmatrix} = 0$  è necessario che il nucleo di  $(zI - H_c)$  contenga valori diversi da 0, il che implica allora che il nucleo deve avere dimensione maggiore di zero, quindi il rango di  $(zI - H_c)$  deve essere minore di  $2n$  e quindi deve valere:

$$\det \left\{ zI - \begin{bmatrix} A_d + B_d \frac{1}{\rho} B_d^T A_d^{-T} & -B_d \frac{1}{\rho} B_d^T A_d^{-T} \\ -A_d^{-T} & A_d^{-T} \end{bmatrix} \right\} = 0$$

che è una equazione che ci fornisce le  $z$  ottime cercate.

Possiamo allora semplificare come segue (facendo uso delle combinazioni proprie della eliminazione di Gauss)

$$\begin{aligned} \det \begin{bmatrix} zI - A_d - B_d \frac{1}{\rho} B_d^T A_d^{-T} & B_d \frac{1}{\rho} B_d^T A_d^{-T} \\ A_d^{-T} & zI - A_d^{-T} \end{bmatrix} &= \det \begin{bmatrix} zI - A_d & B_d \frac{1}{\rho} B_d^T A_d^{-T} \\ zI & zI - A_d^{-T} \end{bmatrix} = \\ &= \det \begin{bmatrix} zI - A_d & B_d \frac{1}{\rho} B_d^T A_d^{-T} \\ 0 & -zI(zI - A_d)^{-1} B_d \frac{1}{\rho} B_d^T A_d^{-T} + zI - A_d^{-T} \end{bmatrix} = \\ &= \det(zI - A_d) \det \left( -zI(zI - A_d)^{-1} B_d \frac{1}{\rho} B_d^T A_d^{-T} + zI - A_d^{-T} \right) = 0 \end{aligned}$$

Allora possiamo continuare, come segue, nelle semplificazioni:

$$\det(zI - A_d) \det \left( -z(zI - A_d)^{-1} B_d \frac{1}{\rho} B_d^T A_d^{-T} + zI - A_d^{-T} \right) = 0$$

$$\begin{aligned}
& \det(zI - A_d) \det \left( (zI - A_d)^{-1} \left( -zB_d \frac{1}{\rho} B_d^T A_d^{-T} + (zI - A_d)(zI - A_d^{-T}) \right) \right) = 0 \\
& \det(zI - A_d) \det((zI - A_d)^{-1}) \det \left( -zB_d \frac{1}{\rho} B_d^T A_d^{-T} + (zI - A_d)(zI - A_d^{-T}) \right) = 0 \\
& \det \left( -zB_d \frac{1}{\rho} B_d^T A_d^{-T} + (zI - A_d)(zI - A_d^{-T}) \right) = 0 \\
& \det(zI - A_d)^{-1} \det \left( -zB_d \frac{1}{\rho} B_d^T A_d^{-T} + (zI - A_d)(zI - A_d^{-T}) \right) \det(zI - A_d^{-T})^{-1} = 0 \\
& \det \left( (zI - A_d)^{-1} \left( -zB_d \frac{1}{\rho} B_d^T A_d^{-T} \right) (zI - A_d^{-T})^{-1} + I \right) = 0 \\
& \det \left( -z \frac{1}{\rho} (zI - A_d)^{-1} B_d B_d^T A_d^{-T} (zI - A_d^{-T})^{-1} + I \right) = 0 \\
& \det \left( \frac{1}{\rho} (zI - A_d)^{-1} B_d B_d^T (-zA_d^{-T}) (zI - A_d^{-T})^{-1} + I \right) = 0 \\
& \det \left( \frac{1}{\rho} (zI - A_d)^{-1} B_d B_d^T (-z^{-1}A_d^T)^{-1} (zI - A_d^{-T})^{-1} + I \right) = 0 \\
& \det \left( \frac{1}{\rho} (zI - A_d)^{-1} B_d B_d^T \left( (zI - A_d^{-T})(-z^{-1}A_d^T) \right)^{-1} + I \right) = 0 \\
& \det \left( \frac{1}{\rho} (zI - A_d)^{-1} B_d B_d^T (z^{-1}I - A_d^T)^{-1} + I \right) = 0 \\
& \det \left( I + \frac{1}{\rho} (zI - A_d)^{-1} B_d B_d^T (z^{-1}I - A_d^T)^{-1} \right) = 0
\end{aligned}$$

Notiamo ora che, nel nostro caso:

$$(zI - A_d)^{-1} B_d = G(z)$$

Ed inoltre:

$$\left( B_d^T (z^{-1}I - A_d^T)^{-1} \right)^T = (z^{-1}I - A_d)^{-1} B_d = G\left(\frac{1}{z}\right)$$

Dal momento che stiamo parlando di sistemi SISO dovrà tuttavia valere:

$$G\left(\frac{1}{z}\right) = G\left(\frac{1}{z}\right)^T = B_d^T (z^{-1}I - A_d^T)^{-1}$$

Allora l'equazione da risolvere diventa:

$$\det \left( 1 + \frac{1}{\rho} G(z) G\left(\frac{1}{z}\right) \right) = 1 + \frac{1}{\rho} G(z) G\left(\frac{1}{z}\right) = 0$$

Le  $z$  che la soddisfano ci consentono di ricavare appunto la posizione desiderata per i poli in ciclo chiuso, che dipenderà dal valore di  $\rho$  scelto.

L'equazione di cui sopra è simile alla seguente:

$$1 + kG'(z) = 0$$

che è appunto l'equazione che descrive l'andamento dei poli in ciclo chiuso del sistema  $G'(z)$  reazionato con un controllore proporzionale che introduce il guadagno  $k$ , e può essere analizzata tracciando il classico luogo delle radici.

Possiamo allora tracciare nuovamente il luogo delle radici, imponendo la condizione:

$$G'(z) = G(z)G\left(\frac{1}{z}\right)$$

Questa porta a tracciare quello che viene chiamato Luogo Reciproco delle Radici (Reciprocal Root Locus, RRL) di  $G(z)$  per ognuno dei nostri sistemi SISO:

$$\begin{cases} G(z) = P_1(z) \\ G(z) = P_2(z) \end{cases}$$

Notiamo inoltre che le soluzioni della equazione:

$$1 + \frac{1}{\rho} G(z)G\left(\frac{1}{z}\right) = 0$$

sono in numero pari a  $2n$  (dove  $n$  è l'ordine del sistema  $G(z)$ ); di queste  $2n$  soluzioni, le  $n$  che descrivono la posizione desiderata dei poli in ciclo chiuso sono solo le  $n$  che si trovano all'interno del cerchio unitario, ossia in zona di stabilità.

Le altre  $n$  soluzioni costituiscono le posizioni simmetriche rispetto alla circonferenza unitaria, che non sono di interesse.

Andiamo ora ad applicare questa tecnica relativamente all'impianto. Si ha che:

$$\begin{cases} P_1(z) = \frac{T}{z-1} = Q(z) \\ P_2(z) = -\frac{T}{z-1} = -Q(z) \end{cases}$$

Ne deriva che:

$$\begin{aligned} 1 + \frac{1}{\rho} P_1(z)P_1\left(\frac{1}{z}\right) &= 1 + \frac{1}{\rho} Q(z)Q\left(\frac{1}{z}\right) = 0 \\ 1 + \frac{1}{\rho} P_2(z)P_2\left(\frac{1}{z}\right) &= 1 + \frac{1}{\rho} (-Q(z))\left(-Q\left(\frac{1}{z}\right)\right) = 1 + \frac{1}{\rho} Q(z)Q\left(\frac{1}{z}\right) = 0 \end{aligned}$$

È quindi evidente che i due luoghi delle radici saranno esattamente identici.

In Matlab è possibile determinare le funzioni di trasferimento di  $P_1\left(\frac{1}{z}\right)$  e  $P_2\left(\frac{1}{z}\right)$  mediante i seguenti comandi:

```
>> z = tf('z', sample_time);
>> Pd1i = Pd1num(2) / (Pd1den(1) * z^(-1) + Pd1den(2))
>> Pd2i = Pd2num(2) / (Pd2den(1) * z^(-1) + Pd2den(2))
```

Il luogo reciproco delle radici può a questo punto essere tracciato, al variare di  $\frac{1}{\rho}$ , digitando:

```
>> sisotool (Pd1*Pd1i)
```

In figura 29 è evidenziato l'output del comando nel caso in cui venga scelto  $\rho = 1$ .

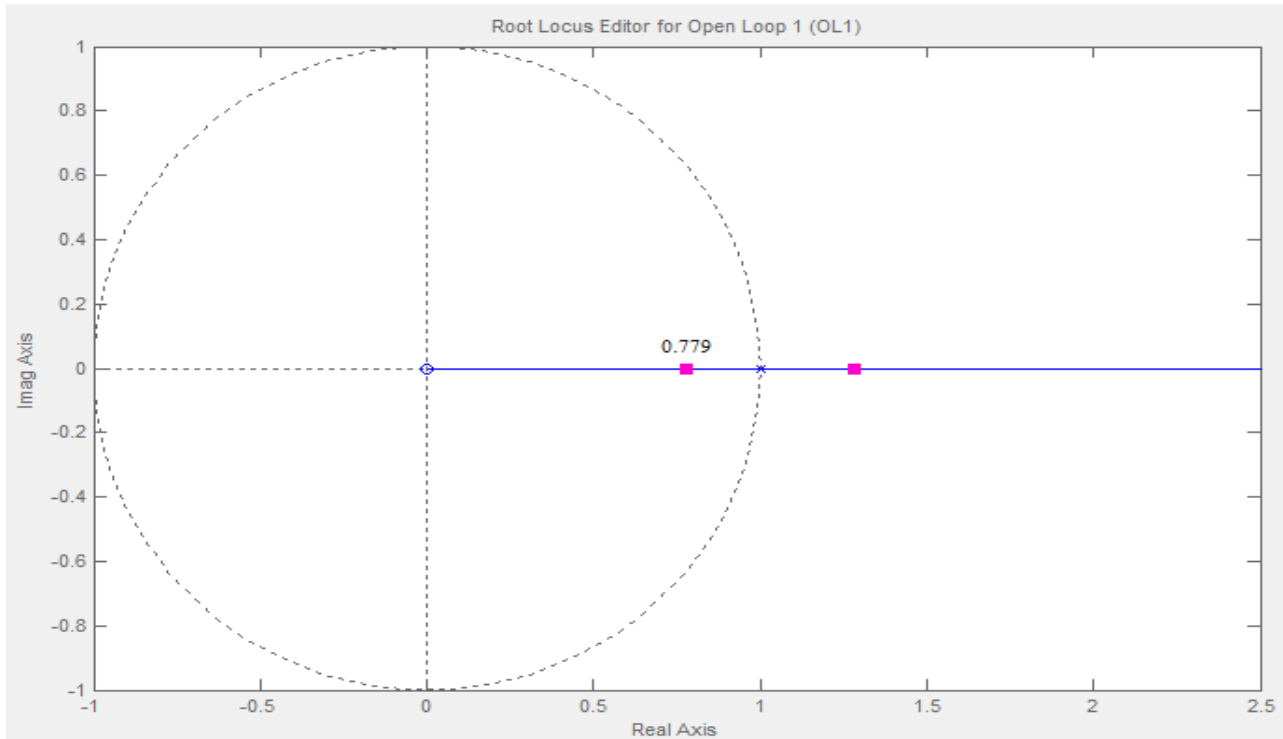


Figura 29: Luogo Reciproco delle Radici per  $P_1(z)$  e  $P_2(z)$  con posizione delle soluzioni se  $\rho = 1$

Ne deriva allora che il polo in ciclo chiuso, sia per  $P_1(z)$  che per  $P_2(z)$  deve essere portato nel punto  $z = 0.779$ . In Matlab è possibile costruire il seguente script che, in base ai valori di  $\rho$  scelti per ogni impianto, determina automaticamente la posizione ottima per i poli, risolvendo le due equazioni sopra descritte.

```
>> locus1 = zero(1+1/roh1*Pd1*Pd1i);
>> locus2 = zero(1+1/roh2*Pd2*Pd2i);
>> for i=1:2 % Poli ottimi (stabili)
>>     if locus1(i) < 1, p1 = locus1(i), end
>>     if locus2(i) < 1, p2 = locus2(i), end
>> end
```

Nota allora la posizione del polo vediamo cosa significa controllare in modo algebrico.

Ora che il sistema è stato disaccoppiato e si è deciso di controllare con due anelli separati, si capisce che il singolo  $P_i(z)$  (con  $i = 1,2$ ) può essere controllato semplicemente con lo schema in figura 30.

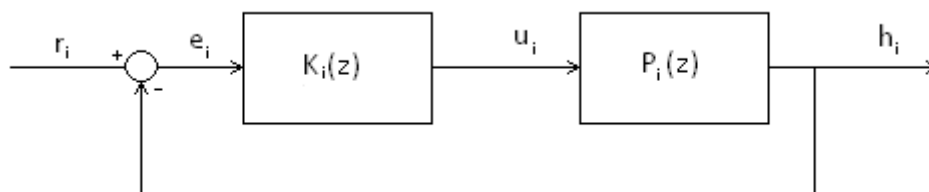


Figura 30: controllo equivalente per  $P_i(z)$

Dallo schema sopra, si deduce che:

$$H_i(z) = P_i(z)K_i(z)(R_i(z) - H_i(z))$$

Da cui:

$$H_i(z) = \frac{P_i(z)K_i(z)}{1 + P_i(z)K_i(z)}R_i(z)$$

Allora la seguente funzione di trasferimento:

$$W_i(z) = \frac{P_i(z)K_i(z)}{1 + P_i(z)K_i(z)}$$

Costituisce la funzione di trasferimento in ciclo chiuso per ognuno dei sottosistemi indipendenti.

Quello che si vuole fare è rendere tale funzione di trasferimento uguale ad un modello di comportamento desiderato per il sistema in ciclo chiuso, scelto a priori, che chiamiamo  $M_i(z)$ .

Allora il controllore cercato si può ricavare dalla seguente equazione:

$$\frac{P_i(z)K_i(z)}{1 + P_i(z)K_i(z)} = M_i(z)$$

Infatti:

$$\begin{aligned} P_i(z)K_i(z) &= M_i(z)(1 + P_i(z)K_i(z)) \\ P_i(z)K_i(z)(1 - M_i(z)) &= M_i(z) \end{aligned}$$

E allora:

$$K_i(z) = \frac{1}{P_i(z)} \frac{M_i(z)}{1 - M_i(z)}$$

Notiamo che in pratica non è possibile scegliere un modello  $M_i(z)$  completamente arbitrario, altrimenti  $K_i(z)$ , avendo al suo interno un termine  $\frac{1}{P_i(z)}$  andrebbe a cancellare perfettamente ogni componente di  $P_i(z)$ .

Siccome la cancellazione perfetta nella pratica non esiste, potrebbero sorgere problemi nel caso in cui si cerchi di cancellare poli instabili o zeri a fase non minima, oppure nel caso si cercasse di eliminare dei ritardi propri del sistema (rendendo il controllore non fisicamente realizzabile).

Ne deriva che:

- Se il sistema  $P_i(z)$  contiene dei poli instabili, questi non devono comparire al numeratore della  $K_i(z)$  e quindi devono comparire in  $(1 - M_i(z))$
- Se il sistema  $P_i(z)$  contiene degli zeri a fase non minima, questi non devono comparire al denominatore della  $K_i(z)$  e quindi devono comparire in  $M_i(z)$

Seguendo le prime due regole,  $K_i(z)$  non andrà mai a cancellare poli instabili o zeri a fase non minima dell'impianto.

Inoltre:

- Se il sistema ha dei ritardi del tipo  $z^{-M}$  questi non possono venire annullati dal controllore  $K_i(z)$ , il quale non dovrà contenere termini del tipo  $z^M$  per la fisica realizzabilità, quindi i termini del tipo  $z^{-M}$  devono comparire in  $M_i(z)$
- Richiedendo che il sistema risponda al gradino con un gradino, si richiede (per il principio del modello interno) che in ciclo aperto la  $P_i(z)K_i(z)$  sia almeno di tipo 1

Nei casi di nostro interesse abbiamo che:

- Il sistema  $P_i(z)$  non contiene dei poli instabili
- Il sistema  $P_i(z)$  non contiene zeri
- Il sistema  $P_i(z)$  non manifesta ritardi
- Il sistema  $P_i(z)$  è già di tipo 1 (sia per  $i = 1$  che per  $i = 2$ )

Allora il modello  $M_i(z)$  nel nostro caso non è soggetto ad ulteriori vincoli.

Avendo trovato la posizione ottima dei poli in ciclo chiuso, possiamo allora scegliere una  $M_i(z)$  semplice, che abbia solo un polo nella posizione desiderata; bisogna inoltre fare in modo che non introduca un guadagno all'infinito (per raggiungere il corretto valore di regime).

Il termine  $A$  serve appunto per evitare che quando si raggiunge la stabilità il valore di regime sia errato. Allora  $M_i(z)$  sarà la seguente:

$$M_i(z) = \frac{A}{z - p_i}$$

Per determinare il corretto valore di  $A$  utilizziamo il teorema del valore finale:

$$\lim_{k \rightarrow \infty} h_i(k) = \lim_{z \rightarrow 1} (z - 1) H_i(z) = \lim_{z \rightarrow 1} (z - 1) M_i(z) \frac{z}{z - 1} = \lim_{z \rightarrow 1} (z - 1) \frac{A}{z - p_i} \frac{z}{z - 1} = \frac{A}{1 - p_i}$$

Allora perché la risposta al gradino unitario si stabilizzi su  $h_i = 1$  è necessario imporre  $A = 1 - p_i$ . Il modello può a questo punto essere descritto in Matlab come segue:

```
>> M1 = (1-p1) / (z-p1)
>> M2 = (1-p2) / (z-p2)
```

Poniamo adesso di scegliere effettivamente, sia per  $P_1(z)$  che per  $P_2(z)$ ,  $\rho = 1$ .

In questo caso abbiamo:

$$M_i(z) = \frac{1 - p_i}{z - p_i} = \frac{0.221}{z - 0.779} = M(z)$$

Per  $i = 1, 2$ .

Da cui si possono derivare i controllori:

$$\begin{cases} K_1(z) = \frac{1}{P_1(z)} \frac{M(z)}{1 - M(z)} = 0.8828 \\ K_2(z) = \frac{1}{P_2(z)} \frac{M(z)}{1 - M(z)} = -0.8828 \end{cases}$$

ottenuti eseguendo in Matlab i comandi:

```
>> K1 = minreal(1/Pd1*M1/(1-M1))
>> K2 = minreal(1/Pd2*M2/(1-M2))
```

Gli andamenti per le altezze nei serbatoi e per gli ingressi al sistema disaccoppiato (sempre in risposta al riferimento  $r = (2, 3)$ ) sono quelli riportati in figura 31 e 32, rispettivamente.

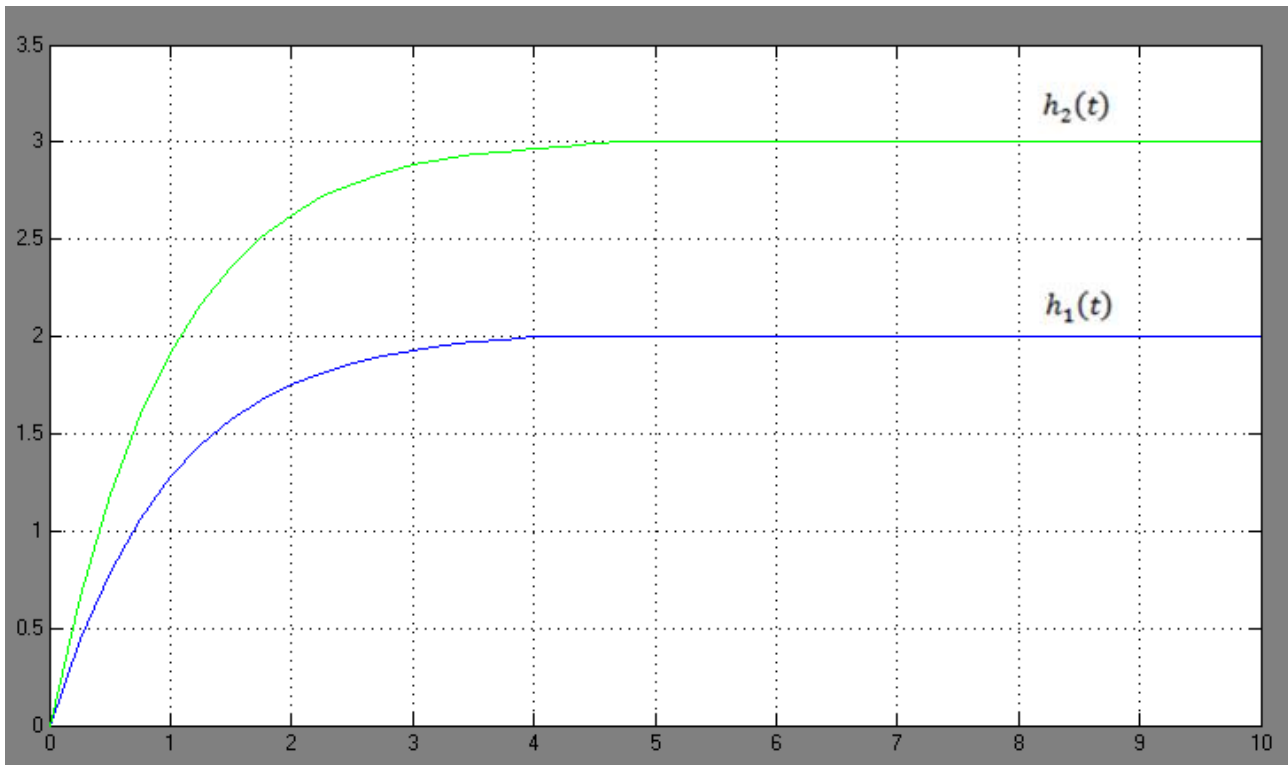


Figura 31: andamenti per le uscite,  $\rho = 1$

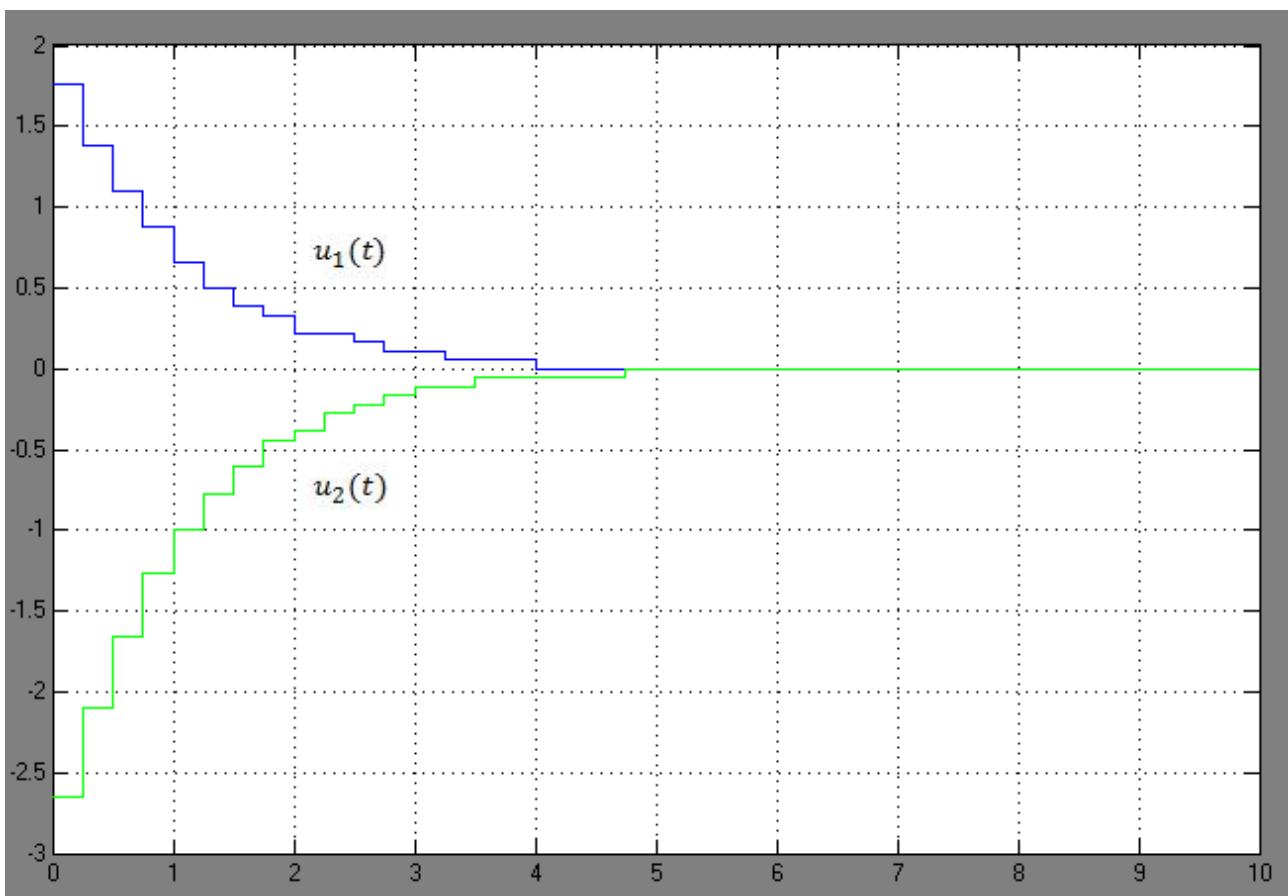


Figura 32: andamenti per gli ingressi,  $\rho = 1$

Notiamo allora alcune cose:

- Il tempo di salita per le uscite risulta troppo breve per il periodo di campionamento scelto
- L'andamento degli ingressi  $u_i(t)$  in questo caso non è vincolato dai problemi meccanici che avevamo per i flussi in quanto possono anche essere pensati come intensità di correnti continue, o qualsiasi altra grandezza continua che comunque tipicamente verrà scelta in modo da non avere forti vincoli
- Per la conversione da digitale ad analogico abbiamo ipotizzato di utilizzare un semplice convertitore che non fa interpolazione, ma semplicemente mantiene in uscita i valori ricevuti convertiti fino a quando non cambiano gli ingressi

Bisogna anche visualizzare l'andamento per i flussi effettivi in ingresso al sistema, per farlo si fa riferimento allo schema Simulink in figura 33.

