

UNIVERSITÀ DEGLI STUDI DI PADOVA

Laurea in Ingegneria dell'Automazione

Corso di Controllo dei Processi

Tesina di tipo A

Simulazione di una colonna di distillazione

Fabio Maran, 547557-IAM

Prof. Giorgio Picci

ANNO ACCADEMICO 2006/2007

Indice

1	Descrizione del problema	1
1.1	Equazioni del modello	1
2	Simulazioni con massa liquida non costante	4
2.1	Configurazione LV	5
2.2	Risultati delle simulazioni	6
3	Simulazioni con massa liquida costante	9
3.1	Risultati delle simulazioni	10
4	Andamento delle temperature	12
5	Filtro di Kalman	14
5.1	Applicazione al caso studiato: modello linearizzato	15
5.2	Taratura del filtro e risultati delle simulazioni	17
	Appendici	21
A	Codice MATLAB	21
A.1	Routine colamod_tb	21
A.2	Routine cola_lv_tb	23
A.3	Routine distillatore	23
A.4	Routine linearization_lv	25
A.5	Routine Kalman_filter	26
A.6	Script kalman	26
	Riferimenti bibliografici	28

1 Descrizione del problema

La distillazione è una delle operazioni più comuni e importanti nei processi industriali di tipo chimico. Consiste nella separazione di una miscela omogenea nei singoli componenti, ottenuta facendo evaporare il composto e successivamente condensando i vapori: il principio di separazione è basato sulla differente temperatura di ebollizione delle componenti (*volatilità*). A livello industriale, tale procedimento viene eseguito all'interno di particolari apparecchiature, le *colonne di distillazione*.

Questi impianti presentano al loro interno una serie di *piatti*, un *condensatore* nella parte superiore (tipicamente considerato il primo piatto), con lo scopo di trasformare i gas in arrivo in cima in liquido, e un *ribollitore* nella parte inferiore che fa evaporare la componente liquida presa dal fondo, almeno in parte. Una parte del flusso liquido prodotto dal condensatore viene estratto come *distillato*, la restante parte viene reimmessa in circolo come *riflusso* per continuare il processo; analogamente, dal fondo viene prelevata una parte di liquido (*residuo*), mentre la restante viene vaporizzata e nuovamente fatta circolare nella colonna sotto forma di gas. Nella zona intermedia è presente inoltre un flusso di alimentazione (*feed*) che immette in un determinato piatto dall'esterno una certa quantità di prodotto, generalmente sia in stato liquido che gassoso: la parte della colonna superiore all'alimentazione è detta di *arricchimento*, quella inferiore di *esaurimento*.

Scopo dell'esperienza è lo studio di funzioni MATLAB per la modellizzazione di una colonna di distillazione, che descrivano nel migliore dei modi la dinamica non lineare del processo, osservando come cambia la risposta considerando o meno certe dinamiche, in particolare cosa accade sia considerando la dinamica del liquido nei vari piatti che la variazione della massa totale su ognuno di essi. Si è in particolare interessati a osservare il modo con cui il sistema va a regime.

1.1 Equazioni del modello

In figura 1 è riportato lo schema di una colonna di distillazione; in tabella 1 è presente la notazione usata (da notare in particolare che il riflusso viene indicato con L piuttosto che con R , per coerenza nei confronti dei file utilizzati).

$F(t)$	flusso di alimentazione [kmol/min]
z_F	composizione dell'alimentazione [frazione molare]
q_F	frazione liquida nell'alimentazione
$D(t)$	flusso di distillato [kmol/min]
$B(t)$	flusso di residuo [kmol/min]
$x_D(t)$	concentrazione della componente più volatile nel distillato (fase liquida) [frazione molare]
$x_B(t)$	concentrazione della componente più volatile nel residuo (fase liquida) [frazione molare]
$L = L_T = L_1$	riflusso liquido [kmol/min]
$V = V_B = V_N$	riflusso gassoso [kmol/min]
N	numero totale di piatti (inclusi condensatore e ribollitore)
i	numero del livello (1 = condensatore, N = ribollitore, N_F = alimentazione)
$L_i(t)$ e $V_i(t)$	flusso liquido e gassoso dal livello i [kmol/min]
$x_i(t)$ e $y_i(t)$	concentrazione liquida e gassosa della componente leggera al livello i [frazione molare]
$M_i(t)$	quantità di materia al livello i [kmol] (M_D condensatore, M_B ribollitore)
α	volatilità relativa fra componente leggera e pesante
τ_L	costante di tempo per la dinamica del flusso liquido ad ogni livello [min]
λ	costante per l'effetto del flusso di vapore nella dinamica del flusso liquido

Tabella 1: Notazione.

Come ipotesi semplificativa, si considera il feed completamente liquido, quindi sarà sempre $q_F = 1$; inoltre, la miscela è supposta binaria, e detti $x_i(t)$ e $y_i(t)$ le concentra-

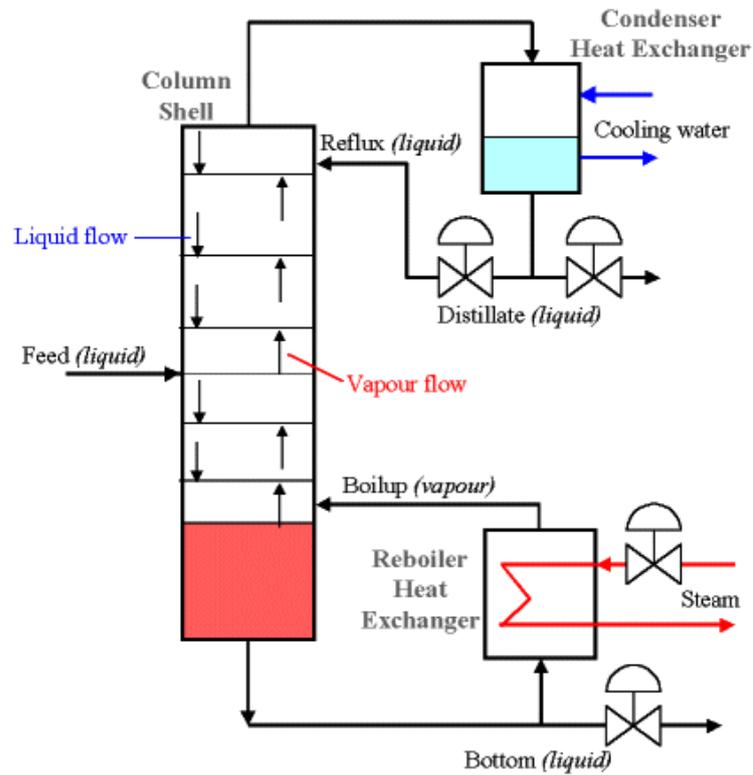


Figura 1: Schema generico della colonna di distillazione.

zioni rispettivamente liquida e gassosa della componente più volatile al piatto i -esimo, si suppongono legate dalla relazione (ideale)

$$y_i(t) = \frac{\alpha x_i(t)}{1 + (\alpha - 1)x_i(t)}, \quad (1)$$

con α coefficiente di volatilità relativa; i livelli sono numerati in ordine crescente dall'alto verso il basso.

Un primo modello, semplificato, considera la massa totale M_i costante in ogni piatto, compresi condensatore e ribollitore. Ad ogni piatto si hanno flussi di liquido e vapore sia in entrata che in uscita, come descritto in figura 2 e quindi si derivano le seguenti equazioni

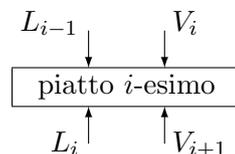


Figura 2: Dinamica di un piatto.

