

Corso di Laurea Specialistica in Ingegneria Elettronica Università degli Studi di Cagliari



Osservatori per sistemi a commutazione:

sviluppo metodologico e applicazione ad un modello fisico con tre serbatoi

Tesi di Laurea di

Stefano Salis

Relatore:

Prof. Alessandro Giua

Anno Accademico 2007/2008



Corso di Laurea Specialistica in Ingegneria Elettronica Università degli Studi di Cagliari



Osservatori per sistemi a commutazione:

sviluppo metodologico e applicazione ad un modello fisico con tre serbatoi

Tesi di Laurea di

Stefano Salis

Relatore:

Prof. Alessandro Giua

Anno Accademico 2007/2008

...alla mia famiglia

Sommario

Lo scopo della tesi è quello di progettare un osservatore dello stato discreto per sistemi a commutazione autonomi non-lineari noto lo stato continuo del sistema. In una seconda fase si cerca di estendere il problema al caso in cui lo stato continuo del sistema non sia noto ma anche esso debba essere osservato. L'osservatore proposto viene applicato a un sistema fisico con tre serbatoi.

Il lavoro svolto è idealmente diviso in due parti: una parte riguardante l'identificazione e modellazione del sistema fisico con tre serbatoi e una seconda dedicata al progetto dell'osservatore e alla simulazione sul sistema fisico e sul modello in ambiente Matlab/Simulink[®].

Ringraziamenti

Ringrazio la mia famiglia per avermi sempre sostenuto durante tutto il mio percorso di studi.

Ringrazio il Prof. Alessandro Giua per avermi seguito e guidato sempre con grande disponibilità, e per i suoi preziosi suggerimenti.

Infine ringrazio tutti i ragazzi del laboratorio di Automatica, con cui è stato piacevole condividere le ore di lavoro.

Indice

1	Intro	oduzior	ne	1
2	Con	ncetti fondamentali		
	2.1	Model	li in variabili di stato	5
	2.2	Sistem	ii ad eventi discreti	7
	2.3	Sistem	ni Ibridi	7
	2.4	Come	costruire un osservatore	8
		2.4.1	Osservatore di Luenberger	8
		2.4.2	Filtro di Kalman	9
		2.4.3	Osservatori per sistemi non lineari	10
	2.5	Contro	ollore LQR	11
	2.6	Identificazione del modello del sistema		12
		2.6.1	Forma predittiva dei modelli per l'identificazione	13
		2.6.2	Algoritmi utilizzati	13
3	Desc	crizione	e del processo fisico	17
	3.1	Multit	ank System	17
	3.2	Modello matematico		19
		3.2.1	Flusso laminare di un fluido ideale	21
		3.2.2	Modello per n serbatoi in cascata	22
		3.2.3	Modello non-lineare del sistema three-tank	23
		3.2.4	Modello linearizzato del sistema three-tank	25
		3.2.5	Modello Switching System	26
	3.3	Model	lo Matlab-Simulink	28

4	Iden	ntificazione del processo		33
	4.1	Curve caratteristiche dei sensori di livello		33
	4.2	Identificazione delle valvole		35
	4.3	Identificazione della pompa		36
	4.4	Identificazione dei parametri del modello		37
	4.5	Confronto tra modello identificato e sistema reale		43
	4.6	Considerazioni sul processo		47
5	Osse	servazione del modo discreto		53
	5.1	Metodologia		53
		5.1.1 Il blocco degli isolatori di modo		53
		5.1.2 I residui		55
	5.2	Simulazioni: confronto tra sistema reale e modello	э	65
	5.3	Simulazioni aggiuntive		69
6	Osse	servazione del modo continuo e discreto		77
	6.1	Metodologia		77
		6.1.1 Il blocco degli isolatori di modo		78
		6.1.2 Osservatore continuo (CO)		78
	6.2	Simulazioni sul modello		79
7	Con	nclusioni 8		85
Ар	pend	dici		87
A	Lista	tati Matlab		89
	A.1	Identificazione dei parametri del modello		89
		A.1.1 Procedura 1		89
		A.1.2 Procedura 2		92
	A.2	Isolatori di modo - funzioni embedded		96
		A.2.1 Caso in cui lo stato continuo è noto		96
		A.2.2 Caso in cui anche lo stato continuo deve e	ssere osservato	104
	A.3	Osservatore Continuo - funzione embedded		116
	A.4	4 Osservatore di Modo		
		A.4.1 Algoritmo 1 - funzione embedded		120
		A.4.2 Algoritmo 2 - funzione embedded		121

INDICE	vii
B Acronimi	123
Bibliografia	125

Elenco delle figure

1.1	Multitank System 2
2.1	Descrizione in variabili di stato
3.1	Multitank System
3.2	Multitank System - particolare Valvole 18
3.3	Multitank System - Schema
3.4	Sistema con controllo della pompa
3.5	Deflusso di un fluido ideale
3.6	Geometria dei tre serbatoi
3.7	Sistema three-tank - diagramma a stati
3.8	Three-tank - modello INTECO 29
3.9	Three-tank - modello Simulink open loop 29
3.10	Three-tank - modello Simulink under mask
3.11	Comportamento del modello three-tank quando il livello minimo è fissato a 1 mm
4.1	Caratteristica Sensore 1 - pallini per le misurazioni, linea continua per la retta interpolante
4.2	Cratteristica di una valvola controllata
4.3	Curva caratteristica sperimentale della pompa in continua
4.4	Upper tank Confronto dati raccolti - modello Procedura 1
4.5	Middle tank Confronto dati raccolti - modello Procedura 1
4.6	Lower tank Confronto dati raccolti - modello Procedura 1
4.7	Upper tank Confronto dati raccolti - modello Procedura 2
4.8	Middle tank Confronto dati raccolti - modello Procedura 2

4.9	Lower tank Confronto dati raccolti - modello Procedura 2	42
4.10	Upper tank - Confronto tra sistema reale e modello in Simulink $\ldots \ldots \ldots$	44
4.11	Middle tank - Confronto tra sistema reale e modello in Simulink $\ldots \ldots$	44
4.12	Lower tank - Confronto tra sistema reale e modello in Simulink $\ldots \ldots \ldots$	45
4.13	Comportamento imprevisto della pompa	46
4.14	Pannello Manual Setup	46
4.15	Livelli sistema controllato	48
5.1	I residui r_0 , r_1 , r_2 , r_3 al variare dei modi	58
5.2	Le derivate dei residui r_0 , r_1 , r_2 , r_3 al variare dei modi $\ldots \ldots \ldots \ldots \ldots$	59
5.3	Simulazione finalizzata all'individuazione della soglia s_1 associata a $d_1 \ldots \ldots$	60
5.4	Simulazione finalizzata all'individuazione della soglia s_0 associata a $d_0 \ldots$	61
5.5	Simulazione finalizzata all'individuazione della soglia s_2 associata a d_2	62
5.6	Simulazione finalizzata all'individuazione della soglia s_3 associata a d_3	62
5.7	Andamento dei residui R_0 , R_1 , R_2 e R_3 , definiti nell'algoritmo 2, al variare dei modi	64
5.8	Modi osservati sul modello del three-tank utilizzando l'algoritmo 1	66
5.9	Modi osservati sul sistema three-tank reale utilizzando l'algoritmo 1 $\ldots \ldots$	66
5.10	Modi osservati sul modello del three-tank utilizzando l'algoritmo 2	67
5.11	Modi osservati sul sistema three-tank reale utilizzando l'algoritmo 2	68
5.12	Confronto dei risultati ottenuti osservando il sistema reale utilizzando i due algoritmi proposti	69
5.13	Simualzione 1 - modi osservati utilizzando l'algoritmo 1	71
5.14	Simulazione 1 - modi osservati utilizzando l'algoritmo 2	71
5.15	Simulazione 1 - Livelli nei tre serbatoi	72
5.16	Simulazione 1 - Ingresso di controllo	73
5.17	Simualzione 2 - modi osservati utilizzando l'algoritmo 1	75
5.18	Simualzione 2 - modi osservati utilizzando l'algoritmo 2	75
5.19	Simulazione 2 - Livelli nei tre serbatoi	76
5.20	Simulazione 2 - Ingresso di controllo	76
6.1	Sistema three-tank - diagramma a stati	79
6.2	Modello three-tank senza controllo - modi osservati $\ldots \ldots \ldots \ldots$	80
6.3	Segnale di controllo	81

6.4	Osservazione del modo continuo e discreto - Upper tank	82
6.5	Osservazione del modo continuo e discreto - Middle tank	82
6.6	Osservazione del modo continuo e discreto - Lower tank	83

Elenco delle tabelle

3.1	Eventi	27
4.1	livelli e frequenze dei sensori associate	34
4.2	parametri <i>C</i> e α identificati	43
5.1	Modello del three-tank - Valori assunti dai residui nei diversi modi a regime .	56
5.2	Sistema reale three-tank - Valori assunti dai residui nei diversi modi a regime	56
5.3	Soglie per il modello e per il sistema reale	63
5.4	Pesi α e β del residuo <i>R</i> per il modello e per il sistema reale	65

Capitolo 1

Introduzione

Molti sistemi tra quelli che si possono incontrare nella realtà sono caratterizzati dall'accoppiamento di dinamiche continue con dinamiche ad eventi discreti. I sistemi in cui queste dinamiche coesistono e interagiscono vengono chiamati *ibridi*. I *sistemi ibridi* costituiscono un'area di ricerca relativamente nuova e molto attiva in questi ultimi anni. In questo lavoro ci occuperemo di *sistemi a commutazione*, una sottoclasse dei sistemi ibridi.

In particolare, lo scopo principale della tesi è quello di progettare un osservatore dello stato discreto per sistemi a commutazione autonomi non-lineari noto lo stato continuo del sistema. Con sistema a commutazione autonomo, o switching system, ci riferiamo a un sistema dinamico costituito da un numero finito di sottosistemi a tempo continuo che commutano tra di essi a causa di un insieme di eventi non controllabili che avvengono a istanti sconosciuti.

L'osservatore dello stato discreto che proponiamo è costituito da due blocchi principali. Un blocco è composto da un banco di isolatori di modo in parallelo che genera un numero di residui pari al numero di stati discreti del sistema. Il secondo blocco è costituito da un insieme di regole logiche applicate ai residui per generare una stima dello stato discreto del sistema. Per quel che riguarda quest'ultimo blocco logico, due diversi algoritmi vengono proposti.

Inoltre il lavoro che presentiamo è caratterizzato da una forte connotazione applicativa. Difatti l'osservatore progettato viene applicato ad un sistema fisico a tre serbatoi, vedi Figura 1.1, in cui vengono effettuate diverse simulazioni per verificare la validità delle soluzioni proposte.

Del sistema fisico a tre serbatoi, denominato Multitank System, daremo una descrizione fisica e identificheremo le varie componenti (valvole, pompa e sensori). Disegneremo un modello del sistema e ne identificheremo i parametri con l'ausilio dell'*Optimization Toolbox* e del *System Identification Toolbox* di Matlab[®].



Figura 1.1: Multitank System

Infine verrà esteso il problema dell'osservazione del sistema al caso in cui lo stato continuo non sia noto, presenteremo un osservatore del modo continuo e discreto di cui mostreremo alcune verifiche sperimentali sul modello del sistema soggetto ad alcune restrizioni. Questa parte della tesi è stata introdotta per completezza, ma non verrà corredata da un'adeguato studio dell'osservabilità del sistema e della stabilità dell'osservatore proposto.

Ora in breve diamo una descrizione della struttura della tesi capitolo per capitolo.

Nel Capitolo 2 illustreremo i concetti fondamentali presenti in letteratura che sono alla base di questo lavoro. Si daranno le definizioni di *modello in variabili di stato*, di *sistema ad eventi discreti* e di *sistema ibrido* per introdurre le notazioni che utilizzeremo per descrivere il nostro sistema. Descriveremo gli osservatori dello stato più comuni in letteratura, Luenberger e Kalman, e le loro estensioni ai sistemi non-lineari. Mostreremo come si progetta un controllore LQR. Infine descriveremo i passi principali da seguire per l'identificazione e la modellazione di un sistema reale.

Nel Capitolo 3 introdurremo il sistema fisico a tre serbatoi. Descriveremo il sistema con un modello in variabili di stato non-lineare e col suo corrispettivo modello linearizzato. Inoltre daremo una descrizione ibrida del sistema.

L'identificazione del processo è trattata nel Capitolo 4. Le fasi di identificazione sono state le seguenti:

- curve caratteristiche dei sensori di livello,
- curva caratteristica della pompa in continua,
- curva caratteristica delle valvole,
- identificazione dei parametri del modello matematico.

Inoltre al termine del capitolo faremo alcune considerazioni sul processo e sulle strategie di controllo adottate.

Nel Capitolo 5 presenteremo l'osservatore dello stato discreto, verrà illustrata la metodologia utilizzata per il suo progetto e proporremo due algoritmi per il trattamento dei residui; le scelte prese verranno confortate e accompagnate dai dati sperimentali ricavati nelle simulazioni sul sistema a tre serbatoi.

Nel Capitolo 6 verrà trattato il caso in cui lo stato continuo non sia misurabile direttamente ma anche esso debba essere osservato. Presenteremo un osservatore del modo continuo e discreto accompagnato da alcuni dati sperimentali.

Per concludere, nel Capitolo 7, verrà data una breve analisi dei risultati ottenuti e degli obiettivi raggiunti col presente lavoro di tesi.

Capitolo 2

Concetti fondamentali

In questo capitolo illustreremo i concetti fondamentali presenti in letteratura che sono alla base di questo lavoro. Si daranno le definizioni di *modello in variabili di stato*, di *sistema ad eventi discreti* e di *sistema ibrido* per introdurre le notazioni che utilizzeremo per descrivere il nostro sistema. Descriveremo gli osservatori dello stato più comuni in letteratura, Luenberger e Kalman, e le loro estensioni ai sistemi non-lineari. Mostreremo come si progetta un controllore LQR. Infine descriveremo i passi principali da seguire per l'identificazione e la modellazione di un sistema reale.

2.1 Modelli in variabili di stato

Il primo passo per poter applicare delle tecniche formali allo studio dei sistemi consiste nella descrizione del comportamento del sistema mediante grandezze che evolvono nel tempo. Nel caso dei sistemi ad avanzamento temporale, due sono le possibili descrizioni:

- descrizione ingresso uscita (IU),
- descrizione in variabili di stato (VS).

Le grandezze alla base di una descrizione IU sono le cause esterne (ingressi) al sistema e gli effetti (uscite). Le cause esterne sono delle grandezze che si generano al di fuori del sistema; la loro evoluzione influenza il comportamento del sistema ma non dipende da esso. Gli effetti invece sono delle grandezze la cui evoluzione dipende dalle cause esterne al sistema e dalla natura del sistema stesso.

Dunque, in termini di IU, dato uno specifico andamento degli ingressi, attraverso il sistema S risulta individuato un ben preciso andamento delle grandezze in uscita. Tuttavia in generale l'uscita di un sistema in un certo istante di tempo t non dipende non dipende dal solo ingresso al tempo t, ma dipende anche dall'evoluzione precedente

del sistema. Di questo fatto è possibile tenere conto introducendo una grandezza intermedia tra ingressi e uscite, chiamata stato del sistema (descrizione in VS). Lo stato del sistema gode della proprietà di concentrare in sé l'informazione sul passato e sul presente del sistema.



Figura 2.1: Descrizione in variabili di stato

Un modello in variabili di stato descrive come:

- 1. l'evoluzione dello stato $\vec{x}(t)$ dipende dallo stato $\vec{x}(t) \in \mathbb{R}^n$ e dall'ingresso $\vec{u}(t) \in \mathbb{R}^p$ [equazione di stato];
- 2. l'uscita $\vec{y}(t) \in \mathbb{R}^m$ dipende dallo stato $\vec{x}(t) \in \mathbb{R}^n$ e dall'ingresso $\vec{u}(t) \in \mathbb{R}^p$ [trasformazione d'uscita].

Detto $\dot{\vec{x}}(t) = [\dot{x}_1(t) \dots \dot{x}_n(t)]^T \in \mathbb{R}^n$ con n = dimensione di x = ordine del sistema, vale:

$$\begin{cases} \dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), t) \\ \vec{y}(t) = \vec{g}(\vec{x}(t), \vec{u}(t), t) \end{cases}$$

In questo lavoro prenderemo in considerazione un modello in variabili di stato Single-Input Multiple-Output, non-lineare, dinamico, stazionario, strettamente proprio e a parametri concentrati. Dunque un sistema della forma:

$$\begin{cases} \dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t)) \\ \vec{y}(t) = \vec{g}(\vec{x}(t)) \end{cases}$$

che nel caso di sistemi tempo-discreti si scrive:

$$\begin{cases} \vec{x}(k+1) = \vec{f}(\vec{x}(k), \vec{u}(k)) \\ \vec{y}(k) = \vec{g}(\vec{x}(k)) \end{cases}$$

2.2 Sistemi ad eventi discreti

Intuitivamente, un sistema ad eventi discreti (SED) si può definire come un sistema dinamico il cui comportamento è caratterizzato dall'occorrenza di eventi istantanei con un cadenzamento irregolare non necessariamente noto. Diamo ora una definizione più formale di SED.

Definizione 2.1 (Sistema ad Eventi Discreti). *Un sistema ad eventi discreti è una tupla* $D = (Q, Q_0, E, \Psi, \eta)$ *tale che:*

- Q è un insieme finito di N stati discreti.
- $Q_0 \subseteq Q \ e \ l'insieme \ delle \ condizioni \ iniziali.$
- $E \subseteq Q \times Q$ è una collezione di archi (edges). Ogni arco $e \in E$ è dato da una coppia ordinata di stati discreti, la prima componente è la sorgente s(e), mentre la seconda è il target t(e).
- Ψ è l'insieme finito di simboli discreti d'uscita. Include anche la stringa vuota ϵ che corrisponde a uscite non osservabili.
- $\eta: E \to \Psi$ è la funzione d'uscita, associa ad ogni arco un simbolo discreto d'uscita.

2.3 Sistemi Ibridi

Un sistema ibrido è un sistema il cui comportamento viene descritto mediante un modello che unisce dinamiche ad avanzamento temporale con dinamiche ad eventi discreti. Questo tipo di sistemi dinamici tipicamente possiede variabili che assumono valori appartenenti ad un insieme continuo (solitamente l'insieme dei numeri reali) e variabili che assumono valori che appartengono ad un insieme discreto non necessariamente numerico (l'insieme dei simboli). Un modello di questo tipo può essere utilizzato per descrivere in maniera accurata un'ampia gamma di processi industriali e il loro sistema di controllo e monitoraggio associato.

In questo lavoro ci concentreremo su una sottoclasse dei sistemi ibridi, chiamata sistemi a commutazione (SS: switching systems), dove a ogni stato discreto viene associato un sistema dinamico. Dunque, col termine SS, ci riferiamo a un sistema dinamico costituito da un numero finito di sottosistemi a tempo-continuo e a un insieme di regole logiche che guidano le commutazioni tra di essi.

Definizione 2.2 (Sistema a commutazione). Uno Sistema a commutazione è una tupla

$$S = (D, X, X_0, U, Y, \varepsilon)$$

tale che:

- $D = (Q, Q_0, E, \Psi, \eta)$ è un sistema ad eventi discreti, come in **Definizione 2.1**.
- $X \subseteq \mathbb{R}^n$ è lo spazio dello stato continuo.
- $X_0 \subseteq X$ è l'insieme delle condizioni iniziali nel continuo.
- $U \subseteq \mathbb{R}^p \ e \ Y \subseteq \mathbb{R}^m$ sono, rispettivamente, l'insieme degli ingressi continui di controllo e l'insieme delle uscite osservabili.
- $\{\varepsilon_q\}_{q \in \Omega}$ associa ad ogni stato discreto $q \in Q$ le dinamiche continue tempo-invarianti

$$\varepsilon_q : \vec{x} = f_q(\vec{x}, \vec{u}) \tag{2.1}$$

con uscita $\vec{y} = g_q(\vec{x})$.

2.4 Come costruire un osservatore

L'osservatore dello stato è un sistema dinamico con lo scopo di stimare l'evoluzione di stato di un sistema da osservare. Viene impiegato in tutte le applicazioni in cui lo stato del sistema non è completamente accessibile o di dimensione troppo elevata per predisporre altrettanti trasduttori. Può essere utilizzato anche per grandezze non misurabili direttamente, il cui valore deve essere ricavato attraverso la misura di altre grandezze e l'utilizzo di un modello matematico. In questi casi l'osservatore permette di correggere gli errori introdotti dalle incertezze sul modello.

2.4.1 Osservatore di Luenberger

L'osservatore di Luenberger è uno stimatore dello stato per sistemi lineari. Si consideri il sistema dinamico ingresso-stato-uscita:

$$\begin{cases} \vec{x}(k+1) = \mathbf{A}\vec{x}(k) + \mathbf{B}\vec{u}(k) \\ \vec{y}(k) = \mathbf{C}\vec{x}(k) \end{cases}$$
(2.2)

dove le matrici sono di dimensione opportuna: $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times p}$ e $\mathbf{C} \in \mathbb{R}^{m \times n}$.

La grandezza da stimare x(k) è lo stato del sistema, è quindi tempo-variante. Se il sistema è osservabile, l'output del sistema, y(k), può essere utilizzato per stimare lo stato.