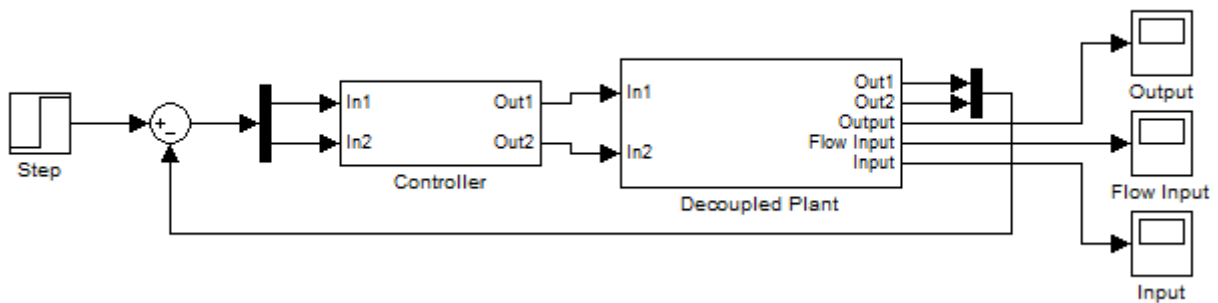


Figura 33: Schema per il sistema disaccoppiato controllato

Il contenuto del blocco 'Controller' è mostrato in figura 34

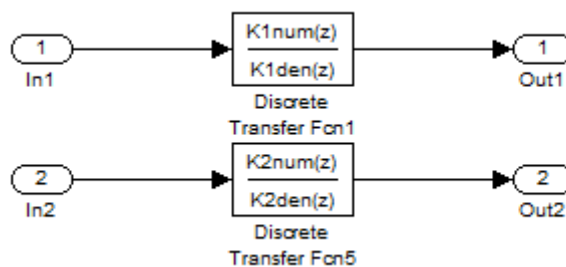


Figura 34: Contenuto del blocco 'Controller'

Il contenuto del blocco 'Decoupled Plant' è mostrato in figura 35

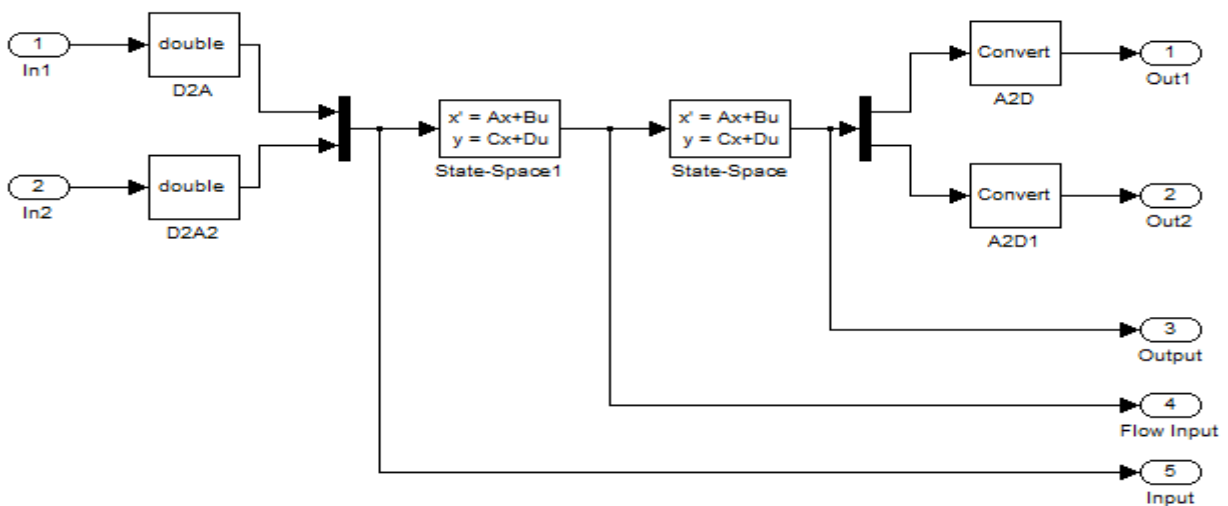


Figura 35: Contenuto del blocco 'Decoupled Plant'

L'andamento dei flussi è riportato in figura 36.

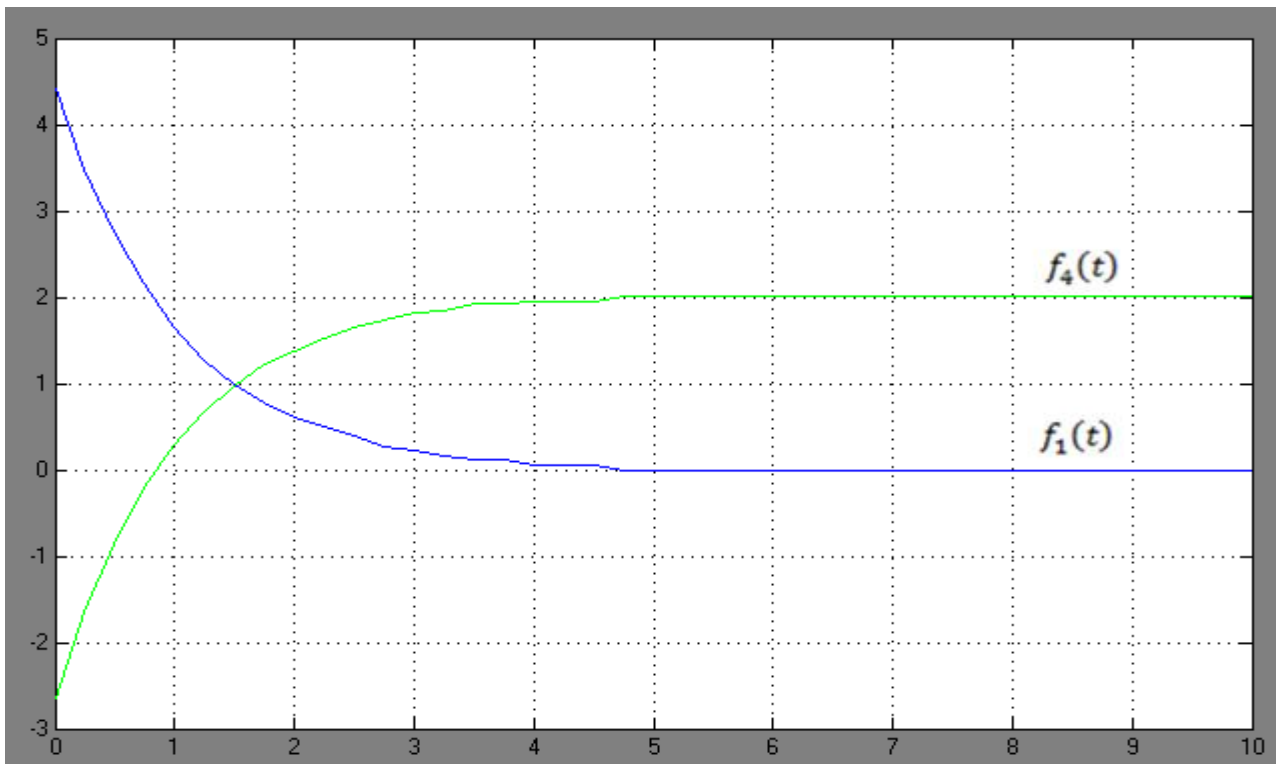


Figura 36: andamento dei flussi in ingresso all'impianto (accoppiato),  $\rho = 1$

Notiamo allora che:

- Anche in questo caso il flusso  $f_4(t)$  assume valori inferiori a -1, i quali però non sono fisicamente ottenibili dalla pompa di ricircolo.

Si può allora tentare di modificare la situazione cambiando i valori per  $\rho$ .

Non è immediato capire quale effetto abbia una modifica per la  $\rho$  sull'andamento dei flussi, mediante simulazione però si può cercare di aumentare  $\rho$  per aumentare il peso dato agli ingressi  $u(k)$ , in questo modo essi tenderanno ad avere una evoluzione più lenta, rallentando anche le uscite.

Prendiamo adesso in esame i singoli casi.

### $P_1(z)$

Ponendo  $\rho_1 = 6$  (ossia il  $\rho$  utilizzato per il calcolo della posizione ottima dei poli in ciclo chiuso per il sistema  $P_1(z)$ ), si ottiene il RRL mostrato in figura 37.

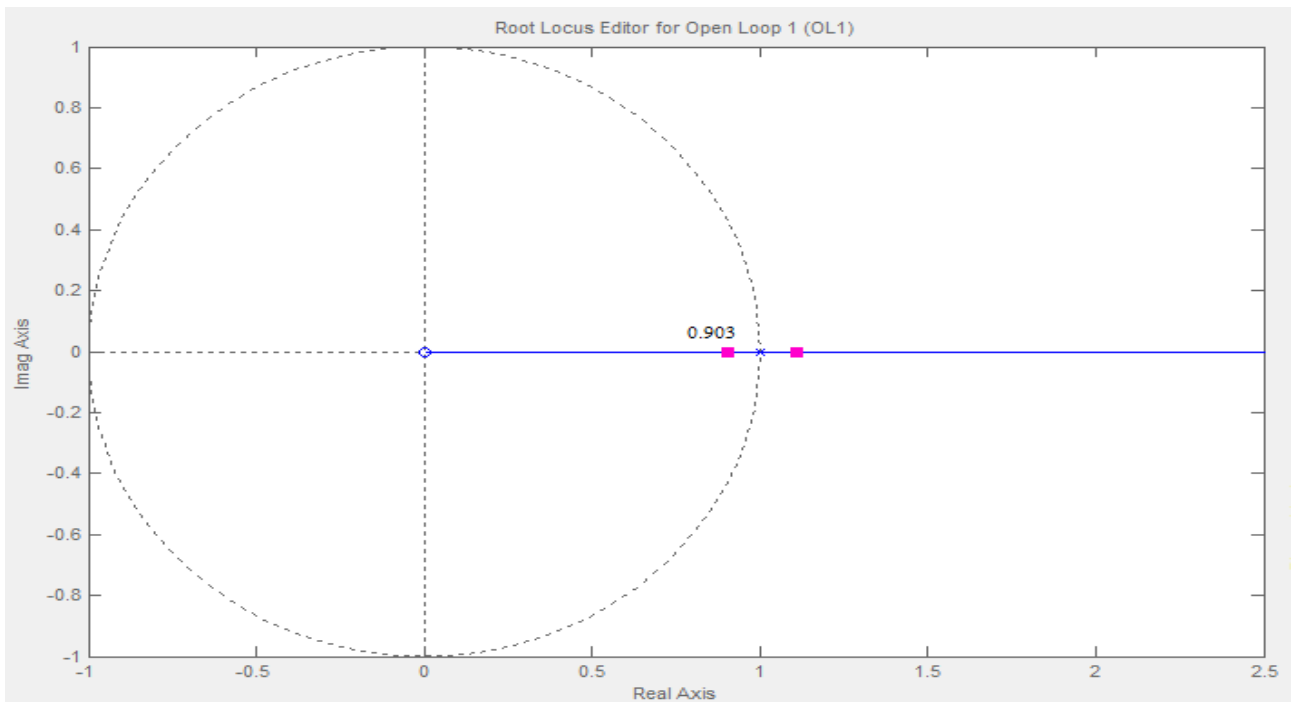


Figura 37: RRL relativo a  $P_1(z)$ , con posizione del polo per  $\rho = 6$

### $P_2(z)$

Ponendo  $\rho_2 = 10$  (ossia il  $\rho$  utilizzato per il calcolo della posizione ottima dei poli in ciclo chiuso per il sistema  $P_2(z)$ ), si ottiene il RRL mostrato in figura 38.

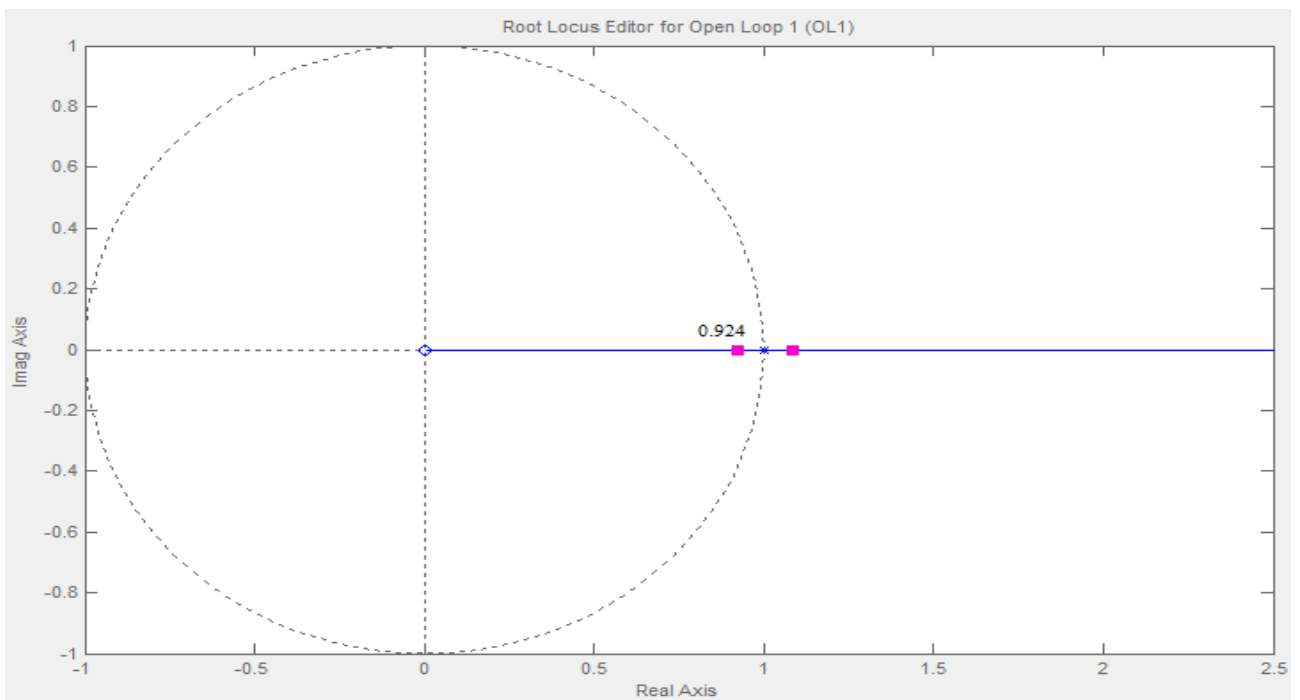


Figura 38: RRL relativo a  $P_2(z)$ , con posizione del polo per  $\rho = 10$

Con osservazioni simili a quelle viste sopra per  $\rho = 1$ , si possono calcolare come al solito i modelli desiderati nei due casi, che sono i seguenti:

$$\begin{cases} M_1(z) = \frac{0.097}{z - 0.903} \\ M_2(z) = \frac{0.076}{z - 0.924} \end{cases}$$

E da questi si ricavano come sopra i controllori finali:

$$\begin{cases} K_1(z) = 0.3879 \\ K_2(z) = -0.304 \end{cases}$$

Osserviamo allora, sempre con riferimento alle figure 33, 34 e 35 e in risposta al solito riferimento  $r = (2, 3)$ , gli andamenti per le uscite riportati in figura 39.

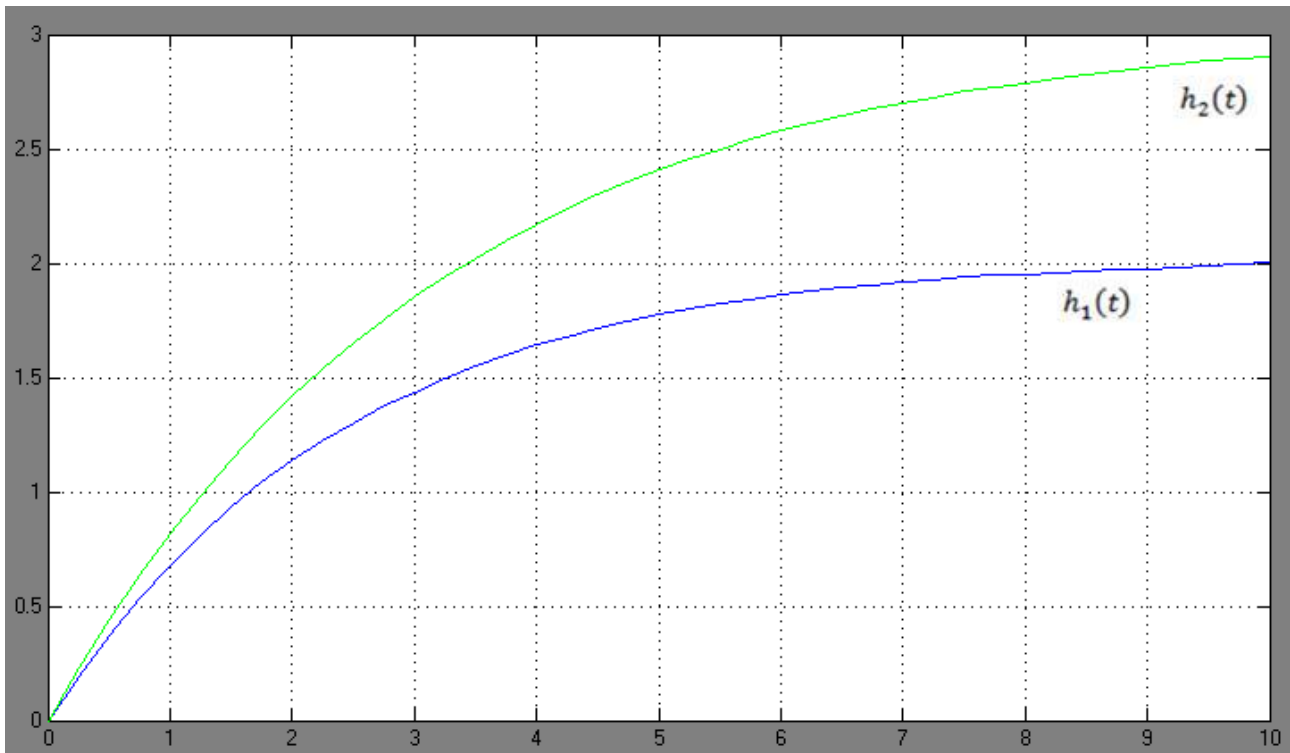


Figura 39: andamenti per le uscite,  $\rho_1 = 6$  e  $\rho_2 = 10$

In cui possiamo osservare una crescita più graduale rispetto agli andamenti in figura 31.

Gli andamenti per gli ingressi del sistema disaccoppiato (gli  $u_i(t)$ ) sono riportati in figura 40.

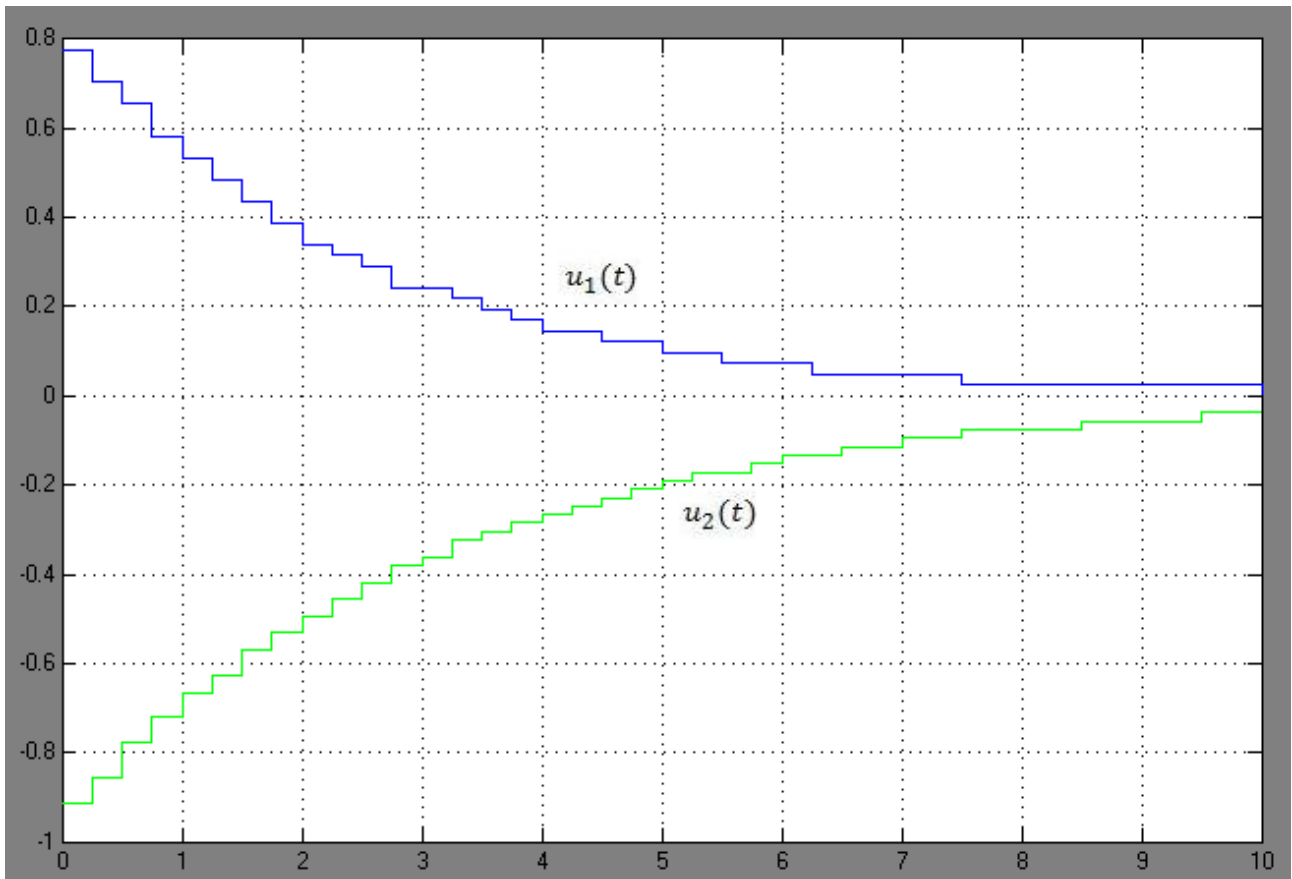


Figura 40: andamenti per gli ingressi,  $\rho_1 = 6$  e  $\rho_2 = 10$

I nuovi andamenti per i flussi effettivi sono invece riportati in figura 41.