(omettendo l'indice temporale t per alleggerire la notazione):

- *bilancio di materia totale al piatto i* ($2 \leq i \leq N_F - 1$, $N_F + 1 \leq i \leq N - 1$)

$$\frac{dM_i}{dt} = L_{i-1} - L_i + V_{i+1} - V_i \quad (2)$$

- *bilancio di materia per la componente leggera al piatto i ($2 \leq i \leq N_F - 1$, $N_F + 1 \leq i \leq N - 1$)*

$$\frac{dM_i x_i}{dt} = L_{i-1} x_{i-1} - L_i x_i + V_{i+1} y_{i+1} - V_i y_i; \quad (3)$$

da queste due equazioni si ricava l'andamento della variazione di concentrazione $x_i(t)$ come

$$\frac{dx_i}{dt} = \frac{1}{M_i} \left(\frac{dM_i x_i}{dt} - x_i \frac{dM_i}{dt} \right) \quad (4)$$

(derivata di funzione composta, valida $\forall i = 1, \dots, N$)

- *equazioni per il condensatore ($i = 1$)*

$$\frac{dM_D}{dt} = V_2 - L - D \quad (5)$$

$$\frac{dM_D x_D}{dt} = V_2 y_2 - L x_D - D x_D \quad (6)$$

- *equazioni per il piatto di alimentazione ($i = N_F$)*

$$\frac{dM_{N_F}}{dt} = L_{N_F-1} - L_{N_F} + V_{N_F+1} - V_{N_F} + F \quad (7)$$

$$\frac{dM_{N_F} x_{N_F}}{dt} = L_{N_F-1} x_{N_F-1} - L_{N_F} x_{N_F} + V_{N_F+1} y_{N_F+1} - V_{N_F} y_{N_F} + z_F F \quad (8)$$

- *equazioni per il ribollitore ($i = N$)*

$$\frac{dM_B}{dt} = L_{N-1} - V - B \quad (9)$$

$$\frac{dM_B x_B}{dt} = L_{N-1} x_{N-1} - V y_B - B x_B \quad (10)$$

- inoltre, poiché si suppone $q_F = 1$, si ha che

$$V_i = V_{i+1} \quad \forall i \quad (11)$$

$$L_i = \begin{cases} L & \forall i < N_F \\ L + F & \forall i \geq N_F \end{cases} \quad (12)$$

Con questo modello, a regime si ha

$$\begin{cases} V(t) = V \\ R(t) = R \\ F(t) = F \end{cases} ; \quad (13)$$

volendo rendere il modello più accurato, si possono considerare le quantità di materia M_i sui piatti non costanti, introducendo la dinamica del flusso liquido sui vari livelli (*liquid flow dynamics*) aggiungendo $\forall i$ l'equazione

$$L_i = L_{0i} + (M_i - M_{0i})/\tau_L + \lambda(V_{i+1} - V_{0,i+1}) \quad (14)$$

dove M_{0i} , L_{0i} sono rispettivamente quantità di materia e flusso di liquido nominali del piatto i , quest'ultimo definito come (sempre nell'ipotesi $q_F = 1$)

$$L_{0i} = \begin{cases} L & \forall i < N_F \\ L + F & \forall i \geq N_F \end{cases} \quad (15)$$

Il termine λ esprime l'influenza del flusso di vapore nella dinamica del liquido: nei test si considererà sempre $\lambda = 0$. In questo caso, all'equilibrio, oltre a (13) si ha

$$\frac{dM_i}{dt} = 0 \quad \forall i. \quad (16)$$

2 Simulazioni con massa liquida non costante

Il modello non lineare della colonna che tiene conto della dinamica del liquido nei piatti è realizzato dalla funzione MATLAB `colamod` scaricabile dal sito [1], assieme a molte altre implementate per effettuare simulazioni e utilizzate in questa esperienza. La routine rappresenta il secondo membro di un'equazione differenziale del tipo $\dot{x} = f(x, u, t)$, dove lo stato x comprende come prime N componenti le concentrazioni di composto volatile in fase liquida negli N piatti, e come seconde N componenti (da $N + 1$ a $2N$) le quantità di materia totale sui vari piatti, considerate in questo caso tempo-varianti, come visto nelle equazioni precedenti (14). La funzione è della forma `xprime=colamod(t,X,U)`, dove t è l'indice temporale (non usato all'interno della funzione, ma necessario per le altre routine che calcoleranno le soluzioni), X è lo stato di cui va calcolata la derivata, U è il vettore di ingressi, definito come in tabella 2. La routine effettua il calcolo dell'uscita spezzando

U(1)	U(2)	U(3)	U(4)	U(5)	U(6)	U(7)
L_T	V_B	D	B	F	z_F	q_F

Tabella 2: Ingressi di `colamod.m`.

il vettore di ingresso X in due sottovettori, uno contenente le concentrazioni liquide della componente volatile, l'altro le quantità di materia ai vari piatti, e a partire da questi dati e dagli ingressi svolge le equazioni per il calcolo di $\frac{dx_i}{dt}$, $\frac{dM_i}{dt}$, restituendole. A differenza di quanto visto a lezione, l'ordinamento dei piatti dato da Skogestad è di tipo bottom-top, ovvero dal basso all'alto: il piatto numero 1 è il ribollitore, l' N -esimo è il condensatore, e lo scambio di materia viene considerato come in figura 3. Come da richiesta, è stato quindi

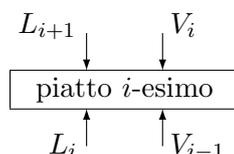


Figura 3: Dinamica di un piatto con ordinamento bottom-top.

modificato il file realizzando `colamod_tb` che ordina i piatti dall'alto verso il basso, ponendo quindi come primo il condensatore e ultimo il ribollitore. Le modifiche fondamentali rispetto alla routine originale sono

- gli indici del tipo

```
i=1:NT-1;
```

diventano

```
i=2:NT;
```

essendo riguardanti operazioni che coinvolgono i piatti dal ribollitore a quello subito prima il condensatore (si noti che $NT = N$)

- gli indici del tipo

```
i=NF:NT-1;
```

`i=NF+1:NT-1;`

che riguardano i piatti nella sezione di arricchimento, condensatore escluso, diventano

`i=2:NF;`
`i=2:NF-1;`

a seconda che comprendano o meno il piatto di alimentazione; per la sezione di esaurimento il procedimento è del tutto analogo con gli opportuni indici

- infine, si scambiano gli indici di condensatore e ribollitore, essendo il primo il piatto numero 1 e il secondo l' N -esimo.

Il file completo è riportato in appendice.

2.1 Configurazione LV

Dopo aver realizzato la funzione con la rinumerazione dall'alto al basso dei piatti, sono stati condotti alcuni test, prima di tutto per verificare che gli output delle due funzioni siano gli stessi (considerando opportunamente gli elementi del vettore risultante da entrambe). Le simulazioni possono essere svolte con diversi approcci, a seconda di quali variabili sono scelte come indipendenti per il controllo di concentrazione [3]. Uno dei metodi più usati è la *configurazione LV*, in cui il riflusso L è usato per controllare la concentrazione di testa x_D e il flusso di vapore V per controllare x_B : L e V sono quindi le variabili su cui si agisce per controllare la dinamica interna alla colonna, mentre D e B sono usate per regolare il livello all'interno del condensatore e ribollitore (M_D e M_B), per garantire stabilità al sistema. Altre configurazioni sono

- *configurazione DV*: in questo caso D è usata per controllare x_D e V per controllare x_B , mentre il controllo di livello è fatto da L e B rispettivamente per condensatore e ribollitore;
- *configurazione LB*: L e B sono usate per il controllo di concentrazione di testa e di coda;
- *configurazione DB*: si usano D e B per controllare x_D , x_B ;
- *configurazione (L/D)(V/B)*: in questo caso si usano i rapporti (L/D) , (V/B) per il controllo di concentrazione: si verifica [3] che con questa struttura il controllo è migliore, ma a scapito della complessità molto maggiore per realizzarlo.

Per quest'ultimo motivo, spesso nell'industria di processo di preferisce utilizzare la configurazione LV , più semplice ed economica da implementare e che fornisce risultati buoni, rapidità di controllo e maggior robustezza alle variazioni di flusso.

Nell'esperienza, le simulazioni sono state condotte con quest'ultima, utilizzando entrambi i modelli analizzati in precedenza, con ordinamento bottom-top e top-bottom.

In MATLAB, le funzioni che risolvono problemi ai valori iniziali per equazioni differenziali ordinarie del prim'ordine hanno forma del tipo

```
[T X] = odeXX(odefun,tspan,x0)
```

dove **XX** viene sostituito dall'identificativo del metodo di integrazione scelto (23, 45, 15s...)¹, **tspan** è l'intervallo temporale in cui si vuole effettuare l'integrazione, **x0** il valore iniziale e **odefun** il membro destro dell'equazione differenziale, cioè la funzione che valuta

¹Nel nostro caso, è stato usato **ode15s**, come suggerito in [1], per l'efficienza sia in precisione che in tempo di esecuzione.