L'output dell'osservatore può essere sottratto dall'output del sistema e poi moltiplicato per una matrice **L** (matrice di guadagno); questo termine è poi aggiunto alle equazioni dello stato dell'osservatore ottenendo le seguenti equazioni (indicheremo le variabili stimate con $\hat{x} \in \hat{y}$):

$$\begin{cases} \vec{x}(k+1) = \mathbf{A}\vec{x}(k) + \mathbf{L}[\vec{y}(k) - \vec{y}(k)] + \mathbf{B}\vec{u}(k) \\ \vec{y}(k) = \mathbf{C}\vec{x}(k) \end{cases}$$
(2.3)

L'osservatore è asintoticamente stabile se l'errore $e(k) = x(k) - \hat{x}(k)$ converge a zero quando $k \to \infty$. Per un osservatore di Luenberger, l'errore soddisfa la relazione $e(k+1) = (\mathbf{A} - \mathbf{LC})e(k)$. Esso è dunque asintoticamente stabile quando la matrice $\mathbf{A} - \mathbf{LC}$ ha tutti gli autovalori all'interno del cerchio unitario.

2.4.2 Filtro di Kalman

Il filtro di Kalman è un efficiente filtro ricorsivo che valuta lo stato di un sistema dinamico a partire da una serie di misure soggette a rumore. Per le sue caratteristiche intrinseche è un filtro ottimo per rumori e disturbi agenti sul sistema gaussiani a media nulla. Trova utilizzo come osservatore dello stato.

Consideriamo il sistema visto in precedenza soggetto a rumore di processo $\vec{v}_x(k)$ e rumore di misura $\vec{v}_y(k)$ (con $\vec{v}_x(k)$ e $\vec{v}_y(k)$ incorrelati, gaussiani a media nulla):

$$\begin{cases} \vec{x}(k+1) = \mathbf{A}\vec{x}(k) + \mathbf{B}\vec{u}(k) + \vec{v}_x(k) \\ \vec{y}(k) = \mathbf{C}\vec{x}(k) + \vec{v}_y(k) \end{cases}$$
(2.4)

Dato il rumore, scriviamo una matrice V, detta matrice di covarianza, come:

$$\mathbf{V} = \left[\begin{array}{cc} \mathbf{Q} & \mathbf{Z} \\ \mathbf{Z}^T & \mathbf{R} \end{array} \right]$$

dove:

 $\mathbf{Q} = \operatorname{Var}[\vec{v}_x(k)]$ di dimensione $n \times n$,

 $\mathbf{Z} = \operatorname{Cov}[\vec{v}_x(k), \vec{v}_y(k)]$ di dimensione $n \times m$,

 $\mathbf{R} = \operatorname{Var}[\vec{v}_{v}(k)]$ di dimensione $m \times m$.

Come ipotesi aggiuntive si prendono: $\mathbf{R} > 0$ (definita positiva), ovvero la componente di rumore ha covarianza non nulla su ogni uscita, $\mathbf{Z} = 0$, ovvero i rumori su stato e uscita sono incorrelati.

Il filtro di Kalman funzina secondo due fasi: la fase di predizione (prediction step), dove date le misurazioni precedenti viene predetto il prossimo stato del sistema, e la fase di aggiornamento (update step), dove viene stimato lo stato corrente del sistema date le misurazioni nel passo corrente. I due passi corrispondono alle seguenti equazioni:

• Predizione:

$$\vec{\hat{x}}(k+1|k) = \mathbf{A}(k)\vec{\hat{x}}(k|k) + \mathbf{B}(k)\vec{u}(k)$$
 (2.5)

$$\mathbf{P}(k+1|k) = \mathbf{A}(k)\mathbf{P}(k|k)\mathbf{A}^{T}(k) + \mathbf{Q}(k)$$
(2.6)

• Aggiornamento:

$$\vec{\hat{x}}(k+1|k+1) = \vec{\hat{x}}(k+1|k) + \mathbf{K}(k+1)(\vec{\gamma}(k+1) - \mathbf{C}\vec{\hat{x}}(k+1|k))$$
(2.7)

$$\mathbf{P}(k+1|k+1) = (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{P}(k+1|k)$$
(2.8)

con:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{C}^{T}\mathbf{V}^{-1}(k+1)$$
$$\mathbf{V}(k+1) = \mathbf{C}\mathbf{P}(k+1|k)\mathbf{C}^{T} + \mathbf{R}(k+1)$$

dove:

- $\vec{x}(k+1|k)$ e **P**(k+1|k) sono, rispettivamente, lo stato predetto e la covarianza dello stato al passo k+1 prima di vedere le misurazioni.
- $\vec{x}(k+1|k+1) \in \mathbf{P}(k+1|k+1)$ sono, rispettivamente, lo stato stimato e la covarianza dello stato al passo k + 1 dopo aver visto le misurazioni.
- V(k+1) è la predizione della covarianza della misura al passo k+1.
- $\mathbf{K}(k+1)$ è il guadagno del filtro di Kalman, che ci dice di quanto la predizione deve essere corretta al passo k+1.

Il valore iniziale della covarianza dello stato viene scelto arbitrariamente, per esempio pari alla matrice identità $\mathbf{P}(0) = \mathbf{I}_{n \times n}$.

2.4.3 Osservatori per sistemi non lineari

La linearizzazione del modello è il modo più semplice di estendere ai sistemi nonlineari l'applicabilità delle tecniche lineari. Dall'altro lato bisogna ricordare che queste tecniche danno buoni risultati solo quando le differenze tra il sistema linearizzato e sistema originale non-lineare sono piccole.

Possiamo distinguere due tipi di linearizzazione, linearizzazione attorno a uno stato costante (i.e. lo stato di equilibrio) e linearizzazione attorno alla stima corrente dello stato. Risulta ovvio che il secondo tipo di linearizzazione di solito porta a risultati migliori (in questo lavoro utilizzeremo quest'ultimo metodo).

Consideriamo il seguente sistema non-lineare a tempo discreto:

$$\begin{cases} \vec{x}(k+1) = \vec{f}(\vec{x}(k), \vec{u}(k)) \\ \vec{y}(k) = \vec{g}(\vec{x}(k)) \end{cases}$$
(2.9)

Per poter applicare le tecniche lineari descritte in precedenza è necessario linearizzare le eq. (2.9) attorno a un valore costante o attorno alla stima dello stato corrente $\hat{\vec{x}}(k)$. Quest'ultimo approccio è il più appropriato poiché l'accuratezza dell'approssimazione migliora tanto più $\hat{\vec{x}}(k)$ tende a $\vec{x}(k)$. Dunque si realizza un'approssimazione di questo tipo:

$$\mathbf{A}(k) = \left. \frac{\partial \vec{f}(\vec{x}(k), \vec{u}(k))}{\partial \vec{x}(k)} \right|_{\vec{x}(k) = \vec{x}(k)}, \mathbf{C}(k) = \left. \frac{\partial \vec{g}(\vec{x}(k))}{\partial \vec{x}(k)} \right|_{\vec{x}(k) = \vec{x}(k)}$$
(2.10)

Osservatore di Luenberger Esteso

L'osservatore di Luenberger esteso ai sistemi non-lineari ha la stessa forma del caso lineare eq. (2.3) dove le matrici $\mathbf{A}(\mathbf{k})$ e $\mathbf{C}(\mathbf{k})$ vengono calcolate linearizzando il sistema, vedi eq. (2.10).

Poiché le matrici A(k) e C(k) solitamente hanno la caratteristica di essere molto variabili nel tempo, spesso risulta difficile trovare una matrice di guadagno L(k+1) dalla forma appropriata. Per questo motivo questo approccio viene poco usato.

Filtro di Kalman Esteso

Il filtro di Kalman esteso (EKF) ai sistemi non-lineari mantiene la stessa forma del caso lineare descritto nella Sezione 2.4.2, con l'eccezione dello stato predetto al passo k + 1 prima di vedere le misurazioni, $\vec{x}(k+1|k)$, eq. (2.5), che vien sostituito da:

$$\vec{\hat{x}}(k+1|k) = \vec{f}(\vec{\hat{x}}(k|k), \vec{u}(k))$$
(2.11)

Questo approccio funziona bene quando il modello linearizzato non si discosta troppo dal comportamento non-lineare del sistema.

L'EKF può anche essere usato come osservatore in sistemi deterministici. In questo caso i valori delle matrici di covarianza possono essere scelti in maniera pressoché arbitraria in modo da aumentare la convergenza dell'osservatore.

2.5 Controllore LQR

Quando la dinamica di un sistema è descritta da un insieme di equazioni differenziali lineari, $\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t)$, e la funzione costo è una funzione quadratica allora si parla di problema LQ. Il regolatore lineare quadratico (LQR) è un compensatore ottenuto a seguito della minimizzazione di un indice di costo J(x, u) funzione dello stato x(t) e del controllo u(t):

$$J(x, u) = \frac{1}{2} \int_0^\infty \left[x^T(\tau) \mathbf{Q} x(\tau) + u^T(\tau) \mathbf{R} u(\tau) \right]$$

dove:

- **Q** è una matrice definita non-negativa $\mathbf{Q} \ge 0$, $\mathbf{Q} = \mathbf{Q}^T$,
- **R** è una matrice definita positiva $\mathbf{R} > 0, \mathbf{R} = \mathbf{R}^T$,
- la coppia (A,B) è controllabile.

Il controllo scalare LQR ottimo u^* è dato da $u^* = -K^*x$, dove $K^* = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}$ è la matrice di retroazione dello stato ottimale. Il problema di controllo ottimo è quello di trovare il

guadagno *K* tale che il controllo *u* minimizzi la funzione costo. La soluzione ottimale si trova risolvendo l'equazione algebrica di Riccati:

$$\mathbf{S}\mathbf{A} + \mathbf{A}^T \mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{S} + \mathbf{Q} = \mathbf{0}$$

2.6 Identificazione del modello del sistema

Il problema dell'identificazione si può ricondurre alla costruzione di modelli matematici di sistemi dinamici sulla base di dati estratti dal sistema stesso. In alcuni casi, l'ingresso non è presente, in questo caso si fa riferimento all'identificazione di serie temporali.

In generale, il procedimento di identificazione può essere scomposto in diverse fasi:

Raccolta dati. In questa fase si deve decidere quali dati utilizzare per impostare il procedimento di identificazione: che serie storica di ingressi fornire al sistema, con che ampiezza e frequenza di campionamento, ecc. Spesso non è possibile impostare un esperimento, oppure la serie storica desiderata non è compatibile con i limiti fisici del sistema stesso e si deve ricorrere a dati acquisiti durante il normale funzionamento dell'impianto (o durante l'osservazione del fenomeno).

Scelta del modello. E' difficile pensare ad un'identificazione in cui del fenomeno da identificare non si abbiano informazioni di alcun tipo, anche solo dalla osservazione delle strisciate di dati si può cominciare a trarre informazioni utili che condizionano la scelta del modello da utilizzare. Scelte più ponderate possono essere fatte nel caso in cui il fenomeno abbia una forte caratterizzazione fisica e sia possibile scriverne (semplici) equazioni dinamiche, si parla in questo caso di identificazione a scatola grigia (questo è il nostro caso). Si definisce identificazione a scatola nera il caso in cui non ci siano informazioni preliminari sul modello matematico. Un aspetto importante nella scelta del modello è l'uso che si deve fare del modello identificato, se finalizzato al controllo o al filtraggio, ecc.

Identificazione dei parametri del modello. Una volta scelto il modello e i dati da utilizzare si procede con il calcolo dei parametri del modello. Esistono algoritmi diversi per uno stesso modello ed i problemi da risolvere in questa fase sono essenzialmente di natura numerica.

Validazione del modello. Il modello appena identificato deve essere validato. Si devono quindi individuare una serie di indicatori che forniscano una sorta di misura della bontà dell'identificazione. In caso di mancata validazione il procedimento va rianalizzato in toto per capire il motivo del fallimento dell'identificazione stessa.

Nel nostro caso la conoscenza del sistema è completa, la scelta del modello è legata esclusivamente alle caratteristiche fisiche del sistema, perciò in questa sezione ci concentreremo unicamente sulla fase di identificazione dei parametri del modello.

2.6.1 Forma predittiva dei modelli per l'identificazione

Dato un modello parametrico $\mathbf{M}(\vec{x}) \operatorname{con} \vec{x}$ vettore di parametri incogniti, il procedimento di identificazione vuole individuare il vettore dei parametri che meglio rappresenta il fenomeno in esame. Forniti degli \vec{x}_i di tentativo deve essere possibile confrontare il modello individuato con i dati e validare o meno la bontà dei coefficienti stessi. Si rende quindi necessario introdurre una misura di errore legata alla misura dei dati a disposizione, una delle possibilità è quella di mettere il modello in forma predittiva. Cioè, utilizzando i dati a disposizione (tipicamente ingressi e/o uscite misurate), si vuole generare una predizione per il passo successivo e verificare quanto questa predizione sia numericamente vicina al valore misurato. Si definisce come predizione al passo kusando i dati fino al passo k - 1 la variabile:

$$\hat{y}(k|k-1)$$

l'errore di predizione (ad un passo) è allora dato da:

$$\varepsilon(k) = y(k) - \hat{y}(k|k-1).$$
 (2.12)

In maniera analoga può essere definito l'errore a n passi. Un modello 'ottimo' è quindi il modello che, su una serie di dati, minimizza una cifra di merito. Normalmente si considera come cifra di merito l'errore quadratico medio:

$$J = \frac{1}{N} \sum_{k=j}^{j+N-1} \varepsilon^2(k)$$
 (2.13)

2.6.2 Algoritmi utilizzati

In letteratura sono presenti numerosi algoritmi utilizzabili per minimizzare una cifra di merito, qui di seguito accenneremo solo ai due utilizzati in questo lavoro.

Algoritmo di Nelder-Mead

Il metodo di Nelder-Mead [2] tenta di minimizzare una funzione non-lineare di n variabili reali con valore scalare usando solo valori di funzione, senza l'utilizzo delle derivate. Questo metodo fa parte di una sotto-classe dei metodi di ricerca diretta in cui viene mantenuto ad ogni passo un simplesso non-degenerativo, una figura geometrica a n dimensioni di volume diverso da zero che è l'inviluppo convesso di n + 1 vertici.

L'algoritmo utilizzato si basa sull'iterazione. Partendo da un punto iniziale \vec{x}_0 appartenente ad \mathbb{R}^n , in corrispondenza del quale si valuta la funzione obiettivo $J(\vec{x}_0)$, si definisce una regione geometrica convessa composta da n + 1 vertici denominata simplesso. Ad ogni iterazione si determina un nuovo punto \vec{x}_i , nel quale viene valutata la funzione J. Questa è confrontata con i valori della funzione ai vertici e di solito uno di quest'ultimi è sostituito dal nuovo punto, restituendo un nuovo Simplesso. Questo passo è ripetuto fin quando il diametro di tale regione è minore di un valore specificato, ovvero di un parametro di soglia al di sotto del quale si stabilisce non sia necessario affinare l'indagine.

Algoritmo a minimi quadrati

Gli algoritmi a minimi quadrati sono numerosi, noi ci concentreremo sul *Large Scale Trust-Region Reflective non-lineare* che è basato sul metodo *interior-reflective Newton* descritto in [3].

Per capire l'approccio Trust Region all'ottimizzazione consideriamo il problema di minimizzazione senza vincoli, minimizzare $f(\vec{x})$, dove la funzione f prende in ingresso un vettore e restituisce uno scalare. Supponiamo di essere in un punto \vec{x} in uno spazio n-dimensionale e vogliamo spostarci in un punto in cui la funzione f sia più piccola. L'idea di base è di approssimare la funzione f con una funzione più semplice q, che riflette il comportamento di f in un intorno N di \vec{x} . Questo intorno viene chiamato Trust Region. Viene calcolato un passo di prova minimizzando in N, questo viene chiamato trust-region subproblem,

$$\min_{\vec{s}} \left\{ q(\vec{s}), \vec{s} \in N \right\}$$

Il punto corrente sarà $\vec{x} + \vec{s}$ se $f(\vec{x} + \vec{s}) < f(\vec{x})$; altrimenti, il punto corrente rimane \vec{x} e la trust-region *N* viene ristretta e viene ripetuto il passo di prova.

Nel metodo trust-region standard, l'approssimazione quadratica q è definita dai primi due termini della serie di Taylor di f in \vec{x} ; l'intorno N viene solitamente scelto di forma sferica o ellissoidale. Matematicamente il trust-region subproblem viene posto nel modo seguente:

$$\min\left\{\frac{1}{2}\vec{s}^{T}\mathbf{H}\vec{s} + \vec{s}^{T}g : \|\mathbf{D}\vec{s}\| \le \Delta\right\},$$
(2.14)

dove *g* è il gradiente di *f* nel punto corrente \vec{x} , **H** è l'Hessiano (la matrice simmetrica delle derivate seconde), **D** è una matrice diagonale scalare, Δ è uno scalare positivo, e $\|\cdot\|$ è la norma-2. Gli algoritmi per risolvere l'eq. (2.14) normalmente implicano il calcolo di un sistema autovalore-autovettore e un processo di Newton applicato all'equazione caratteristica:

$$\frac{1}{\Delta} - \frac{1}{\|\vec{s}\|} \tag{2.15}$$

Ma per problemi in larga scala il trust-region subproblem viene ristretto a un sottospazio bidimensionale *S*.

Nel caso non-lineare

$$\min_{\vec{x}} \sum_{i} f_{i}^{2}(\vec{x}) = \min_{\vec{x}} \|F(\vec{x})\|_{2}^{2}$$

per definire il sottospazio bidimensionale viene utilizzata la direzione dell'approssimazione di Gauss-Newton, cioè la soluzione \vec{s} di

$$\min\|\mathbf{J}\vec{s} + F(\vec{x})\|_2^2$$

dove **J** è lo Jacobiano di $F(\vec{x})$.

Si può dunque schematizzare la minimizzazione senza vincoli secondo il metodo trustregion in questo modo:

- 1. Formulare il trust-region subproblem bidimensionale.
- 2. Risolvere l'eq. (2.14) per determinare il passo di prova \vec{s} .
- 3. Se $f(\vec{x} + \vec{s}) < f(\vec{x})$, allora $\vec{x} = \vec{x} + \vec{s}$.
- 4. Adattare Δ .

Questi quattro passi vengono ripetuti sino alla convergenza. Le dimensioni della trustregion Δ vengono ridotte se il passo di prova non viene accettato, cioè se $f(\vec{x} + \vec{s}) = f(\vec{x})$.

Capitolo 3

Descrizione del processo fisico

In questo capitolo descriveremo il sistema reale su cui sono state fatte le simulazioni e ne daremo un modello matematico. Inoltre mostreremo alcuni particolari del modello del sistema in ambiente Matlab/Simulink[®].

3.1 Multitank System

Si è utilizzato il sistema ibrido multitank disponibile nel laboratorio di Automatica del Dipartimento di Ingegneria Elettrica e Elettronica dell'Università di Cagliari. Il Multitank System è prodotto dalla Inteco Ltd. [4] compagnia Polacca fondata nel 1997 dai ricercatori e ingegneri dell'Università Mineraria e Metallurgica di Cracovia. Il lavoro della Inteco Ltd. è principalmente focalizzato nella produzione di dispositivi elettromeccanici utilizzati nell'ambito della ricerca. Dal 1999 è partner ufficiale di Matlab[®].

Il Multitank System [1], è composto da tre serbatoi in cascata, Figura 3.1, ciascuno dotato sul fondo di due valvole di scarico , una manuale e l'altra controllata, di cui vediamo un particolare in Figura 3.2.

L'ulteriore serbatoio alla base della struttura serve da riserva d'acqua per il sistema.

Il serbatoio superiore (Upper tank) ha una forma rettangolare e quindi una sezione costante, il serbatoio di mezzo (Middle tank) ha una forma conica e il serbatoio inferiore (Lower tank) ha una forma sferica, dunque entrambi hanno una sezione non costante e sono perciò la causa di molte non-linearità del sistema.

Una pompa a velocità variabile, azionata da un motore in continua, viene utilizzata per riempire l'upper tank, il liquido fuoriesce a causa della gravità dai serbatoi attraverso le valvole, che agiscono come resistori di flusso. L'apertura delle valvole è controllata e può essere utilizzata per variare la caratteristica di deflusso.



Figura 3.1: Multitank System



Figura 3.2: Multitank System - particolare Valvole
Ogni serbatoio è dotato di un sensore di livello basato sulla pressione esercitata dal liquido (trasduttori di pressione). I segnali in frequenza dei sensori di livello sono collegati agli ingressi digitali della scheda I/O RT-DAC/PCI multipurpose.

Dalla scheda vengono inviati verso il sistema multitank quattro segnali di controllo: tre segnali per le valvole e uno per la pompa. I segnali di controllo PWM (Modulazione di Larghezza d'Impulso) appropriati vengono trasmessi dalle uscite digitali della scheda I/O all'interfaccia di potenza che a sua volta li trasmette alle valvole e al motore in continua. La velocità della pompa è controllata da una sequenza di PWM configurati e generati dal chip XILINX[®] della scheda RT-DAC.

3.2 Modello matematico



Figura 3.3: Multitank System - Schema

In questa sezione analizzeremo il nostro sistema non come sistema ibrido ma come se avesse un unico modo di funzionamento, e di questo modo andremo a trovare un modello in variabili di stato. Successivamente estenderemo le nostre considerazioni al sistema ibrido. Più precisamente il modo di funzionamento che andiamo a considerare è quello con tutte le valvole manuali chiuse.

Il sistema multitank, descritto in Figura 3.3, ha come variabili di stato i tre livelli dei serbatoi, H_1 , H_2 , H_3 , e ha quattro ingressi controllati: la portata in ingresso della pompa q e l'apertura delle tre valvole. In questo lavoro considereremo come unico ingresso di controllo la portata della pompa, mentre le tre valvole controllate avranno un'apertura costante, dunque lavoreremo su un sistema con controllo della pompa, Figura 3.4.