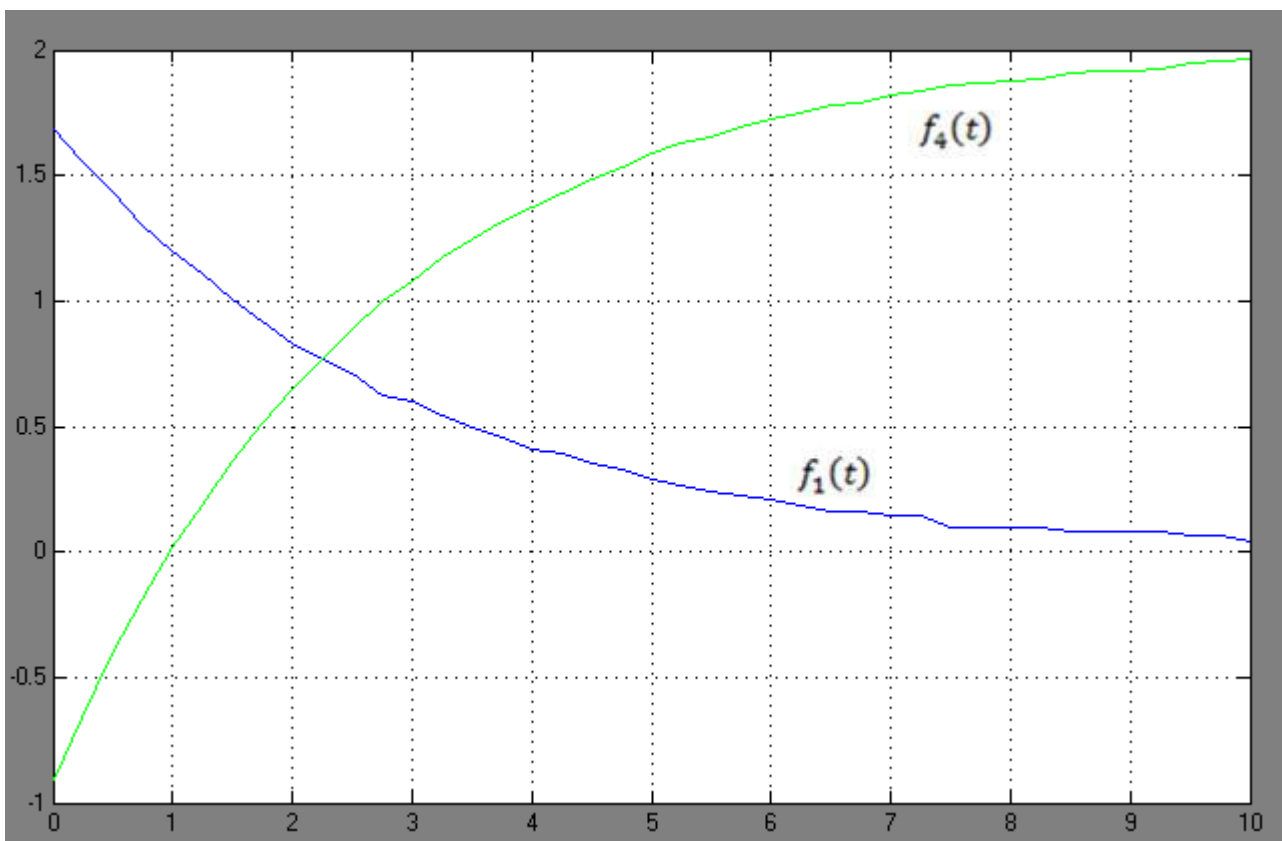


Figura 41: andamento dei flussi in ingresso all'impianto (accoppiato),  $\rho_1 = 6$  e  $\rho_2 = 10$

Stavolta osserviamo che il problema sperimentato dal flusso  $f_4(t)$  viene risolto in quanto non si scende mai sotto il -1.

Notiamo che al solito però questa soluzione non ha valore generale, infatti se si cambiano i riferimenti si può ripresentare il problema dei flussi negativi e in modulo superiore ad 1.

Fare riferimento alle sezione 10 per una possibile soluzione generale a questo problema, che però porta ad alcuni inconvenienti.

## SEZIONE 9: CONTROLLO PROPORZIONALE

Dato l'impianto disaccoppiato e discretizzato, questo può essere controllato agevolmente con un semplice controllore proporzionale. Consideriamo difatti un possibile schema di controllo in ciclo chiuso mediante anelli separati:

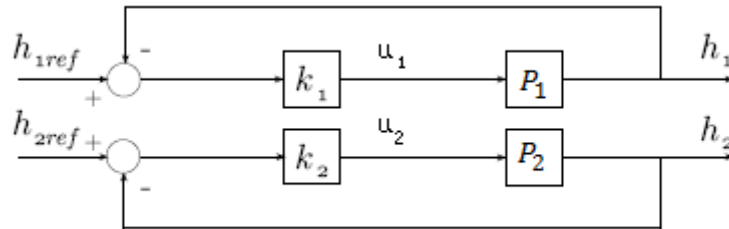


Figura 42: Controllo P su anelli separati

Dal momento che:

$$P_1(z) = \frac{T}{z-1}$$

Andiamo a tracciare il luogo delle radici relativo al primo impianto. L'evoluzione è riportata al variare di  $K_1 > 0$ :

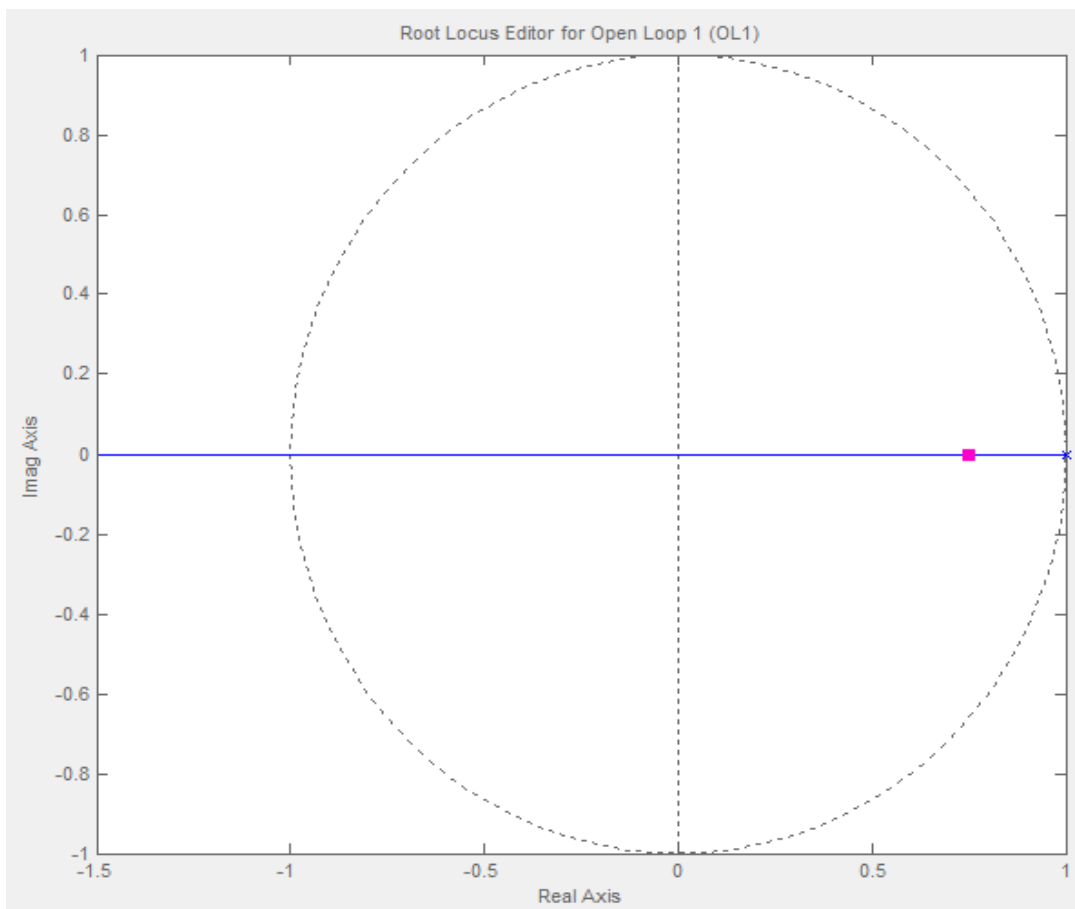


Figura 43: Luogo delle radici P1

Si può notare immediatamente che se la costante di guadagno assumesse valori negativi, allora il sistema risulterebbe instabile. Viceversa per  $K_1$  positivi l'instabilità viene raggiunta solo se  $K_1 > 8$ . L'analisi può essere effettuata in modo semplice mediante Matlab usando il comando:

```
>> sisotool(Pd1)
```

Relativamente al secondo impianto, essendo  $P_1(z) = -P_2(z)$ , allora l'andamento del luogo è lo stesso se considero  $K_2 < 0$ , mentre risulta instabile per altri valori di  $K_2$ .

Dal momento che tutti i valori:

$$0 < K_1 < 8, -8 < K_2 < 0$$

sono ammissibili, cioè rendono il sistema stabile, vanno determinati quelli che meglio gli si adattano sfruttando qualche altra caratteristica. In questo caso prendiamo in considerazione il tempo di salita  $T_s$ , che useremo come parametro di progetto imponendo  $T_s \cong 5s$ .

I conti che faremo in seguito si riferiscono al primo sistema, ricordando che per il secondo valgono analoghe considerazioni.

La funzione di trasferimento in ciclo chiuso del sistema risulta:

$$G(z) = \frac{K_1 P_1(z)}{1 + K_1 P_1(z)} = \frac{K_1 T}{z + K_1 T - 1}$$

Allora l'uscita, ponendo in ingresso un gradino di ampiezza  $r$ , risulterà

$$\begin{aligned} Y(z) = G(z)U(z)r &= \frac{K_1 T}{z + K_1 T - 1} \frac{z}{z - 1} r = r K_1 T z \left( \frac{1/K_1 T}{z - 1} - \frac{1/K_1 T}{z + K_1 T - 1} \right) = \\ &= r \left( \frac{1}{1 - z^{-1}} - \frac{1}{1 - (1 - K_1 T)z^{-1}} \right) \end{aligned}$$

Da cui, antitrasformando:

$$y(t) = y(kT) = r(1 - (1 - K_1 T)^k)$$

Per semplificare i conti si impone ora che il tempo di salita sia pari al tempo che impiega  $y(t)$  per portarsi dal suo stato iniziale al 90% del suo valore di regime  $r$ .

Allora andremo a scrivere che:

$$t = 5, T = 0.25 \rightarrow k = 20$$

$$\begin{aligned} y(kT) &= r(1 - (1 - K_1 T)^k) = 0.9r \\ K_1 &= \frac{1 - \sqrt[k]{0.1}}{T} = \frac{1 - \sqrt[20]{0.1}}{0.25} = 0.435 \end{aligned}$$

Mediante analoghi ragionamenti deduciamo subito che il valore di  $K_2$  che porta il tempo di salita ad essere pari a 5 secondi è:  $K_2 = -0.435$ .

Possiamo a questo punto verificare i risultati ottenuti mediante Simulink, ottenendo i seguenti grafici.

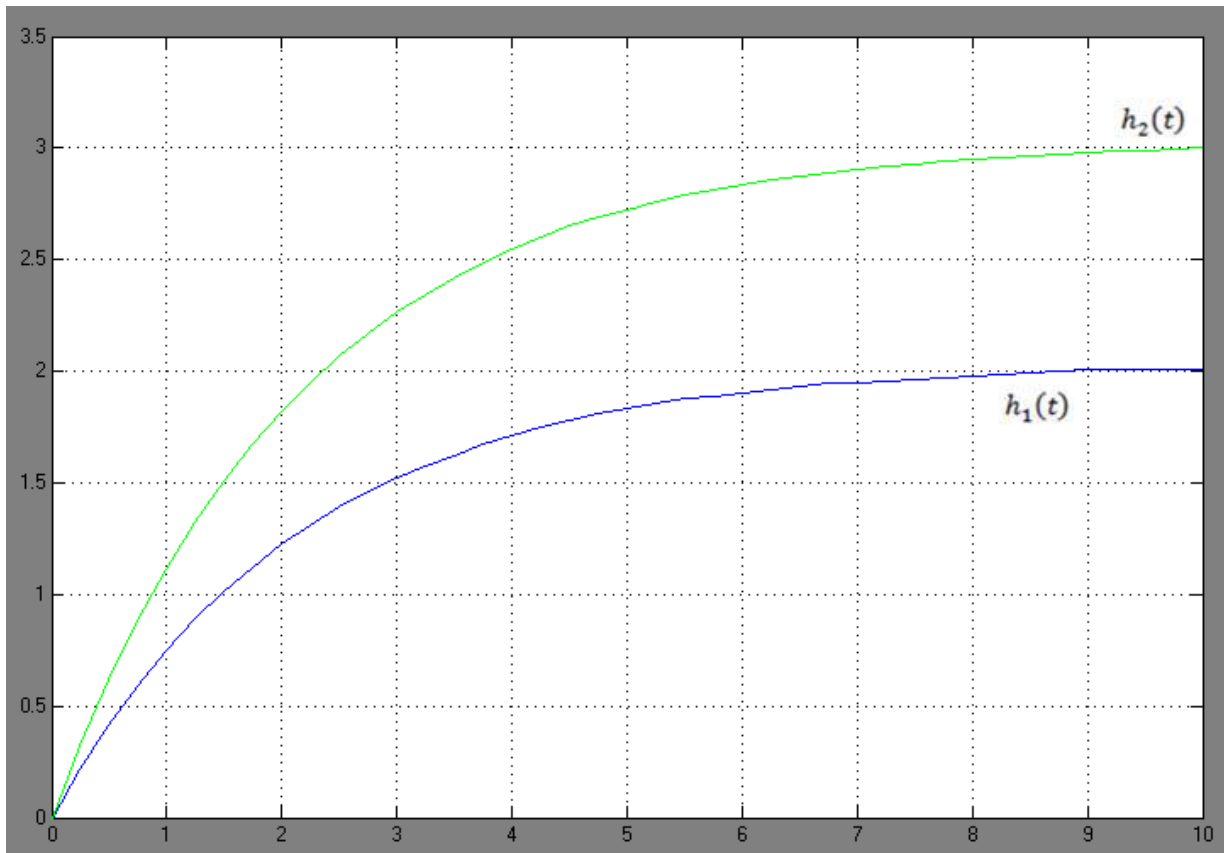


Figura 44: Altezze dei Serbatoi, Proporzionale

Come ci si poteva aspettare, il tempo di salita è perfettamente rispettato da entrambi gli impianti.

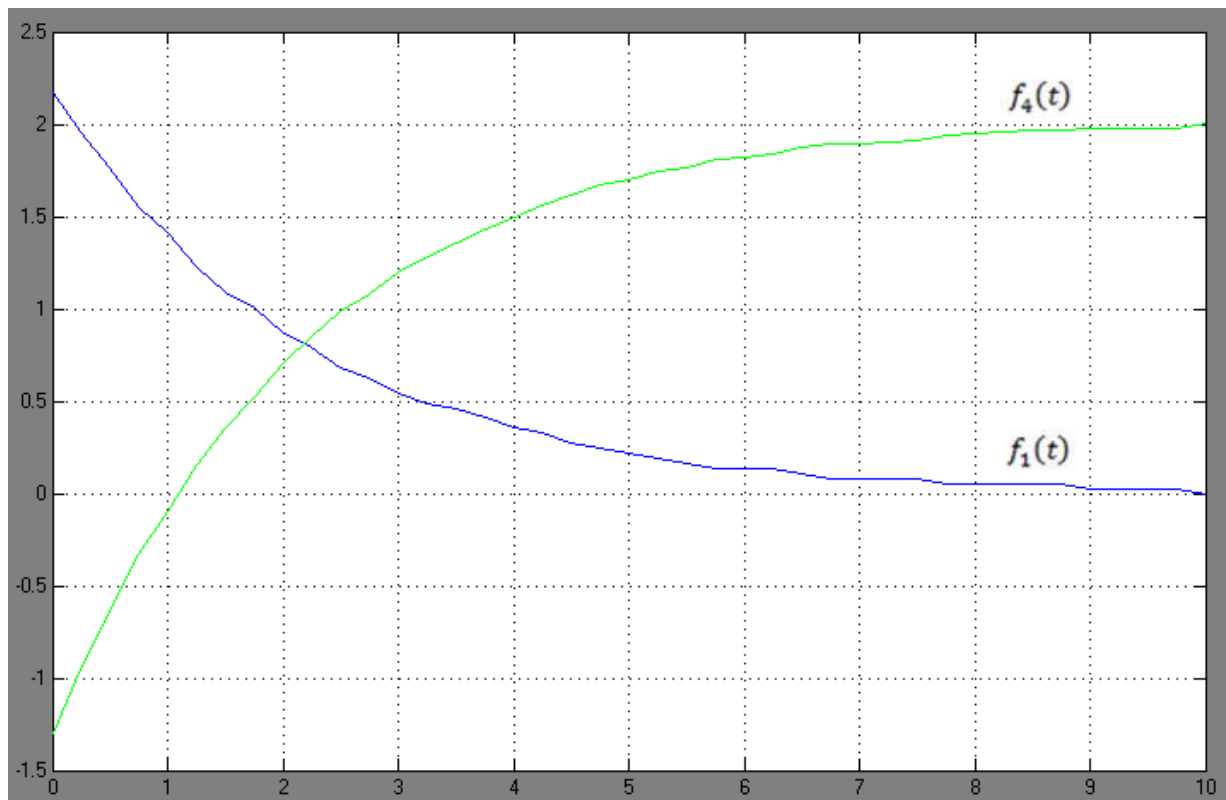


Figura 45: Flussi in ingresso, Proporzionale

In questo caso non c'è controllo sui flussi, visto che il parametro ottimizzato non ne tiene conto. Difatti si incappa nel solito problema di eventuali flussi negativi ed in modulo minori di 1. La risoluzione del problema può essere ottenuta manipolando ulteriormente  $K_2$ , tuttavia non abbiamo una tecnica che ci suggerisca un valore corretto.

Procedendo in modo non rigoroso, si trova che un possibile valore corretto per  $K_2$ , dati gli ingressi  $r = (2, 3)$  considerati, è:

$$K_2 = -0.3$$

che ovviamente aumenta il tempo di salita per il secondo impianto, tuttavia fornisce valori di flusso abbastanza buoni, come è possibile osservare nella simulazione:

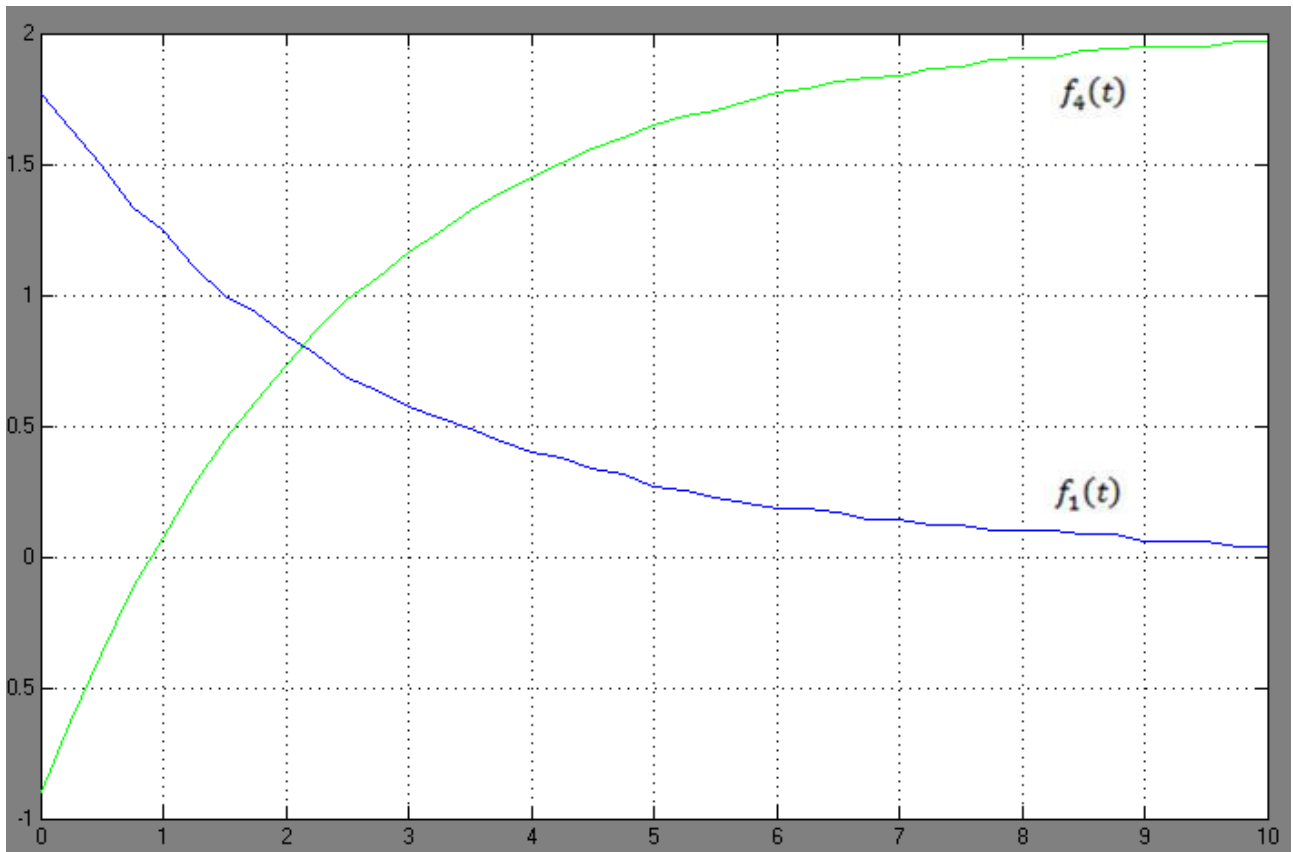


Figura 46: Flussi d'ingresso, Proporzionale Modificato

Come al solito però questa soluzione non ha valore generale, infatti se si cambiano riferimenti, il problema può ripresentarsi.

Fare riferimento alla prossima sezione per affrontare questa tematica.

## SEZIONE 10: ELIMINAZIONE DEI FLUSSI NON CONSENTITI

Durante l'analisi della risposta del sistema al variare dei controllori applicati, abbiamo più volte osservato che il controllo introduceva la possibilità che in ingresso all'impianto si presentassero dei flussi non compatibili con la natura fisica dello stesso.