$f(x, t)$. Come parametro `odefun` non si può usare direttamente `colamod` (o rispettivamente `colamod_tb`), in quanto è richiesto che la funzione passata a `odeXX` abbia come parametri solo \mathbf{X} e \mathbf{t} , mentre le due funzioni necessitano anche del vettore \mathbf{U} precedentemente illustrato, e inoltre non effettuano il controllo di livello richiesto dalla configurazione LV desiderata: è quindi necessaria un'altra routine messa a disposizione sempre in [1], `cola_lv`, che

1. setta gli ingressi desiderati creando il vettore \mathbf{U} ;
2. effettua il controllo di livello su M_D , M_B usando rispettivamente D e B ;
3. richiama la funzione `colamod` restituendone l'uscita;

dall'esterno la routine prende solo \mathbf{t} e \mathbf{X} e quindi può essere usata per il calcolo. Particolarmente interessante è osservare come viene effettuato il controllo di livello: vengono usati due controllori proporzionali con coefficienti $K_{cD} = K_{cB} = 10$. Il codice corrispondente è

```
KcB=10; KcD=10;           % controller gains
MDs=32.1; MBs=10;        % Nominal holdups
Ds=0.5; Bs=0.5;         % Nominal flows
MB=X(NT+1); MD=X(2*NT); % Actual reboiler and condenser holdup
D=Ds+(MD-MDs)*KcD;      % Distillate flow
B=Bs+(MB-MBs)*KcB;      % Bottoms flow
```

in cui, dopo aver settato i coefficienti dei controllori e i valori nominali di M_{Ds} , M_{Bs} , D_s , B_s , si prelevano i valori attuali di M_D , M_B dalle corrispondenti posizioni del vettore di stato (con ordinamento bottom-top) e si valuta il valore di D e B da imporre in base alla differenza fra valori attuali e nominali delle quantità di materia: è evidente che con livelli superiori a quelli nominali aumentino i flussi di distillato e residuo, viceversa per livelli inferiori. Anche questo file è stato adattato per la versione con ordinamento top-bottom, semplicemente scambiando le posizioni del vettore di stato da cui prelevare M_D , M_B (file `cola_lv_tb` in appendice).

2.2 Risultati delle simulazioni

Essendo queste simulazioni destinate a testare il comportamento del modello della colonna in transitorio nella configurazione LV , tutti gli ingressi (L , V) e i disturbi (F , z_F , q_F) sono supposti costanti e non vengono modificati, mentre verranno variate le condizioni iniziali su $M_i \forall i$ per osservare il comportamento del sistema: i dati usati sono riportati in tabella 3 (tratti da [1, 3]). Si noti che si è preso $N = 21$ per fare in modo che con entrambi gli ordinamenti il piatto di alimentazione sia quello centrale, $N_F = 11$; le capacità nominali di condensatore e ribollitore assumono valori più grandi e diversi da quelle dei piatti intermedi per rendere le simulazioni il più possibile fedeli ad un caso reale [1].

N	N_F	F	z_F	q_F	α	L	V	D_s	B_s	$M_{Ds} = M_{0,1}$	$M_{Bs} = M_{0,21}$	L_{0i}	M_{0i}	τ_L
21	11	1	0.5	1	1.5	2.706	3.206	0.5	0.5	32.1	10	0.5	0.5	0.063

Tabella 3: Dati per le simulazioni.

In primo luogo, si è verificato che le routine modificate, `colamod_tb` e `cola_lv_tb` diano gli stessi risultati delle originali, a condizione di invertire gli indici del vettore di stato: per fare questo, si sono condotte simulazioni a partire da condizioni iniziali coincidenti ai valori nominali per M_i , mentre per le concentrazioni si sono prese c.i. $x_{0i} = 0.5 \forall i$ (e in tutta l'esercitazione sono sempre state usati questi come valori iniziali delle concentrazioni), e sono state plottate le concentrazioni di testa, di coda, di alimentazione e di due piatti intermedi, uno nella zona di arricchimento, uno in quella di esaurimento. In figura 4 sono

riportate le uscite del file originale (4(a)), del file modificato considerando gli opportuni indici corrispondenti ($X(1) = x_D$, $X(21) = x_B$, 4(b)) e considerando invece gli stessi indici (4(c)) per mostrare la specularità. È evidente come i primi due grafici siano identici, a conferma della perfetta uguaglianza fra i due file a condizione di prendere il vettore d'uscita nel giusto ordine, mentre confrontando con il terzo si vede proprio come prendendo gli indici nello stesso ordine per entrambi gli ordinamenti si ottengano risultati esattamente speculari, come deve essere. Data la perfetta equivalenza fra le due routine, d'ora in avanti si userà sempre quella modificata (ordinamento top-bottom).

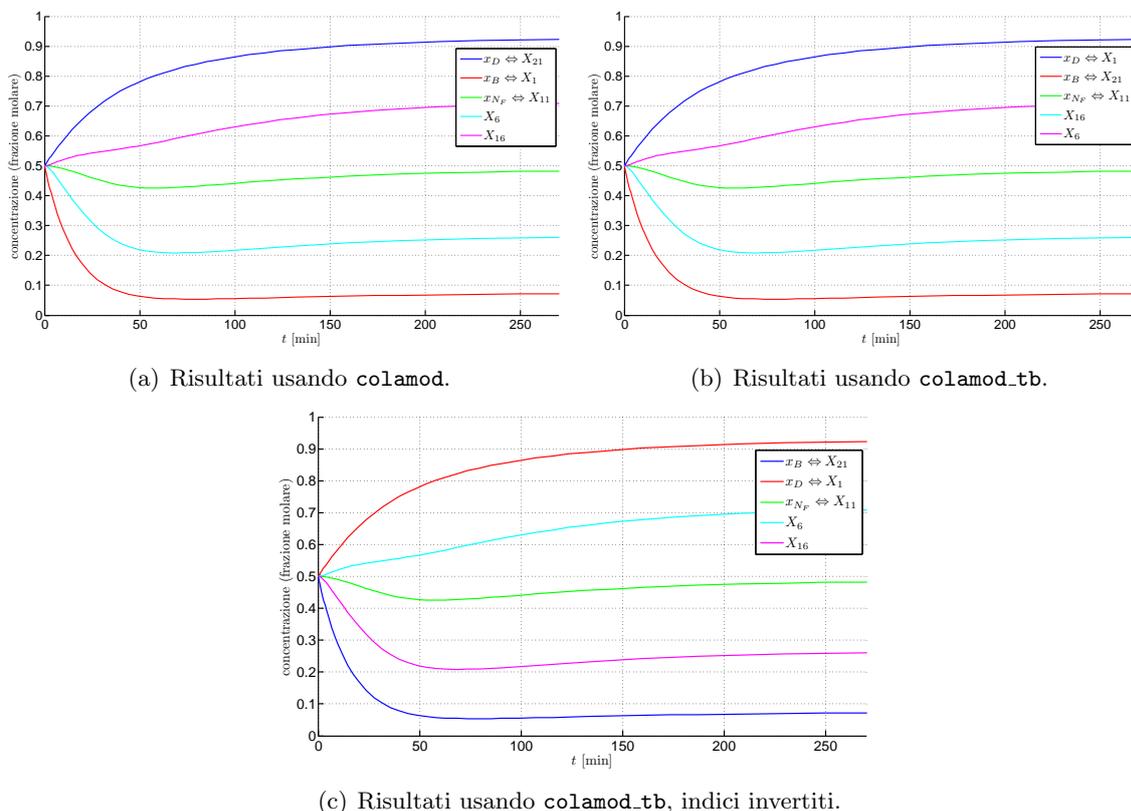


Figura 4: Confronto fra file colamod e colamod.tb.

Si è poi provato a variare le condizioni iniziali sulla massa totale presente nei vari piatti, in modo da introdurre l'effetto della liquid flow dynamics e osservare le ripercussioni sul transitorio. In primo luogo sono state prese c.i. minori dei valori nominali per ogni piatto, in particolare $M_{0D} = 10$ [kmol], $M_{0B} = 1$ [kmol], $M_{0i} = 0.1$ [kmol], $2 \leq i \leq N - 1$. In figura 5 è riportato il confronto con il caso precedente (masse già a regime): come si vede dall'ingrandimento, ciò che varia è soltanto un breve transitorio iniziale, dovuto sia alla compensazione del livello di liquido nel condensatore e nel ribollitore, sia al ritardo introdotto dalla dinamica dei liquidi nei piatti per portare le portate L_i ai valori di regime. Si noti che c'è comunque convergenza agli stessi valori a transitorio esaurito, come c'è da aspettarsi, essendo i flussi L e V gli stessi: è importante osservare però che ciò è reso possibile proprio grazie al controllo di livello al primo e all'ultimo piatto, senza il quale il sistema diventa instabile, come si può vedere in figura 6(a) dove si è fatta girare una simulazione annullando l'azione proporzionale su condensatore e ribollitore: la concentrazione nell'ultimo piatto diventa addirittura negativa, indice che il processo è fuori controllo. Sono proprio questi controlli proporzionali che permettono inoltre di esaurire rapidamente il transitorio, grazie al valore piuttosto elevato: riducendo i guadagni da 10 a 1, si nota una diminuzione delle prestazioni, con tempo di assestamento maggiore (a fronte di un maggior risparmio

nel realizzare il segnale di controllo, che risulterebbe più debole): per la comparazione fra i due valori delle costanti proporzionali si veda figura 6(b).