Figura 3.4: Sistema con controllo della pompa

Si possono individuare diversi elementi che possono ostacolare un controllo accurato dei livelli nei serbatoi:

- non-linearità causate dalla forma dei serbatoi,
- non-linearità dovute alla saturazione, introdotta dai livelli massimi e minimi consentiti nei serbatoi,
- non-linearità introdotta dalla geometria delle valvole e dalle dinamiche di flusso,
- non-linearità introdotte dalle caratteristiche I/O di pompa e valvole.

3.2.1 Flusso laminare di un fluido ideale

Il flusso laminare di un 'fluido ideale' in uscita da un serbatoio, Figura 3.5, è descritto dalla legge di Bernoulli. L'equazione si ricava da un semplice calcolo dell'energia potenziale e dell'energia cinetica del fluido [5]

$$Q_r = \mu S \sqrt{2gH_0} \tag{3.1}$$

dove:

- *S* è l'area del foro d'uscita,
- μ è il coefficiente di flusso del foro
- H_0 è il livello del fluido
- g è l'accelerazione di gravità
- Q_r è il flusso laminare in uscita.



Figura 3.5: Deflusso di un fluido ideale

3.2.2 Modello per n serbatoi in cascata

Se consideriamo *n* serbatoi in cascata, nel caso di un 'fluido ideale', la dinamica del processo può essere descritta come:

$$\frac{dV_1}{dt} = q - C_1 \sqrt{H_1}$$

$$\frac{dV_2}{dt} = C_1 \sqrt{H_1} - C_2 \sqrt{H_2}$$
.....
$$\frac{dV_n}{dt} = C_{n-1} \sqrt{H_{n-1}} - C_n \sqrt{H_n}$$
(3.2)

dove:

- V_1, V_2, \ldots, V_n volume di fluido negli *n* serbatoi,
- C_1, C_2, \ldots, C_n resistenza del foro,
- H_1, H_2, \ldots, H_n livello del fluido negli *n* serbatoi,
- q portata del fluido in ingresso.

Dalle eq. (3.2) possiamo scrivere:

$$\frac{dV_1}{dH_1}\frac{dH_1}{dt} = q - C_1 H_1^{\alpha_1}
\frac{dV_2}{dH_2}\frac{dH_2}{dt} = C_1 H_1^{\alpha_1} - C_2 H_2^{\alpha_2}
\dots
\frac{dV_n}{dH_n}\frac{dH_n}{dt} = C_{n-1} H_{n-1}^{\alpha_{n-1}} - C_n H_n^{\alpha_n}$$
(3.3)

Come già detto nel caso di 'fluido ideale' $\alpha_i = 1/2$, mentre nel sistema reale la turbolenza e l'accelerazione del liquido non possono essere trascurati e i valori dei coefficienti α_i devono essere identificati.

3.2.3 Modello non-lineare del sistema three-tank

Come già detto considereremo il caso di sistema pump-controlled. Scrivendo le eq. (3.3) per n = 3 si ha:

$$F_{1}(q, H_{1}) = \frac{dH_{1}}{dt} = \frac{1}{\beta_{1}(H_{1})}q - \frac{1}{\beta_{1}(H_{1})}C_{1}H_{1}^{\alpha_{1}}$$

$$F_{2}(H_{1}, H_{2}) = \frac{dH_{2}}{dt} = \frac{1}{\beta_{2}(H_{2})}C_{1}H_{1}^{\alpha_{1}} - \frac{1}{\beta_{2}(H_{2})}C_{2}H_{2}^{\alpha_{2}}$$

$$F_{3}(H_{2}, H_{3}) = \frac{dH_{3}}{dt} = \frac{1}{\beta_{3}(H_{3})}C_{2}H_{2}^{\alpha_{2}} - \frac{1}{\beta_{3}(H_{3})}C_{3}H_{3}^{\alpha_{3}}$$
(3.4)

dove, facendo riferimento alla Figura 3.6:

- $\beta_1(H_1) = aw$ sezione orizzontale dell'Upper tank,
- $\beta_2(H_2) = cw + \frac{H_2}{H_{2max}}bw$ sezione orizzontale del Middle tank,
- $\beta_3(H_3) = w\sqrt{R^2 (H_{3max} H_3)^2}$ sezione orizzontale del Lower tank,
- C_i resistenza del foro di uscita relativa all'*i*-esimo serbatoio,
- α_i coefficiente di flusso relativo all'*i*-esimo serbatoio.

I parametri *a*, *b*, *c*, *w*, H_{2max} , H_{3max} , *R* sono relativi alla geometria dei tre serbatoi, il loro significato e il loro valore in centimetri è illustrato in Figura 3.6.

Notiamo come la sezione dell'Upper tank è costante mentre quelle del Middle e del Lower tank variano a seconda dell'altezza in cui le calcoliamo.

Definiamo gli stati ammissibili del processo come:

$$\aleph_i = \{H_i : 0 \le H_i \le H_{max}\}, i = 1, 2, 3$$

e, l'insieme di ingressi di controllo ammissibili:

$$Q = \left\{ q : 0 \le q \le q_{max} \right\}.$$



Figura 3.6: Geometria dei tre serbatoi

Considerando il modello in eq. (3.4), per un valore fissato di $q = q_0$, possiamo definire uno stato di equilibrio, come:

$$H_0 = \{H_i : F_1(q_0, H_1) = 0, F_i(H_{i-1}, H_i) = 0, i = 2, 3\} q_0 \in Q, H_i \in \aleph_i$$

Lo stato di equilibrio può essere calcolato dalle equazioni

$$q_0 = C_1 H_{10}^{\alpha_{10}} = C_2 H_{20}^{\alpha_{20}} = C_3 H_{30}^{\alpha_{30}}$$
(3.5)

Nelle nostre simulazioni utilizzeremo una portata massima pari a $q_{max} = 11 \cdot 10^{-5} [m^3/s]$ e una portata di riferimento pari a $q_0 = 2.1 \cdot 10^{-5} [m^3/s]$.

Per quel che riguarda la trasformazione d'uscita, le uscite sono date dalle misurazioni dei sensori di livello, perciò coincidono con lo stato continuo:

$$\vec{y} = \mathbf{C}\vec{x} \tag{3.6}$$

con

$$\vec{x} = \left[\begin{array}{c} H_1 \\ H_2 \\ H_3 \end{array} \right] e \mathbf{C} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right].$$

3.2.4 Modello linearizzato del sistema three-tank

Il modello linearizzato si ottiene attraverso l'espansione di Taylor della eq. (3.4) attorno allo stato corrente stimato, $\hat{x} = \hat{H}$,

$$\frac{dh}{dt} = \mathbf{J}_H h + \mathbf{J}_q u \tag{3.7}$$

dove $h = H - \hat{H}$, $u = q - q_0$ e, $J_H e J_q$ sono gli Jacobiani utilizzati, come già descritto nella eq. (2.10), per linearizzare le eq. (3.4):

$$\mathbf{J}_{H} = \left[\frac{\partial F(H,q)}{\partial H}\right]_{H=\hat{H},q}, \mathbf{J}_{q} = \left[\frac{\partial F(H,q)}{\partial q}\right]_{H=\hat{H},q}$$
(3.8)

cioè

$$\mathbf{J}_{H} = \begin{bmatrix} \frac{-C_{1}}{\hat{H}_{1}^{1-\alpha_{1}}} \frac{\alpha_{1}}{\beta_{1}(\hat{H}_{1})} & \mathbf{0} & \mathbf{0} \\ \frac{C_{1}}{\hat{H}_{1}^{1-\alpha_{1}}} \frac{\alpha_{1}}{\beta_{2}(\hat{H}_{2})} & \frac{-C_{2}}{\hat{H}_{2}^{1-\alpha_{2}}} \frac{\alpha_{2}}{\beta_{2}(\hat{H}_{2})} & \mathbf{0} \\ \mathbf{0} & \frac{C_{2}}{\hat{H}_{2}^{1-\alpha_{2}}} \frac{\alpha_{2}}{\beta_{3}(\hat{H}_{3})} & \frac{-C_{3}}{\hat{H}_{3}^{1-\alpha_{3}}} \frac{\alpha_{3}}{\beta_{3}(\hat{H}_{3})} \end{bmatrix}$$

e

$$\mathbf{J}_q = \begin{bmatrix} \frac{1}{\beta_1(\hat{H}_1)} & \\ 0 & \\ 0 & \end{bmatrix}$$

3.2.5 Modello Switching System

Fino ad ora si è descritto il sistema con un modello in variabili di stato, questo tipo di modello è adatto a descrivere solo un modo del sistema a commutazione che ora andremo a illustrare. Infatti le eq. (3.4) sono riferite a una configurazione del sistema in cui una sola valvola per serbatoio è aperta. In questo lavoro abbiamo deciso di individuare quattro configurazioni del sistema che corrispondono ai quattro modi dello switching system.

Dunque un modo del sistema, quello delle eq. (3.4), che chiameremo M_0 è quello in cui in ogni serbatoio la valvola manuale è chiusa mentre la valvola controllabile è aperta con un'apertura costante durante tutto il processo.

Il modo M_1 corrisponde alla configurazione in cui nell'Upper tank entrambe le valvole, sia quella controllabile che quella manuale, sono aperte, mentre negli altri due serbatoi solo le valvole controllabili sono aperte.

Il modo M_2 corrisponde alla configurazione in cui nel Middle tank entrambe le valvole, sia quella controllabile che quella manuale, sono aperte, mentre negli altri due serbatoi solo le valvole controllabili sono aperte.

Il modo M_3 corrisponde alla configurazione in cui nel Lower tank entrambe le valvole, sia quella controllabile che quella manuale, sono aperte, mentre negli altri due serbatoi solo le valvole controllabili sono aperte.

Dunque il modello in variabili di stato per i modi M_1, M_2 e M_3 hanno la stessa forma delle eq. (3.4), con la differenza che,

per il modo M_1 , la F_1 e la F_2 hanno la forma:

$$F_{1}(q, H_{1}) = \frac{1}{\beta_{1}(H_{1})}q - \frac{1}{\beta_{1}(H_{1})}C_{1}H_{1}^{\alpha_{1}} - \frac{1}{\beta_{1}(H_{1})}C_{m1}H_{1}^{\alpha_{m1}}$$

$$F_{2}(H_{1}, H_{2}) = \frac{1}{\beta_{2}(H_{2})}C_{1}H_{1}^{\alpha_{1}} + \frac{1}{\beta_{2}(H_{2})}C_{m1}H_{1}^{\alpha_{m1}} - \frac{1}{\beta_{2}(H_{2})}C_{2}H_{2}^{\alpha_{2}}$$
(3.9)

per il modo M_2 , la F_2 e la F_3 hanno la forma:

$$F_{2}(H_{1}, H_{2}) = \frac{1}{\beta_{2}(H_{2})} C_{1} H_{1}^{\alpha_{1}} - \frac{1}{\beta_{2}(H_{2})} C_{2} H_{2}^{\alpha_{2}} - \frac{1}{\beta_{2}(H_{2})} C_{m2} H_{2}^{\alpha_{m2}}$$

$$F_{3}(H_{2}, H_{3}) = \frac{1}{\beta_{3}(H_{3})} C_{2} H_{2}^{\alpha_{2}} + \frac{1}{\beta_{3}(H_{3})} C_{m2} H_{2}^{\alpha_{m2}} - \frac{1}{\beta_{3}(H_{3})} C_{3} H_{3}^{\alpha_{3}}$$
(3.10)

per il modo M_3 , la F_3 ha la forma:

$$F_3(H_2, H_3) = \frac{1}{\beta_3(H_3)} C_2 H_2^{\alpha_2} - \frac{1}{\beta_3(H_3)} C_3 H_3^{\alpha_3} - \frac{1}{\beta_3(H_3)} C_{m3} H_3^{\alpha_{m3}}$$
(3.11)

Dove C_{m1} , C_{m2} , C_{m3} e α_{m1} , α_{m2} , α_{m3} sono, rispettivamente, resistenza del foro e coefficiente di flusso associati alle valvole manuali dei tre serbatoi, mentre indicheremo con C_1 , C_2 , C_3 e α_1 , α_2 , α_3 quelli relativi alle valvole controllabili.

Con un linguaggio più formale, introdotto nella Sezione 2.3, possiamo descrivere il sistema a commutazione three-tank con la tupla $S = (D, X, X_0, U, Y, \varepsilon)$, dove:

- Il sistema ad eventi discreti è descritto dalla tupla $D = (Q, Q_0, E, \Psi, \eta)$, dove:
 - $Q = Q_0 = \{M_0, M_1, M_2, M_3\}$ è l'insieme finito di modi;
 - E è l'insieme degli eventi che corrispondono agli archi in Figura 3.7, vedi Tabella 3.1;

Evento	s()	t()
	modo di partenza	modo di destinazione
a	$s(a) = M_1$	$t(a) = M_0$
b	$s(b) = M_0$	$t(b) = M_1$
С	$s(c) = M_1$	$t(c) = M_2$
d	$s(d) = M_2$	$t(d) = M_1$
е	$s(e) = M_2$	$t(e) = M_3$
f	$s(f) = M_3$	$t(f) = M_2$
g	$s(g) = M_3$	$t(g) = M_0$
h	$s(h) = M_0$	$t(h) = M_3$
i	$s(i) = M_0$	$t(i) = M_2$
1	$s(l) = M_2$	$t(l) = M_0$
m	$s(m) = M_3$	$t(m) = M_1$
n	$s(n) = M_1$	$t(n) = M_3$

Tabella 3.1: Eventi



Figura 3.7: Sistema three-tank - diagramma a stati

- $\Psi = \{0, 1, 2, 3\}$ è insieme dei simboli d'uscita;
- η è la funzione d'uscita, ad ogni evento è associato un simbolo d'uscita che, nel nostro caso, corrisponde per tutti gli eventi alla stringa vuota ε. Questo perché gli eventi *E* non sono controllabili e avvengono in istanti di tempo sconosciuti, in questi casi si parla di *sistemi a commutazione autonomi*.
- $X \subset \mathbb{R}^3$ è lo spazio dello stato continuo;
- $X_0 \subseteq X$ è l'insieme delle condizioni iniziali nel continuo;
- ${\varepsilon_q}_{q \in Q}$, che associa ad ogni modo le dinamiche continue, è già stato specificato nelle eq. (3.4), eq. (3.9), eq. (3.10) e eq. (3.11).

Descriviamo brevemente a cosa corrisponde nel sistema reale ogni singolo evento:

- *a*: viene chiusa la valvola manuale dell'Upper tank,
- *b*: viene aperta la valvola manuale dell'Upper tank,
- c: viene chiusa la valvola manuale dell'Upper tank e aperta quella del Middle tank,
- d: viene aperta la valvola manuale dell'Upper tank e chiusa quella del Middle tank,
- e: viene chiusa la valvola manuale del Middle tank e aperta quella del Lower tank,
- *f*: viene aperta la valvola manuale del Middle tank e chiusa quella del Lower tank,
- g: viene chiusa la valvola manuale del Lower tank,
- h: viene aperta la valvola manuale del Lower tank,
- *i*: viene aperta la valvola manuale del Middle tank,
- *l*: viene chiusa la valvola manuale del Middle tank,
- *m*: viene aperta la valvola manuale dell'Upper tank e chiusa quella del Lower tank,
- *n*: viene chiusa la valvola manuale dell'Upper tank e aperta quella del Lower tank.

3.3 Modello Matlab-Simulink

Abbiamo modificato il modello presentato dalla INTECO in [1], Figura 3.8, per avere un modello Simulink adatto a simulare il nostro processo.

Come si vede in Figura 3.9 abbiamo inserito la possibilità di manovrare le valvole manuali ON/OFF per simulare i cambiamenti di Modo e abbiamo aggiunto le valvole controllate dotando il nostro modello di lookup-tables che descrivono le curve caratteristiche dei parametri C_i e α_i al variare del segnale di controllo, Figura 3.10, in cui abbiamo inserito i parametri identificati, vedi Sezione 4.4.



Tank 3 Open-Loop

Figura 3.8: Three-tank - modello INTECO



Tank 3 Simulation Model

Figura 3.9: Three-tank - modello Simulink open loop



Figura 3.10: Three-tank - modello Simulink under mask

Tank 3 Simulation Model

Le modifiche fatte dunque sono dirette a trasformare il modello INTECO in un modello adatto a descrivere uno SS.

Inoltre abbiamo modificato il limite di saturazione inferiore degli integratori all'interno del modello, che determina il livello minimo di liquido che un serbatoio può avere. Nel modello fornito dalla INTECO questi è fissato a $10^{-3}m$, questo valore non è adatto a descrivere il nostro processo poiché in alcuni istanti della nostra simulazione si verifica la situazione in cui l'Ipper tank è vuoto e la pompa è spenta. In questi casi un valore del livello minimo di $10^{-3}m$, fa si che il Middle tank continua ad avere un flusso di liquido in ingresso proveniente dall'Upper tank anche quando quest'ultimo è vuoto.



Figura 3.11: Comportamento del modello three-tank quando il livello minimo è fissato a 1 mm

In Figura 3.11 mostriamo l'andamento del livello del liquido nel Middle tank che si ottiene con la soglia a $10^{-3}m$, quando il sistema funziona nel modo M_1 (tutte le valvole controllate aperte e la valvola manuale dell'Upper tank aperta) e la pompa è spenta. Si può notare come il livello del liquido del Middle tank cresce nonostante l'Upper tank sia vuoto e la pompa spenta, mentre ci aspetteremmo che il Middle tank rimanesse vuoto.

Abbiamo corretto questo comportamento abbassando il limite inferiore di saturazione a $10^{-5}m$, con questa configurazione il comportamento del sistema è quello aspettato, cioè il livello del Middle tank rimane intorno allo 0.

Capitolo 4

Identificazione del processo

In questo capitolo descriveremo i passi seguiti per l'identificazione delle varie componenti del sistema e per l'identificazione dei parametri del suo modello.

L'identificazione del sistema three-tank consiste nei seguenti passi:

- curve caratteristiche dei sensori di livello,
- curva caratteristica della pompa in continua,
- curva caratteristica valvole,
- identificazione dei parametri C_i e α_i del modello matematico.

Inoltre al termine del capitolo faremo alcune considerazioni sul processo e sulle strategie di controllo adottate.

4.1 Curve caratteristiche dei sensori di livello

I tre serbatoi sono dotati di trasduttori di pressione piezo-resistivi per misurare il livello del liquido. I sensori di pressione forniscono un output in frequenza proporzionale alla pressione esercitata dal liquido. La curva caratteristica dei sensori di livello che descrive il livello del liquido in funzione della frequenza, può essere considerata lineare:

$$Level = Gain * (Freq - freq_{bias})$$
(4.1)

dove:

- Level è il livello del liquido nel serbatoio,
- Freq è la frequenza che andremo a leggere nella scheda I/O,

- Gain è un valore reale costante che indica la pendenza della curva caratteristica,
- *freq*_{bias} valore reale costante che indica un disturbo sistematico.

I parametri della curva caratteristica si trovano misurando il livello del liquido e l'uscita in frequenza del trasduttore corrispondente per almeno due punti e cercando la curva caratteristica interpolante.

In Tabella 4.1 sono indicati i livelli misurati e le corrispondenti uscite in frequenza per i tre serbatoi.

Level	Sensor 1	Sensor 2	Sensor 3
[<i>m</i>]	freq1[Hz]	freq2[Hz]	freq3[Hz]
0.05	2631	2612	2588
0.15	3351	3326	3276
0.20	3719	3686	3631
0.25	4070	4014	3972

Tabella 4.1: livelli e frequenze dei sensori associate



Figura 4.1: Caratteristica Sensore 1 - pallini per le misurazioni, linea continua per la retta interpolante

Trasferendo in Matlab i dati della Tabella 4.1, col comando *polyfit(freq1,Level,1)* si ottengono, per il Sensore 1, i coefficienti *a* e *b* dell'equazione interpolante Level = a * Frequency + b; confrontando quest'ultima con l'eq (4.1) risulta:

 $Gain_1 = 1.38695569107 \cdot 10^{-4} \text{ e } Bias_1 = 2.271119204247147 \cdot 10^3$

e allo stesso modo per i sensori 2 e 3

$$Gain_2 = 1.42015749811 \cdot 10^{-4} \text{ e } Bias_2 = 2.265010715149862 \cdot 10^3$$
$$Gain_3 = 1.43613168237 \cdot 10^{-4} \text{ e } Bias_3 = 2.237238161679126 \cdot 10^3$$

4.2 Identificazione delle valvole

Il fondo di ogni serbatoio è dotato di una valvola controllata. L'apertura delle valvole viene impostata usando il segnale PWM dell'interfaccia di potenza. In Figura 4.2 viene mostrata la curva caratteristica di una valvola, ottenuta, mantenendo la pressione del liquido costante, misurando il flusso in uscita al variare del segnale di controllo PWM.



Figura 4.2: Cratteristica di una valvola controllata

Notiamo come la caratteristica possa essere approssimata con un andamento lineare per i valori di controllo che vanno da 0.65 a 1. Per i valori di controllo che vanno da 0 a 0.65 (non mostrati in figura) la portata della pompa assume un valore nullo.

4.3 Identificazione della pompa

La pompa in continua provvede al trasporto del liquido dalla riserva d'acqua che sta alla base del sistema verso l'Upper tank. La sua funzione è quella di regolare il flusso in ingresso in accordo con il segnale di controllo.