In particolare, il sistema è stato linearizzato supponendo di avere come punti di equilibrio i seguenti:

$$\begin{cases} h_{1,0} = 1 \\ h_{2,0} = 1 \\ f_{1,0} = 1 \\ f_{2,0} = 2f_{1,0} = 2 \\ f_{3,0} = f_{1,0} = 1 \\ f_{4,0} = f_{1,0} = 1 \end{cases}$$

Osserviamo in particolare che, con riferimento agli ingressi del sistema, si ha  $f_{4,0} = f_{1,0} = 1$ . Allora le variazioni sui flussi dovranno essere limitate. In particolare si dovrà avere sempre:

$$\begin{cases} f_1(t) \geq 0 \\ f_4(t) \geq 0 \\ \tilde{f}_1(t) = f_1(t) - f_{1,0} \\ \tilde{f}_4(t) = f_4(t) - f_{4,0} \end{cases} \xrightarrow{\text{yields}} \begin{cases} \tilde{f}_1(t) \geq -1 \\ \tilde{f}_4(t) \geq -1 \end{cases}$$

Inoltre, altre limitazioni fisiche derivano dai macchinari utilizzati, che probabilmente non sono in grado di trasportare quantità arbitrariamente grandi di flusso. Si è allora posto un altro limite, stavolta superiore:

$$\begin{cases} \tilde{f}_1(t) \leq 5 \\ \tilde{f}_4(t) \leq 5 \end{cases}$$

Per implementare le limitazioni qui imposte, è sufficiente porre un saturatore, così che gli ingressi del sistema si mantengano sempre all'interno di quegli *upper e lower bound*. Considerando allora il nostro sistema discreto accoppiato, la situazione può essere quella mostrata in figura 47.

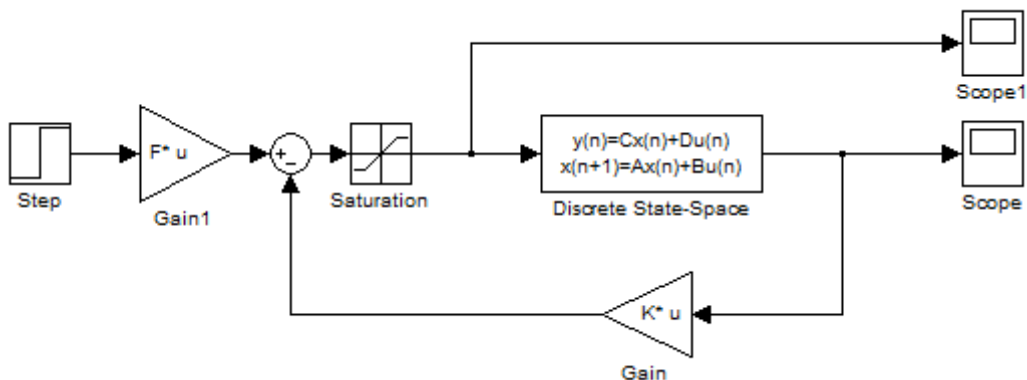


Figura 47: Introduzione del saturatore

Tornando alla notazione utilizzata durante il controllo, secondo cui  $f_i(k)$  rappresenta la variazione dal punto di equilibrio, il saturatore farà in modo che:

$$-1 < f_i(k) < 5$$

Andiamo a vedere il comportamento del sistema quando gli ingressi sono limitati. Utilizziamo il semplice controllo in retroazione basato sull' assegnamento dei poli, e per renderci bene conto di cosa accade con e senza la presenza del saturatore, supponiamo di piazzare i poli in modo che gli ingressi tendano a salire oltre il limite di saturazione:  $P = (0.7, 0.7)$ .

Le matrici di controllo relative risultano:

$$K = \begin{pmatrix} 1.2 & 1.2 \\ -0.8438 & -1.2 \end{pmatrix}; F = \begin{pmatrix} 1.2 & 1.2 \\ 0.1562 & -1.2 \end{pmatrix}$$

Andando a simulare, senza saturatore si ottiene:

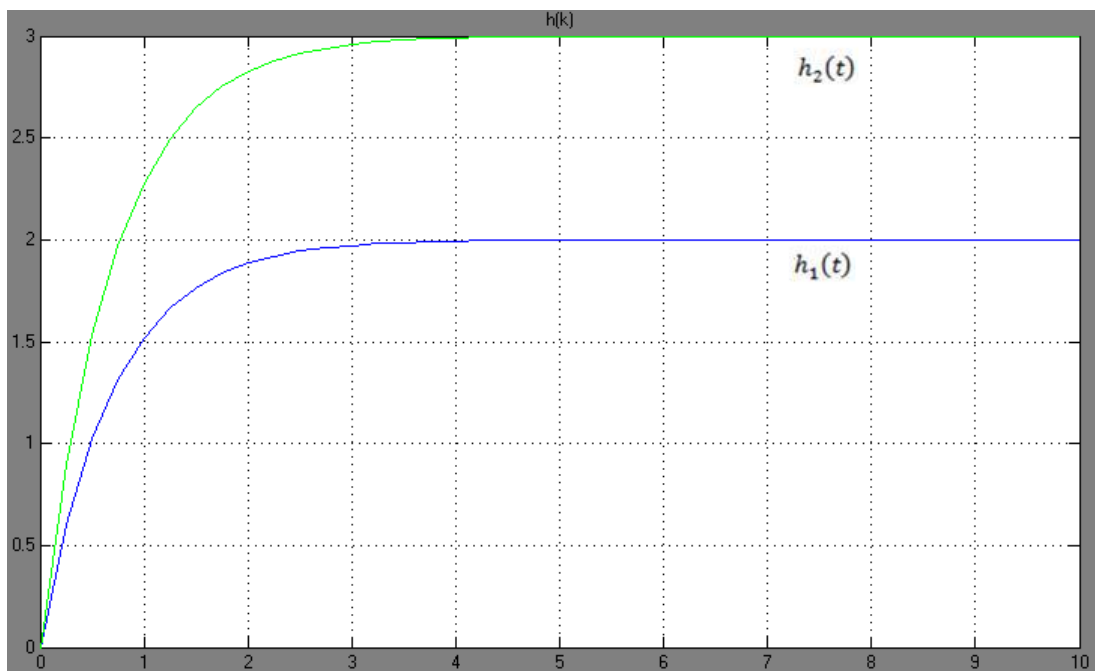


Figura 48: Altezze dei serbatoi,  $P = [0.7 \ 0.7]$

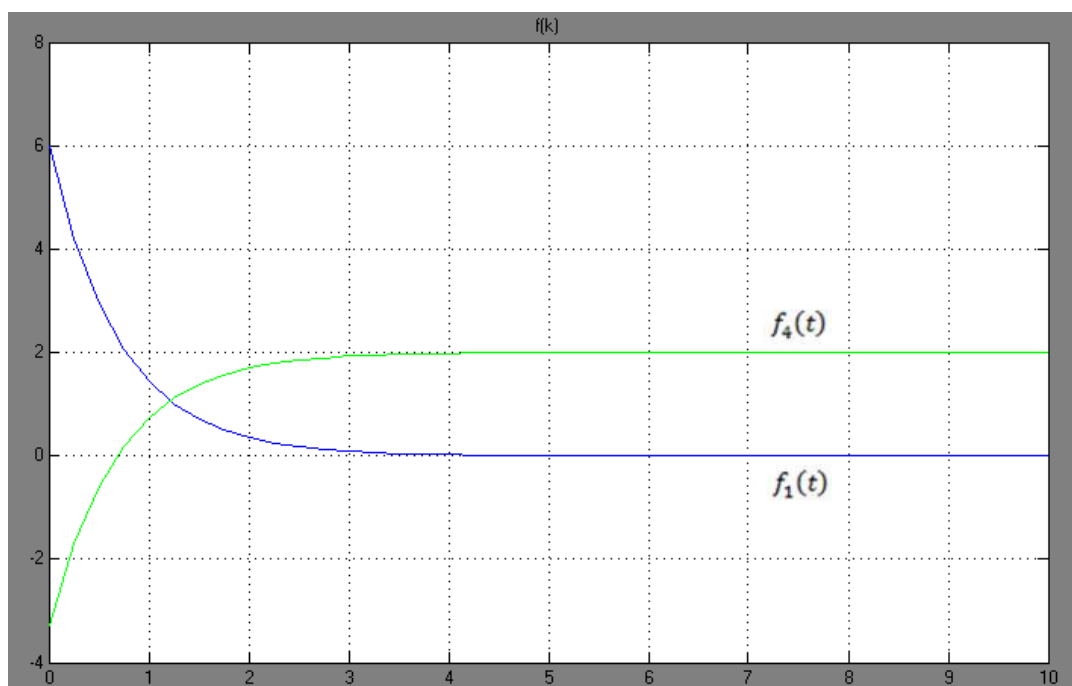


Figura 49: Flussi di ingresso,  $P = [0.7 \ 0.7]$

L'evoluzione è ovviamente molto veloce, vista la lontananza dei poli dalla circonferenza unitaria. Tuttavia quest' esempio ci verrà utile per capire cosa accade in presenza di saturatore.

Per quanto riguarda l'evoluzione dei flussi, come si può notare nella figura seguente, questa supera i limiti imposti, sia per quanto riguarda  $f_1(t) > 5$ , che  $f_4(t) < -1$ .

Andiamo ora a simulare l'andamento che si ottiene in presenza di saturatore. Per quanto riguarda i flussi di ingresso si ha la figura seguente.

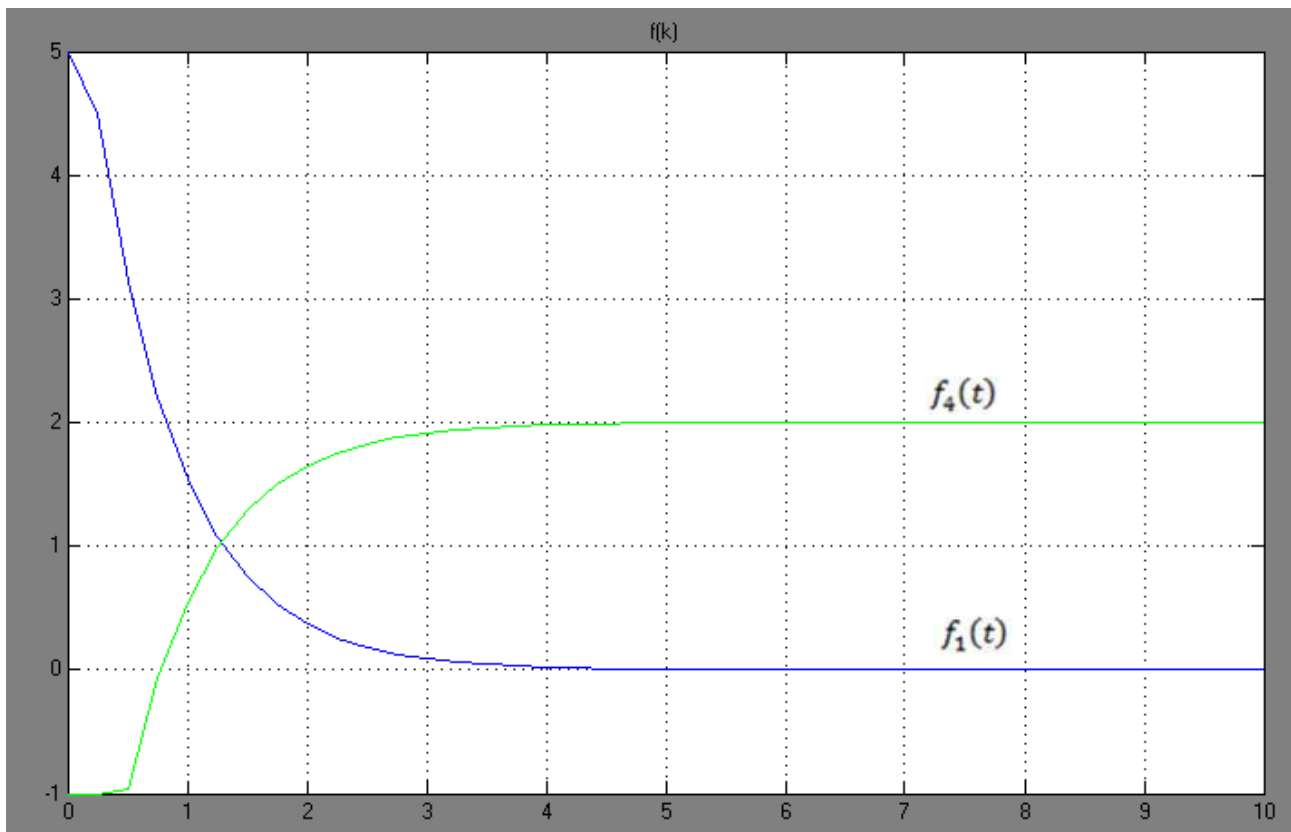


Figura 50: Flussi di ingresso saturati

Il saturatore compie il suo lavoro. In particolare notiamo che nel caso considerato il suo effetto viene sentito molto di più da  $f_4(t)$ . In effetti durante l'analisi della risposta ai vari controllori, ci eravamo resi conto che il flusso più "critico" era proprio quest' ultimo.

A questo punto resta da valutare come si modifica l'andamento dell'altezza di livello del liquido nei serbatoi quando gli ingressi sono stati modificati come appena descritto.

L'andamento in questione è riportato nella figura di pagina seguente.

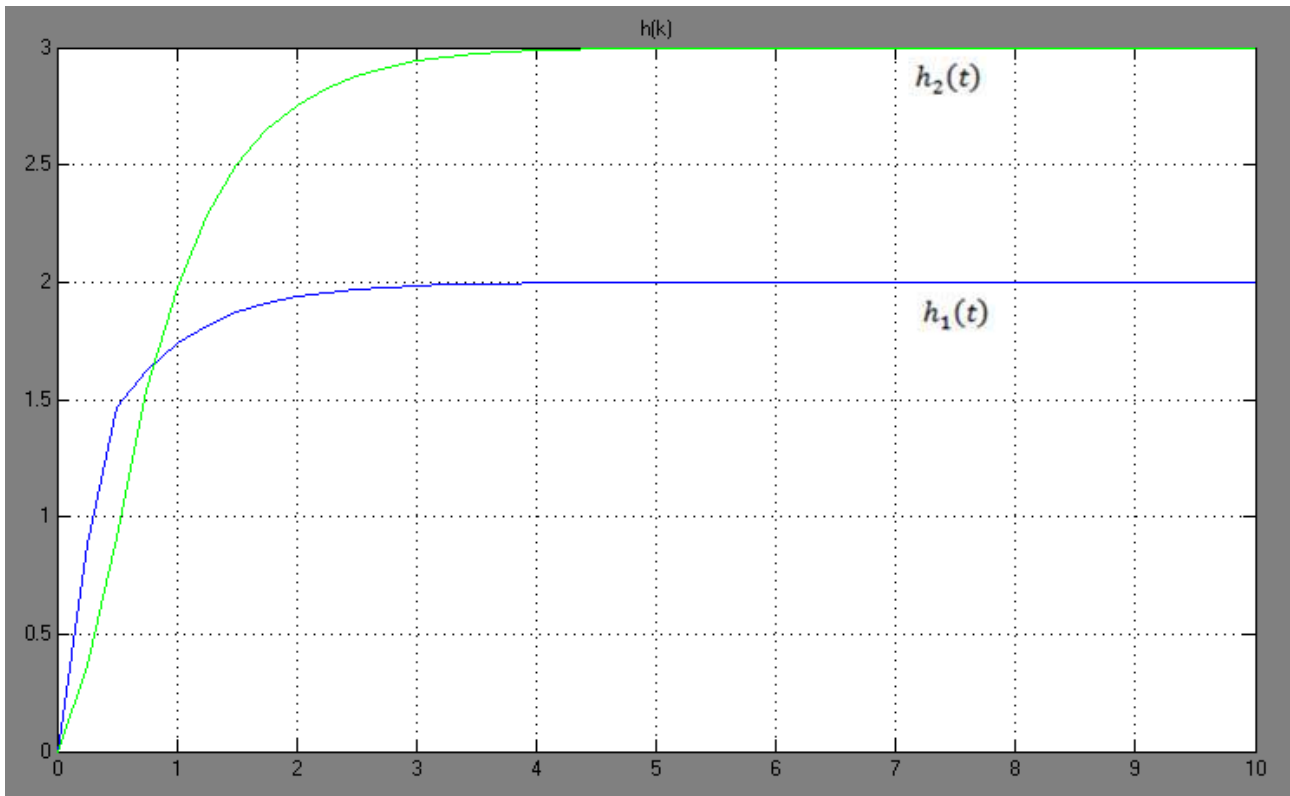


Figura 51: Altezze dei serbatoi, ingressi saturati

Dalla figura si osserva che  $h_1(t)$  risente particolarmente della saturazione avvenuta.

Questo si deve sostanzialmente al fatto che  $f_4(t)$  non è più in grado di “svuotare” il primo serbatoio per riempire il secondo, visto che si impone che un comportamento di questo tipo non è fisicamente ottenibile: allora  $h_1(t)$  tenderà a riempirsi più rapidamente del previsto.

Come sappiamo questo fenomeno non è particolarmente gradito vista l’osservazione finora fatta sui tempi di salita.

Si deduce quindi che l’uso di un saturatore, benché consigliato caldamente per evitare che si verifichino condizioni che le macchine non sono in grado di sopportare, non può essere visto come soluzione definitiva al problema, visto che il permanere di uno stato di saturazione fa assumere un andamento non ottimale per quel che concerne l’evoluzione delle altezze dei serbatoi.

Pertanto è opportuno associare al saturatore un buon controllo che eviti il verificarsi della saturazione, secondo il sano principio che prevenire è meglio che curare.

## SEZIONE 11: STUDIO DELLA SENSIBILITÀ AL DISTURBO

---

Nella sezione 1 abbiamo visto che nel sistema fisico preso in considerazione vi è un flusso  $f_3(t)$  che rappresenta la perdita di liquido dal secondo serbatoio.

La dinamica era la seguente:

$$\begin{cases} \dot{\tilde{h}}_1(t) = -\frac{f_{2,0}}{2A_1h_{1,0}}\tilde{h}_1(t) + \frac{\tilde{f}_1(t)}{A_1} + \frac{\tilde{f}_4(t)}{A_1} \\ \dot{\tilde{h}}_2(t) = \frac{f_{2,0}}{2A_2h_{1,0}}\tilde{h}_1(t) - \frac{\tilde{f}_4(t)}{A_2} - \frac{\tilde{f}_3(t)}{A_2} \end{cases}$$

Il flusso differenziale  $\tilde{f}_3(t)$ , non controllabile, rappresenta appunto un disturbo che nella analisi condotta fino ad ora è stato ipotizzato trascurabile, imponendo  $\tilde{f}_3(t) = 0$ .

In questa sezione si propone uno studio dell'effetto che avrebbe il disturbo  $\tilde{f}_3(t)$  (non più trascurabile) su alcuni dei controllori elaborati nella precedente trattazione.

Dovendo stavolta considerare anche il disturbo abbiamo che:

$$\begin{pmatrix} \dot{\tilde{h}}_1(t) \\ \dot{\tilde{h}}_2(t) \end{pmatrix} = \begin{pmatrix} -\frac{f_{2,0}}{2A_1h_{1,0}} & 0 \\ \frac{f_{2,0}}{2A_2h_{1,0}} & 0 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_1(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{A_1} & \frac{1}{A_1} \\ 0 & -\frac{1}{A_2} \end{pmatrix} \begin{pmatrix} \tilde{f}_1(t) \\ \tilde{f}_4(t) \end{pmatrix} + \begin{pmatrix} 0 \\ -\frac{1}{A_2} \end{pmatrix} \tilde{f}_3(t)$$

E quindi, sostituendo i valori del punto di equilibrio utilizzati nella stessa sezione 1:

$$\begin{cases} h_{1,0} = 1 \\ h_{2,0} = 1 \\ f_{1,0} = 1 \\ f_{2,0} = 2f_{1,0} = 2 \\ f_{3,0} = f_{1,0} = 1 \\ f_{4,0} = f_{1,0} = 1 \end{cases}$$

E gli stessi valori utilizzati per  $A_1$  e  $A_2$  ( $A_1 = 1$ ,  $A_2 = 1$ ), si ottiene:

$$\begin{pmatrix} \dot{\tilde{h}}_1(t) \\ \dot{\tilde{h}}_2(t) \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_1(t) \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \tilde{f}_1(t) \\ \tilde{f}_4(t) \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} \tilde{f}_3(t)$$

A questo punto dunque il sistema viene ad avere tre ingressi (flussi) di cui però solo due sono controllabili, riscriviamo:

$$\begin{cases} \begin{pmatrix} \dot{\tilde{h}}_1(t) \\ \dot{\tilde{h}}_2(t) \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_2(t) \end{pmatrix} + \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} \tilde{f}_1(t) \\ \tilde{f}_4(t) \\ \tilde{f}_3(t) \end{pmatrix} \\ (y_1(t) \\ y_2(t)) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{h}_1(t) \\ \tilde{h}_2(t) \end{pmatrix} \end{cases}$$

Da cui si ricava una modifica nella matrice  $B$  rispetto ai casi precedenti, applichiamo le modifiche anche agli script Matlab mediante i seguenti comandi (c2dm è il comando automatico Matlab per la discretizzazione):

```
>> Ac = [-1 0; 1 0]
>> Bc = [1 1 0; 0 -1 -1]
>> Cc = [1 0; 0 1]
>> Dc = [0 0 0; 0 0 0]
>> [Ad,Bd,Cd,Dd] = c2dm(Ac,Bc,Cc,Dc,sample_time)
```

Non avendo informazioni sul comportamento atteso del disturbo si suppone quanto segue:

- Essendo il punto di equilibrio per  $f_3(t) = f_{3,0} + \tilde{f}_3(t) = 1$ , si ipotizza che il disturbo sia in modulo minore a quest'ultimo, pari al massimo a 0.15
- Operando su un sistema meccanico, si suppone che il disturbo abbia una pulsazione piuttosto bassa, approssimativamente sotto 1 rad/sec

Vediamo allora quale sarebbe l'influenza del disturbo sui sistemi controllati in modo accoppiato.

Siamo interessati alla forma discreta per la funzione di trasferimento disturbo-uscita del sistema in ciclo chiuso (funzione di sensitività al disturbo), quindi per la analisi a seguire partiamo dal visualizzare quale è la situazione con l'inserimento dell'anello di reazione (figura 52).

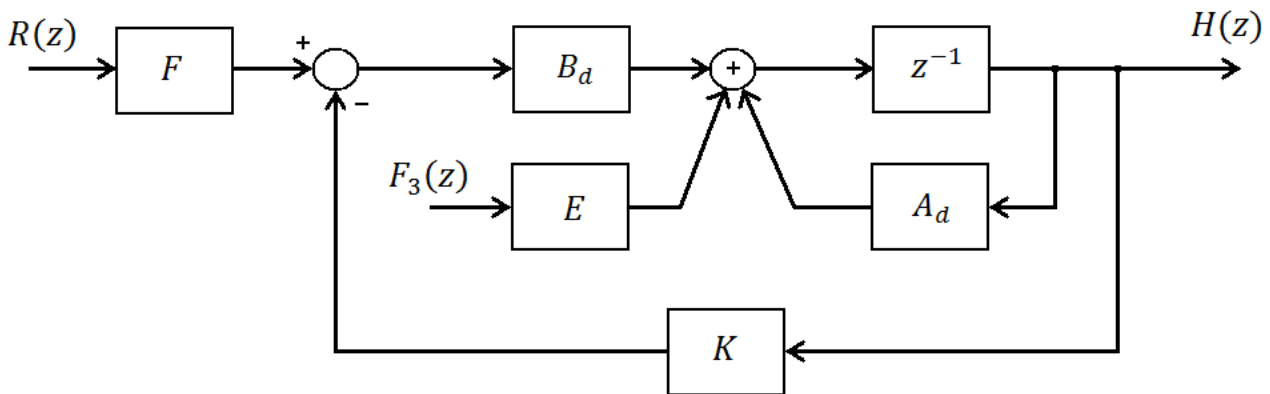


Figura 52: controllo accoppiato con iniezione del disturbo

Notiamo che in realtà il flusso di disturbo agisce direttamente sul sistema analogico.

Controllando digitalmente il sistema, siamo però interessati solo alla funzione di sensitività di tipo discreto e quindi si può considerare tranquillamente un segnale di disturbo discreto per valutare la sensibilità del sistema.