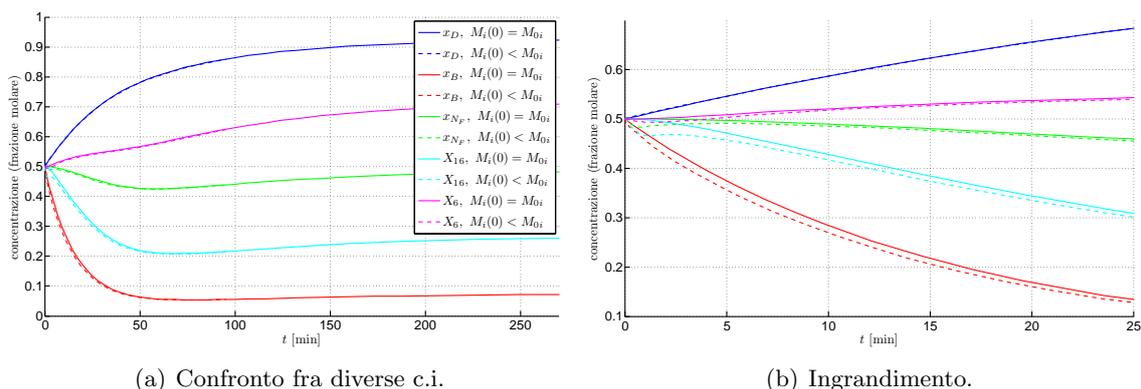


Figura 5: Confronto fra c.i. uguali e inferiori a M_{0i} per le masse.

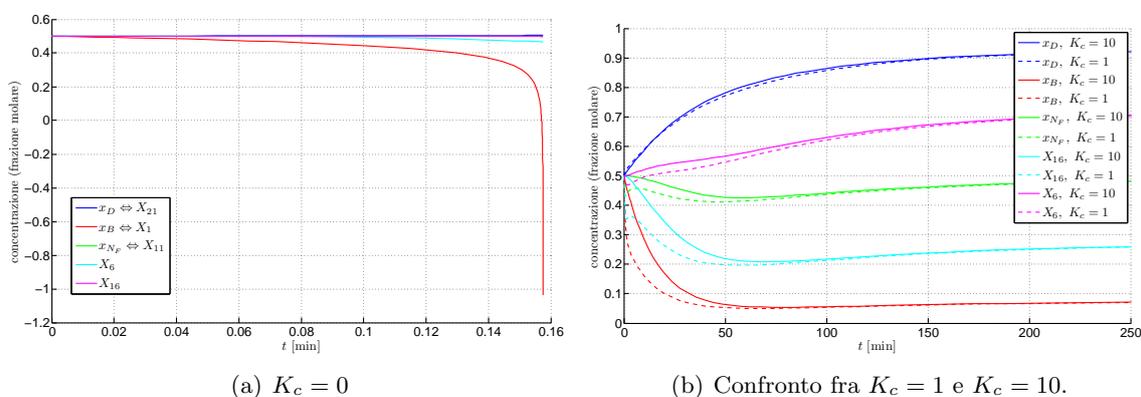


Figura 6: Variazioni dei guadagni nei controllori di livello.

Infine, si è provato a prendere condizioni iniziali superiori alle capacità nominali ($M_{0D} = 90$ [kmol], $M_{0B} = 25$ [kmol], $M_{0i} = 1.5$ [kmol], $2 \leq i \leq N - 1$): anche in questo caso si notano solo piccole variazioni nel transitorio iniziale, grazie all'azione di compensazione del controllo di livello. I grafici relativi sono in figura 7.

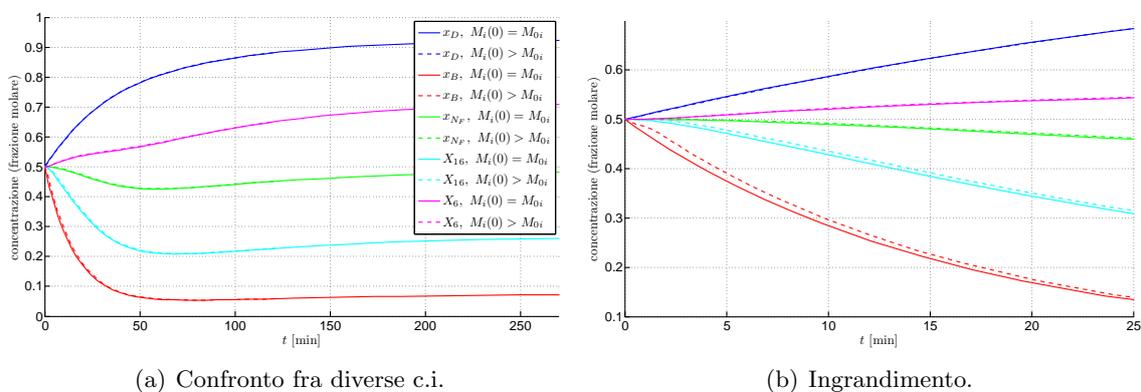


Figura 7: Confronto fra c.i. uguali e maggiori a M_{0i} per le masse.

In tutti i casi (tranne quello di instabilità) si ha convergenza a determinati valori per

le concentrazioni, riportati in tabella 4, mentre in figura 8 è riportata una simulazione completa di raggiungimento dello steady-state.

x_D	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{N_F}
0.9263	0.8934	0.8551	0.8119	0.7647	0.7148	0.664	0.6143	0.5673	0.5245	0.4867
x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_B	
0.4501	0.4085	0.3628	0.3147	0.2662	0.2191	0.1753	0.1361	0.1022	0.0737	

Tabella 4: Valori di regime.

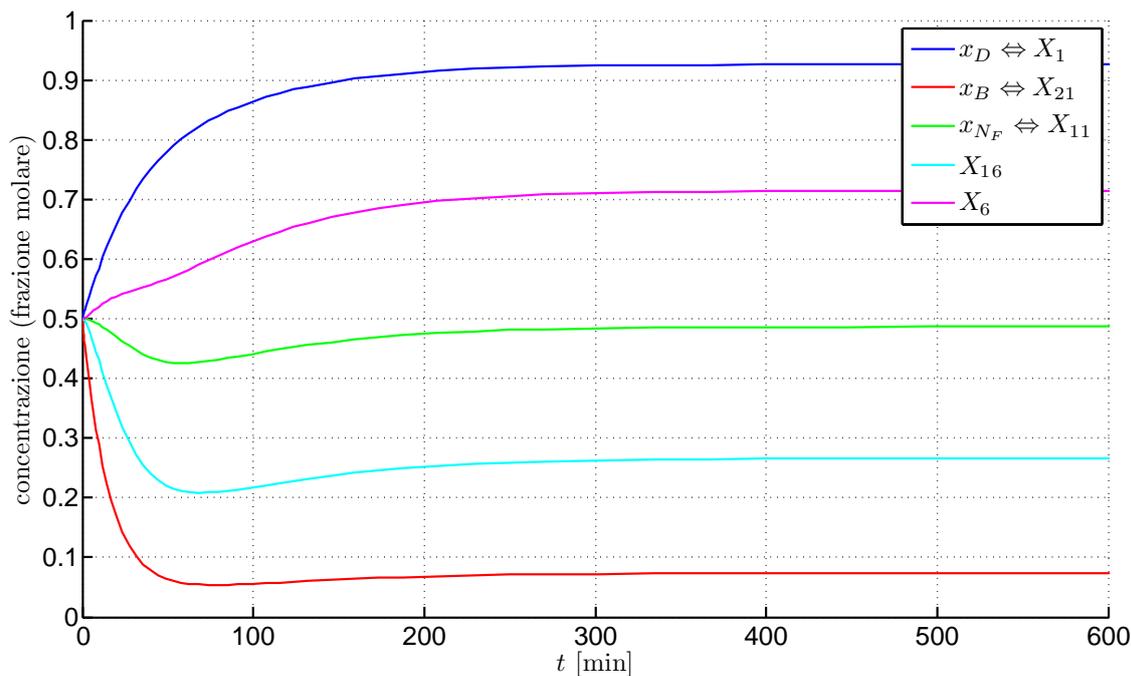


Figura 8: Andamento a regime delle concentrazioni.

3 Simulazioni con massa liquida costante

La dinamica interna della colonna può essere semplificata considerando le masse totali su tutti i piatti costanti, condensatore e ribollitore compreso: in questo modo sparisce il controllo di livello, appunto perché esso rimane costante per ipotesi. A differenza delle simulazioni precedenti, in questo paragrafo i test verranno fatti usando SIMULINK, o meglio una funzione MATLAB, `distillatore`, simile alle precedenti ma realizzata in modo da sfruttare il tool grafico.