Abbiamo costruito la curva caratteristica sperimentale della pompa in continua, Figura 4.3, misurando la portata in ingresso fornita dalla pompa al variare del segnale di controllo. Più precisamente, per ogni segnale di controllo abbiamo misurato i tempi di riempimento dell'Upper tank, e poiché le dimensioni del serbatoio sono note, con dei semplici passaggi matematici si risale alla portata.



Figura 4.3: Curva caratteristica sperimentale della pompa in continua

I dati relativi alle prove di riempimento possono essere consultati all'indirizzo WEB: http://www.diee.unica.it/giua/TESI/09_Stefano.Salis/Matlab_3Tank.rar.

Notiamo come anche la curva caratteristica della pompa sia perlopiù lineare ma con una saturazione per i valori di controllo che vanno da 0 a 0.15 in cui la portata della pompa risulta nulla.

Utilizzeremo la curva inversa rispetto a quella in Figura 4.3 nella 'look-up table' del controllore LQR che andremo a progettare per il nostro sistema.

4.4 Identificazione dei parametri del modello

Nella Sezione 3.2.2 abbiamo introdotto i parametri del modello matematico del threetank system:

- C_i resistenza del foro d'uscita dell'*i*-esimo serbatoio,
- α_i coefficiente di flusso dell'*i*-esimo serbatoio.

Abbiamo identificato sperimentalmente i parametri $C_i \in \alpha_i$ per valori fissati di apertura delle valvole. Per ogni serbatoio e per ogni valore di controllo delle valvole considerato è stata effettuata una prova di svuotamento, che consiste nella raccolta dei dati corrispondenti al livello del liquido nel serbatoio al variare del tempo, partendo dall'istante in cui si apre la valvola a serbatoio pieno fino all'istante in cui il serbatoio è vuoto. I dati collezionati vengono confrontati con i dati del modello e si cercano i parametri ottimali minimizzando la cifra di merito, come già illustrato nella Sezione 2.6.1.

I dati relativi alle prove di svuotamento possono essere consultati all'indirizzo WEB: http://www.diee.unica.it/giua/TESI/09_Stefano.Salis/Matlab_3Tank.rar

Per identificare i parametri C_i e α_i abbiamo utilizzato due procedure differenti, in cui ci siamo serviti dell'*Optimization Toolbox* e del *System Identification Toolbox* di Matlab [7].

Procedura 1

Per calcolare la predizione del livello del liquido all'istante *i* utilizziamo il dato raccolto (misurato) corrispondente al livello all'istante precedente, *i* – 1. Consideriamo, per esempio, il caso dell'Upper tank, se L_i è il dato che vogliamo predire, utilizziamo il dato L_{i-1} per definire la predizione \hat{L}_i come:

$$\hat{L}_{i} = L_{i-1} - \frac{1}{aw} C_{1} L_{i-1}^{\alpha_{1}} \Delta T$$

dove C_1 e α_1 sono i parametri da identificare e ΔT è il passo con cui i dati sono stati raccolti; nel nostro caso si è utilizzato un $\Delta T = 0.1$ secondi. La cifra di merito è stata definita come:

$$J(C_1, \alpha_1) = \sum_{i=1}^{N-1} (\hat{L}_i - L_i)^2, i = 1, 2 \dots N - 1$$
(4.2)

Per trovare i parametri che minimizzano la cifra di merito si è utilizzato il comando Matlab *fminsearch()* che è una procedura di minimizzazione che si basa sull'algoritmo di Nelder-Mead descritto nella Sezione 2.6.2. La stessa procedura è stata adoperata per i due restanti serbatoi. Nel Middle tank la predizione viene calcolata come:

$$\hat{L}_{i} = L_{i-1} - \frac{1}{w(c + bL_{i-1}/H_{2max})} C_2 L_{i-1}^{\alpha_2} \Delta T$$

e nel Lower tank:

$$\hat{L}_{i} = L_{i-1} - \frac{1}{w\sqrt{R^{2} - (H_{3max} - L_{i-1})^{2}}}C_{3}L_{i-1}^{\alpha_{3}}\Delta T$$

Di seguito mostriamo, per ogni serbatoio, il confronto tra i valori raccolti e i valori predetti dal modello identificato con la **Procedura 1**. Le curve mostrate fanno riferimento alla configurazione in cui in ogni serbatoio solo la valvola manuale è aperta. Nella legenda delle figure è indicato il valore di **fit**, che è un parametro che indica la percentuale di output misurato (dati raccolti) che combacia col modello.



Figura 4.4: Upper tank Confronto dati raccolti - modello Procedura 1



Figura 4.5: Middle tank Confronto dati raccolti - modello Procedura 1



Figura 4.6: Lower tank Confronto dati raccolti - modello Procedura 1

Procedura 2

La **Procedura 2** si serve del *System Identification Toolbox*, e più precisamente delle procedure di questo tool dedicate all'identificazione di modelli grey-box non-lineari.

Se chiamiamo Y i dati raccolti, il primo passo da eseguire è quello di creare un Data Object col comando *DAT* = *IDDATA(Y*,[],*Ts*) in cui *Ts* è il passo di campionamento utilizzato per raccogliere i dati. L'oggetto creato serve al *System Identification Toolbox* per elaborare le serie temporali di dati raccolti.

Il secondo passo consiste nel descrivere la struttura del modello del serbatoio in linguaggio C. Con struttura si intende il modello in variabili di stato che descrive la dinamica di un singolo serbatoio nel continuo, per il Middle tank:

```
{
    double C2, Alfa2;
    C2 = p[0][0];
    Alfa2 = p[1][0];
    /* State equations. */
    dx[0] = - (C2*(pow( x[0],Alfa2) )/(w*( c+x[0]*b/H2max )));
    y[0] = x[0];
}
```

Il codice completo per il Middle tank e per gli altri due serbatoi lo si trova nell'Appendice A.1.2. Il modello in C viene poi 'mexato', cioè dal file in C si passa ad un MEX-file (Matlab EXecutable) utilizzando il comando MATLAB *mex 'nomefile*'. I MEX-file sono delle subroutine prodotte da codice sorgente C o Fortran che possono essere eseguite in ambiente Matlab allo stesso modo degli M-file e delle funzioni built-in.

Il terzo passo consiste nel creare un oggetto Idnlgrey col comando *NLGR* = *IDNL*-*GREY(filename, order, parameters, initialstates, Ts)*, dove:

- filename specifica il nome del MEX-file in cui si descrive la struttura del modello,
- *order* è l'ordine del sistema, [101] nel nostro caso, cioè un'uscita, zero ingressi e uno stato,
- parametrs specifica il valore dei parametri con cui iniziare la procedura,
- *initialstates* indica lo stato iniziale che deve essere uguale al primo dato raccolto Y(0).

L'ultimo passo consiste nel calcolo dei parametri ottimali per minimizzare l'errore di predizione. Questo viene fatto col comando *NLGR = pem(DAT, NLGR)*, che prende in ingresso il Data Object con i dati raccolti e l'oggetto Idnlgrey che specifica il modello del sistema con dei parametri di prova non ottimali e ci restituisce in uscita l'oggetto

Idnlgrey configurato con i parametri ottimali; il metodo di ricerca dei parametri ottimali si basa sull'algoritmo ai minimi quadrati Trust-Region Reflective Newton (denominato LSQNONLIN in Matlab) descritto nella Sezione 2.6.2.

Di seguito mostriamo le curve corrispondenti alla Figura 4.4, Figura 4.5 e Figura 4.6 ottenute con la **Procedura 2**



Figura 4.7: Upper tank Confronto dati raccolti - modello Procedura 2



Figura 4.8: Middle tank Confronto dati raccolti - modello Procedura 2



Figura 4.9: Lower tank Confronto dati raccolti - modello Procedura 2

Facendo un confronto fra le due procedure si può notare che i risultati ottenuti sono molto simili, i valori di **fit** trovati si aggirano tutti intorno al 97 ÷ 99%. Abbiamo identificato i parametri con entrambe le procedure non solo nel caso di valvole manuali aperte ma anche nelle configurazioni in cui, le valvole manuali sono chiuse e le valvole controllate sono aperte con un controllo 1, 0.9 e 0.8 nelle diverse prove. Si è osservato che, nelle prove sul Middle tank, con la **Procedura 1** si sono ottenuti valori di **fit** più alti rispetto a quelli ottenuti con la **Procedura 2**, mentre si rileva un comportamento opposto per l'Upper ed il Lower tank. Dunque i parametri che abbiamo adottato nel nostro modello sono stati scelti basandosi sul valore di **fit** più elevato, indipendentemente dalla procedura utilizzata per ottenerli. In Tabella 4.2 sono indicati i valori dei parametri *C* e α scelti nel caso di valvola manuale aperta ($C_{mi} e \alpha_{mi}$) e valvola di controllo completamente aperta ($C_i e \alpha_i$), che sono le uniche due configurazioni utilizzate nel nostro lavoro.

	Upper Tank	Middle Tank	Lower Tank
C _{mi}	$1.505 \cdot 10^{-4}$	$1.804 \cdot 10^{-4}$	$1.704 \cdot 10^{-4}$
α_{mi}	0.263	0.312	0.314
C_i	$3.884 \cdot 10^{-5}$	$4.315 \cdot 10^{-5}$	$3.890 \cdot 10^{-5}$
α_i	0.291	0.310	0.301

Tabella 4.2: parametri *C* e α identificati

4.5 Confronto tra modello identificato e sistema reale

Di seguito mostriamo un confronto tra modello identificato e sistema reale per valutare la bontà dei parametri identificati. I dati utilizzati per il confronto sono i livelli nei tre serbatoi raccolti durante un processo in cui il sistema lavora in tutti e quattro i modi possibili e in cui si verificano tutti gli eventi illustrati in Tabella 3.1.

Si può osservare che i dati del modello in Simulink sono molto vicini a quelli del sistema reale. Nel particolare, per l'Upper tank, Figura 4.10, si nota che per grandi tratti le curve sono sovrapposte e solo in alcuni tratti si discostano di pochi millimetri con un massimo di 4*mm*. Si osserva un andamento simile nel Middle e nel Lower tank, Figura 4.11 e Figura 4.12, con una differenza massima tra modello e sistema reale di 6*mm* e 7*mm* rispettivamente; questi risultati leggermente inferiori rispetto all'Upper tank sono probabilmente legati alla maggiore non-linearità nella geometria del Middle e Lower tank. Inoltre si può notare che, quando il livello del liquido è prossimo allo zero, nei dati raccolti dal sistema reale si leggono valori di poco sotto lo zero, massimo $1 \div 2mm$, questo andamento si osserva maggiormente nel Middle e Lower tank ed è dovuto a piccole imprecisioni nell'identificazione dei sensori.



Figura 4.10: Upper tank - Confronto tra sistema reale e modello in Simulink



Figura 4.11: Middle tank - Confronto tra sistema reale e modello in Simulink



Figura 4.12: Lower tank - Confronto tra sistema reale e modello in Simulink

Si vuole sottolineare che il sistema reale si è dimostrato sensibile alle variazioni ambientali, in particolare i sensori di livello hanno manifestato un forte legame con la temperatura ambiente e del liquido. Per questo motivo abbiamo svolto le nostre prove in un arco di tempo relativamente breve e immediatamente successivo all'identificazione del sistema, e in alcuni casi abbiamo dovuto procedere a una nuova identificazione.

Come ultima osservazione, si vuole documentare un comportamento imprevisto del sistema che abbiamo incontrato durante una prova. La prova consisteva nel trovare sperimentalmente il livello di equilibrio nell'Upper tank, fissati la portata in ingresso e il segnale di controllo della valvola. Dunque siamo in una situazione in cui i segnali in ingresso son costanti. Come si può notare dalla Figura 4.13 il livello del liquido sembra aver raggiunto una condizione di equilibrio, fino a che non si arriva ad un istante (\approx campione 7400) in cui il livello del liquido aumenta decisamente. Abbiamo supposto che questo comportamento sia dovuto a una sorta di interferenza che si verifica tra il pannello di simulazione real-time di Simulink e il pannello di manual setup presente nella GUI (graphical user interface) di Matlab fornita dalla INTECO, Figura 4.14. Questa interferenza influisce sul segnale di controllo della pompa, che non rimane costante come impostato nella simulazione.



Figura 4.13: Comportamento imprevisto della pompa

🖁 Tank3 Manual Setup				
RT-DAC4/PCI board No of detected boards: Board: Board 1 Bus number: Slot number: Base addiess: 54272 Logic version: Application: I/O driver status:	1 2 10 7 0xD 400 105 Tank OK	Control Valve 1: Valve 2: Valve 3: Pump: J	5 T C	1.0000
Levels [cm] Current Tank 1: -0.02 Tank 2: -0.12 Tank 3: 17.04	Safety levets [cn Max level 28.91 29.26 28.91	n] and flags - Min level 27.67 28.00 27.66	Activity flag	Overflow alert Safe
Display 1/0	Ŀ	jelp		CLOSE

Figura 4.14: Pannello Manual Setup

4.6 Considerazioni sul processo

Prima di entrare nel dettaglio sull'osservazione dello stato continuo e discreto, facciamo alcune considerazioni sulle caratteristiche del processo che andiamo a studiare e sulle strategie di controllo adottate.

Dato il sistema a commutazione three-tank descritto, l'obiettivo di questo lavoro è quello di progettare un osservatore dello stato discreto (MO: mode observer) che sia in grado di capire in che modo il sistema sta lavorando. Il MO genera un segnale di uscita che indica il modo osservato, questo segnale va in ingresso ad un blocco di controllo che applica al sistema un controllore diverso per ogni stato discreto (gain scheduling). Il blocco di controllo ha come obiettivo quello di portare il sistema all'equilibrio, mediante la regolazione della velocità della pompa, nel minor tempo possibile e una volta raggiunto deve cercare di mantenerlo. Ogni volta che si verifica un cambiamento di modo il sistema viene destabilizzato e il blocco di controllo deve portarlo in un nuovo stato di equilibrio; ci sono tanti stati di equilibrio quanti sono gli stati discreti. Nelle eq. (3.5) abbiamo mostrato come trovare i livelli di equilibrio fissata l'apertura delle valvole e fissato un valore di riferimento della portata della pompa. Si ha che, quando in uno dei tre serbatoi sia la valvola controllata che quella manuale sono aperte, per il valore di portata di riferimento che abbiamo scelto $(q_0 = 2.1 \cdot 10^{-5} [m^3/s])$, il livello di equilibrio corrisponde al serbatoio vuoto. Perciò se per il modo M_0 i livelli di equilibrio sono $H_1 = H_{10}$, $H_2 = H_{20}$ e $H_3 = H_{30}$, rispettivamente, per l'Upper, il Middle e il Lower tank, allora si ha che per il modo:

- M_1 i livelli di equilibrio sono $H_1 = 0$, $H_2 = H_{20}$ e $H_3 = H_{30}$;
- M_2 i livelli di equilibrio sono $H_1 = H_{10}$, $H_2 = 0$ e $H_3 = H_{30}$;
- M_3 i livelli di equilibrio sono $H_1 = H_{10}$, $H_2 = H_{20}$ e $H_3 = 0$.

Si vuole sottolineare che la fase di controllo del processo è stata introdotta per completezza, ma non è stata sviluppata approfonditamente. Questo poiché il processo studiato, quando il sistema funziona in uno dei tre modi a valvola manuale aperta, non permette di mantenere il livello del liquido del serbatoio in cui la valvola manuale è aperta a un valore al di sopra dello zero. Infatti se si aumentasse la portata della pompa per avere un livello maggiore di zero, si provocherebbe un superamento del livello massimo consentito negli altri serbatoi, che corrisponde nel sistema reale alla fuoriuscita del liquido dai serbatoi in cui la valvola manuale è chiusa.

Per il blocco di controllo abbiamo deciso di utilizzare dei controllori LQR, vedi Sezione 2.5. Abbiamo risolto il problema LQ per il sistema linearizzato attorno allo stato di equilibrio, illustrato nella Sezione 3.2.4, utilizzando i quattro modelli in variabili di stato dei quattro modi considerati. Per la risoluzione del problema LQ ci siamo serviti della funzione *lqry* del Matlab Control System Toolbox.

Per i valori di controllo fissati i livelli di equilibrio risultano: $H_{10} = 0.12$, $H_{20} = 0.098$ e $H_{30} = 0.128$ metri.



Figura 4.15: Livelli sistema controllato

In Figura 4.15 mostriamo l'azione di controllo quando il sistema lavora nel modo M_0 .

Notiamo come il segnale di controllo, quando i livelli raggiungono lo stato di equilibrio, si assesta al valore di ~ 0.27 che corrisponde alla portata di riferimento scelta, $q_0 = 2.1 \cdot 10^{-5} [m^3/s]$, come si può vedere nella 'look-up table' della pompa in Figura 4.3.

Inoltre osserviamo come il sistema raggiunga l'equilibrio piuttosto lentamente, in circa 700 secondi, per questo, nelle simulazioni che effettueremo, le transizioni da un modo di funzionamento all'altro avverranno con intervalli di tempo piuttosto lunghi, mediamente intorno ai 200 secondi, un tempo necessario affinché il sistema sia almeno vicino all'equilibrio.

Per concludere andiamo a descrivere più nel dettaglio la forma che assume il sistema e la strategia di controllo facendo un'analisi modo per modo.

Modo M_0

Livelli di equilibrio:

$$H_{10} = 0.12, H_{20} = 0.098 \text{ e } H_{30} = 0.128 \text{ [}m\text{]}.$$

Guadagno ottimo del controllore LQR:

$$K_0^* = [0.2360 \cdot 10^{-3}, 0.1014 \cdot 10^{-3}, 0.0570 \cdot 10^{-3}]^T.$$

Ingesso di controllo di riferimento:

$$q_0 = 2.1 \cdot 10^{-5} [m^3/s].$$

Insieme di ingressi di controllo ammissibili:

$$Q = \{q : 0 \le q \le 5 \cdot 10^{-5}\},\$$

l'ingresso di controllo viene saturato superiormente per evitare la fuoriuscita del liquido dai serbatoi.

Forma del sistema linearizzato intorno all'equilibrio, facendo riferimento alla notazione introdotta in Sezione 3.2.4 con $J_H = A e J_q = B$:

$$\mathbf{A} = \begin{bmatrix} -0.0058 & 0 & 0\\ 0.0073 & -0.0096 & 0\\ 0 & 0.0066 & -0.0049 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 114.2857\\ 0\\ 0 \end{bmatrix}$$
$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1 \end{bmatrix}.$$

Test di controllabilità:

$$rango([\mathbf{B} \ \mathbf{AB} \ \mathbf{A}^2\mathbf{B}]) = 3$$

sistema completamente controllabile.

Modo M_1

Livelli di equilibrio:

$$H_{10} = 0, H_{20} = 0.098 \text{ e } H_{30} = 0.128 [m].$$

In questo caso poiché, come già spiegato, non ci è possibile agire sul livello del serbatoio in cui la valvola manuale è aperta, dal punto di vista del controllo lavoreremo come se il sistema fosse costituito solo dai due serbatoi in cui la valvola manuale è chiusa (il Middle e il Lower tank). Dunque i livelli da controllare risultano H_2 e H_3 e si suppone che la pompa porti il liquido direttamente al Middle tank. Guadagno ottimo del controllore LQR:

$$K_1^* = [0.2188 \cdot 10^{-3}, 0.1138 \cdot 10^{-3}]^T.$$

Ingesso di controllo di riferimento:

$$q_0 = 2.1 \cdot 10^{-5} [m^3/s].$$

Insieme di ingressi di controllo ammissibili:

$$Q = \{q : 0 \le q \le 11 \cdot 10^{-5}\}.$$

Forma del sistema a due serbatoi linearizzato intorno all'equilibrio:

$$\mathbf{A}_{1} = \begin{bmatrix} -0.0096 & 0\\ 0.0066 & -0.0049 \end{bmatrix}$$
$$\mathbf{B}_{1} = \begin{bmatrix} 144.7094\\ 0 \end{bmatrix}$$

Test di controllabilità:

 $rango([\mathbf{B}_1 \ \mathbf{A}_1\mathbf{B}_1]) = 2,$

sistema a due serbatoi completamente controllabile.

Modo M_2

Livelli di equilibrio:

$$H_{10} = 0.12, H_{20} = 0 \text{ e } H_{30} = 0.128 \text{ [}m\text{]}$$

In questo caso poiché, come già spiegato, non ci è possibile agire sul livello del serbatoio in cui la valvola manuale è aperta, dal punto di vista del controllo lavoreremo come se il sistema fosse costituito solo dai due serbatoi in cui la valvola manuale è chiusa (l'Upper e il Lower tank). Dunque i livelli da controllare risultano H_1 e H_3 .

Guadagno ottimo del controllore LQR:

$$K_2^* = \left[0.2275 \cdot 10^{-3}, 0.0932 \cdot 10^{-3}\right]^T$$

Ingesso di controllo di riferimento:

$$q_0 = 2.1 \cdot 10^{-5} [m^3/s].$$

Insieme di ingressi di controllo ammissibili:

$$Q = \{q : 0 \le q \le 7 \cdot 10^{-5}\}.$$

Forma del sistema a due serbatoi linearizzato intorno all'equilibrio:

$$\mathbf{A}_{2} = \begin{bmatrix} -0.0058 & 0\\ 0.0050 & -0.0049 \end{bmatrix}$$
$$\mathbf{B}_{2} = \begin{bmatrix} 114.2857\\ 0 \end{bmatrix}$$

Test di controllabilità:

 $rango(\begin{bmatrix} \mathbf{B}_2 & \mathbf{A}_2\mathbf{B}_2 \end{bmatrix}) = 2,$

sistema a due serbatoi completamente controllabile.