Dalla figura 52 si deduce che:

$$\begin{aligned} H(z) &= z^{-1}(EF_3(z) + B_d(FR(z) - KH(z)) + A_dH(z)) \\ zH(z) &= EF_3(z) + B_dFR(z) - B_dKH(z) + A_dH(z) \\ (zI - A_d + B_dK)H(z) &= B_dFR(z) + EF_3(z) \end{aligned}$$

E dunque la funzione di sensitività risulta:

$$G_d(z) = (zI - A_d + B_d K)^{-1} E$$

Che è allora facilmente calcolabile, dove:

$$\left\{ \begin{array}{l} A_d = \begin{pmatrix} 0.7788 & 0 \\ 0.2212 & 1 \end{pmatrix} \\ B_d = \begin{pmatrix} 0.2212 & 0.2212 \\ 0.0288 & -0.2212 \end{pmatrix} \\ E = \begin{pmatrix} 0 \\ -0.25 \end{pmatrix} \\ K, F \text{ derivanti dal controllo} \end{array} \right.$$

In Matlab si va a dividere la nuova  $B_d$  ( $2 \times 3$ ) nelle due matrici  $B_d$  ( $2 \times 2$ ) ed  $E$  ( $2 \times 1$ ) sopra riportate, al fine di determinare la funzione di sensitività:

```
>> E = Bd(:,3)
>> Bd = Bd(:,1:2)
>> z = tf('z',sample_time)
>> Gd = minreal((z*I-Ad+Bd*K)^(-1)*E)
```

Vediamo cosa succede allora con alcuni dei controllori sviluppati precedentemente.

### Controllo mediante assegnamento dei poli

Utilizzando il controllore che piazza i poli in:

$$\begin{aligned} \lambda_1 &= 0.9 \\ \lambda_2 &= 0.95 \end{aligned}$$

Si ha che le matrici relative al controllo risultano:

$$K = \begin{pmatrix} 0.4 & 0.2 \\ -0.9479 & -0.2 \end{pmatrix}; F = \begin{pmatrix} 0.4 & 0.2 \\ 0.0521 & -0.2 \end{pmatrix}$$

Ne deriva che il calcolo della  $G_d(z)$  porta a:

$$G_d(z) = (zI - A_d + B_d K)^{-1} E = \begin{pmatrix} \frac{1.665 * 10^{-16}}{z^2 - 1.85z + 0.855} \\ \frac{-0.25}{z - 0.95} \end{pmatrix}$$

Si riportano in figura 53 i due diagrammi di Bode relativi al modulo delle due componenti di  $G_d(z)$ .

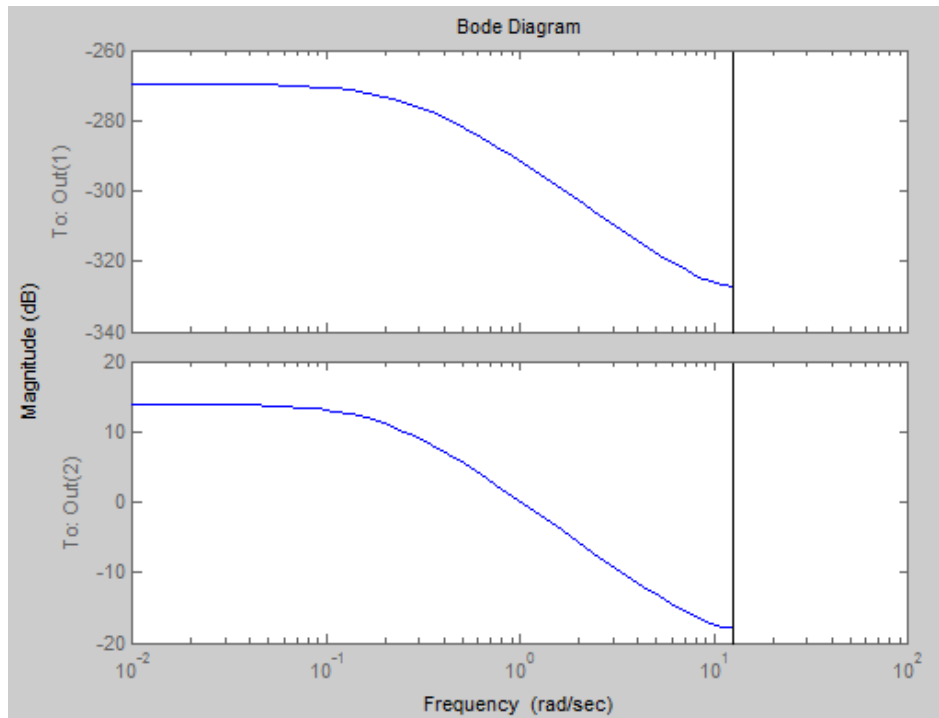


Figura 53: Diagrammi di modulo per le componenti di  $G_d(z)$  (piazamento dei poli)

Notiamo che la prima delle due ha al numeratore un valore del tipo  $1.665 \cdot 10^{-16}$ , questo significa che la componente è molto vicina allo zero (in realtà quel valore al numeratore potrebbe anche essere un errore derivante dalle approssimazioni numeriche svolte da Matlab, in realtà potrebbe quindi essere 0).

Generando il disturbo in modo random (anche se con i vincoli descritti sopra per frequenza e ampiezza), si può simulare mediante lo schema Simulink in figura 54.

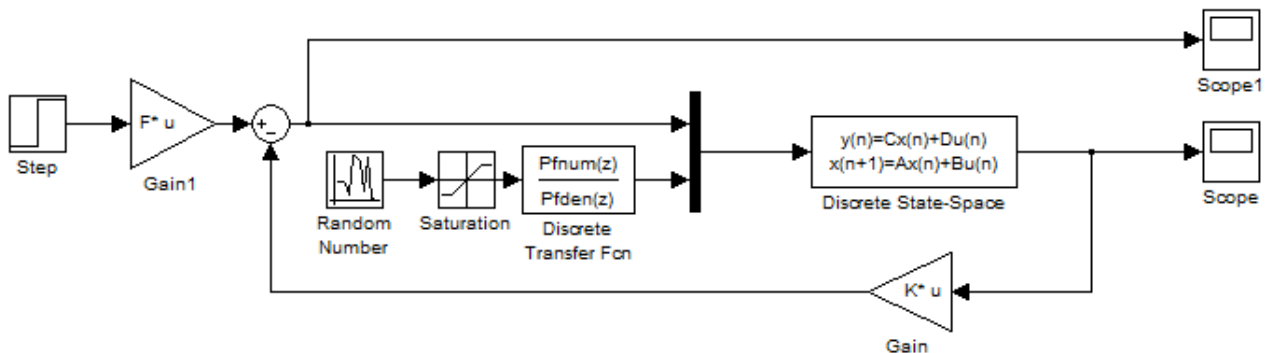


Figura 54: Schema per le simulazioni dell'effetto del disturbo sui controlli accoppiati

In figura si può notare che i requisiti richiesti per il disturbo vengono soddisfatti applicando alla sequenza di numeri random generati un saturatore (che ne limita l'ampiezza) ed un filtro passa basso del tipo:

$$P_f(z) = \frac{0.111z + 0.111}{z - 0.7778}$$

che introduce un guadagno unitario ed ha pulsazione di taglio pari a 1 rad/sec.

Il filtro viene ottenuto come discretizzazione secondo Tustin di un semplice filtro passa-basso continuo. In Matlab si esegue la procedura qui riportata:

```
>> s = tf('s')
>> Pf = c2d(1/(s+1), sample_time, 'tustin')
```

Gli andamenti ottenuti sono descritti in figura 55 per le altezze dei liquidi.

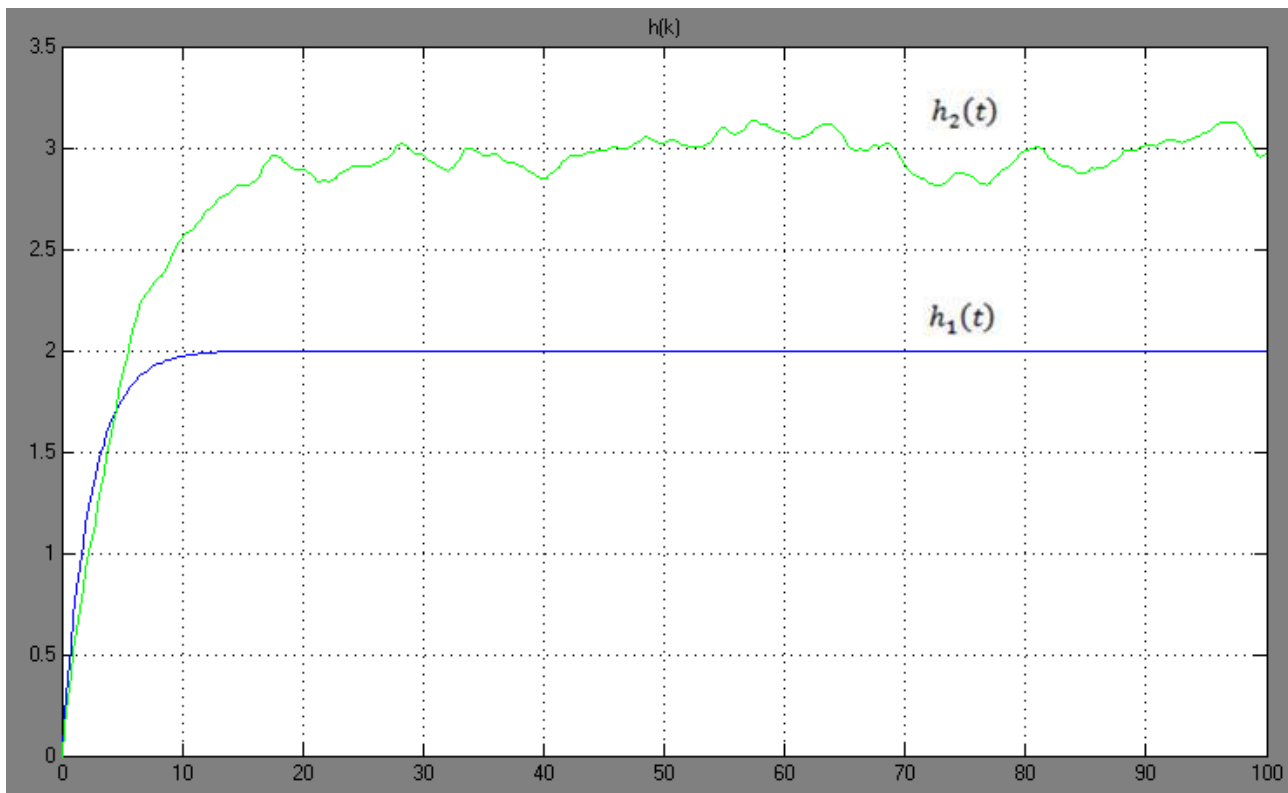


Figura 55: uscite con disturbo (piazamento dei poli)

Dove si può osservare che il disturbo non sembra avere influenza su  $h_1(t)$ , comportamento che ci si poteva aspettare osservando l'andamento del diagramma di Bode per la relativa componente di  $G_d(z)$ .

Notiamo invece che l'andamento di  $h_2(t)$  è pesantemente influenzato dal disturbo, dal diagramma di modulo per la relativa componente di  $G_d(z)$  si vede infatti che non solo non attenua l'effetto del disturbo, ma addirittura lo amplifica per le basse frequenze (come quelle ipotizzate per il disturbo).

In figura 56 si riporta l'ingrandimento dell'evoluzione per  $h_2(t)$ .

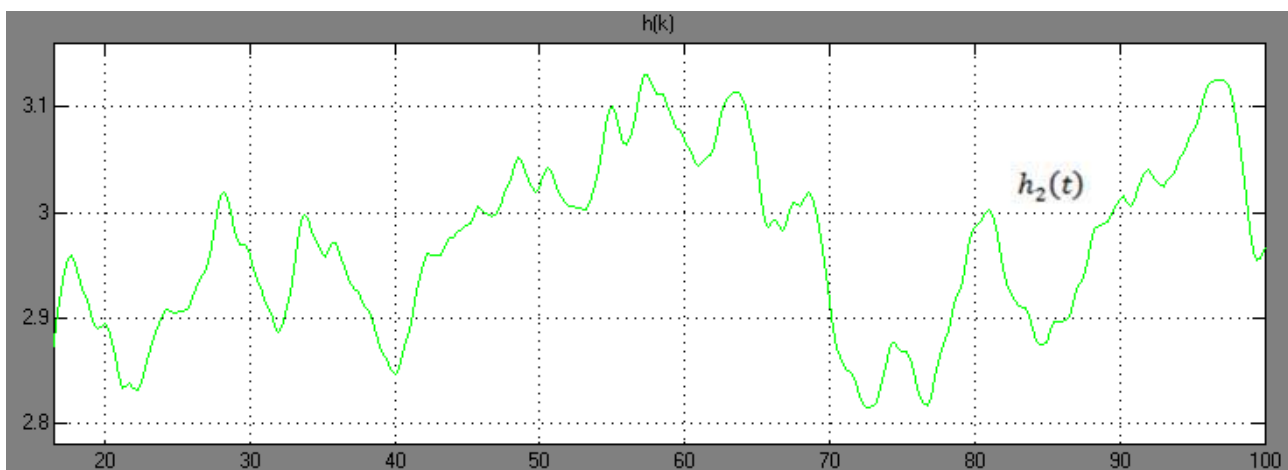


Figura 56: ingrandimento dell'evoluzione di  $h_2(t)$  (piazamento dei poli)

In figura 57 è inoltre riportato l'andamento per i flussi in ingresso al sistema

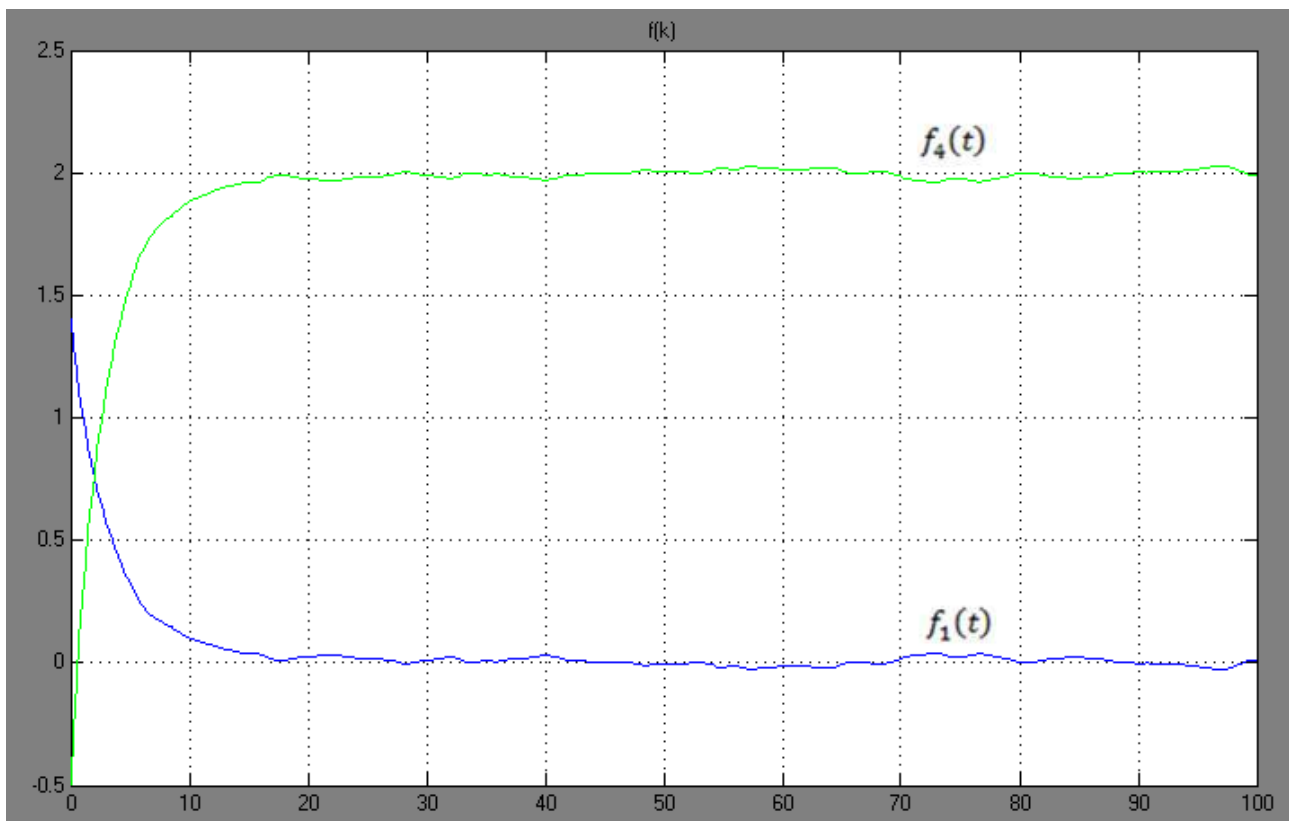


Figura 57: flussi con disturbo (piazzamento dei poli)

### Controllo Linear Quadratic Regulator (o Controllo di Riccati)

Nella Sezione 5 eravamo arrivati alla conclusione che il miglior controllo possibile fosse quello con:

$$K = \begin{pmatrix} 0.3056 & 0.2919 \\ 0.0295 & -0.5521 \end{pmatrix}; F = \begin{pmatrix} 0.3056 & 0.2919 \\ 1.0295 & -0.5521 \end{pmatrix}$$

Ne deriva che il calcolo della  $G_d(z)$  porta a:

$$G_d(z) = (zI - A_d + B_dK)^{-1}E = \begin{pmatrix} \frac{-0.01439}{z^2 - 1.574z + 0.6001} \\ \frac{-0.25z + 0.1762}{z^2 - 1.574z + 0.6001} \end{pmatrix}$$

Si riportano in figura 58 i due diagrammi di Bode relativi al modulo delle due componenti della nuova  $G_d(z)$ .

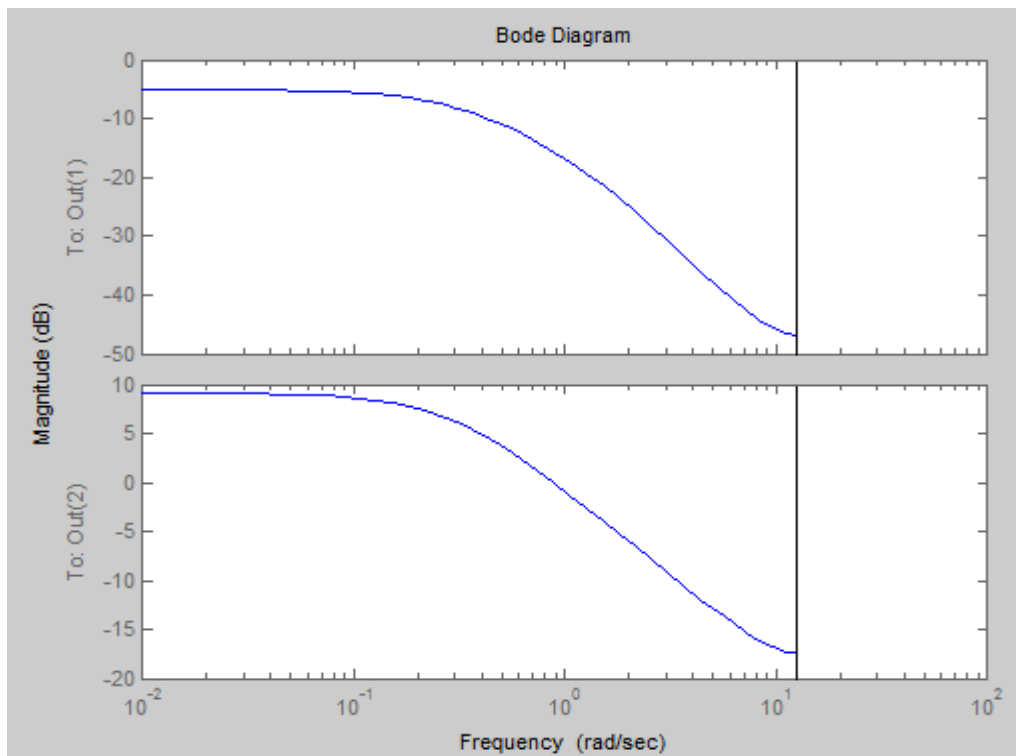


Figura 58: Diagrammi di modulo per le componenti di  $G_d(z)$  (Controllo LQR)

Dove notiamo che, a differenza del caso precedente, entrambe le componenti evidenziano una certa sensibilità al disturbo, specialmente a bassa frequenza.

In effetti però entrambe evidenziano che il comportamento è migliore rispetto al comportamento che aveva  $h_2(t)$  nel caso di assegnamento dei poli (l'amplificazione a bassa frequenza del disturbo è infatti inferiore).

Sempre effettuando le simulazioni come descritto in precedenza e col medesimo schema, si ottiene l'andamento per le uscite riportato in figura 59.

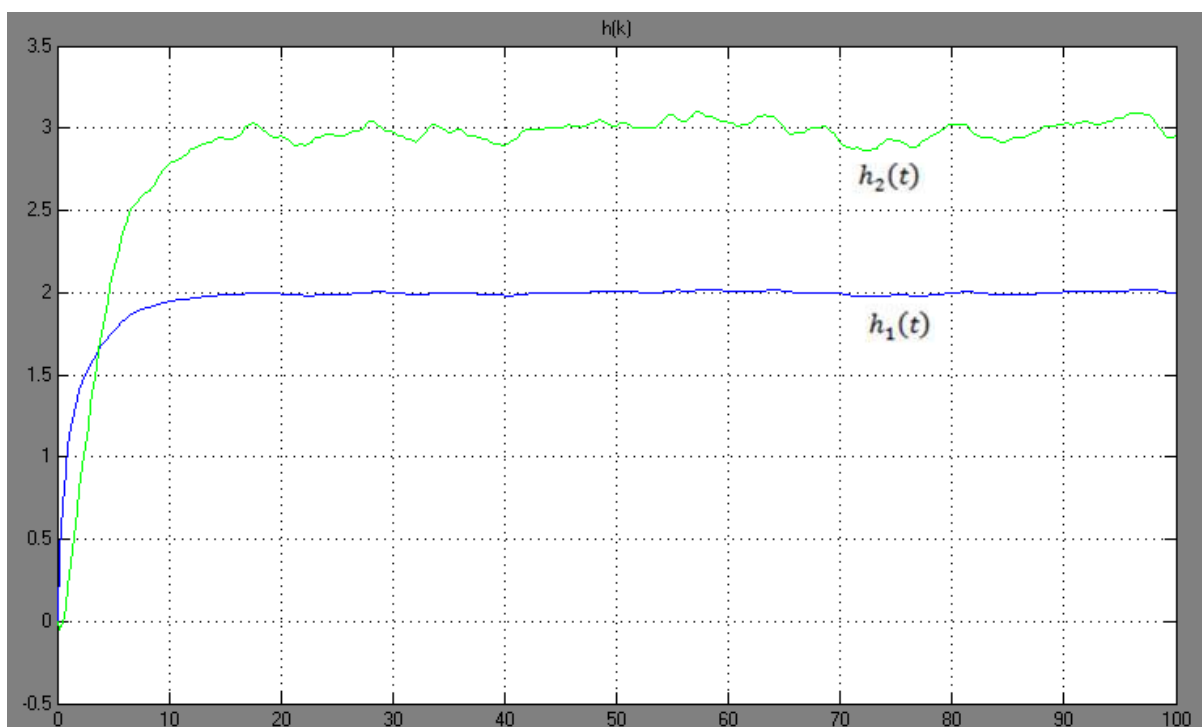


Figura 59: uscite con disturbo (Controllo LQR)

Notiamo che in effetti è possibile vedere una influenza sia su  $h_1(t)$  che su  $h_2(t)$ , a differenza di quanto accadeva nel caso di piazzamento dei poli (e questo comportamento era atteso osservando le componenti della  $G_d(z)$ ).

In figura 60 e 61 vengono mostrati gli ingrandimenti delle evoluzioni per  $h_1(t)$  e  $h_2(t)$ , rispettivamente.