La routine è una versione modificata di quella scritta da Bequette, `dist_dyn` (reperibile in [2]), realizzata per essere lanciata da riga di comando mediante funzioni risolutive `odeXX` come già descritto in precedenza. `dist_dyn` riceve in ingresso il vettore x degli stati correnti (questa volta solo le concentrazioni) e svolge il calcolo di $\frac{dx_i}{dt}$ esattamente come `colamod`, omettendo di calcolare le equazioni della liquid flow dynamics in quanto le masse sono supposte costanti e uguali ai valori nominali; l'ordinamento dei piatti è top-bottom. Il parametro τ non è stavolta necessario solo per i risolutori di tipo `odeXX`, ma è funzionale alla routine stessa, che prevede possibili variazioni a gradino dei parametri L, V, F, z_F : è quindi necessario un indice temporale per stabilirne il tempo d'inizio. I parametri necessari al calcolo (numero di piatti, piatto d'alimentazione, flussi di liquido e vapore, masse, ampiezza

dei gradini ecc. . .) devono essere dichiarati esternamente in una variabile globale vettoriale `DIST_PAR`: la funzione la preleva dal workspace e ne usa i singoli componenti per i calcoli. Con `distillatore` si è voluto modificare la funzione originale per renderla più elastica, ampliando i possibili tipi di variazioni da applicare alle quattro variabili sopra citate. In particolare, le modifiche principali sono

1. eliminazione della variabile globale `DIST_PAR`: i valori usati sono memorizzati una volta per tutte all'interno della funzione stessa;
2. uso di un solo parametro di ingresso, il vettore `xu`, di lunghezza $N + 4$, le cui prime N componenti sono le concentrazioni, e le ultime quattro sono le variazioni rispetto al valore nominale di L , V , z_F , F all'istante corrente;
3. sostituzione dei cicli per le variazioni a gradino, del tipo

```
if t < tstepr;
    reflux = refluxi;
else
    reflux = refluxi + stepr;
end
```

con comandi per le variazioni impostate dall'esterno, tipo

```
reflux = refluxi + xu(ns+1);
```

il modello SIMULINK usato è in figura 9 (la routine è incapsulata in un blocco `Matlab function`): poiché si è interessati a osservare il transitorio piuttosto che gli effetti di variazioni dei flussi sui valori di regime, per il momento tutti e quattro gli ingressi esterni sono stati posti a 0.

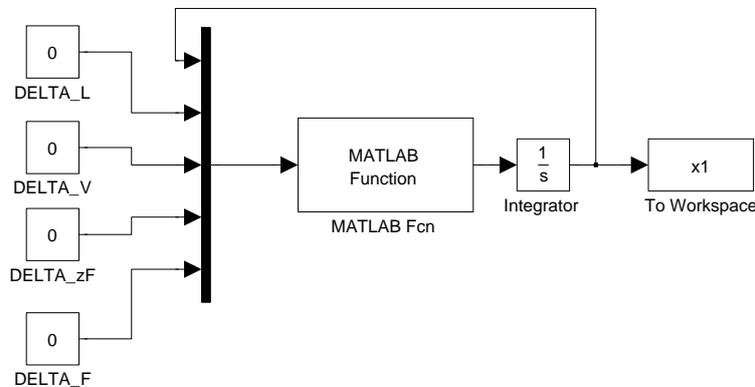


Figura 9: Modello SIMULINK usato.

3.1 Risultati delle simulazioni

I dati usati per le simulazioni rimangono quelli in tabella 3, con la differenza che non essendoci dinamica dei fluidi nei vari piatti, i valori nominali delle quantità di materia M_{0i} diventano i valori effettivi delle masse totali ad ogni livello, quindi $M_i = M_{0i} \forall i$ (compresi ribollitore e condensatore): poiché per conformità con i test precedenti si considereranno come c.i. per le concentrazioni sempre $x_{0i} = 0.5 \forall i$, necessariamente le uscite avranno un andamento univoco, riportato in figura 10.

È logico aspettarsi che i grafici ottenuti con il modello a massa costante coincidano con quelli del modello a massa variabile, nel caso in cui questo abbia condizioni iniziali sulle

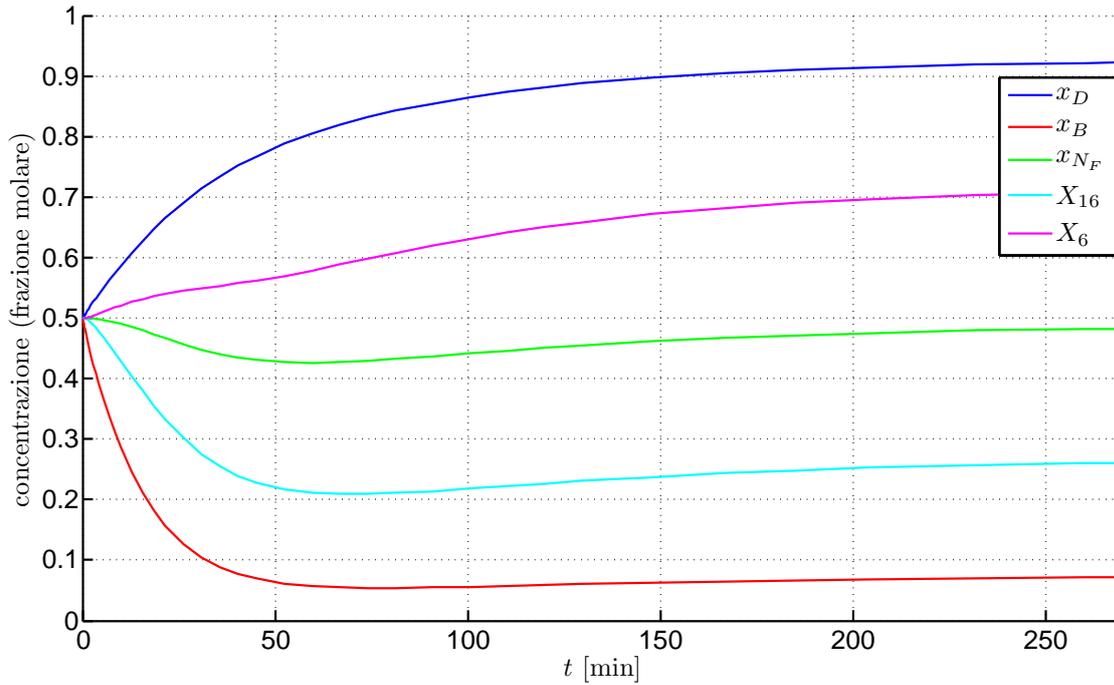
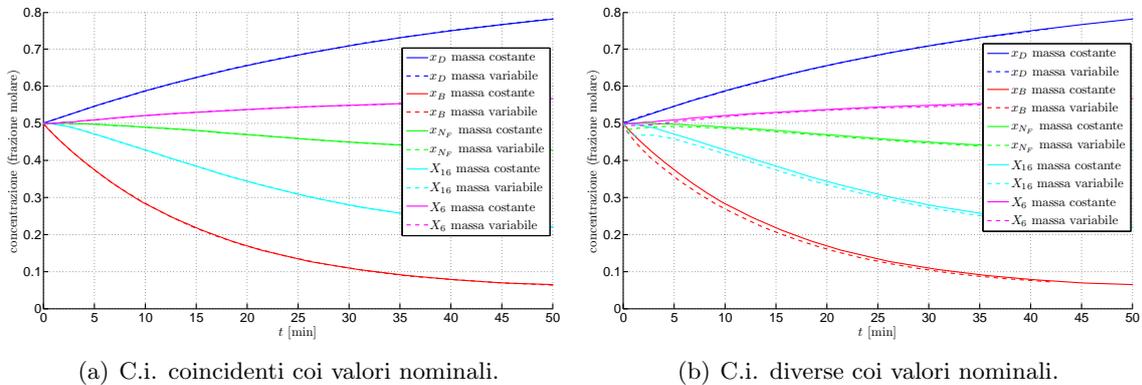


Figura 10: Andamento delle concentrazioni con `distillatore` via SIMULINK.

masse coincidenti ai valori nominali: la mancanza di transistori su M_i e quindi su L_i (assenza di dinamica dei liquidi nei piatti) porta necessariamente a risultati identici, a condizione di usare come valori costanti per M_i nel modello semplificato le capacità nominali M_{0i} . Ciò è confermato dai grafici in figura 11, dove sono messi a confronto le uscite del modello SIMULINK con quelle di `colamod_tb` al variare delle c.i. sulle masse di quest'ultimo.