Modo M_3

Livelli di equilibrio:

$$H_{10} = 0.12, H_{20} = 0.098 \text{ e } H_{30} = 0 [m].$$

In questo caso poiché, come già spiegato, non ci è possibile agire sul livello del serbatoio in cui la valvola manuale è aperta, dal punto di vista del controllo lavoreremo come se il sistema fosse costituito solo dai due serbatoi in cui la valvola manuale è chiusa (l'Upper e il Middle tank). Dunque i livelli da controllare risultano H_1 e H_2 .

Guadagno ottimo del controllore LQR:

$$K_3^* = [0.2279 \cdot 10^{-3}, 0.0656 \cdot 10^{-3}]^T.$$

Ingesso di controllo di riferimento:

$$q_0 = 2.1 \cdot 10^{-5} [m^3/s]$$

Insieme di ingressi di controllo ammissibili:

$$Q = \{q : 0 \le q \le 5.5 \cdot 10^{-5}\}.$$

Forma del sistema a due serbatoi linearizzato intorno all'equilibrio:

$$\mathbf{A}_{3} = \begin{bmatrix} -0.0058 & 0\\ 0.0073 & -0.0096 \end{bmatrix}$$
$$\mathbf{B}_{3} = \begin{bmatrix} 114.2857\\ 0 \end{bmatrix}$$

Test di controllabilità:

$$rango([\mathbf{B}_3 \ \mathbf{A}_3\mathbf{B}_3]) = 2,$$

sistema a due serbatoi completamente controllabile.

Capitolo 5

Osservazione del modo discreto

In questo capitolo descriveremo la metodologia che abbiamo seguito per progettare l'osservatore dello stato discreto nel caso in cui lo stato continuo sia misurabile. Proporremo due algoritmi per stabilire il modo stimato e li verificheremo sperimentalmente sul sistema fisico three-tank e sul suo modello in ambiente Matlab/Simulink.

5.1 Metodologia

L'Osservatore di Modo (MO) che proponiamo consiste di un banco di *N* Isolatori di Modo (MI) in parallelo, con *N* il numero di modi del sistema. L'isolatore di modo per il modo M_i , che chiamiamo MI_i , è un osservatore progettato secondo l'equazione dello stato che descrive la dinamica del modo f_i . Tale osservatore prende in ingresso lo stato continuo corrente misurato ed è in grado di produrre una stima dell'uscita corrente che il sistema dovrebbe avere se stesse lavorando nel modo M_i , che indichiamo con \hat{y}_i . Si ottengono così tante stime dell'uscita del sistema quanti sono i modi. Ciascuna delle N stime \hat{y}_i viene confrontata con l'uscita misurata y, ottenendo così N residui, dove il residuo i-esimo generato dal MI_i è definito come $res_i = y - \hat{y}_i$. Nel caso in cui y sia un vettore, calcoleremo il nostro residuo sommando le componenti al quadrato del vettore res_i e lo chiameremo $r_i = (res_{1i}^2 + res_{2i}^2 + ... + res_{mi}^2) = ||res||^2$.

L'osservatore di modo è composto, oltre che dal blocco degli MI, da un'altro blocco che prende in ingresso gli *N* residui e li elabora utilizzando uno dei due algoritmi proposti per decidere in quale modo il sistema si trovi.

5.1.1 Il blocco degli isolatori di modo

Ogni singolo elemento del blocco degli MI ha la funzione di simulare la dinamica del sistema secondo un particolare modo e confrontarla con la dinamica reale misurata ad ogni passo di campionamento. Da questo confronto genera un residuo che è indice di quanto il sistema reale si discosta dalla dinamica simulata. Il residuo ha un valore piccolo quando il modo reale di funzionamento coincide col modo simulato.

Per generare la dinamica simulata per ogni modo, il blocco degli MI utilizza i valori misurati del passo precedente a quello da stimare. Poiché si lavora con dati campionati andiamo subito a descrivere il sistema con un modello a tempo-discreto. Partendo dal modello non-lineare nel continuo, descritto nella Sezione 3.2.3, se indichiamo con ΔT il passo di campionamento utilizzato (0.1 secondi nelle nostre simulazioni), possiamo scrivere la dinamica del sistema a tempo-discreto corrispondente al modo M_0 come:

$$\begin{aligned} H_{1}(k+1) &= H_{1}(k) + \left[\frac{1}{aw}q - \frac{1}{aw}C_{1}H_{1}^{\alpha_{1}}(k)\right]\Delta T \\ H_{2}(k+1) &= H_{2}(k) + \left[\frac{1}{cw + bw\frac{H_{2}(k)}{H_{2max}}}C_{1}H_{1}^{\alpha_{1}}(k) - \frac{1}{cw + bw\frac{H_{2}(k)}{H_{2max}}}C_{2}H_{2}^{\alpha_{2}}(k)\right]\Delta T \\ H_{3}(k+1) &= H_{3}(k) + \left[\frac{C_{2}H_{2}^{\alpha_{2}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}} - \frac{C_{3}H_{3}^{\alpha_{3}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}}\right]\Delta T \end{aligned}$$
(5.1)

Per il modo M_1 :

$$H_{1}(k+1) = H_{1}(k) + \left[\frac{1}{aw}q - \frac{C_{1}H_{1}^{\alpha_{1}}(k) + C_{m1}H_{1}^{\alpha_{m1}}(k)}{aw}\right]\Delta T$$

$$H_{2}(k+1) = H_{2}(k) + \left[\frac{C_{1}H_{1}^{\alpha_{1}}(k) + C_{m1}H_{1}^{\alpha_{m1}}(k)}{cw + bw\frac{H_{2}(k)}{H_{2max}}} - \frac{1}{cw + bw\frac{H_{2}(k)}{H_{2max}}}C_{2}H_{2}^{\alpha_{2}}(k)\right]\Delta T$$

$$H_{3}(k+1) = H_{3}(k) + \left[\frac{C_{2}H_{2}^{\alpha_{2}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}} - \frac{C_{3}H_{3}^{\alpha_{3}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}}\right]\Delta T$$
(5.2)

Per il modo M_2 :

$$H_{1}(k+1) = H_{1}(k) + \left[\frac{1}{aw}q - \frac{1}{aw}C_{1}H_{1}^{\alpha_{1}}(k)\right]\Delta T$$

$$H_{2}(k+1) = H_{2}(k) + \left[\frac{C_{1}H_{1}^{\alpha_{1}}(k)}{cw + bw\frac{H_{2}(k)}{H_{2max}}} - \frac{C_{2}H_{2}^{\alpha_{2}}(k) + C_{m2}H_{2}^{\alpha_{m2}}(k)}{cw + bw\frac{H_{2}(k)}{H_{2max}}}\right]\Delta T$$

$$H_{3}(k+1) = H_{3}(k) + \left[\frac{C_{2}H_{2}^{\alpha_{2}}(k) + C_{m2}H_{2}^{\alpha_{m2}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}} - \frac{C_{3}H_{3}^{\alpha_{3}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}}\right]\Delta T$$
(5.3)
Per il modo M_3 :

$$H_{1}(k+1) = H_{1}(k) + \left[\frac{1}{aw}q - \frac{1}{aw}C_{1}H_{1}^{\alpha_{1}}(k)\right]\Delta T$$

$$H_{2}(k+1) = H_{2}(k) + \left[\frac{1}{cw + bw\frac{H_{2}(k)}{H_{2max}}}C_{1}H_{1}^{\alpha_{1}}(k) - \frac{1}{cw + bw\frac{H_{2}(k)}{H_{2max}}}C_{2}H_{2}^{\alpha_{2}}(k)\right]\Delta T$$

$$H_{3}(k+1) = H_{3}(k) + \left[\frac{C_{2}H_{2}^{\alpha_{2}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}} - \frac{C_{3}H_{3}^{\alpha_{3}}(k) + C_{m3}H_{3}^{\alpha_{m3}}(k)}{w\sqrt{R^{2} - (H_{3max} - H_{3}(k))^{2}}}\right]\Delta T$$
(5.4)

In questa Sezione, come già detto, consideriamo il caso in cui la dinamica reale del sistema è quella misurata. In altre parole, consideriamo il caso in cui lo stato continuo viene direttamente osservato poiché l'uscita \vec{y} coincide con esso:

$$\vec{y} = \mathbf{H} = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix}$$

Con questa configurazione del sistema, lo stato 'predetto' $\hat{\mathbf{H}}_{M_i}(k+1)$ dall'elemento i-esimo del blocco degli MI viene calcolato come

$$\widehat{\mathbf{H}}_{M_i}(k+1) = f_{M_i}(q(k), \vec{y}(k))$$

dove:

- f_{M_i} è una tra le eq. (5.1), eq. (5.2), eq. (5.3) e eq. (5.4),
- q(k) è la portata della pompa al passo k,
- $\vec{y}(k)$ è l'uscita del sistema misurata al passo k, cioè nelle eq. (5.1), eq. (5.2), eq. (5.3) e eq. (5.4) si pone $\mathbf{H}(k) = \vec{y}(k)$.

I residui vengono calcolati come la differenza tra livelli misurati e livelli stimati, $\vec{y}(k+1) - \hat{\mathbf{H}}_{M_i}(k+1)$, e le componenti di questa differenza vengono sommate al quadrato per evitare compensazioni di segno. In questo modo otteniamo un residuo r_i per ogni modulo MI_i .

5.1.2 I residui

Ora abbiamo quattro residui r da cui dobbiamo capire in che modo il sistema sta funzionando.

In Tabella 5.1 e Tabella 5.2 mostriamo i valori assunti dai quattro residui nei quattro diversi modi a regime, rispettivamente, nel modello del three-tank in Simulink e nel sistema reale.

	M_0	M_1	M_2	M_3
<i>r</i> ₀	$\sim 10^{-10}$	$\sim 10^{-7}$	$\sim 2 \cdot 10^{-7}$	$\sim 2 \cdot 10^{-7}$
r_1	$\sim 3 \cdot 10^{-6}$	$\sim 10^{-12}$	$\sim 10^{-5}$	$\sim 3 \cdot 10^{-6}$
r_2	$\sim 3 \cdot 10^{-6}$	$\sim 3 \cdot 10^{-6}$	$\sim 10^{-10}$	$\sim 10^{-5}$
<i>r</i> ₃	$\sim 10^{-6}$	$\sim 10^{-6}$	$\sim 10^{-6}$	$\sim 4 \cdot 10^{-10}$

Tabella 5.1: Modello del three-tank - Valori assunti dai residui nei diversi modi a regime

	M_0	M_1	M_2	M_3
<i>r</i> ₀	$\sim 2 \cdot 10^{-8}$	$\sim 4 \cdot 10^{-7}$	$\sim 9 \cdot 10^{-7}$	$\sim 10^{-6}$
r_1	$\sim 6 \cdot 10^{-6}$	$\sim 7 \cdot 10^{-8}$	$\sim 7.5 \cdot 10^{-5}$	$\sim 7 \cdot 10^{-6}$
r_2	$\sim 7.5 \cdot 10^{-6}$	$\sim 9 \cdot 10^{-6}$	$\sim 1.2 \cdot 10^{-7}$	$\sim 2.5 \cdot 10^{-5}$
<i>r</i> ₃	$\sim 2 \cdot 10^{-6}$	$\sim 2 \cdot 10^{-6}$	$\sim 3.5 \cdot 10^{-6}$	$\sim 2.5\cdot 10^{-7}$

Tabella 5.2: Sistema reale three-tank - Valori assunti dai residui nei diversi modi a regime

Notiamo subito che i valori dei residui che corrispondono al modo esatto di funzionamento (in **grassetto** nelle tabelle) risultano di qualche ordine di grandezza, nel caso del modello, e di quasi un ordine di grandezza, nel caso del sistema reale, più piccoli degli altri residui.

Si può inoltre osservare che il residuo r_0 , se escludiamo i valori in **grassetto**, risulta mediamente più piccolo di quasi un ordine di grandezza rispetto agli altri residui; questo si può spiegare con il fatto che r_0 è il residuo associato all'isolatore di modo MI_0 , cioè quello che simula la dinamica del sistema per il caso in cui tutte le valvole manuali sono chiuse. Questa configurazione si può vedere come centrale rispetto alle altre configurazioni in cui anche una valvola manuale è aperta. In altre parole il modo M_0 è quello che si distanzia meno dagli altri modi poiché si differenzia da questi per la diversa configurazione di una sola valvola. Questo, invece, non si verifica per i modi M_1 , M_2 e M_3 che tra di loro hanno due valvole configurate diversamente.

Infine osserviamo che il residuo r_3 presenta valori lievemente inferiori rispetto a quelli di r_1 e r_2 , poiché l'apertura della valvola manuale del Lower tank cambia la dinamica del solo Lower tank e non ha effetti sui livelli degli altri serbatoi.

Partendo dalle considerazioni fatte proponiamo due algoritmi con cui elaboriamo i residui per ottenere una stima del modo di funzionamento del sistema.

Algoritmo 1

Nelle seguenti considerazioni parleremo di 'transitorio' riferendoci agli istanti successivi a un evento che provoca un cambiamento di modo del sistema e parleremo di 'regime' riferendoci alle altre fasi di funzionamento del sistema. L'**algoritmo 1** nasce dalla considerazione, confermata dai dati sperimentali, Tabella 5.1 e Tabella 5.2, che, quando il sistema è a 'regime', i residui mantengono un andamento costante e il residuo più piccolo è quello associato al modo corrente del sistema reale. In altri termini, se il sistema funziona nel modo M_i allora il residuo r_i è il più piccolo tra gli N residui. Questo invece non vale quando il sistema è in una situazione di 'transitorio'; in questi casi i residui variano velocemente e in particolare in una transizione $M_i \rightarrow M_i$ si ha che il residuo r_i è quello che normalmente decresce più velocemente fino a diventare il residuo più piccolo, il residuo r_i cresce velocemente e gli altri residui hanno anch'essi variazioni rapide, alcuni crescenti altri decrescenti. Quindi, durante gli istanti successivi a un cambiamento di modo, normalmente, il residuo associato al modo corrente non è necessariamente quello più piccolo. Per tenere conto di queste variazioni è quindi utile inserire come termini di decisione anche le derivate dei residui d. Più precisamente si fissano delle soglie sperimentali s_i (ovviamente minori di zero), una per ogni residuo, e si stabilisce che, quando la derivata di un residuo sta al di sotto della sua soglia, allora il modo stimato è quello associato a questo residuo. Riepilogando, in una situazione di 'regime' si ha che il modo stimato \hat{M} è quello associato al residuo r più piccolo, mentre, in una situazione di 'transitorio' il modo stimato \hat{M} è associato alla derivata più piccola tra le derivate dei residui d quando questa è inferiore alla sua soglia s.

Poiché si lavora su dati campionati è opportuno precisare cosa intendiamo con la derivata d dei residui. Calcoliamo la derivata del residuo r come:

$$d = \frac{dr}{dt}$$

dove dr è la variazione del residuo e dt è la variazione di tempo rispetto al passo precedente di simulazione. L'accuratezza dell'approssimazione dipende dal passo di campionamento utilizzato nella simulazione. Più è piccolo il passo di campionamento più la derivata ottenuta risulta liscia e accurata. Abbiamo inizialmente svolto simulazioni con passo 0.1 e 0.01 secondi senza riscontrare differenze significative, perciò si è scelto di lavorare unicamente con passo di campionamento 0.1 secondi. In ambiente Matlab/Simulink per il calcolo di d ci siamo serviti del blocco derivatore. Il blocco derivatore ha come uscita iniziale il valore zero.

Diamo ora una definizione formale dell'algoritmo 1:

-
$$\rho := \arg \min_{j=0,1...,m} r_j(k)$$

- $\delta := \arg \min_{j=0,1...,m} d_j(k)$
- $\mathbf{if}(d_{\delta}(k) \le s_{\delta})$
 $i^* = \delta;$
else
 $i^* = \rho;$
end
- $\hat{M}(k) := M_{i^*}(k)$

dove:

- $\rho = \operatorname{argmin}_{i=0,1,\dots,m} r_i(k)$ è l'indice del residuo più piccolo al passo k,
- $\delta = \operatorname{argmin}_{j=0,1,\dots,m} d_j(k)$ è l'indice del residuo con la derivata più piccola al passo k,
- $\hat{M}(k) = M_{i^*}(k)$ è il modo stimato al passo k.

Di seguito mostriamo l'andamento dei residui, Figura 5.1, e delle derivate di questi, Figura(5.2), durante i diversi modi di lavoro del sistema; i dati illustrati si riferiscono a una prova effettuata sul modello del sistema in cui l'osservatore dei modi (MO) elabora i residui secondo l'**algoritmo 1**.



Figura 5.1: I residui r_0 , r_1 , r_2 , r_3 al variare dei modi

Osservando le derivate dei residui sono evidenti i picchi che si presentano ad ogni transizione di modo, e in particolare i picchi negativi dei residui associati al modo di arrivo della transizione.



Figura 5.2: Le derivate dei residui r_0 , r_1 , r_2 , r_3 al variare dei modi

Le Soglie

Finora abbiamo definito la soglia s_{δ} come un numero reale negativo per cui, se la derivata del residuo d_{δ} ha un valore inferiore a s_{δ} , allora, il modo di funzionamento del sistema è il modo M_{δ} . Normalmente la derivata del residuo d_{δ} raggiunge un valore inferiore alla soglia ad esso associata quando avviene una commutazione da un generico modo di partenza verso il modo δ -esimo, questo poiché, come è logico aspettarsi, il residuo r_{δ} decrescerà velocemente fino a diventare il residuo più piccolo.

Vogliamo ora dare alcune indicazioni su come le soglie possono essere ricavate. Non forniamo un algoritmo con cui calcolare le soglie ottimali ma forniamo una procedura che attraverso l'analisi del comportamento assunto dalle derivate *d* ad ogni transizione permette di ricavare un valore preliminare di *s* congruo, che poi dovrà essere affinato procedendo per tentativi.

Per maggior chiarezza, concentriamoci ora sul comportamento di una sola derivata, la derivata d_1 , per mostrare come sia possibile ottenere un valore preliminare della soglia s_1 attraverso una prima simulazione, in seguito estenderemo le nostre considerazioni alle restanti derivate dei residui. Si considera il sistema a tre serbatoi con il segnale di controllo in ingresso costante e fissato al valore di riferimento q_0 . Si effettua una simulazione in cui tutte le transizioni devono essere testate e possibilmente devono essere ripetute più volte differenziandosi di volta in volta per il livello del liquido al

momento della transizione. In Figura 5.3 mostriamo l'andamento della sola derivata d_1 , abbiamo contrassegnato con dei pallini tutti i minimi di d_1 significativi per la scelta della soglia, e chiamiamo D_1 l'insieme che contiene tutti i minimi. Consideriamo l'insieme $D_1^+ \subset D_1$ dei minimi (contrassegnati in figura con un pallino nero) che cadono negli istanti immediatamente successivi a una transizione che ha come modo di arrivo un modo diverso da M_1 . È possibile tracciare una retta (tratteggiata in figura) all'altezza $s_1^* = -51.02 \cdot 10^{-7}$ corrispondente al minimo più piccolo tra i minimi appartenenti a D_1^+ .



Figura 5.3: Simulazione finalizzata all'individuazione della soglia s_1 associata a d_1

Notiamo come i minimi rimanenti, cioè quelli appartenenti all'insieme $D_1^- = D_1 - D_1^+$, che corrispondono agli istanti immediatamente successivi alle transizioni in cui il modo di arrivo è M_1 (contrassegnati in figura con pallini rossi), cadono tutti al di sotto di s_1^* . Ora se consideriamo tutti i minimi dell'insieme D_1^- possiamo tracciare una retta (tratteggiata in figura) all'altezza $d_1^* = -97.9 \cdot 10^{-7}$ corrispondente al minimo più grande tra i minimi appartenenti a D_1^- . Dunque succede che tutti i minimi appartenenti a D_1^+ cadono al di sopra di d_1^* e che tutti i minimi appartenenti a D_1^- cadono al di sotto di s_1^* . Abbiamo quindi individuato una regione $S_1 = \{d_1 : d_1^* < d_1 < s_1^*\}$ (contenuta tra le due rette tratteggiate in figura) che separa completamente gli insiemi D_1^+ e D_1^- . La soglia preliminare deve essere scelta all'interno della regione S_1 . Abbiamo quindi descritto una semplice procedura che ci indica come separare i picchi d'interesse dagli altri picchi.

Ora mostriamo in Figura 5.4, Figura 5.5 e Figura 5.6 come la procedura sia valida anche per le altre tre derivate dei residui.



Figura 5.4: Simulazione finalizzata all'individuazione della soglia s_0 associata a d_0

Dunque in generale nel nostro sistema $\forall d_i$ riusciamo sempre a trovare una soglia $s_i \in S_i = \{d_i : d_i^* < d_i < s_i^*\}$ tale che:

$$d_{i\,i}^{-} < s_i < d_{i\,k}^{+}$$

 $\forall d_{ij}^- \in D_i^- \in \forall d_{ik}^+ \in D_i^+$, dove:

- D_i^+ è l'insieme di tutti i minimi di d_i che si verificano negli istanti immediatamente successivi a una transizione che ha come modo di arrivo un modo diverso da M_i ,
- D_i^- è l'insieme di tutti i minimi di d_i che si verificano negli istanti immediatamente successivi a una transizione che ha come modo di arrivo M_i ,
- $d_i^* = \max(D_i^-)$,
- $s_i^* = \min(D_i^+).$



Figura 5.5: Simulazione finalizzata all'individuazione della soglia s_2 associata a d_2



Figura 5.6: Simulazione finalizzata all'individuazione della soglia s_3 associata a d_3

Facciamo ora un riepilogo dei risultati ottenuti con le procedure illustrate in Figura 5.3, Figura 5.4, Figura 5.5 e Figura 5.6 definendo le regioni individuate per la scelta delle soglie:

$$S_{0} = \{d_{0}: -2.85 \cdot 10^{-7} < d_{0} < -1.81 \cdot 10^{-7}\},$$

$$S_{1} = \{d_{1}: -97.9 \cdot 10^{-7} < d_{1} < -51.02 \cdot 10^{-7}\},$$

$$S_{2} = \{d_{2}: -91.8 \cdot 10^{-7} < d_{2} < -39.8 \cdot 10^{-7}\},$$

$$S_{3} = \{d_{3}: -50.95 \cdot 10^{-7} < d_{3} < -11.8 \cdot 10^{-7}\}.$$

In Tabella 5.3 mostriamo i valori delle soglie che abbiamo utilizzato nell'MO del sistema reale e nell'MO del modello del sistema.