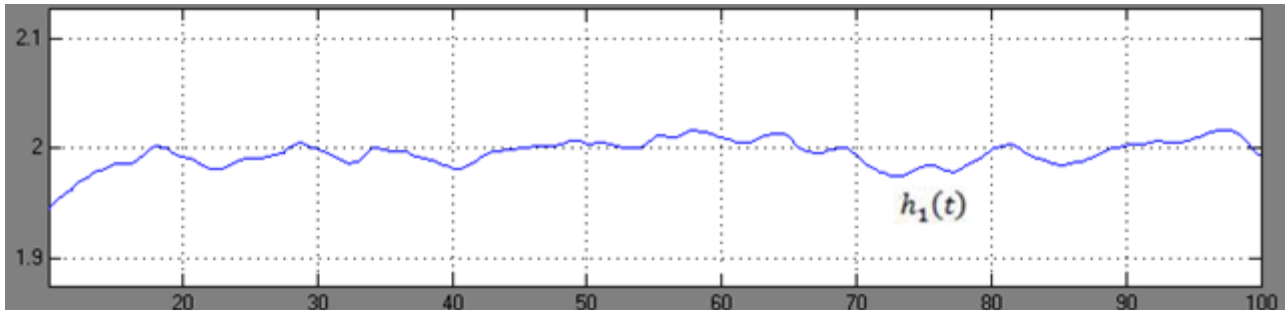


Figura 60: ingrandimento della evoluzione di  $h_1(t)$  (controllo LQR)

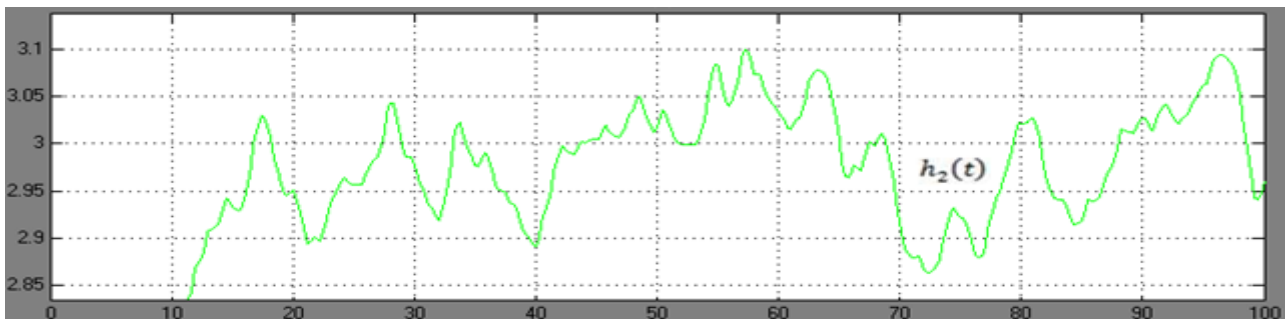


Figura 61: ingrandimento della evoluzione di  $h_2(t)$  (controllo LQR)

Dove si può osservare un andamento per  $h_2(t)$  leggermente migliore rispetto a quello ottenuto per il controllo mediante piazzamento dei poli.

In figura 62 si riporta inoltre il nuovo andamento per i flussi di ingresso in presenza del disturbo.

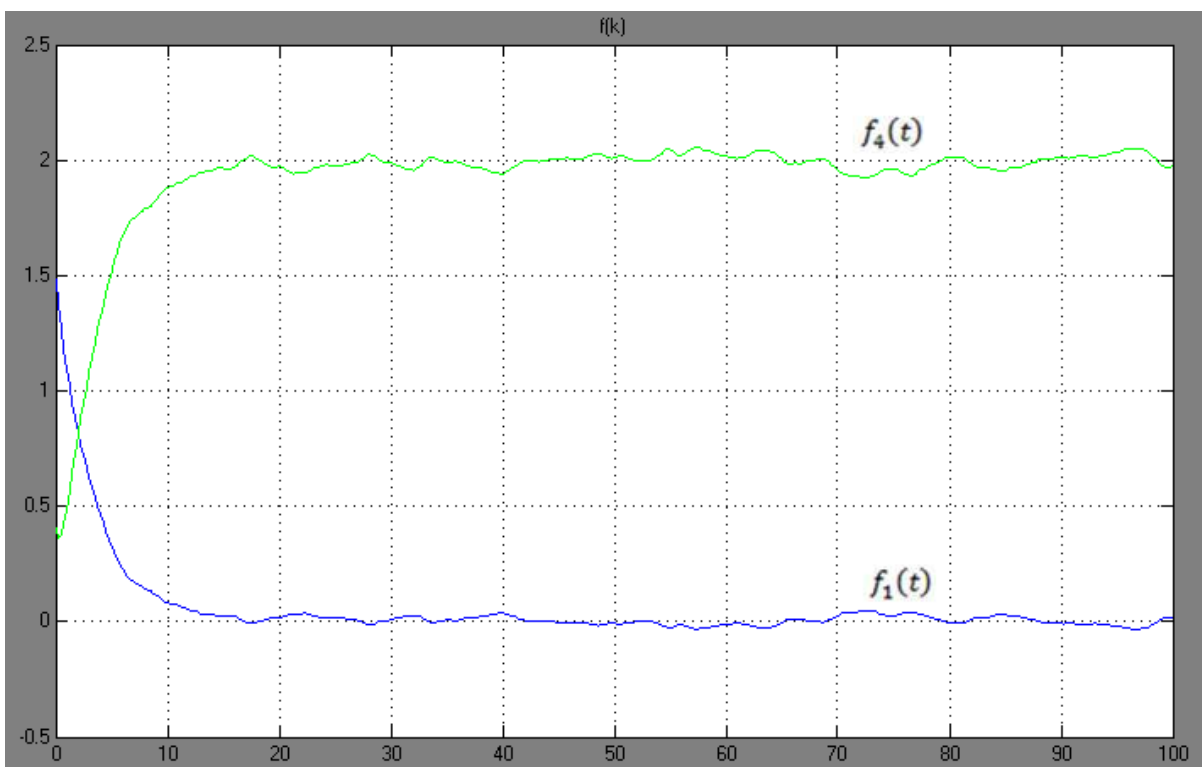


Figura 62: flussi con disturbo (Controllo LQR)

Adesso esaminiamo quale è la influenza di questo ipotetico disturbo nel caso di controllo disaccoppiato dell'impianto

Siamo ancora interessati alla sola forma discreta per la funzione di trasferimento disturbo uscita del sistema in ciclo chiuso (funzione di sensitività al disturbo), quindi per la analisi a seguire facciamo riferimento al seguente schema dove viene inserito il controllore e il tutto viene chiuso in reazione (figura 63).

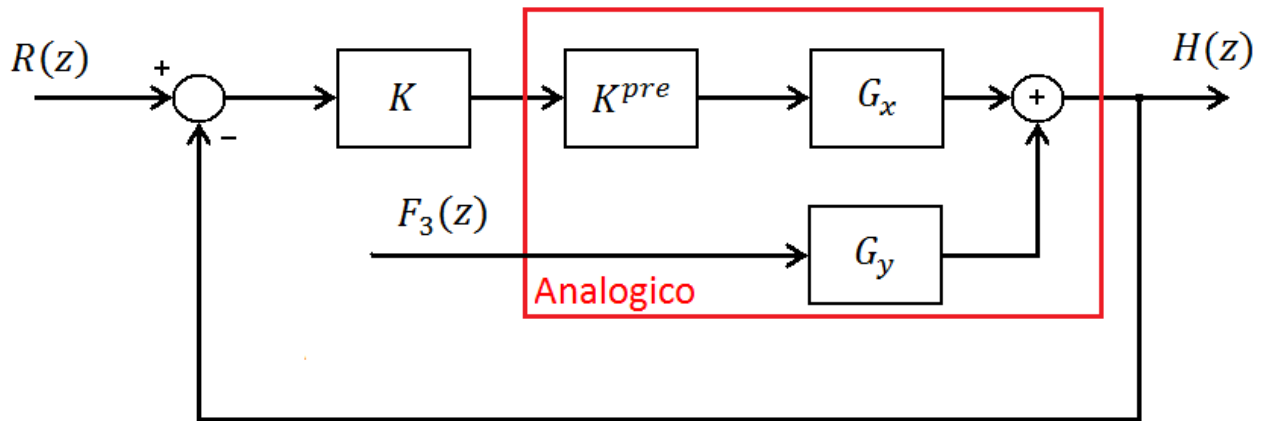


Figura 63: controllo disaccoppiato con iniezione del disturbo

Dove, dato il controllo disaccoppiato, la K avrà la seguente forma:

$$K(z) = \begin{pmatrix} K_1(z) & 0 \\ 0 & K_2(z) \end{pmatrix}$$

E invece avremo che :

$$G(s) = C(zI - A)^{-1}B = \begin{pmatrix} \frac{1}{s+1} & \frac{1}{s+1} & 0 \\ \frac{1}{s(s+1)} & -\frac{1}{s+1} & -\frac{1}{s} \end{pmatrix} = \begin{pmatrix} G_{11}(s) & G_{12}(s) & G_{13}(s) \\ G_{21}(s) & G_{22}(s) & G_{23}(s) \end{pmatrix}$$

Dove A, B e C sono le matrici tali che:

$$\begin{cases} \dot{\tilde{h}}(t) = A\tilde{h} + B \begin{pmatrix} \tilde{f}_1(t) \\ \tilde{f}_4(t) \\ \tilde{f}_3(t) \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} \tilde{h} + \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} \tilde{f}_1(t) \\ \tilde{f}_4(t) \\ \tilde{f}_3(t) \end{pmatrix} \\ y(t) = C\tilde{h} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tilde{h} \end{cases}$$

In particolare la G(s) è tale che:

$$\begin{aligned} H(s) = G(s) \begin{pmatrix} F_1(s) \\ F_4(s) \\ F_3(s) \end{pmatrix} &= \begin{pmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{pmatrix} \begin{vmatrix} G_{13}(s) \\ G_{23}(s) \end{vmatrix} \begin{pmatrix} F_1(s) \\ F_4(s) \\ F_3(s) \end{pmatrix} = \\ &= (G_x(s) \mid G_y(s)) \begin{pmatrix} F_1(s) \\ F_4(s) \\ F_3(s) \end{pmatrix} = G_x(s) \begin{pmatrix} F_1(s) \\ F_4(s) \end{pmatrix} + G_y(s)F_3(s) \end{aligned}$$

E quindi in figura 63 sono state rappresentate separatamente le due componenti. Indichiamo adesso con  $disc(F(s))$  la operazione di discretizzazione della  $F(s)$  secondo la tecnica esatta della invarianza al gradino, descritta precedentemente, e il risultato sia  $F(z)$ .

Dalla figura 63 deriva allora che:

$$H(z) = disc(G_x(s)K^{pre}(s))K(z)(R(z) - H(z)) + disc(G_y(s))F_3(z)$$

$$(I + disc(G_x(s)K^{pre}(s))K(z))H(z) = disc(G_x(s)K^{pre}(s))K(z)R(z) + disc(G_y(s))F_3(z)$$

E dunque la funzione di sensitività al disturbo è la seguente:

$$G_d(z) = (I + disc(G_x(s)K^{pre}(s))K(z))^{-1} disc(G_y(s)) =$$

$$= (I + disc(P(s))K(z))^{-1} disc(G_y(s))$$

Dove si è indicato, come nella sezione 6,  $P(s) = G_x(s)K^{pre}(s)$  (anche se la notazione  $G_x(s)$  è stata introdotta solo in questa sezione, in sezione 6 si parlava infatti semplicemente di  $G(s)$ ).

In definitiva la funzione di sensitività è la seguente:

$$G_d(z) = (I + P(z)K(z))^{-1}G_y(z)$$

La  $G_y(z)$  è stata ottenuta come discretizzazione della funzione di trasferimento fra il terzo ingresso  $f_3(t)$  (il disturbo), e l'uscita in ciclo aperto. In Matlab può quindi essere ottenuto mediante i comandi:

```
>> [NUM3,DEN3] = ss2tf(Ac,Bc,Cc,Dc,3)
>> Gy = minreal(tf({NUM3(1,:);NUM3(2,:)},{DEN3;DEN3}))
>> Gyd = c2d(Gy,sample_time)
```

Ottenendo così:

$$G_y(z) = \begin{pmatrix} 0 \\ -\frac{0.25}{z-1} \end{pmatrix}$$

Per  $P(z)$  si può sia rieseguire la procedura sopra indicata per gli altri due ingressi, sia riutilizzare le funzioni di trasferimento già ottenute nella sezione 7, ottenendo:

$$P(z) = \begin{pmatrix} \frac{0.25}{z-1} & 0 \\ 0 & -\frac{0.25}{z-1} \end{pmatrix}$$

A questo punto può essere effettuato agevolmente, sempre mediante Matlab, il calcolo della funzione di sensitività:

```
>> z = tf('z',sample_time)
>> Gd = minreal((I+[Pd1 0; 0 Pd2]*[K1 0; 0 K2])^(-1)*Gyd)
```

Vediamo allora cosa succede con uno dei controllori sviluppati precedentemente per il sistema disaccoppiato.

## Controllo algebrico del sistema disaccoppiato

Il controllore ricavato nella sezione 8 era il seguente:

$$K(z) = \begin{pmatrix} 0.3879 & 0 \\ 0 & -0.304 \end{pmatrix}$$

Ne deriva che il calcolo della  $G_d(z)$  porta a:

$$G_d(z) = (I + P(z)K(z))^{-1}G_y(z) = \begin{pmatrix} 0 \\ 0.25 \\ -\frac{0.25}{z - 0.924} \end{pmatrix}$$

Si nota subito che pare si abbia una reiezione perfetta del disturbo su  $h_1(z)$ , infatti la relativa componente della funzione di sensitività risulta proprio pari a 0, osserviamo tuttavia che questo comportamento deriva dal fatto che si è assunto il disaccoppiatore in grado di effettuare un disaccoppiamento perfetto, il che difficilmente si verificherà nella pratica (lasciando un certo accoppiamento che porterà ad una influenza del disturbo anche su  $h_1(z)$ ).

In figura 64 si riporta il diagramma di Bode (solo il modulo) della seconda componente della  $G_d(z)$ .

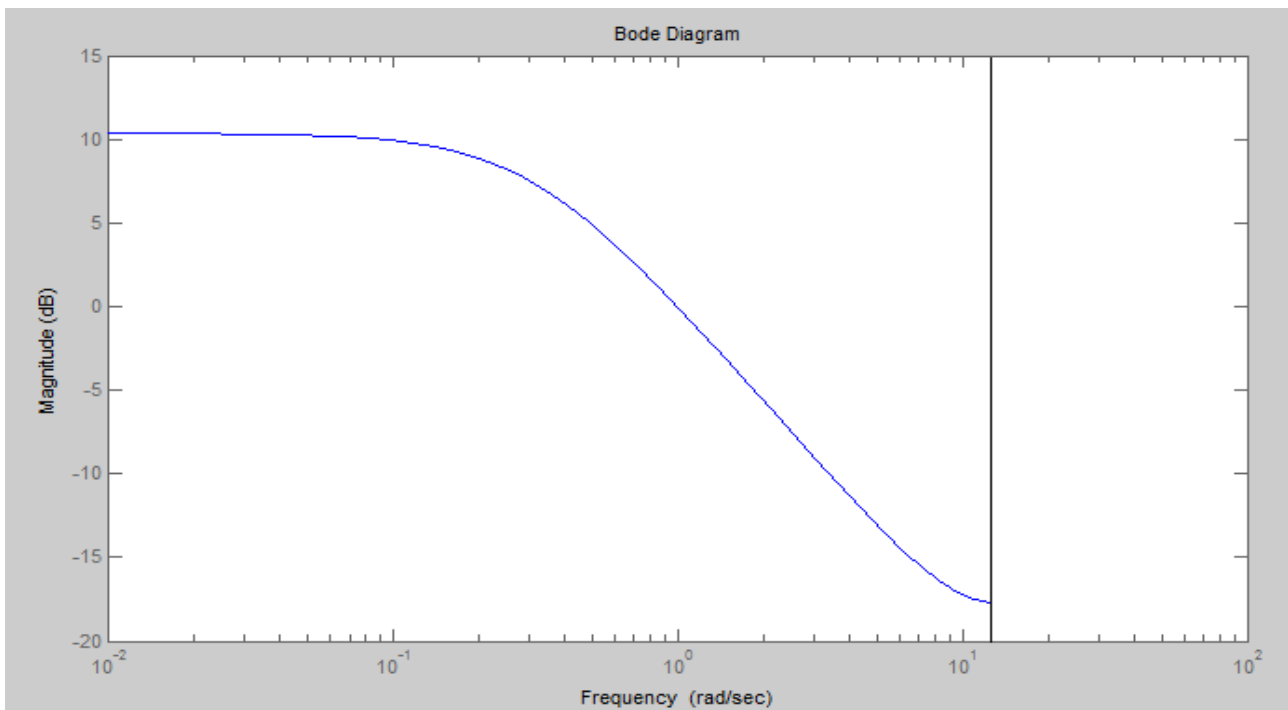


Figura 64: Diagrammi di modulo per la seconda componente di  $G_d(z)$  (Controllo algebrico disaccoppiato)

Notiamo allora che per basse frequenze il disturbo risulta amplificato in uscita, mentre si ha una certa reiezione ad alte frequenze, comportamento molto simile a quello osservato negli altri controlli esaminati in questa sezione.

Generando il disturbo in modo random (anche se con i vincoli descritti sopra per frequenza e ampiezza), si può simulare mediante lo schema Simulink in figura 65.

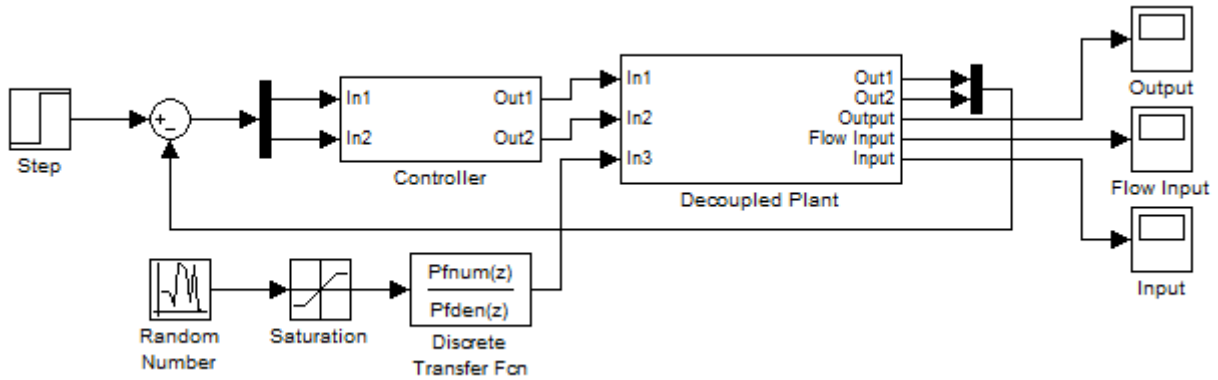


Figura 65: Schema per le simulazioni dell'effetto del disturbo sul controllo disaccoppiato

Dove il contenuto del blocco 'Decoupled Plant' è quello riportato in figura 66.

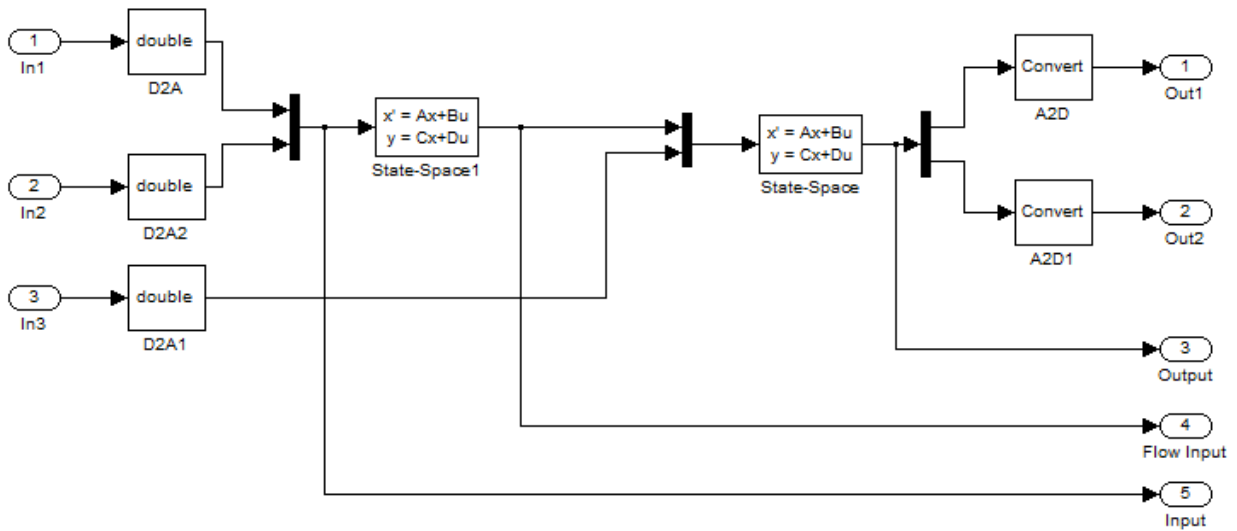


Figura 66: contenuto del blocco 'decoupled plant'

I risultati ottenuti per gli andamenti delle altezze sono mostrati in figura 67.

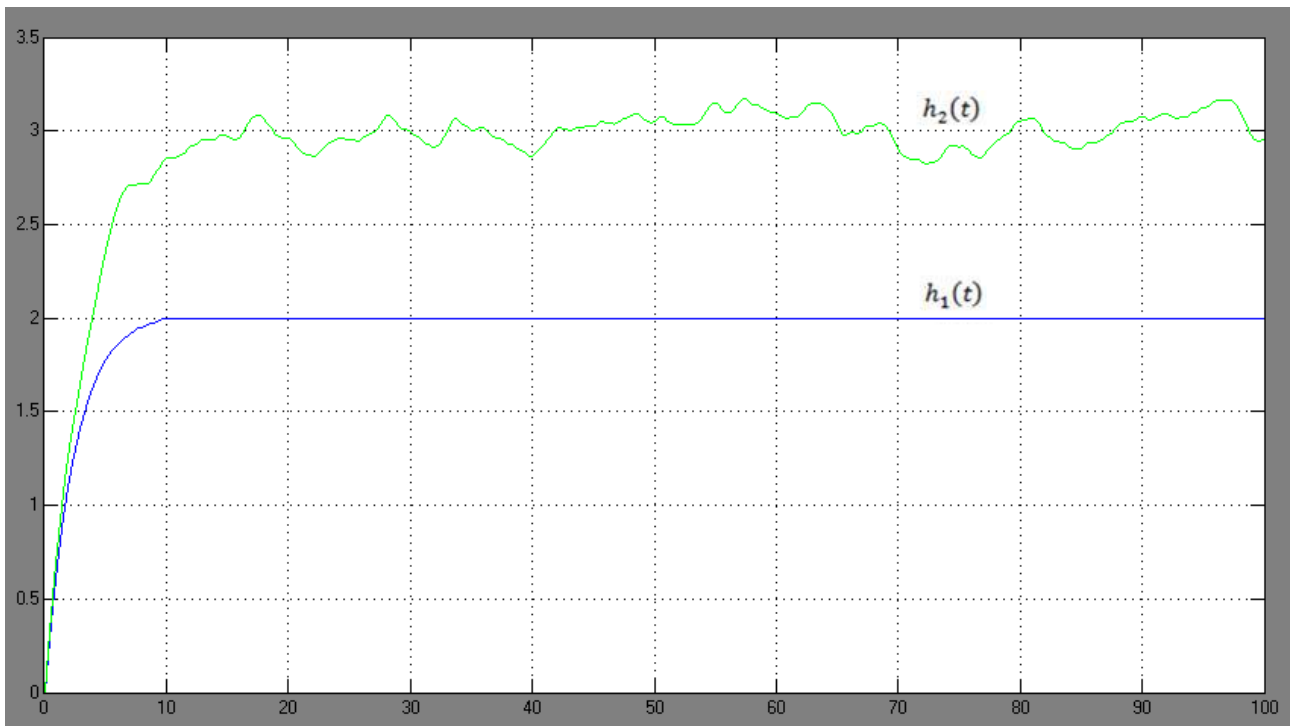


Figura 67: uscite con disturbo (controllo algebrico disaccoppiato)

Dove si può osservare che il disturbo non ha appunto nessuna influenza su  $h_1(t)$ , comportamento che ci si aspettava osservando la relativa componente di  $G_d(z)$ .