(a) C.i. coincidenti coi valori nominali.

(b) C.i. diverse coi valori nominali.

Figura 11: Confronto fra modelli a massa costante e variabile.

Infine, è interessante mettere a confronto le risposte del modello a massa costante usando diversi valori per M_D e M_B : in particolare, in figura 12 sono messi a confronto simulazioni con $M_D = 32.1$ [kmol], $M_B = 10$ [kmol] come suggerito da Skogestad [1, 3] (linee continue), e $M_D = M_B = 5$ [kmol] (linee tratteggiate), valori suggeriti da Bequette [2]. È evidente che con valori più bassi il transitorio si esaurisce molto più rapidamente. Questo è facilmente spiegabile osservando l'equazione (4): la derivata temporale di x_i esprime quanto rapidamente vari la concentrazione nel tempo, ed essendo inversamente proporzionale ad M_i , maggiore è quest'ultima, più lentamente varierà la concentrazione, quindi un aumento di M_D e M_B rende più lento l'esaurimento del transitorio per x_D e x_B , con ovvie ripercussioni

su tutte le restanti concentrazioni.

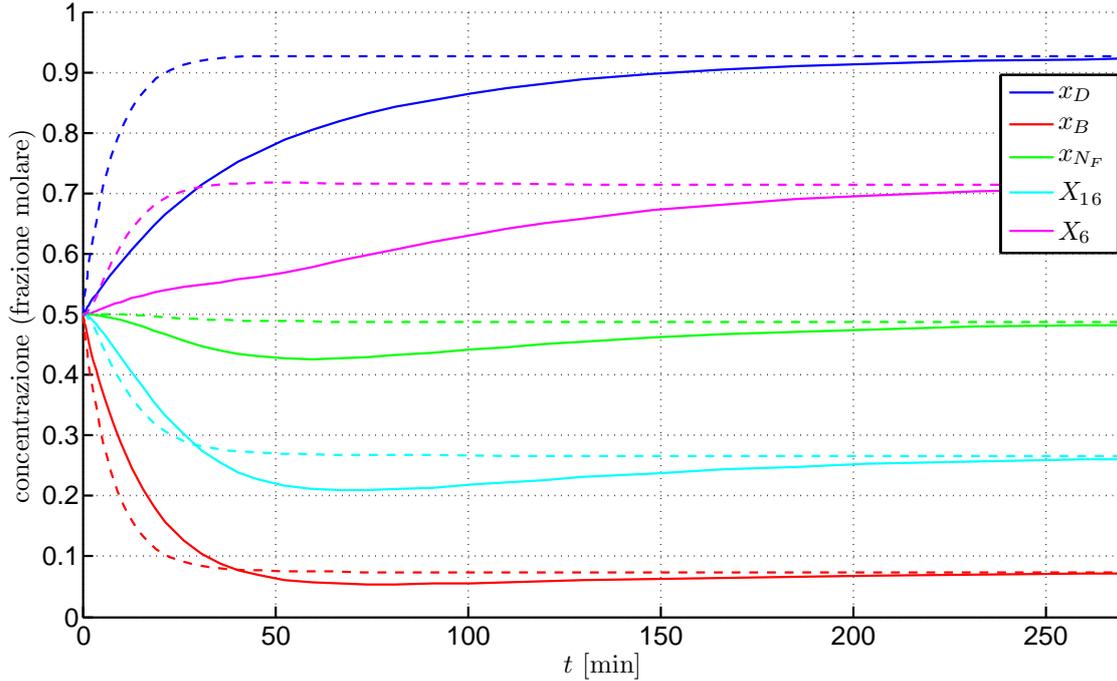


Figura 12: Andamento delle concentrazioni al variare di M_D , M_B ($M_D = 32.1$, $M_B = 10$ continuo; $M_D = M_B = 5$ tratteggiato).

Gli steady-state coincidono con quelli del caso precedente, essendo stati usati gli stessi valori per gli ingressi L , V e i disturbi z_F , F .

4 Andamento delle temperature

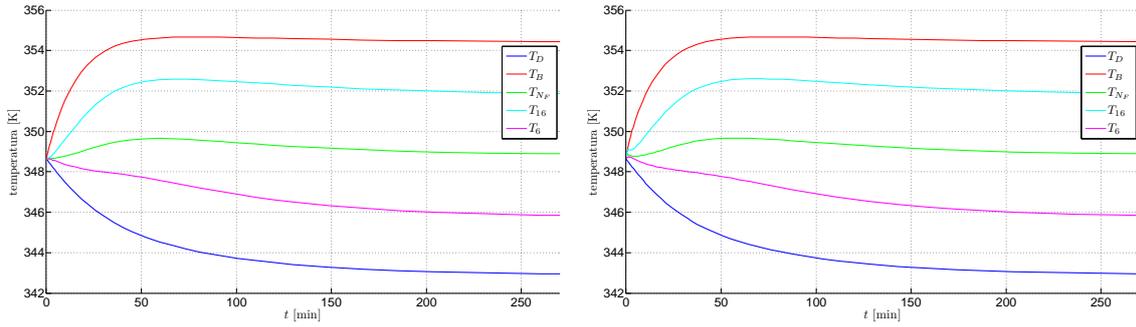
La dinamica delle concentrazioni di composto all'interno dei piatti ne influenza la temperatura, che varia in maniera proporzionale alla variazione delle concentrazioni di componente leggera e pesante nel liquido. Questa corrispondenza è particolarmente importante per il controllo, perché permette di usare semplici sensori di temperatura per ricavare real-time la concentrazione su determinati piatti e usare tali misure per agire opportunamente sugli ingressi e dare la regolazione migliore. Un'approssimazione molto usata, valida per una gran quantità di miscele binarie, è una relazione lineare del tipo

$$T_i = \underbrace{x_i}_{x_{L,i}} T_{bL} + \underbrace{(1-x_i)}_{x_{H,i}} T_{bH} \quad (17)$$

$$= x_i(T_{bL} - T_{bH}) + T_{bH} \quad (18)$$

dove $x_{L,i}$, $x_{H,i}$ sono le concentrazioni della componente leggera e pesante nel liquido al piatto i -esimo, e T_{bL} , T_{bH} le rispettive temperature di ebollizione. Poiché si ha che $T_{bL} < T_{bH}$ (la componente più volatile ha sempre temperatura di ebollizione inferiore rispetto a quella più pesante, principio base per il processo di distillazione [4]), è evidente dalla formula che all'aumentare di x_i diminuisca T_i , quindi i piatti più vicini al condensatore avranno temperatura inferiore rispetto a quelli più in basso, presentando una concentrazione di componente leggera maggiore. In figura 13 sono riportati i grafici con l'andamento delle temperature usando i dati ottenuti sia da `distillatore` via SIMULINK, sia con `colamod_tb` (con c.i. sulle masse inferiori ai valori nominali), e in figura 14 è plottato il confronto fra i due. L'andamento è del tutto speculare a quello delle concentrazioni, come c'era da

aspettarsi dato il tipo di legame lineare fra T_i e x_i ; per le temperature di ebollizione sono stati usati i valori suggeriti in [1], $T_{bL} = 341.9$ [K], $T_{bH} = 355.4$ [K].



(a) Risultati ottenuti con modello a massa costante. (b) Risultati ottenuti con modello a massa variabile.

Figura 13: Andamento delle temperature nei due tipi di modello.