	<i>s</i> ₀	s_1	<i>s</i> ₂	<i>s</i> ₃	
modello	$-2.8 \cdot 10^{-7}$	$-47.5 \cdot 10^{-7}$	$41.3 \cdot 10^{-7}$	$-12.4 \cdot 10^{-7}$	
sistema reale	$-12 \cdot 10^{-7}$	$-40 \cdot 10^{-7}$	$-29 \cdot 10^{-7}$	$-21 \cdot 10^{-7}$	

Tabella 5.3: Soglie per il modello e per il sistema reale

Notiamo subito come il valore di s_1 mostrato in tabella ($s_1 = -47.5 \cdot 10^{-7}$) sia diverso da quello calcolato in precedenza ($s_1 = -51.02 \cdot 10^{-7}$), questo perché, come già detto, quest'ultimo è un valore preliminare ricavato dalla sola prima simulazione. Per affinare il valore della soglia è quindi importante svolgere ulteriori simulazioni con il sistema controllato con il controllore descritto nella Sezione 4.6 e osservato con l'osservatore configurato con le soglie preliminari. L'aggiunta del controllo modifica leggermente l'andamento delle derivate e nel nostro caso abbiamo potuto sollevare la soglia s_1 ottenendo il valore mostrato in tabella. Per quel che riguarda le altre soglie abbiamo che i valori scelti appartengono alle regioni S_i individuate.

La procedura per il calcolo delle soglie qui descritta presuppone che il progettista abbia la possibilità di agire sul sistema per poter simulare tutte le transizioni di modo nei tempi da lui voluti, cioè che nelle simulazioni finalizzate all'individuazione delle soglie abbia la possibilità di controllare gli eventi discreti. Questo è naturalmente possibile nelle simulazioni sul modello in ambiente Matlab/Simulink e nel nostro caso anche sul sistema reale dato che è possibile agire sulle valvole manuali.

Algoritmo 2

Le considerazioni fatte per motivare l'**algoritmo 1** valgono anche in questo caso, ma, al posto di utilizzare due parametri di riferimento, i residui e le loro derivate, si definisce un unico parametro come somma pesata di questi:

$$R_i(k) = \alpha_i r_i(k) + \beta_i d_i(k)$$

con $\alpha_i \in \mathbb{R}$ costante utilizzata per rendere dello stesso ordine di grandezza i residui e $\beta_i \in \mathbb{R}$ del tipo:

$$\beta_i(d_i) = \begin{cases} \beta_i^+ & \text{se } d_i(k) > 0\\ \beta_i^- & \text{se } d_i(k) \le 0 \end{cases}$$

utilizzato per aumentare in alcuni casi il contributo della derivata.

Formalmente, definiamo il modo stimato al passo k, come:

$$\hat{M}(k) = M_{i^*}(k), \text{ con } i^* = \arg\min_{j=1,2,\dots,m} \{R_j(k)\}.$$

In riferimento all'**algoritmo 2**, mostriamo l'andamento dei parametri R_i costruiti come somma pesata del residuo e della sua derivata.



Figura 5.7: Andamento dei residui R_0 , R_1 , R_2 e R_3 , definiti nell'algoritmo 2, al variare dei modi

	α_0	α_1	α_2	α_3	β_0^+	β_1^+	β_2^+	β_3^+	eta_0^-	β_1^-	β_2^-	β_3^-
modello	3	1	1	3	2.5	1	1	2	1	1	1	1
sistema reale	1.5	1	1	1.5	1	1	1	1	1	1.1	1.1	1.4

In Tabella 5.4 mostriamo i pesi α e β utilizzati nell'osservatore del sistema reale e nell'osservatore del modello del sistema.

Anche in questo caso non forniamo un algoritmo per il calcolo dei pesi, i quali sono stati ricavati utilizzando le informazioni che provengono dall'analisi dei residui r e delle derivate di questi d. Possiamo osservare che i valori scelti per i pesi $\alpha_0 e \alpha_3$ sono finalizzati a ridurre la distanza, documentata in Tabella 5.1 e in Tabella 5.2, dei residui $r_0 e r_3$ dai residui $r_1 e r_2$. I valori di $\beta_0^+ e \beta_3^+$ utilizzati nel modello servono per pesare maggiormente le derivate $d_0 e d_3$ quando queste sono maggiori di zero, questo serve per cercare di eliminare le osservazioni errate durante le transizioni dovute sempre al fatto che $r_0 e r_3$ rimangono sempre relativamente 'basse'. Questo tipo di osservazioni errate saranno mostrate nella prossima sezione dedicata alle simulazioni. Per ultimo i valori dei β^- nel sistema reale vengono utilizzati per pesare maggiormente le derivate d_3 in modo da minimizzare i ritardi di osservazione.

5.2 Simulazioni: confronto tra sistema reale e modello

In questa sezione mostreremo i risultati ottenuti nelle simulazioni sul sistema reale e sul modello applicando i due algoritmi.

Nelle simulazioni, utilizzate per testare l'osservatore discreto, abbiamo sperimentato tutti gli eventi possibili, vedi Tabella 3.1, in modo da verificare il comportamento del MO in tutte le transizioni.

In Figura 5.8 mostriamo il modo osservato sul modello del sistema utilizzando l'**algoritmo 1**, mentre in Figura 5.9 mostriamo i risultati corrispondenti ottenuti sul sistema reale. Abbiamo indicato col tratteggio il modo di funzionamento reale del sistema e con la linea piena il modo osservato.

Per quanto riguarda i risultati sul modello, si osserva che, nell'arco di massimo un secondo dalla transizione di modo, l'osservatore dello stato discreto (MO) capisce che è avvenuta una transizione e identifica correttamente il nuovo modo. In alcune transizioni, precisamente nelle transizioni $M_1 \rightarrow M_2$, $M_1 \rightarrow M_3$ e $M_3 \rightarrow M_2$, per alcuni decimi di secondo viene osservato un modo sbagliato. Il modo errato è in tutti e tre i casi l' M_0 , questo si spiega con il fatto che il residuo r_0 , come osservato nella Sezione precedente, ha mediamente un valore inferiore di quasi un ordine di grandezza rispetto agli altri residui. Dunque, se consideriamo la transizione $M_1 \rightarrow M_2$, si ha che nell'istante



Figura 5.8: Modi osservati sul modello del three-tank utilizzando l'algoritmo 1



Figura 5.9: Modi osservati sul sistema three-tank reale utilizzando l'algoritmo 1

di commutazione il residuo r_1 diventa maggiore di r_0 e allo stesso tempo il residuo r_2 non è ancora inferiore ad r_0 , inoltre la derivata \dot{r}_2 impiega qualche decimo di secondo per raggiungere la soglia s_2 , e questi pochi decimi di secondo sono gli istanti in cui si verifica l'errore di osservazione.

Le osservazioni sul sistema reale sono, come è normale aspettarsi, qualitativamente inferiori rispetto a quelle sul modello, anche se i risultati ottenuti non sono molto distanti. Si presentano lo stesso tipo di errori di osservazione dovuti alla natura del residuo r_0 , ma sul sistema reale il ritardo di osservazione è mediamente intorno ai $3 \div 4$ secondi, e non dell'ordine dei decimi di secondo come nel caso del modello.

In Figura 5.10 e Figura 5.11 illustriamo i risultati ottenuti, eseguendo la stessa simulazione sul modello e sul sistema reale, utilizzando l'**algoritmo 2**.



Figura 5.10: Modi osservati sul modello del three-tank utilizzando l'algoritmo 2



Figura 5.11: Modi osservati sul sistema three-tank reale utilizzando l'algoritmo 2

I risultati sul modello sono molto simili a quelli ottenuti con l'**algoritmo 1**. Questo invece non accade nell'osservazione del sistema reale che si presenta qualitativamente lievemente inferiore a quella ottenuta con l'**algoritmo 1**, per il numero maggiore di osservazioni errate.

In Figura 5.12 mostriamo un confronto dei risultati ottenuti sul sistema reale coi due diversi algoritmi.

Concentriamoci sui primi secondi della simulazione, si nota che l'osservatore progettato secondo l'**algoritmo 2** osserva il modo sbagliato, poiché crede di essere nel modo M_1 quando in realtà il sistema è nel modo M_0 . Questo comportamento è dovuto al fatto che nel sistema reale la risposta della pompa ha un ritardo rispetto al segnale di controllo. Questo implica che all'inizio della simulazione l'osservatore vede la pompa in funzione quando in realtà il flusso non è ancora iniziato. Dunque l'osservatore interpreta la mancanza di flusso in ingresso come un maggior flusso in uscita, che equivale all'apertura della valvola manuale dell'Upper tank, modo M_1 . Questo effetto non ha influenza sull'osservatore disegnato secondo l'**algoritmo 1** poiché con la scelta delle soglie si riesce a rendere il MO meno sensibile a questo comportamento. Sempre alla scelta delle soglie è dovuto il minor numero di errori di osservazione dell'**algoritmo 1** rispetto all'**algoritmo 2**. Dall'altra parte l'**algoritmo 2** presenta dei ritardi di osservazione lievemente inferiori, pochi decimi di secondo in meno, e questo si spiega col fatto



Figura 5.12: Confronto dei risultati ottenuti osservando il sistema reale utilizzando i due algoritmi proposti

che le soglie introdotte nell'**algoritmo 1** provocano l'aumento del tempo di risposta dell'osservatore alle transizioni.

5.3 Simulazioni aggiuntive

In questa sezione descriveremo in breve i risultati di due ulteriori simulazioni sul modello del three-tank in cui abbiamo applicato il MO presentato in precedenza.

Simulazione 1

La prima simulazione che presentiamo si differenzia dalla precedente per il blocco di controllo. Facendo riferimento al controllore presentato nella Sezione 4.6 il controllore utilizzato in questa simulazione si differenzia per una diversa portata di riferimento:

$$q_0 = 1.791 \cdot 10^{-5} [m^3/s].$$

In questo caso si ottengono i seguenti valori:

Sistema nel modo M_0

Livelli di equilibrio:

$$H_{10} = 0.07, H_{20} = 0.058 \text{ e } H_{30} = 0.076 \text{ [}m\text{]}.$$

Guadagno ottimo del controllore LQR:

$$K_0^* = [0.2313 \cdot 10^{-3}, 0.0901 \cdot 10^{-3}, 0.0529 \cdot 10^{-3}]^T.$$

Sistema nel modo M_1

Livelli di equilibrio:

$$H_{10} = 0, H_{20} = 0.058 \text{ e } H_{30} = 0.076 [m].$$

Guadagno ottimo del controllore LQR:

$$K_1^* = [0.2032 \cdot 10^{-3}, 0.1075 \cdot 10^{-3}]^T$$

Sistema nel modo M_2

Livelli di equilibrio:

$$H_{10} = 0.07, H_{20} = 0 \text{ e } H_{30} = 0.076 [m].$$

Guadagno ottimo del controllore LQR:

$$K_2^* = \left[0.2178 \cdot 10^{-3}, 0.0853 \cdot 10^{-3}\right]^T$$
.

Sistema nel modo M_3

Livelli di equilibrio:

Guadagno ottimo del controllore LQR:

$$K_3^* = [0.2178 \cdot 10^{-3}, 0.0559 \cdot 10^{-3}]^T.$$

In Figura 5.13 e in Figura 5.14 mostriamo, rispettivamente, i risultati dell'osservazione del sistema ottenuti utilizzando l'**algoritmo 1** e l'**algoritmo 2**.



Figura 5.13: Simualzione 1 - modi osservati utilizzando l'algoritmo 1



Figura 5.14: Simulazione 1 - modi osservati utilizzando l'algoritmo 2

Si vuole sottolineare che il MO ha come stato iniziale il modo M_0 . Si può notare come in pochi decimi di secondo il MO si accorge di essere nel modo M_1 .

Per l'algoritmo 1 sono state utilizzate le seguenti soglie:

$$s = [-4.6 \cdot 10^{-7}, -34 \cdot 10^{-7}, -41 \cdot 10^{-7}, -9.9 \cdot 10^{-7}].$$

Per l'algoritmo 2 abbiamo utilizzato i pesi illustrati in Tabella 5.4.

Mostriamo ora i livelli nei tre serbatoi e l'ingresso di controllo, rispettivamente, in Figura 5.15 e in Figura 5.16.



Figura 5.15: Simulazione 1 - Livelli nei tre serbatoi



Figura 5.16: Simulazione 1 - Ingresso di controllo

Simulazione 2

In questa simulazione abbiamo utilizzato un blocco di controllo che usa una portata di riferimento q_{0i} diversa per ogni modo di funzionamento M_i .

In questo caso si ottengono i seguenti valori:

Sistema nel modo M₀

Portata di riferimento:

$$q_{00} = 2.1 \cdot 10^{-5} [m^3/s].$$

Livelli di equilibrio:

$$H_{10} = 0.12, H_{20} = 0.098 \text{ e } H_{30} = 0.128 [m].$$

Guadagno ottimo del controllore LQR:

$$K_0^* = [0.2360 \cdot 10^{-3}, 0.1014 \cdot 10^{-3}, 0.0570 \cdot 10^{-3}]^T.$$

Sistema nel modo M_1

Portata di riferimento:

$$q_{01} = 1.791 \cdot 10^{-5} [m^3/s].$$

Livelli di equilibrio:

$$H_{10} = 0, H_{20} = 0.058 \text{ e} H_{30} = 0.076 [m].$$

Guadagno ottimo del controllore LQR:

$$K_1^* = [0.2032 \cdot 10^{-3}, 0.1075 \cdot 10^{-3}]^T$$

Sistema nel modo M_2

Portata di riferimento:

$$q_{02} = 1.575 \cdot 10^{-5} [m^3/s].$$

Livelli di equilibrio:

$$H_{10} = 0.045, H_{20} = 0 \text{ e } H_{30} = 0.05 [m].$$

Guadagno ottimo del controllore LQR:

$$K_2^* = [0.2078 \cdot 10^{-3}, 0.0768 \cdot 10^{-3}]^T.$$

Sistema nel modo M_3

Portata di riferimento:

$$q_{03} = 2.236 \cdot 10^{-5} [m^3/s].$$

Livelli di equilibrio:

$$H_{10} = 0.15, H_{20} = 0.12 \text{ e } H_{30} = 0 [m].$$

Guadagno ottimo del controllore LQR:

$$K_3^* = [0.2311 \cdot 10^{-3}, 0.0683 \cdot 10^{-3}]^T.$$

In Figura 5.17 e in Figura 5.18 mostriamo, rispettivamente, i risultati dell'osservazione del sistema ottenuti utilizzando l'**algoritmo 1** e l'**algoritmo 2**.

Per l'algoritmo 1 sono state utilizzate le seguenti soglie:

$$s = \left[-8.5 \cdot 10^{-7}, -19.8 \cdot 10^{-7}, -45 \cdot 10^{-7}, -17.5 \cdot 10^{-7}\right].$$

Per l'algoritmo 2 abbiamo utilizzato i pesi illustrati in Tabella 5.4.



Figura 5.17: Simualzione 2 - modi osservati utilizzando l'algoritmo 1



Figura 5.18: Simualzione 2 - modi osservati utilizzando l'algoritmo 2

Mostriamo ora i livelli nei tre serbatoi e l'ingresso di controllo, rispettivamente, in Figura 5.19 e in Figura 5.20.



Figura 5.19: Simulazione 2 - Livelli nei tre serbatoi



Figura 5.20: Simulazione 2 - Ingresso di controllo

Capitolo 6

Osservazione del modo continuo e discreto

In questo capitolo consideriamo per completezza il caso in cui lo stato continuo non sia noto ma debba anch'esso essere osservato. Proponiamo una metodologia che abbiamo verificato sperimentalmente essere valida sotto certe condizioni sul modello del nostro sistema, ma di cui non forniamo un'esauriente spiegazione teorica (quali studio di stabilità e osservabilità del sistema) in grado di darne una dimostrazione di validità generale.

6.1 Metodologia

L'osservatore di modo (MO) deve essere in grado di capire in che modo il sistema sta funzionando essendo noti l'ingresso e l'uscita del sistema. L'uscita continua del sistema, $\vec{y}(k) \in \mathbb{R}^m$, non coincide più con lo stato, $\vec{x}(k) \in \mathbb{R}^n$, ma è della forma:

 $\vec{y}(k) = \mathbf{C}\vec{x}(k)$, con **C** tale che: $rango(\mathbf{C}) < n$.

In questo caso ogni isolatore di modo (MI) deve produrre una stima dello stato continuo a partire dalla stima al passo precedente e non a partire dalla misura al passo precedente. Perciò, per minimizzare l'errore di stima, sostituiamo gli isolatori di modo descritti nella Sezione 5.1.1, con dei Filtri di Kalman Estesi (EKF).

Ogni MI produce una stima dello stato continuo, \vec{x} , dunque si hanno quattro stime diverse a ogni istante di campionamento, di queste stime solo una, quella generata dall'MI associato al modo reale di funzionamento, ha un valore che si avvicina allo stato reale del sistema, le altre stime si possono discostare anche di molto. Per questo motivo, per generare una stima dello stato più accurata, si utilizza un blocco diverso dall'MO. La stima dello stato continuo, \hat{x} , viene prodotta da un osservatore dello stato continuo (CO), che può essere considerato come un sistema che ha più modi di funzionamento la cui dinamica di funzionamento dipende dalla stima del modo, \hat{M} , generata dall'osservatore di modo.

6.1.1 Il blocco degli isolatori di modo

L'MO è formato da un banco di MI, ciascuno dei quali è un EKF:

$$\begin{cases} \hat{x}_{M_i}(k+1|k) = f_{M_i}(\hat{x}_{M_i}(k|k), u(k)) \\ \mathbf{P}_{M_i}(k+1|k) = \mathbf{A}_{M_i}(k)\mathbf{P}_{M_i}(k|k)\mathbf{A}_{M_i}^T(k) + \mathbf{Q}_{M_i}(k) \\ \hat{x}_{M_i}(k+1|k+1) = \hat{x}_{M_i}(k+1|k) + \mathbf{K}_{M_i}(k+1)(y(k+1) - \mathbf{C}\hat{x}_{M_i}(k+1|k)) \\ \mathbf{P}_{M_i}(k+1|k+1) = (\mathbf{I} - \mathbf{K}_{M_i}(k+1)\mathbf{C})\mathbf{P}_{M_i}(k+1|k) \end{cases}$$
(6.1)

per ogni modo M_i .

Dove f_{M_i} è una tra le eq. (5.1), eq. (5.2), eq. (5.3) e eq. (5.4), e

$$\mathbf{A}_{M_i}(k) = \left. \frac{\partial f_{M_i}(x_{M_i}(k), u(k))}{\partial x_{M_i}(k)} \right|_{x_{M_i}(k) = \hat{x}_{M_i}(k)}$$

Abbiamo scelto arbitrariamente una matrice C pari a:

$$\mathbf{C} = \left[\begin{array}{rrr} 0 & 1 & 1 \\ 2 & 4 & 3 \end{array} \right]$$

 $\operatorname{con} rango(\mathbf{C}) = 2 < n = 3.$

Il residuo i-esimo viene calcolato come la norma quadra del vettore

$$y(k+1) - \mathbf{C}\hat{x}_{M_i}(k+1|k).$$

Per determinare il modo stimato si è utilizzato unicamente l'**algoritmo 2**, poiché non risulta immediato trovare delle soglie adeguate all'applicazione dell'**algoritmo 1**.

6.1.2 Osservatore continuo (CO)

Il CO è un sistema a commutazione con tanti modi quanti sono i modi del sistema reale. Anch'esso è progettato come un EKF:

$$\hat{x}(k+1|k) = f_i(\hat{x}(k|k), u(k)) \mathbf{P}(k+1|k) = \mathbf{A}_i(k)\mathbf{P}(k|k)\mathbf{A}_i^T(k) + \mathbf{Q}(k) \hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \mathbf{K}(k+1)(y(k+1) - \mathbf{C}\hat{x}(k+1|k)) \mathbf{P}(k+1|k+1) = (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{P}(k+1|k)$$

$$(6.2)$$

per ogni modo M_i . Dove f_i è una tra le eq. (5.1), eq. (5.2), eq. (5.3) e eq. (5.4), e

$$\mathbf{A}_{i}(k) = \left. \frac{\partial f_{i}(x(k), u(k))}{\partial x(k)} \right|_{x(k) = \hat{x}(k)}$$

L'EKF al passo k calcola la stima per il passo k+1 utilizzando la f_i e la \mathbf{A}_i corrispondenti al modo stimato M_i .

Lo stato continuo e il modo stimato vengono utilizzati dal blocco di controllo LQR per determinare il segnale di controllo.