Notiamo invece che anche stavolta l'andamento di  $h_2(t)$  è pesantemente influenzato dal disturbo, comportamento atteso osservando la forma del diagramma del modulo per la relativa componente di  $G_d(z)$ .

In figura 68 si riporta l'ingrandimento della evoluzione per  $h_2(t)$ .

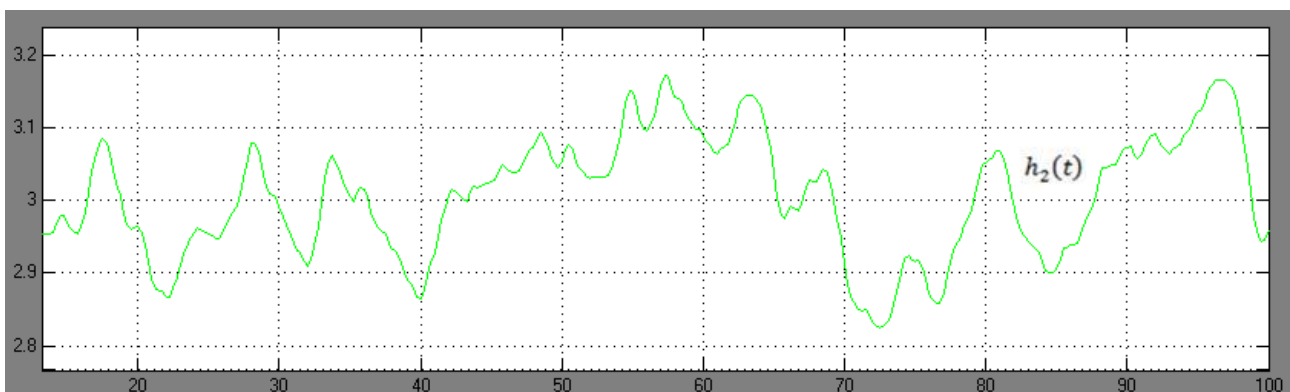


Figura 68: ingrandimento della evoluzione di  $h_2(t)$  (controllo algebrico disaccoppiato)

In figura 69 è inoltre riportato l'andamento per i flussi in ingresso al sistema

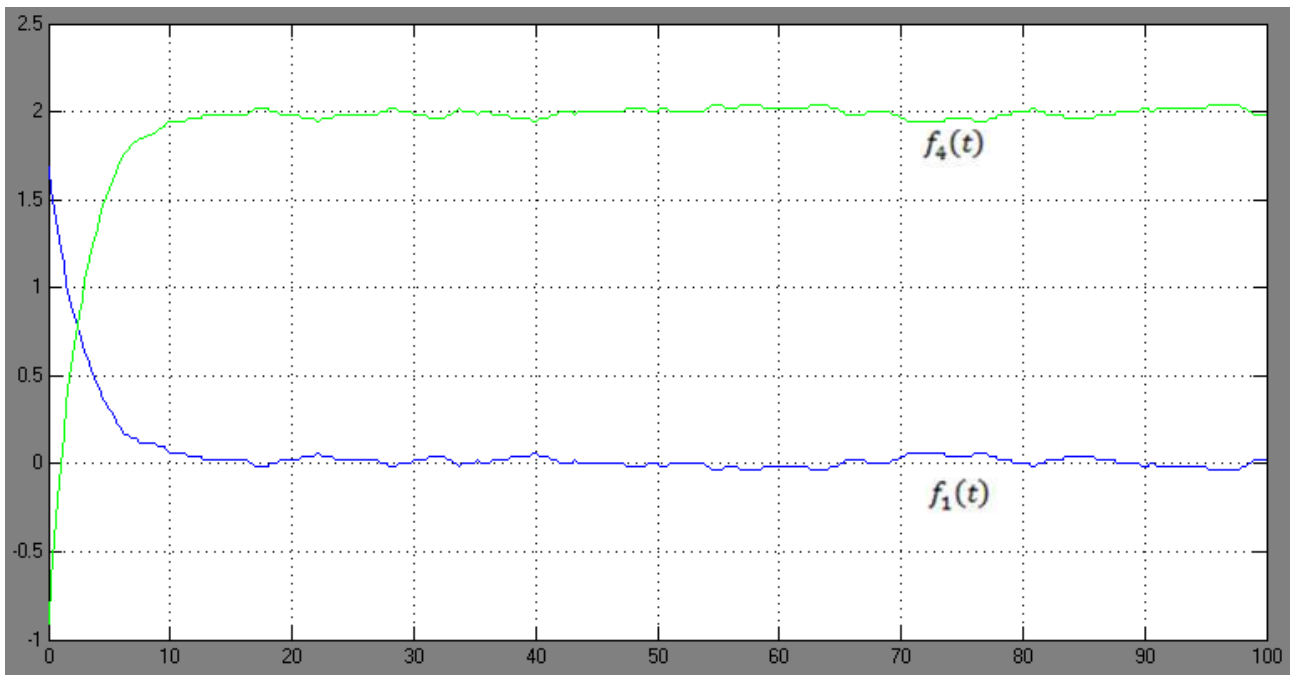


Figura 69: flussi con disturbo (controllo algebrico disaccoppiato)

Si può concludere osservando che, se il disturbo ha una natura verosimile (come nella analisi precedente), le uscite non assumono mai un andamento molto preoccupante, questo deriva anche dal fatto che, per come è fatto l'impianto originale, un flusso (anche di disturbo) subisce una integrazione per andare ad operare sullo stato e sulle uscite (i livelli dei liquidi).

1. File “riccati.m”: esegue l’algoritmo per il calcolo del controllore LQ

Ingressi:     A (Matrice discreta di stato)  
              B (Matrice discreta degli ingressi)  
              Q (Peso dello stato)  
              R (Peso degli ingressi)  
Uscite:       C (Controllore LQ)  
              E (Posizione dei poli in ciclo chiuso)

##### riccati.m #####

```
function [C,E] = riccati(A,B,Q,R)

% Hamiltoniana
Hc = [ A + B * R^(-1) * B' * A'^(-1) * Q , -B * R^(-1) * B' * A'^(-1) ;
      - A'^(-1) * Q , A'^(-1) ];

% Autovalori ed autovettori
[auv,D] = eig(Hc);
j=1;
E=zeros(size(D,1)/2);
V=zeros(size(D,1),size(D,1)/2);
for i=1:size(D)
    if abs(D(i,i)) < 1
        % Autovettori associati ad autovalori stabili
        E(j,j) = D(i,i);
        V(:,j) = auv(:,i);
        j=j+1;
    end
end
X = V(1:size(V,1)/2,:);
L = V(size(V,1)/2+1:size(V,1),:);

% Controllore
C = ( R + B' * L * X^(-1) * B )^(-1) * B' * L * X^(-1) * A;

end
```

2. File “retro\_stato.m” utilizzato per eseguire le simulazioni sul modello accoppiato. Il file esegue la procedura ed invoca direttamente Simulink, configurandolo con i parametri calcolati. Le seguenti variabili sono di interesse all’interno del codice:

- `control` : tipo di controllo utilizzato. Può assumere i seguenti valori:
  - `'pole_place'` % Assegnamento dei poli
  - `'receding_horizon'` % Orizzonte sfuggente
  - `'riccati_lqr'` % Regolatore LQ
- `model` : modello utilizzato. Può assumere i valori:
  - `'retro_stato_model'` % Modello standard, con EMF
  - `'smb_retro_stato'` % Modello standard, solo SMB
  - `'retro_stato_model_sat'` % Modello con saturatore ed EMF
  - `'retro_stato_model'` % Modello con saturatore, solo SMB
- `interf` : valore utilizzato per regolare l’intensità del disturbo. Varia fra 0 ed 1 e se assume il valore 0, indica che il disturbo non è presente.
- `step_in` : riferimento che il sistema dovrà seguire. E’ un vettore di due componenti per indicare le altezze desiderate dei due serbatoi.
- `Ac, Bc, Cc, Dc` : matrici del modello continuo utilizzate, sono 2x2.
- `sample_time` : intervallo di campionamento per la discretizzazione
- Parametri per i vari controlli:
  - `'pole_place'` -> P : poli del sistema in ciclo chiuso, possono assumere valori compresi fra -1 ed 1.
  - `'receding_horizon'` -> N : dimensione della manovra, intero  $\geq 2$
  - `'riccati_lqr'` -> Qin : peso associato allo stato.  
Rin : peso associato agli ingressi.  
Devono essere entrambe matrici 2x2 simmetriche e definite positive.

```
##### retro_stato.m #####
```

```
%% Controllo
```

```
control='riccati_lqr' %Controllo
model='retro_stato_model' %Modello
interf = 0.15
step_in = [2 3] % Ingresso
```

```
%% Modello continuo
```

```
Ac = [-1 0; 1 0];
Bc = [1 1; 0 -1];
Cc = [1 0; 0 1];
Dc = [0 0; 0 0];
```

```
%% Discretizzazione
```

```
%Campionamento
sample_time = 0.25
%Matrici Discrete
fcn = @(tau) expm(Ac*tau);
Ad = fcn(sample_time);
Bd = quadv(fcn,0,sample_time)*Bc;
```

```

Cd=Cc; Dd=Dc;

%% Calcolo Controllore

I=eye(2);
switch(control)
    case 'pole_place' %Piazzamento Poli
        P = [0.9 0.95] %Posizione poli
        K = place(Ad,Bd,P)

    case 'receding_horizon' %Controllo a orizzonte sfuggente
        N = 10 %Dimensione della manovra
        G = 0; %Gramiano di controllabilità
        for i = 1:N
            G = G + Ad^(i-1)*Bd*Bd'*Ad^(i-1);
        end
        K = Bd'*Ad^(N-1)*G^(-1)*Ad^N

    case 'riccati_lqr' % Controllo LQR
        Qin = [1 0; 0 1] %Costo Uscite
        Rin = [7 0; 0 1] %Costo Ingressi
        [K,P] = riccati(Ad,Bd,Qin,Rin) % [K,S,E] = dlqr(Ad,Bd,Q,R)
end

Acc = Ad-Bd*K; Bcc = Bd; %Evoluzione in ciclo chiuso
F = Bcc^(-1)*(I-Acc)*Cd^(-1) %Errore a regime nullo

%% Produzione del disturbo

if interf > 0

    % Aggiunta del disturbo
    Bc = [1 1 0; 0 -1 -1];
    Dc = [0 0 0; 0 0 0];
    sys = c2d(ss(Ad,Bc,Cc,Dc),sample_time);
    E = sys.b(:,3);

    % Fdt del disturbo
    z = tf('z',sample_time); s = tf('s');
    Gd = minreal((z*I-Ad+Bd*K)^(-1)*E)
    % Filtro di frequenza
    Pf = c2d(1/(s+1),sample_time,'tustin');
    Pfnum = Pf.num{1}, Pfden = Pf.den{1}
else
    E = [0 0]';
    Pfnum = 1, Pfden = 1
end

if (strcmp(model,'retro_stato_model_sat') ||
strcmp(model,'smb_retro_stato_sat'))
    Ad, Bd, Cd, Dd
else
    Ad, Bd = [Bd E]
    Cd, Dd = [Dd zeros(2,1)]
end

%% Esecuzione

open_system(model);
sim(model);

```

3. File “decoupled\_control.m” utilizzato per eseguire le simulazioni sul modello SISO. Il file esegue la procedura ed esegue direttamente Simulink, configurandolo con i parametri calcolati. I seguenti valori sono di interesse all’interno del codice:

- control : tipo di controllo utilizzato. Può assumere i seguenti valori:
  - 'algebraic\_rrl'                   % Controllo algebrico
  - 'proportional'                   % Controllo proporzionale
- interf : valore utilizzato per il disturbo. Varia fra 0 ed 1 e se assume il valore 0, indica che il disturbo non è presente.
- step\_in : riferimento che il sistema dovrà seguire. E’ un vettore di due componenti per indicare le altezze dei due serbatoi.
- Ac,Bc,Cc,Dc : matrici del modello continuo utilizzate, sono 2x2.
- sample\_time : intervallo di campionamento per la discretizzazione
- Parametri per i vari controlli:
  - 'algebraic\_rrl' ->           roh1 : costo del primo ingresso.  
                                  roh2 : costo del secondo ingresso.  
                                  Assume valori positivi.
  - 'proportional' ->           Kp1 : guadagno del primo impianto.  
                                  Kp2 : guadagno del secondo impianto.

NOTA: affinché il controllore lasci il sistema stabile deve valere:  
 $0 < Kp1 < 8; -8 < Kp2 < 0$

##### decoupled\_control.m #####

%% Controllo

```
model='decoupled_model' %Modello
control='algebraic_rrl' %Controllo
interf = 0.2
step_in = [2 3] %Ingresso
```

%% Modello continuo

% Matrici

```
Ac = [-1 0; 1 0]
Bc = [1 1; 0 -1]
Cc = [1 0; 0 1]
Dc = [0 0; 0 0]
```

% Fdt

```
[NUM1,DEN1] = ss2tf(Ac,Bc,Cc,Dc,1);
[NUM2,DEN2] = ss2tf(Ac,Bc,Cc,Dc,2);
G = minreal(tf({NUM1(1,:),NUM2(1,:) ; NUM1(2,:),NUM2(2,:)},
               {DEN1,DEN2;DEN1,DEN2}));
```

%% Disaccoppiamento

% Disaccoppiatore

```
Kpre = minreal([1, -G(1,2)/G(1,1); -G(2,1)/G(2,2), 1]);
Ksys = ss(Kpre);
AK = Ksys.a, BK = Ksys.b
CK = Ksys.c, DK = Ksys.d
```

% Fdt disaccoppiata

```
P = minreal(G*Kpre);
```

```

%% Discretizzazione

%Campionamento
sample_time = 0.25
%Fdt discrete
[Pd1num,Pd1den] = c2dm(P(1,1).num{1},P(1,1).den{1},sample_time)
[Pd2num,Pd2den] = c2dm(P(2,2).num{1},P(2,2).den{1},sample_time)
Pd1 = minreal(tf(Pd1num,Pd1den,sample_time));
Pd2 = minreal(tf(Pd2num,Pd2den,sample_time));

%% Calcolo Controllore

I=eye(2);
switch(control)
    case 'algebraic_rrl' %Controllo algebrico
        roh1 = 6; roh2 = 10; % Costi
        % Fdt Inverse
        z = tf('z',sample_time);
        Pd1i = minreal(Pd1num(2)/(Pd1den(1)*z^(-1)+ Pd1den(2)));
        Pd2i = minreal(Pd2num(2)/(Pd2den(1)*z^(-1)+ Pd2den(2)));
        % Risoluzione lrr
        locus1 = zero(1+1/roh1*Pd1*Pd1i);
        locus2 = zero(1+1/roh2*Pd2*Pd2i);
        for i=1:2 %Poli ottimi
            if locus1(i) < 1, p1 = locus1(i), end
            if locus2(i) < 1, p2 = locus2(i), end
        end
        % Modello
        M1 = minreal((1-p1)/(z-p1));
        M2 = minreal((1-p2)/(z-p2));
        M1num = M1.num{1}; M1den = M1.den{1};
        M2num = M2.num{1}; M2den = M2.den{1};
        % Controllore determinato algebricamente
        K1 = minreal(1/Pd1*M1/(1-M1));
        K2 = minreal(1/Pd2*M2/(1-M2));
        K1num = K1.num{1}, K1den = K1.den{1}
        K2num = K2.num{1}, K2den = K2.den{1}

    case 'proportional' %Controllo proporzionale
        Kp1 = 0.435; Kp2 = -0.3;
        K1 = Kp1; K2 = Kp2;
        K1num = Kp1, K1den = 1
        K2num = Kp2, K2den = 1

end

%% Produzione del disturbo

if interf > 0
    % Matrici
    Bc = [1 1 0; 0 -1 -1]
    Dc = [0 0 0; 0 0 0]
    % Fdt
    [NUM3,DEN3] = ss2tf(Ac,Bc,Cc,Dc,3);
    Gy = minreal(tf({NUM3(1,:);NUM3(2,:)},{DEN3;DEN3}));
    Gyd = c2d(Gy,sample_time);
    % Aggiunta del disturbo
    E = Bc(:,3);
    % Fdt del disturbo
    z = tf('z',sample_time); s = tf('s');

```

```

Gd = minreal((I+[Pd1 0; 0 Pd2]*[K1 0; 0 K2])^(-1)*Gyd)
% Filtro di frequenza
Pf = c2d(1/(s+1),sample_time,'tustin');
Pfnum = Pf.num{1}, Pfden = Pf.den{1}
else
Bc = [1 1 0; 0 -1 0]
Dc = [0 0 0; 0 0 0]
Pfnum = 1, Pfden = 1
end

%% Esecuzione

open_system(model);
sim(model);

```

1. Modello “retro\_stato\_model.mdl”. Usa una Embedded Matlab Function (EMF) per simulare il comportamento del sistema, con il seguente codice:

```
function [y,u_k,x_k_1] = discrete_space(A,B,C,D,F,K,r_k,if_k,x_k)

u_k = F*r_k - K*x_k;
x_k_1 = A*x_k + B(:,1:2)*u_k + B(:,3)*if_k;
y = C*x_k + D(:,1:2)*u_k + D(:,3)*if_k;

end
```

La Matrice  $B(:,3)$  diventa una colonna di zeri se non siamo in presenza di disturbo.

Il modello è riportato in figura70.

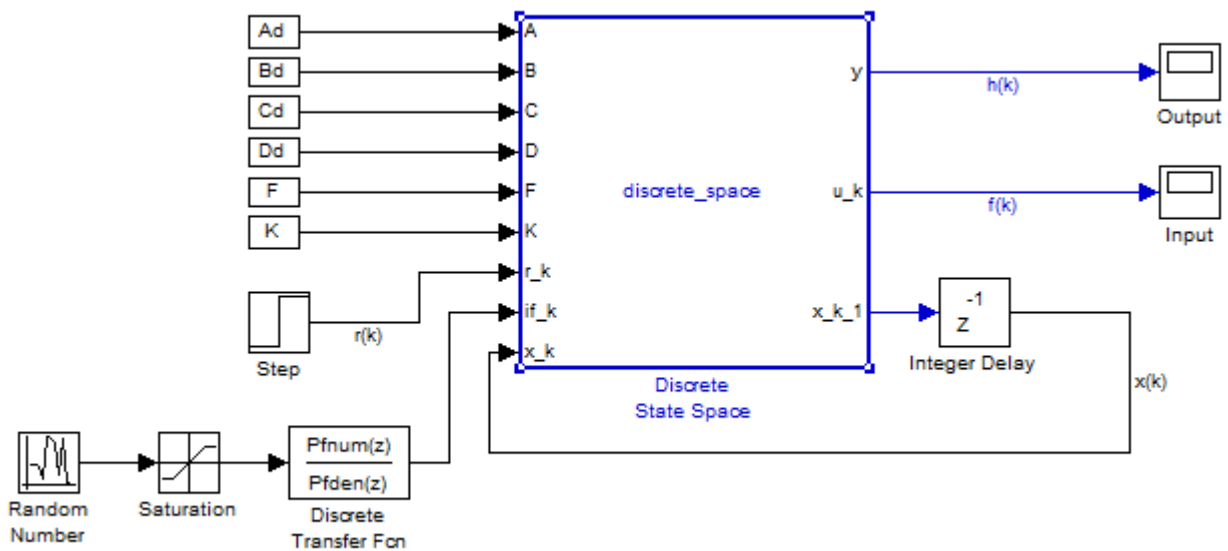


Figura 70: retro\_stato\_model.mdl

2. Modello “retro\_stato\_model\_sat.mdl”. Usa una Embedded Matlab Function per simulare il comportamento del sistema MIMO accoppiato, con il seguente codice:

```
function [y,u_k,x_k_1] = discrete_space(A,B,C,D,F,K,r_k,x_k)

u_k = F*r_k - K*x_k;
for i = 1:2
    if u_k(i) < -1, u_k(i) = -1; end
    if u_k(i) > 5, u_k(i) = 5; end
end

x_k_1 = A*x_k + B*u_k;
y = C*x_k + D*u_k;
end
```

In questo caso non è prevista la presenza di disturbo. Gli ingressi vengono mantenuti all'interno delle soglie di saturazione con una tipica struttura `if ... then ... else`.

Il modello è riportato in figura71.

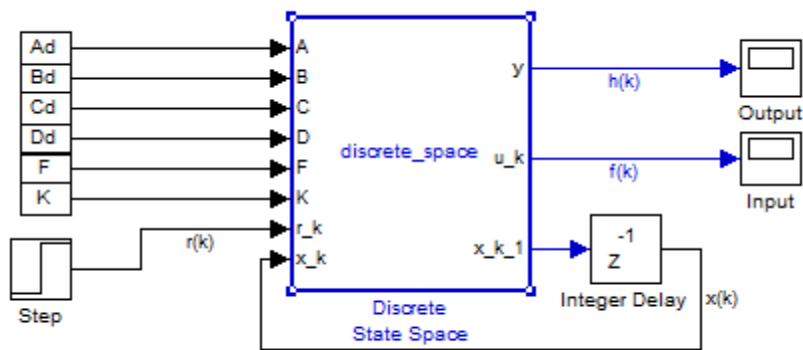


Figura 71: retro\_stato\_model\_sat.mdl

- Modello “smb\_retro\_stato.mdl”. Usa esclusivamente blocchi Simulink per simulare il comportamento del sistema MIMO accoppiato, eventualmente in presenza di disturbo.

Il modello è riportato in figura72.

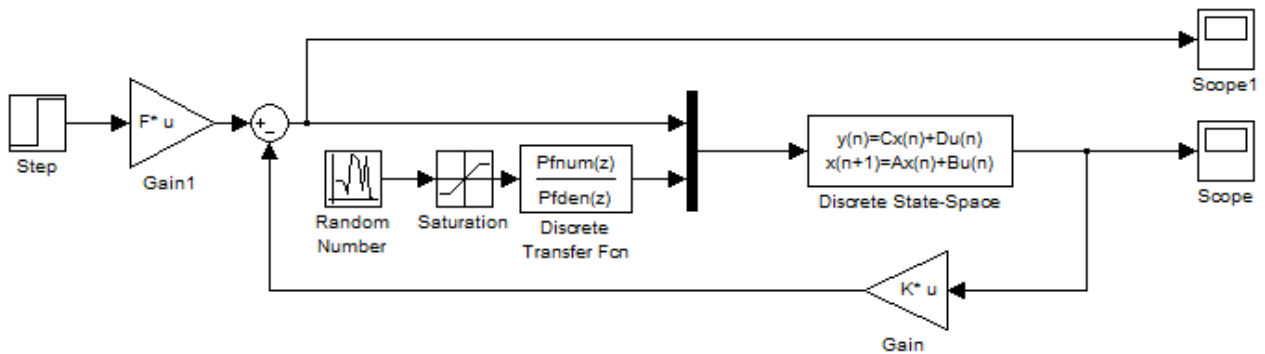


Figura 72: smb\_retro\_stato.mdl

- Modello “smb\_retro\_stato\_sat.mdl”. Usa esclusivamente blocchi Simulink per simulare il comportamento del sistema MIMO accoppiato in presenza di saturatore.

Il modello è riportato in figura73.