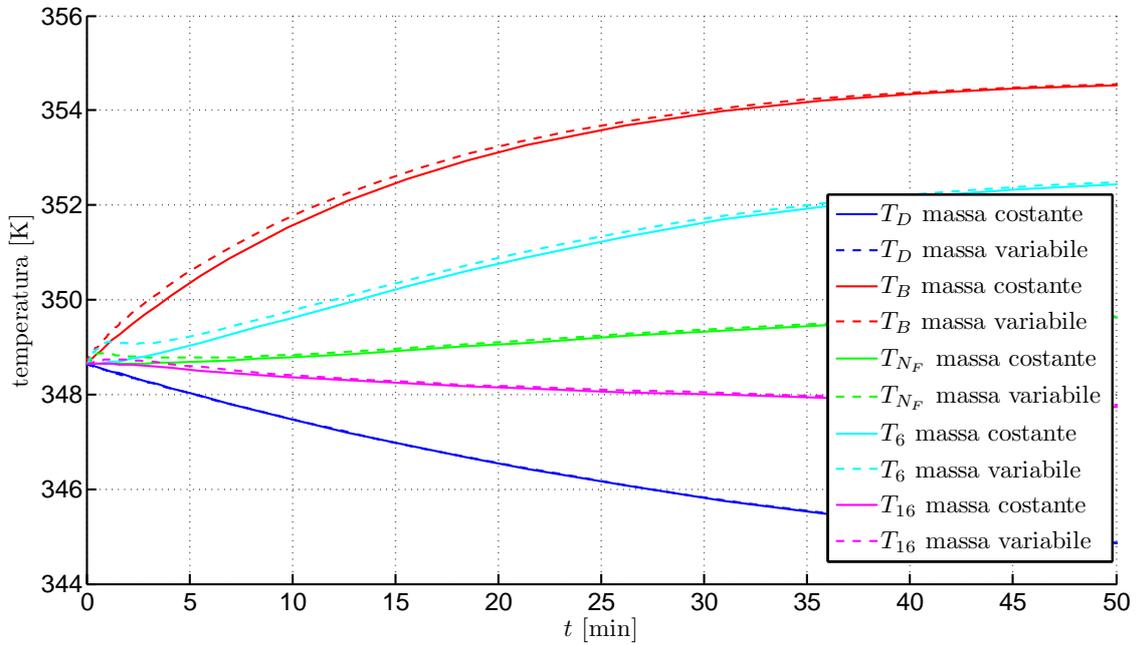


Figura 14: Confronto fra gli andamenti delle temperature nei due tipi di modello considerati.

I valori delle temperature a regime sono in tabella 5.

T_D	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
342.895	343.3395	343.856	344.4394	345.0773	345.7508	346.436	347.1073	347.7412
T_{10}	T_{N_F}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}
348.3191	348.8294	349.3236	349.8855	350.5015	351.1507	351.8067	352.4423	353.0335
T_{19}	T_{20}	T_B						
353.5626	354.0205	354.405						

Tabella 5: Valori delle temperature a regime in [K].

5 Filtro di Kalman

Il *filtro di Kalman* è uno dei metodi più diffusi per la stima a minima varianza d'errore (m.v.) di grandezze rilevate con misure rumorose, ovvero lo scopo è quello di produrre stime con la più piccola dispersione possibile rispetto al valore vero. È un algoritmo *ricorsivo* che si basa sull'ipotesi di poter rappresentare il sistema secondo un modello discreto *lineare* in spazio di stato, del tipo

$$\begin{aligned}\mathbf{x}(t+1) &= A\mathbf{x}(t) + Bu(t) + \mathbf{v}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + \mathbf{w}(t)\end{aligned}$$

(in grassetto si intendono variabili e processi aleatori), dove

- $\mathbf{x}(t)$ è lo stato, tipicamente la grandezza da stimare;
- $u(t)$ sono gli ingressi noti e manipolabili;
- $\mathbf{y}(t)$ è il processo delle misure;
- $\mathbf{v}(t)$ è il *rumore di modello*, che riassume le imprecisioni della modellizzazione ed eventuali disturbi agenti sul sistema;
- $\mathbf{w}(t)$ è il *rumore di misura*, che tiene conto delle imprecisioni nelle misure usate per la stima (tipicamente dovute ai sensori);

inoltre, si fanno le seguenti ipotesi

- $\text{Var } \mathbf{v}(t) = Q$;
- $\text{Var } \mathbf{w}(t) = R > 0$;
- $\text{Cov } \mathbf{v}(t), \mathbf{w}(t) = S$;
- $\mathbf{x}_0 \perp \mathbf{v}(t)$;
- $\mathbb{E}[\mathbf{x}_0] = \bar{x}_0$, $\text{Var } \mathbf{x}_0 = P_0$ note.

Supponendo per semplicità $S = 0$ (vera in gran parte dei casi), l'algoritmo comprende i seguenti passi [5]:

1. *stime a priori* (aggiornamento temporale)

$$\hat{\mathbf{x}}(t+1|t) = A\hat{\mathbf{x}}(t|t) + Bu(t) \quad (19)$$

$$P(t+1|t) = AP(t|t)A^T + Q \quad (20)$$

2. *stime a posteriori* (aggiornamento rispetto alle misure)

$$\hat{\mathbf{x}}(t+1|t+1) = \hat{\mathbf{x}}(t+1|t) + L(t+1)[\mathbf{y}(t+1) - C\hat{\mathbf{x}}(t+1|t)] \quad (21)$$

$$P(t+1|t+1) = P(t+1|t) - (t+1|t)C^T \Lambda(t+1)^{-1}CP(t+1|t) \quad (22)$$

3. *condizioni iniziali*

$$\begin{aligned}\hat{\mathbf{x}}(t_0|t_0-1) &= \bar{x}_0 \\ P(t_0|t_0-1) &= P_0,\end{aligned}$$

tipicamente $t_0 = 0$.

$P(t+1|t)$, $P(t|t)$ sono le varianze degli errori di predizione e di stima, $\Lambda(t)$ è la varianza del processo di innovazione $\mathbf{e}(t) = \mathbf{y}(t) - C\hat{\mathbf{x}}(t|t-1)$,

$$\Lambda(t) = CP(t|t-1)C^T + R$$

e $L(t)$ è il guadagno del filtro

$$L(t) = P(t|t-1)C^T\Lambda(t)^{-1}.$$

Spesso la (22) è scritta in forma simmetrizzata

$$P(t+1|t+1) = [I - L(t+1)C]P(t+1|t)[I - L(t+1)C]^T + L(t+1)RL(t+1)^T$$

per evitare il calcolo dell'inversa di $\Lambda(t)$ (numericamente oneroso e instabile).

5.1 Applicazione al caso studiato: modello linearizzato

La colonna di distillazione è, come si è visto, un modello fortemente non lineare, e questo impedisce di realizzare un filtro di Kalman *lineare* per stimarne l'*intera* dinamica: per far questo si deve usare la versione non lineare dell'algoritmo, il filtro di Kalman *esteso* [5, 4]. Tuttavia è possibile approssimare bene la dinamica del modello non lineare per piccole perturbazioni attorno ad un punto di equilibrio mediante un sistema lineare. Nel nostro caso il sistema è del tipo

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= g(x, u)\end{aligned}$$

dove x è lo stato (comprendente soltanto le concentrazioni, ci si limita al modello a massa costante), u il vettore che comprende sia gli ingressi L , V che i disturbi z_F , F , e y contiene le misure di temperatura ai piatti 2, N_F , $N-1$ come specificato da consegna (secondo l'equazione (17)). Trovato un punto di equilibrio (\bar{x}, \bar{u}) , il comportamento del sistema è ben approssimato *nell'intorno* di tale punto dalle equazioni lineari

$$\begin{aligned}\delta\dot{x} &= A_c\delta x + B_c\delta u \\ \delta y &= C_c\delta x + D_c\delta u\end{aligned}$$

dove le quattro matrici sono le jacobiane calcolate nel punto di equilibrio

$$A_c = \left. \frac{\partial f}{\partial x} \right|_{\bar{x}, \bar{u}}, \quad B_c = \left. \frac{\partial f}{\partial u} \right|_{\bar{x}, \bar{u}}, \quad C_c = \left. \frac{\partial g}{\partial x} \right|_{\bar{x}, \bar{u}}, \quad D_c = \left. \frac{\partial g}{\partial u} \right|_{\bar{x}, \bar{u}};$$

con δx , δu , δy si intendono piccoli spostamenti attorno al punto di equilibrio: nella trattazione seguente i δ saranno omessi, intendendo sempre di essere nel caso appena illustrato.

Il punto di equilibrio \bar{x} dello stato è il suo valore a regime (tabella 4), per \bar{u} si considerano invece come fatto finora $L = 2.706$ [kmol/min], $V = 3.206$ [kmol/min], $z_F = 0.5$, $F = 1$ [kmol/min]; la linearizzazione è ottenuta numericamente mediante la funzione MATLAB `cola_linearize`, messa a disposizione in [1]. Presi in ingresso \bar{x} , \bar{u} e una funzione che calcoli $f(x, u)$ e $g(x, u)$, la routine restituisce le quattro matrici A_c, B_c, C_c, D_c calcolando gli elementi delle jacobiane con il metodo della *differenza finita centrata*: preso un numero δ molto piccolo, la derivata nel punto viene calcolata come

$$f'(x) \approx \frac{f(x+\delta) - f(x-\delta)}{2\delta}$$

ottenendo un'approssimazione di $f'(x)$ di ordine 2 rispetto a δ . In particolare la routine effettua due volte il calcolo, prima con $\delta = 10^{-6}$ e poi con $\delta = 0.5 \cdot 10^{-6}$, facendo una media

pesata dei risultati, con più peso a quello con incremento minore. La funzione che viene passata per valutare f e g è `linearization_lv` (in appendice), implementata in modo da realizzare direttamente la valutazione di g con l'equazione (17), mentre per calcolare f richiama una versione modificata di `distillatore`, `distillatore_mod`, che prevede un secondo parametro di ingresso, u , per settare L , V , z_F , F , cambiando

```

reflux = u(1);
vapor  = u(2);
zfeed  = u(3);
feed   = u(4);

```

Si ottengono così le quattro matrici del modello lineare a tempo continuo: il filtro di Kalman è però un algoritmo a tempo discreto, d'altra parte la consegna prevede che le misure di temperatura siano campionate con campionamento T_c . Occorre quindi discretizzare il sistema (comando `c2d` di MATLAB): idealmente, si è scelto di prendere $T_c = 1$ [min], ma per una questione di stabilità numerica e di tempo di calcolo, i conti effettivi saranno fatti con $T_c^{num} = 0.001$ [s] e le simulazioni su 10 [s], che è del tutto equivalente ad una situazione con tempo di campionamento 1 [min] e tempo di simulazione 10000 [min] (e saranno queste le scale usate nei grafici).