6.2 Simulazioni sul modello

Per verificare la validità dell'osservatore abbiamo utilizzato il modello del sistema a commutazione three-tank presentato nella Sezione 3.2.5, con alcune restrizioni. Le transizioni tra i modi sono sempre da o verso il modo M_0 . In altri termini, utilizzando la notazione introdotta in Tabella 3.1, l'insieme degli eventi abilitati è dato da $\mathbf{E}' = \{a, b, h, g, l, i\} \subset \mathbf{E}$, vedi, in Figura 6.1, l'automa che descrive il sistema a eventi discreti in cui sono indicati gli eventi possibili. Inoltre si è supposto di conoscere lo stato continuo iniziale.



Figura 6.1: Sistema three-tank - diagramma a stati

Controllo costante

In Figura 6.2 mostriamo i risultati ottenuti in una simulazione in cui non abbiamo applicato alcuna strategia di controllo, portata della pompa costante $q_0 = 2.1 \cdot 10^{-5} [m^3/s]$.

Come già detto lo stato continuo iniziale è supposto noto, in questa simulazione è pari a:

$$\vec{H}_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$



Figura 6.2: Modello three-tank senza controllo - modi osservati

Notiamo che il ritardo di osservazione è mediamente di un secondo, con l'eccezione della transizione $M_2 \rightarrow M_0$ in cui il ritardo è di 10 secondi.

Controllo limitato

Abbiamo mostrato il caso di sistema con controllo costante poiché, come ora illustreremo, un ingresso variabile produce dei peggioramenti nell'osservazione del sistema. Ripetiamo dunque la simulazione precedente ma in questo caso con un controllo variabile, anche se con un range di segnale di controllo limitato q $[m^3/s]$. Precisamente utilizzeremo lo stesso blocco di controllo presentato nella Sezione 4.6 ma con il segnale di controllo saturato inferiormente e superiormente per tutti i modi di funzionamento del sistema:

$$Q = \left\{ q : 1.9 \cdot 10^{-5} \le q \le 4 \cdot 10^{-5} \right\}.$$

I valori assunti dall'ingresso di controllo durante la simulazione vengono mostrati in Figura 6.3.



Figura 6.3: Segnale di controllo

Come in precedenza lo stato continuo iniziale è supposto noto:

$$\vec{H}_0 = \begin{bmatrix} 0.5\\0.5\\0.5\end{bmatrix}$$

In Figura 6.4, Figura 6.5 e Figura 6.6 mostriamo il risultato dell'osservazione di modo combinato con lo stato continuo stimato, rispettivamente, nell'Upper tank, nel Middle tank e nel Lower tank.



Figura 6.4: Osservazione del modo continuo e discreto - Upper tank



Figura 6.5: Osservazione del modo continuo e discreto - Middle tank



Figura 6.6: Osservazione del modo continuo e discreto - Lower tank

Notiamo che, rispetto al caso di controllo costante, all'istante 200 della simulazione si è aggiunto un errore nell'osservazione di modo, se aumentassimo l'intervallo Q del segnale di controllo gli errori di osservazione aumenterebbero esponenzialmente.

Per ultimo, notiamo che l'osservazione dello stato continuo ha una buona corrispondenza con l'andamento reale dello stato quando il modo stimato coincide col modo reale di funzionamento del sistema. Questo non accade in presenza di errori o ritardi nell'osservazione di modo, dove, come è normale aspettarsi, l'errore di stima è molto grande.

Abbiamo mostrato solo i risultati sul modello poiché le prove sul sistema three-tank reale non hanno prodotto risultati soddisfacenti. Questo è legato al fatto che gli errori di modello e di identificazione diventano non trascurabili nel momento in cui lo stato continuo non è direttamente misurabile.

Capitolo 7

Conclusioni

Al temine di questo lavoro siamo in grado di valutare i risultati ottenuti e se gli obiettivi prefissati siano stati raggiunti.

Il primo obiettivo è stato quello dell'identificazione del sistema fisico a tre serbatoi. Le problematiche di questa fase del nostro lavoro sono state legate alla forte non linearità della geometria del sistema in esame ed alla sua grande sensibilità ai fattori esterni, in particolar modo alla temperatura del liquido e ambiente. Abbiamo ricavato sperimentalmente le curve caratteristiche delle componenti principali del sistema (pompa, valvole e sensori), e calcolato i parametri del modello utilizzando due diverse procedure che si servono di due diversi algoritmi di ottimizzazione (Nelder-Mead e a Minimi Quadrati), ottenendo dei valori molto vicini. I risultati ottenuti sono più che soddisfacenti, come abbiamo mostrato nel confronto tra sistema reale e modello in cui le differenze tra dinamica reale e dinamica simulata al calcolatore sono molto sottili.

L'osservatore dello stato discreto proposto si è dimostrato sufficientemente veloce nel rilevare le transizioni di modo, con ritardi dell'ordine dei decimi di secondo nel caso del modello e di circa tre secondi nel caso del sistema reale. L'entità del ritardo di osservazione nel caso reale non risulta inaspettata se si tiene conto degli errori di modello e dei disturbi legati alle turbolenze del liquido che spiegano la differenza coi risultati sul modello.

Come ultimo abbiamo progettato un osservatore del modo continuo e discreto di cui non abbiamo svolto uno studio sulla stabilità, e sull'osservabilità del sistema. Uno studio di questo tipo su sistemi a commutazione autonomi sarebbe stato di elevata complessità anche perché in letteratura non sono presenti delle procedure generali con cui trattare questo tipo di analisi. Dell'osservatore proposto vengono comunque mostrate delle simulazioni in cui i risultati ottenuti sono sembrati interessanti.

Appendici

Appendice A

Listati Matlab

A.1 Identificazione dei parametri del modello

A.1.1 Procedura 1

Upper Tank

```
%Funzione costo per l'identificazione
%dei parametri C1 e alpha1 dell'upper-tank
%
%Lev - dati raccolti nella prova di svuotamento
%
%X - vettore parametri da identificare
%X(1)=C1, X(2)=alpha1
function J = J(X, Lev)
deltaT = 0.1;%passo di campionameto dati raccolti
a = 0.25;
w = 0.035;
J=0;
N = length(Lev);
for i = 2:N;
    Hm = Lev(i);%Livello misurato
    %Livello predetto
    Ht =Lev(i-1)-(1/(a*w))*X(1)*(Lev(i-1)^X(2))*deltaT;
    J = J + (Ht-Hm)^2;%funzione costo
end
```

Middle Tank

```
%Funzione costo per l'identificazione
%dei parametri C2 e alpha2 del middle tank
%
%Lev - dati raccolti nella prova di svuotamento
%
%X - vettore parametri da identificare
%X(1)=C2, X(2)=alpha2
function Jsec = Jsec(X, Lev)
deltaT = 0.1;%passo di campionameto dati raccolti
Hmax = 0.35;
b = 0.348;
c = 0.1;
w = 0.035;
Jsec = 0;
N = length(Lev);
for i = 2:N;
    Hm = Lev(i);%Livello misurato
    %Livello predetto
    Ht = Lev(i-1)-X(1)*(Lev(i-1)^X(2))*deltaT/(w*(c+b*Lev(i-1)/Hmax));
    Jsec = Jsec + (Ht-Hm)^2;%funzione costo
end
```
Lower Tank

```
%Funzione costo per l'identificazione
%dei parametri C3 e alpha3 dell'upper tank
%
%Lev - dati raccolti nella prova di svuotamento
%
%X - vettore parametri da identificare
%X(1)=C2, X(2)=alpha2
function Jthd = Jthd(X, Lev)
deltaT = 0.1;%passo di campionameto dati raccolti
R = 0.365;
Hmax = 0.35;
w = 0.035;
Jthd=0;
N = length(Lev);
for i = 2:N;
    Hm = Lev(i);%Livello misurato
    %Livello predetto
    Ht = Lev(i-1)+
    -(1/(w*sqrt(R^2-(Hmax-Lev(i-1))^2)))*X(1)*(Lev(i-1)^X(2))*deltaT;
    Jthd = Jthd + (Ht-Hm)^2;%funzione costo
end
```

A.1.2 Procedura 2

Riportiamo il codice C utilizzato per creare il file .MEX che Matlab utilizza per specificare la struttura del modello grey box non-lineare.

Abbiamo utilizzato un template fornito da Matlab in cui abbiamo inserito le equazioni che descrivono la dinamica del sistema da identificare. Di seguito riportiamo le istruzioni per utilizzare il template:

```
/* Template file for IDNLGREY model specification.
   Use this file to create a MEX function that specifies the model
   structure and equations. The MEX file syntax is
      [dx, y] = mymodel(t, x, u, p1, p2, ..., pn, auxvar)
   where
      * t is the time (scalar).
      * x is the state vector at time t (column vector).
      * u is the vector of inputs at time t (column vector).
      * p1, p2,... pn: values of the estimated parameters specified
        in the IDNLGREY model.
      * auxvar: a cell array containing auxiliary data in any format
        (optional).
      * dx is the vector of state derivatives at time t (column vector).
      * y is the vector of outputs at time t.
   To create the MEX file "mymodel", do the following:
      1) Save this template as "mymodel.c" (replace "mymodel" by the
         name of your choice).
      2) Define the number NY of outputs below.
      3) Specify the state derivative equations in COMPUTE_DX below.
      4) Specify the output equations in COMPUTE_Y below.
      5) Build the MEX file using
            >> mex mymodel.c
```

*/

Dinamica Upper Tank

```
/* Include libraries. */
#include "mex.h"
#include "math.h"
/* Specify the number of outputs here. */
#define NY 1
/* State equations. */
void compute_dx(
    double *dx, /* Vector of state derivatives (length nx). */
                /* Time t (scalar). */
    double t.
    double *x, /* State vector (length nx). */
    double *u, /* Input vector (length nu). */
    double **p, /* p[j] points to the j-th
     estimated model parameters(a double array).*/
    const mxArray *auxvar /* Cell array of additional data. */
   )
{
    double C1, Alfa1;
    C1 = p[0][0];
    Alfa1 = p[1][0];
    dx[0] = - (C1*(pow( x[0],Alfa1) )/(0.035*0.25));
}
/* Output equations. */
void compute_y(
    double *y, /* Vector of outputs (length NY). */
    double t, /* Time t (scalar). */
    double *x, /* State vector (length nx). */
    double *u, /* Input vector (length nu). */
    double **p, /* p[j] points to the j-th
      estimated model parameters(a double array).*/
    const mxArray *auxvar /* Cell array of additional data. */
   )
{
    y[0] = x[0];
}
```

Dinamica Middle Tank

```
/* Include libraries. */
#include "mex.h"
#include "math.h"
/* Specify the number of outputs here. */
#define NY 1
/* State equations. */
void compute_dx(
    double *dx, /* Vector of state derivatives (length nx). */
    double t, /* Time t (scalar). */
    double *x, /* State vector (length nx). */
    double *u, /* Input vector (length nu). */
    double **p, /* p[j] points to the j-th
     estimated model parameters(a double array).*/
    const mxArray *auxvar /* Cell array of additional data. */
   )
{
    double C2, Alfa2;
    C2 = p[0][0];
    Alfa2 = p[1][0];
    dx[0] = -(C2*(pow(x[0],A]fa2))/(0.035*(0.1+x[0]*0.348/0.35)));
}
void compute_y(
    double *y, /* Vector of outputs (length NY). */
    double t, /* Time t (scalar). */
    double *x, /* State vector (length nx). */
    double *u, /* Input vector (length nu). */
    double **p, /* p[j] points to the j-th
     estimated model parameters(a double array).*/
    const mxArray *auxvar /* Cell array of additional data. */
   )
{
    y[0] = x[0];
}
```

Dinamica Lower Tank

```
/* Include libraries. */
#include "mex.h"
#include "math.h"
/* Specify the number of outputs here. */
#define NY 1
/* State equations. */
void compute_dx(
    double *dx, /* Vector of state derivatives (length nx). */
    double t, /* Time t (scalar). */
    double *x, /* State vector (length nx). */
    double *u, /* Input vector (length nu). */
    double **p, /* p[j] points to the j-th
      estimated model parameters(a double array).*/
    const mxArray *auxvar /* Cell array of additional data. */
   )
{
    double C3, Alfa3;
    C3 = p[0][0];
    Alfa3 = p[1][0];
    dx[0] = -(C3*(pow(x[0], Alfa3))/(0.035*sqrt(pow(0.365, 2)...
      - pow((0.35-x[0]),2 ))));
}
/* Output equations. */
void compute_v(
    double *y, /* Vector of outputs (length NY). */
    double t, /* Time t (scalar). */
    double *x, /* State vector (length nx). */
    double *u, /* Input vector (length nu). */
    double **p, /* p[j] points to the j-th
     estimated model parameters(a double array).*/
    const mxArray *auxvar /* Cell array of additional data. */
   )
{
    y[0] = x[0];
}
```

A.2 Isolatori di modo - funzioni embedded

A.2.1 Caso in cui lo stato continuo è noto

Isolatore di modo per il modo M₀

```
%Isolatore di modo per il modo M_0
%
%xmis - livelli misurati
%u - portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=isolMO(xmis,u,C1,C2,C3,a1,a2,a3)
%parametri geometria sistema
a = 0.25;
b = 0.348;
c = 0.1;
w = 0.035;
h2max = 0.35;
h3max = 0.35;
Rag = 0.365;
%passo di campionamento
deltaT = 0.1;
% Initialization
persistent xstim
    if(isempty(xstim))
    xstim = xmis;
    end
%residuo
residual = xmis-xstim;
%
%saturazine per livello inferiore misurato
%utilizzato nel sistema reale
if( xmis (1)<=0)
    xmis(1) = 0.00001;
end
if( xmis (2)<=0)
    xmis(2) = 0.00001;
end
if( xmis (3)<=0)
    xmis(3) = 0.00001;
end
```

```
%calcolo della stima
xstim=xmis+...
[(u-C1*xmis(1)^a1)/(a*w);
  (C1*xmis(1)^a1-C2*xmis(2)^a2)/(c*w+b*w*xmis(2)/h2max);
  (C2*xmis(2)^a2-C3*xmis(3)^a3)/(w*(Rag^2-(h3max-xmis(3))^2)^0.5)]*deltaT;
%saturazine per livello inferiore stimato
if( xstim (1)<=0)
    xstim(1) = 0.00001;
end
if( xstim (2)<=0)
    xstim(2) = 0.00001;
end
if( xstim (3)<=0)
    xstim(3) = 0.00001;
end
%uscite dell'MI_0
residualout = residual;
xstimout = xstim;
```

Isolatore di modo per il modo M₁

```
%Isolatore di modo per il modo M_1
%
%xmis - livelli misurati
%u - portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=isolM1(xmis,u,C1,C2,C3,a1,a2,a3)
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%parametri dell'upper tank con valvola manuale aperta
Cm1 = 0.00015053;
alfam1 = 0.262891;
%passo di campionamento
deltaT = 0.1;
% Initialization
persistent xstim
    if(isempty(xstim))
    xstim = xmis;
    end
%residuo
residual = xmis-xstim;
%
%saturazine per livello inferiore misurato
%utilizzato nel sistema reale
if( xmis (1)<=0)
    xmis(1) = 0.00001;
end
if( xmis (2)<=0)
    xmis(2) = 0.00001;
end
if( xmis (3)<=0)
    xmis(3) = 0.00001;
end
```

```
%calcolo della stima
xstim=xmis+...
[(u-C1*xmis(1)^a1-Cm1*xmis(1)^alfam1)/(a*w);
  (C1*xmis(1)^a1+Cm1*xmis(1)^alfam1-C2*xmis(2)^a2)/(c*w+b*w*xmis(2)/h2max);
  (C2*xmis(2)^a2-C3*xmis(3)^a3)/(w*(Rag^2-(h3max-xmis(3))^2)^0.5)]*deltaT;
%saturazine per livello inferiore stimato
if( xstim (1)<=0)
    xstim(1) = 0.00001;
end
if( xstim (2)<=0)
    xstim(2) = 0.00001;
end
if( xstim (3)<=0)
    xstim(3) = 0.00001;</pre>
```

end

%uscite dell'MI_1
residualout = residual;
xstimout = xstim;

Isolatore di modo per il modo M₂

```
%Isolatore di modo per il modo M_2
%
%xmis - livelli misurati
%u - portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=isolM2(xmis,u,C1,C2,C3,a1,a2,a3)
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%parametri del middle tank con valvola manuale aperta
Cm2 = 0.000180352;
alfam2 = 0.311945;
%passo di campionamento
deltaT = 0.1;
% Initialization
persistent xstim
    if(isempty(xstim))
    xstim = xmis;
    end
%residuo
residual = xmis-xstim;
%
%saturazine per livello inferiore misurato
%utilizzato nel sistema reale
if( xmis (1)<=0)
    xmis(1) = 0.00001;
end
if( xmis (2)<=0)
    xmis(2) = 0.00001;
end
if( xmis (3)<=0)
    xmis(3) = 0.00001;
end
```

100

```
%calcolo della stima
xstim=xmis+
[(u-C1*xmis(1)^a1)/(a*w);
(C1*xmis(1)^a1-C2*xmis(2)^a2-Cm2*(xmis(2)^alfam2))/(c*w+b*w*xmis(2)/h2max);
  (C2*xmis(2)^a2+Cm2*(xmis(2)^alfam2...
     -C3*xmis(3)^a3)/(w*(Rag^2-(h3max-xmis(3))^2)^0.5)]...
*deltaT;
```

```
%saturazine per livello inferiore stimato
if( xstim (1)<=0)
        xstim(1) = 0.00001;
end
if( xstim (2)<=0)
        xstim(2) = 0.00001;
end
if( xstim (3)<=0)
        xstim(3) = 0.00001;
end
%uscite dell'MI_2
```

residualout = residual; xstimout = xstim;

Isolatore di modo per il modo M₃

```
%Isolatore di modo per il modo M_3
%
%xmis - livelli misurati
%u - portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=isolM3(xmis,u,C1,C2,C3,a1,a2,a3)
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%parametri del lower tank con valvola manuale aperta
Cm3 = 0.000170454;
alfam3 = 0.314445;
%passo di campionamento
deltaT = 0.1;
% Initialization
persistent xstim
    if(isempty(xstim))
    xstim = xmis;
    end
%residuo
residual = xmis-xstim;
%
%saturazine per livello inferiore misurato
%utilizzato nel sistema reale
if( xmis (1)<=0)
    xmis(1) = 0.00001;
end
if( xmis (2)<=0)
    xmis(2) = 0.00001;
end
if( xmis (3)<=0)
    xmis(3) = 0.00001;
end
```

102

```
%calcolo della stima
xstim=xmis+
[(u-C1*xmis(1)^a1)/(a*w);
(C1*xmis(1)^a1-C2*xmis(2)^a2)/(c*w+b*w*xmis(2)/h2max);
    (C2*xmis(2)^a2-C3*xmis(3)^a3...
    -Cm3*( xmis(3)^alfam3))/(w*(Rag^2-(h3max-xmis(3))^2)^0.5)]...
*deltaT;
```

```
%saturazine per livello inferiore stimato
if( xstim (1)<=0)
    xstim(1) = 0.00001;
end
if( xstim (2)<=0)
    xstim(2) = 0.00001;
end
if( xstim (3)<=0)
    xstim(3) = 0.00001;
end
%uscite dell'MI_3</pre>
```

```
residualout = residual;
xstimout = xstim;
```

A.2.2 Caso in cui anche lo stato continuo deve essere osservato

Isolatore di modo per il modo M₀

```
%Isolatore di modo per il modo M_0
%
%ymis - uscita del sistema
%u – portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=ekfM0(ymis,u,C1,C2,C3,a1,a2,a3)
%Parametri scelti arbitrariamente, senza significato fisico
q=0.005;
          %std of process
r=0.005;
            %std of measurement
Q=q^2*eye(3); % covariance of process
R=r^2*eye(2); % covariance of measurement
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%passo di campionamento
deltaT =0.1;
%Matrice C della trasformazione d'uscita
C=[0 1 1;2 4 3];
% Initialization
persistent P;
persistent xstim
if isempty(P)
    P = 0.001 * [1 -1 1; -1 1 -1; 1 -1 1];
    xstim = [0.05; 0.05; 0.05];
end
%predizione non-lineare dello stato
x1=xstim+[(u-C1*xstim(1)^a1)/(a*w);
    (C1*xstim(1)^a1-C2*xstim(2)^a2)/(c*w+b*w*xstim(2)/h2max);
```

```
(C2*xstim(2)^a2-C3*xstim(3)^a3)/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)]...
    *deltaT:
%
%linearizazione allo stato corrente
A=[1-C1*a1*deltaT/((xstim(1)^(1-a1))*(a*w)),0,0;
   C1*a1*deltaT/((xstim(1)^(1-a1))*(c*w+b*w*xstim(2)/h2max)),...
     1-C2*a2*deltaT/(xstim(2)^(1-a2)*(c*w+b*w*xstim(2)/h2max)),0;
   0,C2*a2*deltaT/(xstim(2)^(1-a2)*(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),...
     1-C3*a3*deltaT/(xstim(3)^(1-a3)*(w*(Rag^2-(h3max-xstim(3))^2)^0.5))];
%saturazine per livello minimo stima parziale
if( x1 (1)<=0)
    x1(1) = 0.0000001;
end
if( x1 (2)<=0)
    x1(2) = 0.0000001;
end
if( x1 (3)<=0)
    x1(3) = 0.0000001;
end
%saturazine per livello massimo stima parziale
if( x1 (1)>=0.25)
    x1(1) = 0.25;
end
if( x1 (2)>=0.25)
    x1(2) = 0.25;
end
if( x1 (3)>=0.25)
    x1(3) = 0.25;
end
P=A*P*A'+Q;
               %predizione covarianza dello stato
V=C*P*C'+R;
K=P*C'*inv(V); %guadagno Kalman filter
%residuo
residual = ymis-C*x1;
%aggiornameto stima dello stato
xstim=x1+K*(residual);
%saturazine per livello minimo stima
if( xstim (1)<=0)
    xstim(1) = 0.0000001;
end
if( xstim (2)<=0)
    xstim(2) = 0.0000001;
```