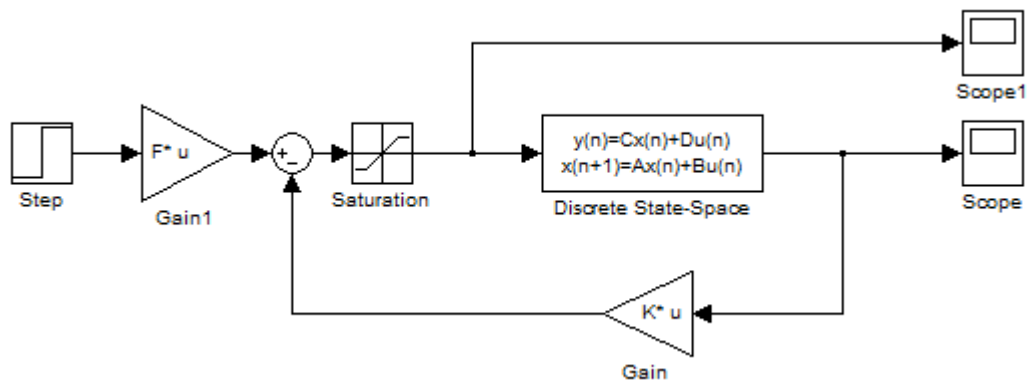


Figura 73: smb\_retro\_stato\_sat.mdl

5. Modello “decupled\_model.mdl”. Simula il comportamento del sistema disaccoppiato (equivalente a due sistemi SISO). La struttura generale è riportata in figura 74.

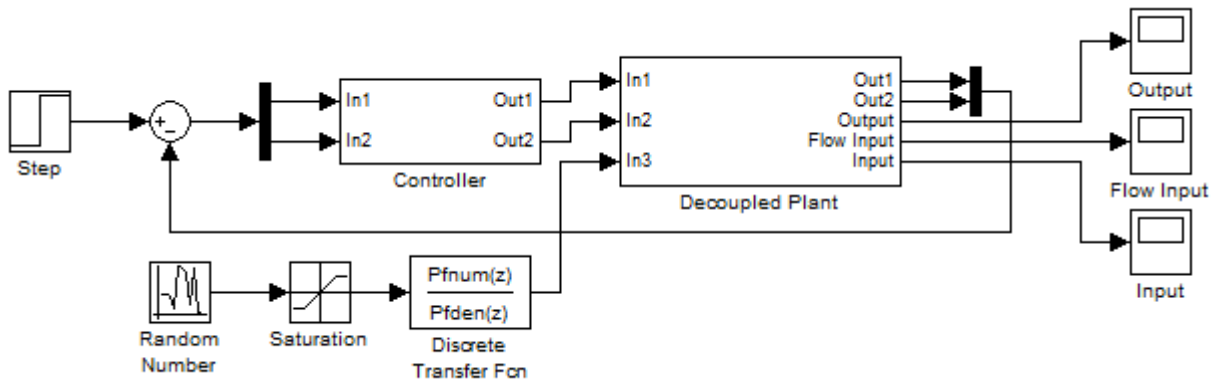


Figura 74: decupled\_model.mdl

Il controllore a sua volta è un sottomodello composto dai due distinti controllori, uno per ogni anello di reazione, come mostrato in figura 75.

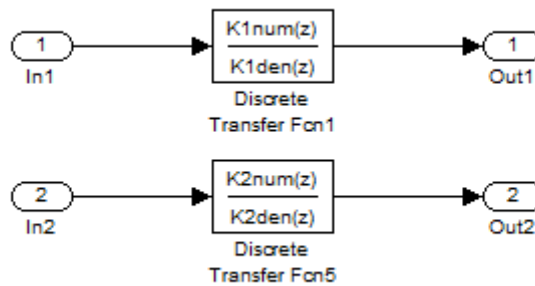


Figura 75: decupled\_model/Controller

Il sottomodello relativo all’impianto disaccoppiato è rappresentato in figura 76.

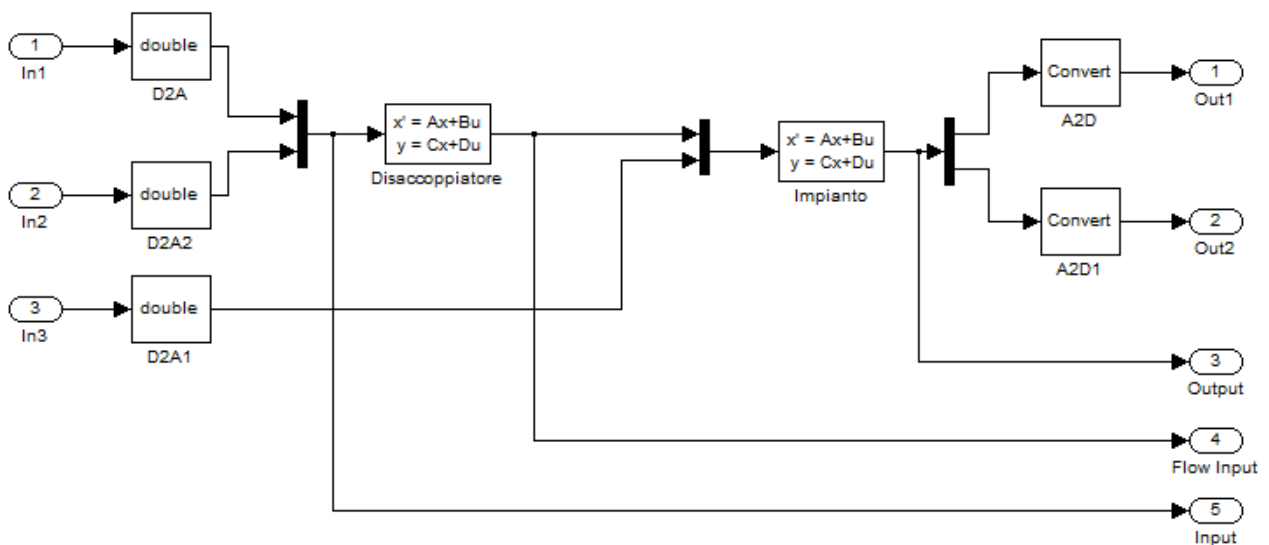


Figura 76: decupled\_model/Decoupled Plant

In figura si osserva come l’impianto disaccoppiato sia in realtà composto dai due blocchi disaccoppiatore/impianto (entrambi analogici) racchiusi all’interno di convertitori D/A ed A/D per interagire col sistema di controllo digitale.

Al progetto è allegata una GUI (Graphical user interface) che permette di testare più rapidamente i risultati ottenuti al variare dei modelli e dei parametri utilizzati. La GUI permette all'utente di interagire in modo semplice e veloce con gli script Matlab ed i modelli Simulink sin qui mostrati.

I parametri manipolabili mediante l'interfaccia grafica sono i seguenti:

- Tipo di sistema:  
sono disponibili sia il caso accoppiato (sistema MIMO) che quello disaccoppiato (due sistemi SISO)
- Tipo di controllore:  
a seconda del sistema scelto sono disponibili:
  - Assegnamento dei poli, che permette la scelta dei poli  $P$
  - Regolatore Lineare Quadratico, che consente la modifica di  $Q$  ed  $R$
  - Controllo ad orizzonte sfuggente, su cui è possibile decidere la dimensione  $N$  della manovra.
  - Controllo algebrico, con scelta dei  $\rho_1$  e  $\rho_2$  per i diversi SISO
  - Controllo Proporzionale, con scelta di  $Kp_1$  e  $Kp_2$
- Modello Simulink:  
oltre al modello standard è eventualmente disponibile un modello che preveda la presenza del saturatore in ingresso. Inoltre in alcuni casi è previsto sia il modello che sfrutta le Embedded Matlab Functions (EMF), sia il corrispettivo realizzato unicamente con blocchi Simulink.
- Riferimenti:  
 $h_{ref1}$  ed  $h_{ref2}$  individuano i riferimenti di altezza dei due serbatoi
- Disturbo:  
introduce il disturbo ove consentito (i modelli con saturatore non lo permettono).
- Tempo di simulazione:  
individua il tempo di durata della simulazione.
- Open Model:  
Apre o meno il Modello richiesto in Simulink.

L'interfaccia grafica, dopo aver eseguito gli script, eseguirà automaticamente la simulazione.

Sono immediatamente disponibili:

- Risultati numerici:  
ovvero il controllore e, se è presente il disturbo, la funzione di sensitività. Nel caso di sistema accoppiato il controllore sarà costituito da 2 matrici  $K$  ed  $F$ , mentre nel caso di sistema disaccoppiato ci sarà un'unica matrice  $K$  diagonale.
- Altezze:  
grafici delle altezze dei serbatoi lungo il tempo di simulazione.
- Flussi:  
grafici dei flussi d'ingresso lungo il tempo di simulazione.
- Disturbo:  
visualizza il diagramma di Bode del modulo della funzione di sensitività  $G_d$ .

Per comodità dell'utente, i campi editabili sono protetti di modo che non possano essere introdotti valori incongruenti. Si faccia riferimento alle figure seguenti, che evidenziano le funzionalità introdotte dall'interfaccia grafica.

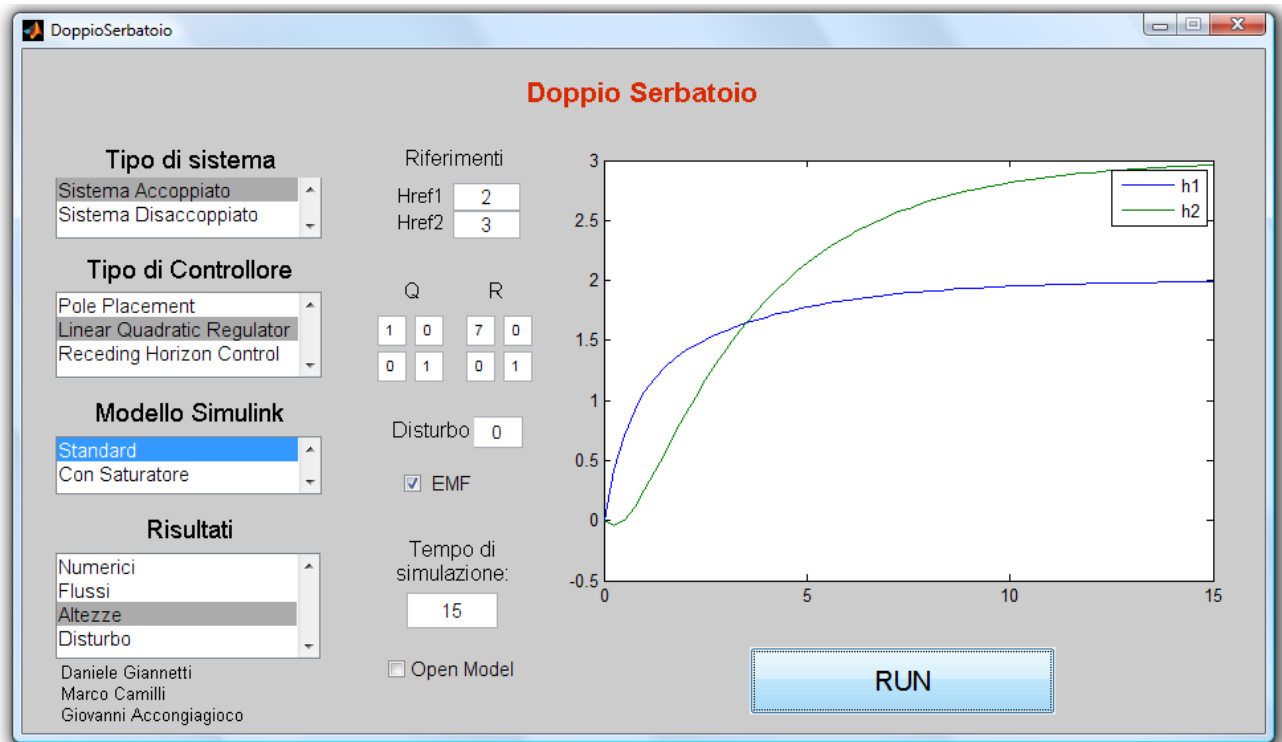


Figura 77: GUI, esempio di visualizzazione delle altezze dei serbatoi

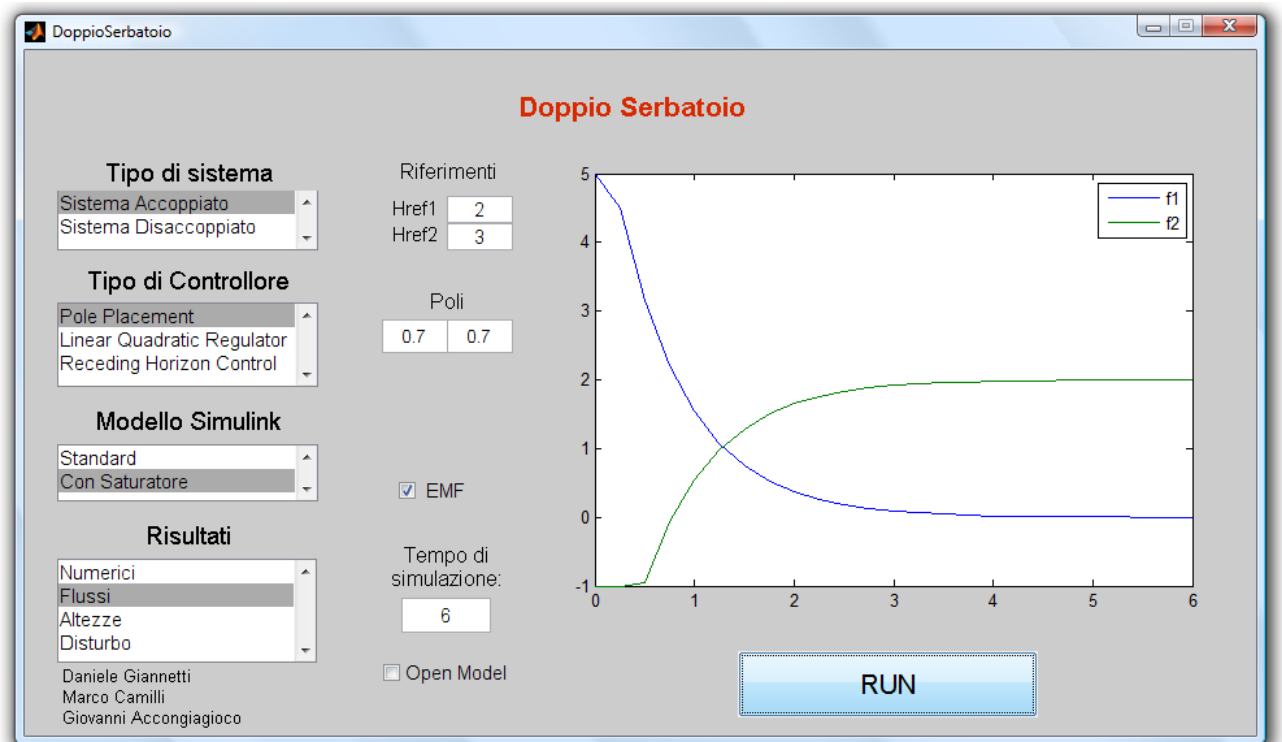


Figura 78: GUI, esempio di visualizzazione dei flussi d'ingresso

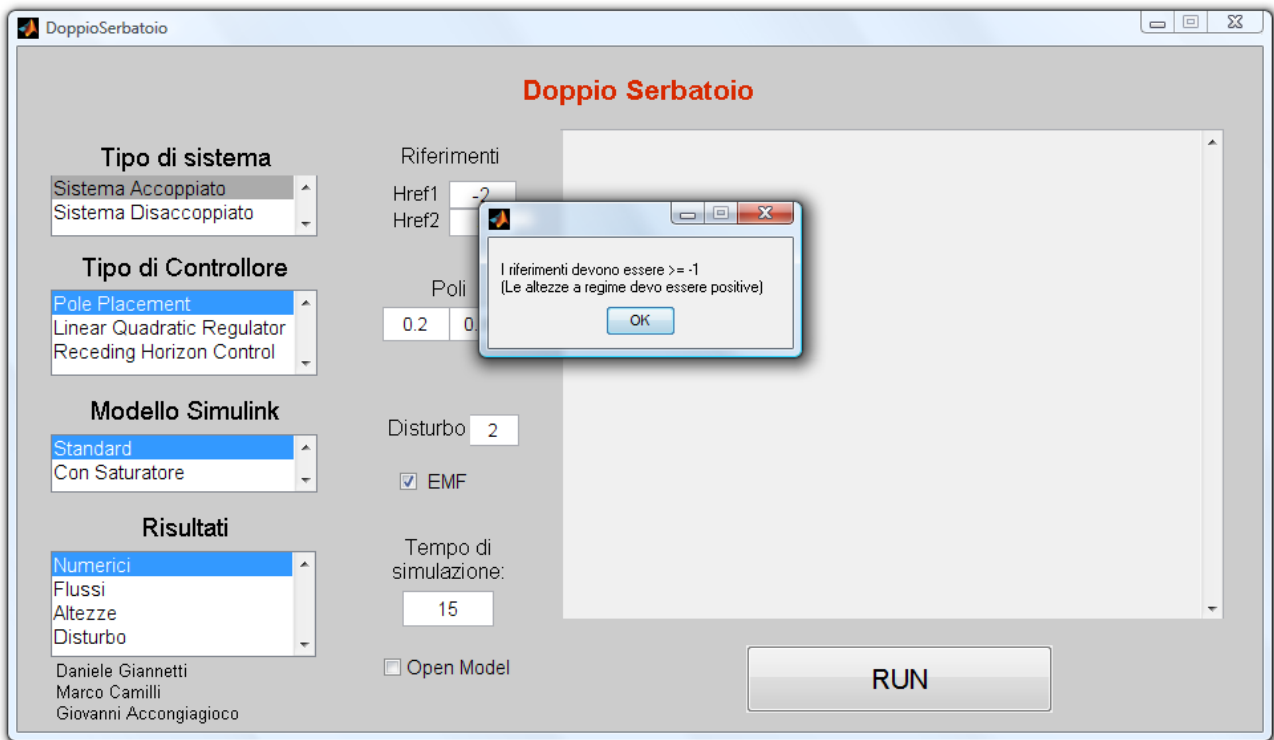


Figura 79: GUI, esempio di errore di input

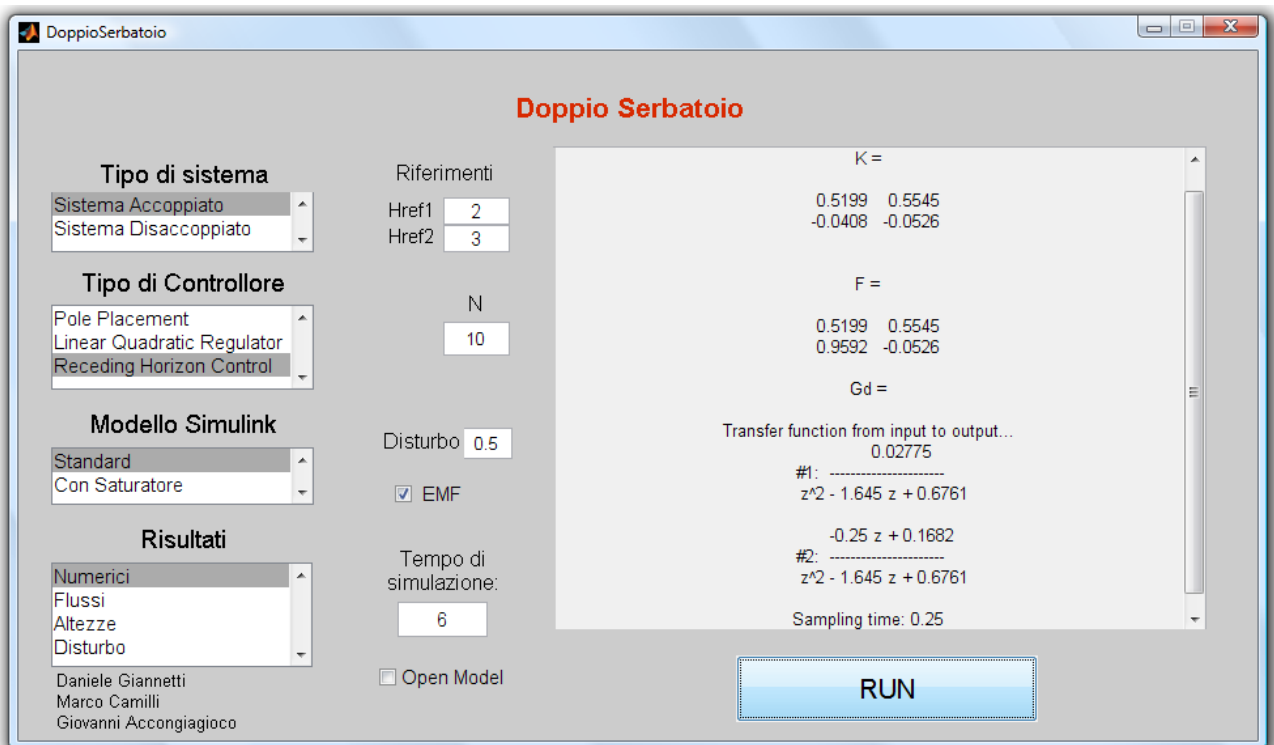


Figura 80: GUI, esempio di visualizzazione dei dati numerici

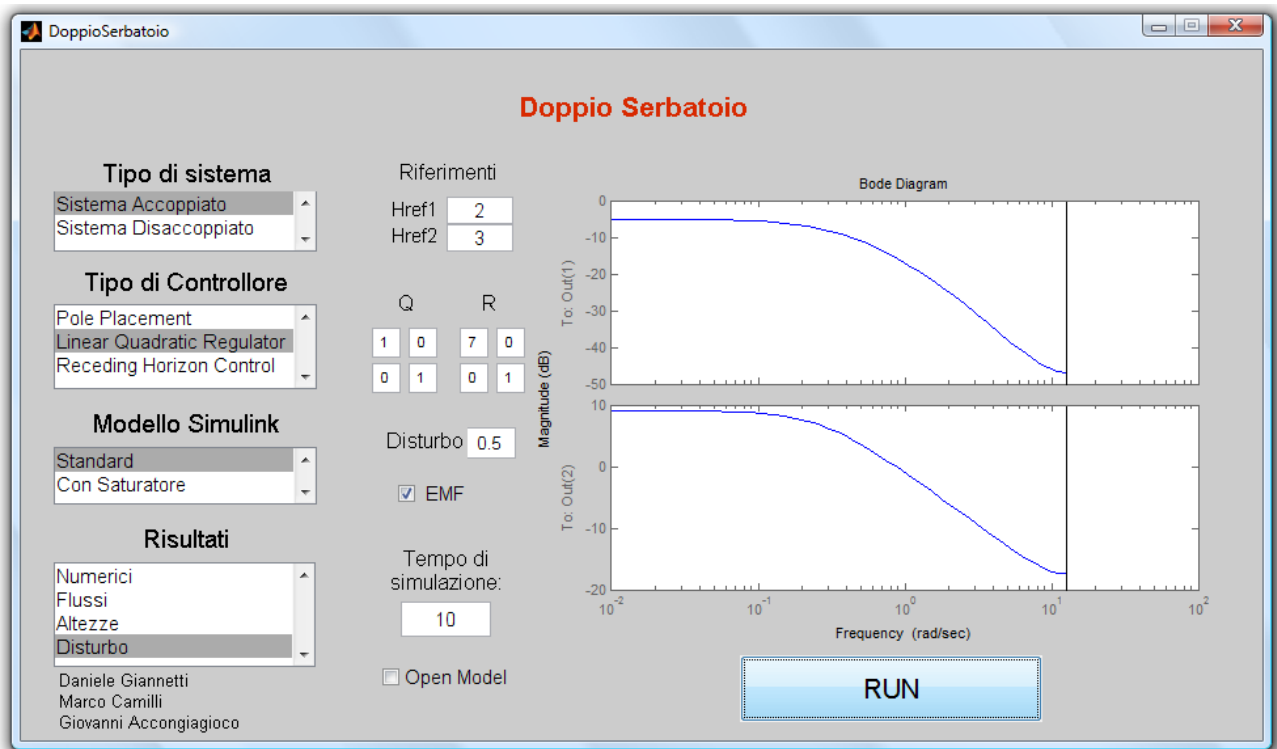


Figura 81: GUI, esempio di visualizzazione dei diagrammi di Bode



Figura 82: GUI, esempio di visualizzazione delle uscite in presenza di disturbo