APPENDICE A. LISTATI MATLAB

```
end
if( xstim (3)<=0)
    xstim(3) = 0.00000001;
end
%aggiornamento covarianza dello stato
P=(eye(3)-K*C)*P;
```

```
%uscite dell'MI_0
residualout=residual;
xstimout = x1;
```

Isolatore di modo per il modo M₁

```
%Isolatore di modo per il modo M_1
%
%ymis - uscita del sistema
%u - portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=ekfM1(ymis,u,C1,C2,C3,a1,a2,a3)
%Parametri scelti arbitrariamente, senza significato fisico
q=0.005; %std of process
          %std of measurement
r=0.005;
Q=q^2*eye(3); % covariance of process
R=r^2*eye(2); % covariance of measurement
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%parametri dell'upper tank con valvola manuale aperta
Cm1 = 0.00015053;
alfam1 = 0.262891;
%passo di campionamento
deltaT =0.1;
%Matrice C della trasformazione d'uscita
C=[0 \ 1 \ 1;2 \ 4 \ 3];
% Initialization
persistent P;
persistent xstim
if isempty(P)
    P = 10^{(-4)} * [1 -1 1; -1 1 -1; 1 -1 1];
    xstim = [0.05; 0.05; 0.05];
end
%predizione non-lineare dello stato
x1=xstim+[(u-C1*xstim(1)^a1)/(a*w)-Cm1*xstim(1)^alfam1/(a*w);
```

```
(C1*xstim(1)^al+Cm1*xstim(1)^alfam1-C2*xstim(2)^a2)/(c*w+b*w*xstim(2)/h2max);
 (C2*xstim(2)^a2-C3*xstim(3)^a3)/(w*(Rag^2 -(h3max-xstim(3))^2)^0.5)]*deltaT;
%linearizazione allo stato corrente
A=[1+(-C1*a1*deltaT/(xstim(1)^(1-a1))...
-(Cm1*alfam1)*deltaT/(xstim(1)^(1-alfam1)))/(a*w),0,0;
 (C1*a1*deltaT/(xstim(1)^(1-a1))...
  +(Cm1*alfam1)*deltaT/((xstim(1)^(1-alfam1))))/(c*w+b*w*xstim(2)/h2max),...
  1+(-C2*a2*deltaT/(xstim(2)^(1-a2)))/(c*w+b*w*xstim(2)/h2max),0;
 0,((C2*a2*deltaT/(xstim(2)^(1-a2)))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),...
  1+(-C3*a3*deltaT/(xstim(3)^(1-a3))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5))];
%saturazine per livello minimo stima parziale
if( x1 (1)<=0)
    x1(1) = 0.0000001;
end
if( x1 (2)<=0)
    x1(2) = 0.0000001;
end
if( x1 (3)<=0)
    x1(3) = 0.0000001;
end
%saturazine per livello massimo stima parziale
if( x1 (1)>=0.25)
    x1(1) = 0.25;
end
if( x1 (2)>=0.25)
    x1(2) = 0.25;
end
if( x1 (3)>=0.25)
    x1(3) = 0.25;
end
               %predizione covarianza dello stato
P=A*P*A'+Q;
V=C*P*C'+R;
K=P*C'*inv(V); %guadagno Kalman filter
%residuo
residual = ymis-C*x1;
%aggiornameto stima dello stato
xstim=x1+K*(residual);
%saturazine per livello minimo stima
if( xstim (1)<=0)
    xstim(1) = 0.0000001;
end
if( xstim (2)<=0)
```

```
xstim(2) = 0.00000001;
end
if( xstim (3)<=0)
    xstim(3) = 0.00000001;
end
%aggiornamento covarianza dello stato
P=(eye(3)-K*C)*P;
%uscite dell'MI_1
```

residualout=residual;

xstimout = x1;

Isolatore di modo per il modo M₂

```
%Isolatore di modo per il modo M_2
%
%ymis - uscita del sistema
%u – portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=ekfM2(ymis,u,C1,C2,C3,a1,a2,a3)
%Parametri scelti arbitrariamente, senza significato fisico
q=0.005; %std of process
            %std of measurement
r=0.005;
Q=q^2*eye(3); % covariance of process
R=r^2*eye(2); % covariance of measurement
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%parametri del middle tank con valvola manuale aperta
Cm2 = 0.000180352;
alfam2 = 0.311945;
%passo di campionamento
deltaT =0.1;
%Matrice C della trasformazione d'uscita
C=[0 1 1;2 4 3];
% Initialization
persistent P;
persistent xstim
if isempty(P)
    P = 10^{(-4)} * [1 -1 1; -1 1 -1; 1 -1 1];
    xstim = [0.05; 0.05; 0.05];
end
```

```
%predizione non-lineare dello stato
x1=xstim+[(u-C1*xstim(1)^a1)/(a*w);
  (C1*xstim(1)^a1-C2*xstim(2)^a2...
   -Cm2*xstim(2)^alfam2)/(c*w+b*w*xstim(2)/h2max);
  (C2*xstim(2)^a2+(Cm2*xstim(2)^alfam2)...
   -C3*xstim(3)^a3)/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)]...
  *deltaT;
%linearizazione allo stato corrente
A=[1+(-C1*a1*deltaT/(xstim(1)^{(1-a1)}))/(a*w),0,0;
  (C1*a1*deltaT/(xstim(1)^(1-a1)))/(c*w+b*w*xstim(2)/h2max),...
  1+(-C2*a2*deltaT/(xstim(2)^(1-a2))...
   -(Cm2*alfam2*deltaT/(xstim(2)^(1-alfam2))))/(c*w+b*w*xstim(2)/h2max),0;
  0,((C2*a2*deltaT/(xstim(2)^(1-a2))...
   +(Cm2*alfam2*deltaT/(xstim(2)^(1-alfam2))))...
   /(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),...
  1+(-C3*a3*deltaT/(xstim(3)^(1-a3))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5))];
%saturazine per livello minimo stima parziale
if( x1 (1)<=0)
    x1(1) = 0.0000001;
end
if( x1 (2)<=0)
    x1(2) = 0.0000001;
end
if( x1 (3)<=0)
    x1(3) = 0.0000001;
end
%saturazine per livello massimo stima parziale
if( x1 (1)>=0.25)
    x1(1) = 0.25;
end
if( x1 (2)>=0.25)
    x1(2) = 0.25;
end
if( x1 (3)>=0.25)
    x1(3) = 0.25;
end
               %predizione covarianza dello stato
P=A*P*A'+Q;
V=C*P*C'+R;
K=P*C'*inv(V); %guadagno Kalman filter
%residuo
residual = ymis-C*x1;
%aggiornameto stima dello stato
xstim=x1+K*(residual);
```

```
%saturazine per livello minimo stima
if( xstim (1)<=0)
    xstim(1) = 0.00000001;
end
if( xstim (2)<=0)
    xstim(2) = 0.00000001;
end
if( xstim (3)<=0)
    xstim(3) = 0.00000001;
end
%aggiornamento covarianza dello stato
P=(eye(3)-K*C)*P;
%uscite dell'MI_2
residualout=residual;
xstimout = x1;
```

Isolatore di modo per il modo M₃

```
%Isolatore di modo per il modo M_3
%
%ymis - uscita del sistema
%u - portata pompa
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function [xstimout,residualout]=ekfM3(ymis,u,C1,C2,C3,a1,a2,a3)
%Parametri scelti arbitrariamente, senza significato fisico
q=0.005; %std of process
          %std of measurement
r=0.005;
Q=q^2*eye(3); % covariance of process
R=r^2*eye(2); % covariance of measurement
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%parametri del lower tank con valvola manuale aperta
Cm3 = 0.000170454;
alfam3 = 0.314445;
%passo di campionamento
deltaT =0.1;
%Matrice C della trasformazione d'uscita
C=[0 \ 1 \ 1;2 \ 4 \ 3];
% Initialization
persistent P;
persistent xstim
if isempty(P)
    P = 10^{(-4)} * [1 -1 1; -1 1 -1; 1 -1 1];
    xstim = [0.05; 0.05; 0.05];
```

end

```
%predizione non-lineare dello stato
x1=xstim+[(u-C1*xstim(1)^a1)/(a*w);
  (C1*xstim(1)^a1-C2*xstim(2)^a2)/(c*w+b*w*xstim(2)/h2max);
  (C2*xstim(2)^a2-C3*xstim(3)^a3...
   -(Cm3*xstim(3)^alfam3))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)]...
  *deltaT;
%linearizazione allo stato corrente
A=[1+(-C1*a1*deltaT/(xstim(1)^(1-a1)))/(a*w),0,0;
  (C1*a1*deltaT/(xstim(1)^(1-a1)))/(c*w+b*w*xstim(2)/h2max),
  1+(-C2*a2*deltaT/(xstim(2)^(1-a2)))/(c*w+b*w*xstim(2)/h2max),0;
  0,((C2*a2*deltaT/(xstim(2)^(1-a2)))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),...
  1+((-C3*a3*deltaT/(xstim(3)^(1-a3))...
    -(Cm3*alfam3*deltaT/(xstim(3)^(1-alfam3))))...
    /(w*(Rag^2-(h3max-xstim(3))^2)^0.5))];
%saturazine per livello minimo stima parziale
if( x1 (1)<=0)
    x1(1) = 0.0000001;
end
if( x1 (2)<=0)
    x1(2) = 0.0000001;
end
if( x1 (3)<=0)
    x1(3) = 0.0000001;
end
%saturazine per livello massimo stima parziale
if( x1 (1)>=0.25)
    x1(1) = 0.25;
end
if( x1 (2)>=0.25)
    x1(2) = 0.25;
end
if( x1 (3)>=0.25)
    x1(3) = 0.25;
end
P=A*P*A'+Q;
               %predizione covarianza dello stato
V=C*P*C'+R;
K=P*C'*inv(V); %guadagno Kalman filter
%residuo
residual = ymis-C*x1;
%aggiornameto stima dello stato
xstim=x1+K*(residual);
%saturazine per livello minimo stima
```

```
if( xstim (1)<=0)
    xstim(1) = 0.00000001;
end
if( xstim (2)<=0)
    xstim(2) = 0.00000001;
end
if( xstim (3)<=0)
    xstim(3) = 0.00000001;
end
%aggiornamento covarianza dello stato
P=(eye(3)-K*C)*P;
%uscite dell'MI_3
residualout=residual;</pre>
```

```
xstimout = x1;
```

A.3 Osservatore Continuo - funzione embedded

```
%Osservatore Continuo
%
%ymis - uscita del sistema
%u - portata pompa
%mode - modo stimato
%C1,C2,C3,a1,a2,a3 parametri del modello dei serbatoi
%al variare delle valvole controllate
function xstimout
                     = ekfCO(ymis,u,mode,C1,C2,C3,a1,a2,a3)
%Parametri scelti arbitrariamente, senza significato fisico
          %std of process
q=0.005;
            %std of measurement
r=0.005;
Q=q^2*eye(3); % covariance of process
R=r^2*eye(2); % covariance of measurement
%parametri geometria sistema
a=0.25;
b=0.348;
c=0.1;
w=0.035;
h2max=0.35;
h3max=0.35;
Rag=0.365;
%parametri dell'upper tank con valvola manuale aperta
Cm1 = 0.00015053;
alfam1 = 0.262891;
%parametri del middle tank con valvola manuale aperta
Cm2 = 0.000180352;
alfam2 = 0.311945;
%parametri del lower tank con valvola manuale aperta
Cm3 = 0.000170454;
alfam3 = 0.314445;
%passo di campionamento
deltaT =0.1;
%Matrice C della trasformazione d'uscita
C=[0 1 1;2 4 3];
% Initialization
persistent P;
persistent xstim
if isempty(P)
```

```
P = 0.0001 * [1 -1 1; -1 1 -1; 1 -1 1];
    xstim = [0.05, 0.05, 0.05]';
end
%Modo M_O valvole manuali tutte chiuse
%predizione non-lineare dello stato
x1=xstim+[(u-C1*xstim(1)^a1)/(a*w);
    (C1*xstim(1)^a1-C2*xstim(2)^a2)/(c*w+b*w*xstim(2)/h2max);
    (C2*xstim(2)^a2-C3*xstim(3)^a3)/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)]...
    *deltaT;
%
%linearizazione allo stato corrente
A=[1-C1*a1*deltaT/((xstim(1)^(1-a1))*(a*w)),0,0;
   C1*a1*deltaT/((xstim(1)^(1-a1))*(c*w+b*w*xstim(2)/h2max)),...
   1-C2*a2*deltaT/(xstim(2)^(1-a2)*(c*w+b*w*xstim(2)/h2max)),0;
   0,C2*a2*deltaT/(xstim(2)^(1-a2)*(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),...
   1-C3*a3*deltaT/(xstim(3)^(1-a3)*(w*(Rag^2-(h3max-xstim(3))^2)^0.5))];
%Modo M_1 valvola manuale dell'Upper tank aperta
if(mode==1)
%predizione non-lineare dello stato
x1=xstim+[(u-C1*xstim(1)^a1)/(a*w);
  (C1*xstim(1)^a1-C2*xstim(2)^a2-Cm2*xstim(2)^alfam2)/(c*w+b*w*xstim(2)/h2max)...
  (C2*xstim(2)^a2+(Cm2*xstim(2)^alfam2)...
   -C3*xstim(3)^a3)/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)]...
  *deltaT:
%linearizazione allo stato corrente
A=[1+(-C1*a1*deltaT/(xstim(1)^(1-a1)))/(a*w),0,0;
  (C1*a1*deltaT/(xstim(1)^(1-a1)))/(c*w+b*w*xstim(2)/h2max),...
  1+(-C2*a2*deltaT/(xstim(2)^(1-a2))...
   -(Cm2*alfam2*deltaT/(xstim(2)^(1-alfam2))))/(c*w+b*w*xstim(2)/h2max),0;
  0,((C2*a2*deltaT/(xstim(2)^(1-a2))...
   +(Cm2*alfam2*deltaT/(xstim(2)^(1-alfam2))))...
   /(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),...
  1+(-C3*a3*deltaT/(xstim(3)^(1-a3))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5))];
else
```

```
%Modo M_2 valvola manuale del Middle tank aperta
if(mode==2)
```

%predizione non-lineare dello stato x1=xstim+[(u-C1*xstim(1)^a1)/(a*w); (C1*xstim(1)^a1-C2*xstim(2)^a2-Cm2*xstim(2)^alfam2)/(c*w+b*w*xstim(2)/h2max)... (C2*xstim(2)^a2+(Cm2*xstim(2)^alfam2)... -C3*xstim(3)^a3)/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)]... *deltaT; %linearizazione allo stato corrente $A=[1+(-C1*a1*deltaT/(xstim(1)^{(1-a1)}))/(a*w),0,0;$ (C1*a1*deltaT/(xstim(1)^(1-a1)))/(c*w+b*w*xstim(2)/h2max),... 1+(-C2*a2*deltaT/(xstim(2)^(1-a2))... -(Cm2*alfam2*deltaT/(xstim(2)^(1-alfam2))))/(c*w+b*w*xstim(2)/h2max),0; 0,((C2*a2*deltaT/(xstim(2)^(1-a2))... +(Cm2*alfam2*deltaT/(xstim(2)^(1-alfam2))))... /(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),... 1+(-C3*a3*deltaT/(xstim(3)^(1-a3))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5))]; else %Modo M_3 valvola manuale del Lower tank aperta if(mode==3) %predizione non-lineare dello stato x1=xstim+[(u-C1*xstim(1)^a1)/(a*w); (C1*xstim(1)^a1-C2*xstim(2)^a2)/(c*w+b*w*xstim(2)/h2max); (C2*xstim(2)^a2-C3*xstim(3)^a3... -(Cm3*xstim(3)^alfam3))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)]... *deltaT; %linearizazione allo stato corrente A=[1+(-C1*a1*deltaT/(xstim(1)^(1-a1)))/(a*w),0,0; (C1*a1*deltaT/(xstim(1)^(1-a1)))/(c*w+b*w*xstim(2)/h2max),... 1+(-C2*a2*deltaT/(xstim(2)^(1-a2)))/(c*w+b*w*xstim(2)/h2max),0; 0,((C2*a2*deltaT/(xstim(2)^(1-a2)))/(w*(Rag^2-(h3max-xstim(3))^2)^0.5)),... 1+((-C3*a3*deltaT/(xstim(3)^(1-a3))... -(Cm3*alfam3*deltaT/(xstim(3)^(1-alfam3))))... /(w*(Rag^2-(h3max-xstim(3))^2)^0.5))]; end end end %saturazine per livello minimo stima parziale if(x1 (1)<=0) x1(1) = 0.0000001;end if(x1 (2)<=0) x1(2) = 0.0000001;end

```
if( x1 (3)<=0)
    x1(3) = 0.0000001;
end
P=A*P*A'+Q;
               %predizione covarianza dello stato
V=C*P*C'+R;
K=P*C'*inv(V); %guadagno Kalman filter
%residuo
residual = ymis-C*x1;
%aggiornameto stima dello stato
xstim=x1+K*(residual);
%saturazine per livello minimo stima
if( xstim (1)<=0)
    xstim(1) = 0.0000001;
end
if( xstim (2)<=0)
    xstim(2) = 0.0000001;
end
if( xstim (3)<=0)
    xstim(3) = 0.0000001;
end
%aggiornamento covarianza dello stato
P=(eye(3)-K*C)*P;
%uscita stimata
xstimout=xstim;
```

A.4 Osservatore di Modo

A.4.1 Algoritmo 1 - funzione embedded

```
%Algoritmo 1
%r - vettore residui
%rdot - vettore derivate residui
function mode = modesel1(r,rdot)
%definizione delle soglie
s=[-2.8*10^{(-7)}; -47.5*10^{(-7)}; -41.3*10^{(-7)}; -12.4*10^{(-7)}];
%funzione argmin
%restituisce l'indice della derivata dei residui minore
[ignore,idot_min]=min(rdot);
%
%restituisce l'indice del residuo minore
[ignore,i_min]=min(r);
%se la derivata minore è inferiore alla soglia associata
%allora restituisce l'indice della derivata minore
%altrimenti restituisce l'indice del residuo minore
if(rdot(idot_min)-s(idot_min)<=0)</pre>
    d=idot_min;
else
    d=i_min;
end
%Modo Stimato
mode = d-1;
```

A.4.2 Algoritmo 2 - funzione embedded

```
%Algoritmo 2
%r - vettore residui
%rdot - vettore derivate residui
function mode = modesel2(r,rdot)
if(r1(1)>0)
    beta0=2.5;
else
    beta0=1;
end
if(r1(4)>0)
    beta3=2;
else
    beta3=1;
end
%Vettore combinazione lineare residui e derivate
R1 = 3 r(1) + beta0 rdot(1);
R2 = r(2) + rdot(2);
R3 = r(3)+rdot(3);
R4 = 3*r(4)+beta3*rdot(4);
R = [R1 \ R2 \ R3 \ R4];
%funzione argmin
%restituisce l'indice dell'argomento R(i) minore
[ignore,d]=min(R);
%Modo Stimato
mode=d-1;
```

Appendice B

Acronimi

- IU Modello Ingresso Uscita
- VS Modello in Variabili di Stato
- SED Sistema ad Eventi Discreti
- SS Sistema a Commutazione
- EKF Filtro di Kalman Esteso
- LQR Regolatore Lineare Quadratico
- PWM Modulazione di Larghezza d'Impulso
- GUI Graphical User Interface
- MO Osservatore di Modo
- MI Isolatore di Modo
- CO Osservatore dello Stato Continuo

Bibliografia

- [1] INTECO Ltd., Multitank System User's Manual, 2007.
- [2] Lagarias J.C., J. A. Reeds, M. H. Wright and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions", *SIAM Journal of Optimization*, vol. 9, 1998.
- [3] Coleman T.F. and Y.Li, "An interior, trust region approach for nonlinear minimization subject to bounds", *SIAM Journal of Optimization*, vol. 6, 1996.
- [4] INTECO Ltd., http://www.inteco.com.pl/
- [5] Street R. L.,G. Z. Watters ,J. K. Vennard , *Elementary Fluid Mechanics*, John Wiley&Sons Inc., 1996.
- [6] Witczak M., *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems*, Springer-Verlag, 2007.
- [7] MathWorks, http://www.mathworks.com.
- [8] Wikipedia, http://it.wikipedia.org.
- [9] Antonelli G., *Note di Identificazione e Filtraggio*, Università degli Studi di Cassino, http://webuser.unicas.it/antonelli/didattica/note_if.pdf, 2007.
- [10] Wang W., L. Li, D. Zhou and K. Liu, "Robust state estimation and fault diagnosis for uncertain hybrid nonlinear systems", *Science Direct Nonlinear analysis: Hybrid systems*, vol. 1, 2007.
- [11] Di Benedetto M.D., S. Di Gennaro and A. D'Innocenzo, "Discrete State Observability of Hybrid Systems", *International Journal of Robust and Nonlinear Control*, 2008.
- [12] Giua A., C. Seatzu, Analisi Dei Sistemi Dinamici, Springer-Verlag, 2005.
- [13] Savkin V.A., R.J. Evans, *Hybrid Dynamical Systems: Controller and Sensor Switching Problems*, Springer-Verlag, 2002.