Si giunge in definitiva ad un sistema nella forma

$$\begin{aligned}x(t+1) &= Ax(t) + \tilde{B}\tilde{u}(t) \\ y(t) &= Cx(t)\end{aligned}$$

(D è stata omessa in quanto nulla). L'ingresso $\tilde{u}(t)$ non è quello vero del sistema in quanto comprende anche i disturbi z_F e F : mentre sul primo non abbiamo informazioni, e quindi non viene considerato, per il secondo viene richiesta una modellizzazione a passeggiata aleatoria del tipo

$$\mathbf{F}(t) = \mathbf{F}(t-1) + \mathbf{v}_F(t)$$

dove $\mathbf{v}_F(t)$ è un rumore bianco di varianza $\sigma_{v_F}^2 = 0.3$: si considera la media del processo nulla (ricordiamo che si sta trattando il sistema linearizzato attorno all'equilibrio, quindi i processi descrivono gli andamenti attorno allo steady-state, *non* assoluti). Poiché gli ingressi veri manipolabili sono le prime due componenti di $\tilde{u}(t)$ e il disturbo dovuto al flusso corrisponde all'ultima componente, si può riscrivere il modello come [5, par. 11.3]

$$\begin{aligned}\mathbf{x}(t+1) &= A\mathbf{x}(t) + Bu(t) + E\mathbf{d}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t)\end{aligned}$$

dove B corrisponde alle prime due colonne di \tilde{B} , E all'ultima colonna, $u(t)$ alle prime due componenti di $\tilde{u}(t)$ e $\mathbf{d}(t)$ tiene conto del disturbo modellato come

$$\mathbf{d}(t) = \mathbf{F}(t) + \tilde{\mathbf{v}}(t)$$

dove $\mathbf{F}(t)$ ha la dinamica appena vista e $\tilde{\mathbf{v}}(t)$, scorrelato da $\mathbf{v}_F(t)$, tiene conto dell'errore di modellizzazione sull'intero sistema. Considerando infine l'errore di misura $\mathbf{w}(t)$ sulle uscite del sistema (bianco a media nulla e varianza $\sigma_w^2 = 1$, a componenti indipendenti) si può riscrivere il sistema *aumentando* lo stato per tener conto della dinamica di $\mathbf{F}(t)$ (minor rischio di divergenza), ottenendo [5, par. 11.3]

$$\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{F}(t+1) \end{bmatrix} = \begin{bmatrix} A & E \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{F}(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} \mathbf{v}_1(t) \\ \mathbf{v}_F(t) \end{bmatrix} \quad (23)$$

$$\mathbf{y}(t) = [C \ 0] \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{F}(t) \end{bmatrix} + \mathbf{w}(t) \quad (24)$$

dove $\mathbf{v}_1(t) = E\tilde{\mathbf{v}}(t)$ è il rumore di modello del sistema originale, supposto a componenti indipendenti e scorrelato da $\{\mathbf{v}_F(t)\}$, di varianza Q_1 da tarare: il rumore di modello per il sistema aumentato $\mathbf{v}(t) \triangleq [\mathbf{v}_1(t) \ \mathbf{v}_F(t)]^T$ avrà quindi varianza $Q = \begin{bmatrix} Q_1 & 0 \\ 0 & \sigma_{v_F}^2 \end{bmatrix}$ diagonale per l'indipendenza delle componenti di $\mathbf{v}(t)$ e dipendente dalla taratura di Q_1 . Il rumore di modello $\mathbf{w}(t)$ ha invece varianza $R = \sigma_w^2 I_3$, diagonale per l'ipotesi di indipendenza fra le componenti.

Lo stato aumentato sarà d'ora in poi indicato con $\mathbf{x}_{ex}(t)$, e le corrispondenti matrici di sistema come A_{ex}, B_{ex}, C_{ex} .

5.2 Taratura del filtro e risultati delle simulazioni

L'algoritmo è implementato mediante una funzione, `Kalman_filter` (in appendice), che presi in ingresso le matrici di sistema, di varianza, le c.i. e ingressi e uscite *nell'intero arco temporale*, restituisce un vettore contenente le stime dello stato; l'implementazione esegue esattamente i passi descritti in precedenza.

Essa è richiamata da un altro script, `kalman` (sempre in appendice), che setta tutti valori delle grandezze considerate. Come prima cosa, richiama la simulazione col modello a massa costante, prendendo l'ultimo valore dello stato come punto di equilibrio per la linearizzazione, e calcolando le corrispondenti temperature di regime (così come lo stato del modello linearizzato mostra l'effetto di piccole variazioni attorno al regime, così le uscite descrivono piccole variazioni attorno alle temperature a transitorio esaurito). Viene quindi calcolata la linearizzazione come spiegato in precedenza (`cola_linearize`), e il modello è discretizzato con tempo di campionamento T_c : la matrice \tilde{B} che si ottiene ha quattro colonne, avendo usato sia gli ingressi L e V che i disturbi z_F e F per il calcolo, quindi va separata nella matrice relativa agli ingressi B (prime due colonne) e E (ultima colonna), e infine da questo si ricava il modello aumentato come precedentemente illustrato.

Come dati per la stima sono stati usati quelli provenienti da un secondo modello SIMULINK, simile a quello di figura 9 ma stavolta con un gradino in Δ_F di ampiezza 0.1 (aumento del 10%) e con condizioni iniziali pari all'equilibrio per simulare l'effetto di piccole perturbazioni. Poiché gli ingressi L e V non vengono modificati, il modello linearizzato è da considerarsi a ingressi nulli, mentre le misure di temperatura sono state ottenute con la relazione (17) a cui sono state aggiunte componenti rumorose di deviazione standard σ_w per simulare l'effetto del rumore di misura dovuto ai sensori: inoltre, essendo come detto il modello lineare la rappresentazione della dinamica *attorno* agli equilibri, alle componenti del processo di misura ottenuto sono state tolti i valori iniziali delle temperature, in modo da ottenere la grandezza $\delta\mathbf{y}(t)$ che è quella corretta per il sistema.

Visto che la varianza di $\mathbf{w}(t)$ è nota, resta da trovare la varianza del rumore di modello Q , o meglio la sottomatrice Q_1 relativa alle componenti dello stato del modello non aumentato, essendo l'ultima componente di Q nota e pari a $\sigma_{v_F}^2$. Per trovare il valore ottimale, è stata effettuata una taratura empirica basata sul *test di Bartlett* [5]: nota R , si cerca di variare Q_1 in modo che il processo di innovazione $\mathbf{e}(t) = \mathbf{y}(t) - C_{ex}\hat{\mathbf{x}}_{ex}(t|t-1)$ sia il più bianco possibile. Poiché nel nostro caso si tratta di un processo vettoriale, la bianchezza va testata in due passi:

1. *bianchezza delle singole componenti*. Si cerca di sfruttare la proprietà di un rumore bianco di avere spettro piatto ad ogni frequenza. Quindi, idealmente l'integrale corrisponde ad una rampa di coefficiente angolare pari alla varianza dell'innovazione. Questa verifica corrisponde al vero e proprio test di Bartlett, o del *periodogramma cumulato*: si approssima lo spettro della componente i -esima del processo ($i = 1, 2, 3$ nel nostro caso) con la tecnica del periodogramma basata sulla FFT, ottenendo un segnale in frequenza molto rumoroso, e si fa una somma integrale che equivale ad un filtraggio passa-basso, verificando quanto il risultato si avvicini alla retta ideale. Si

procede quindi a modificare Q in modo da giungere al miglior esito possibile per il test;

2. *incorrelazione fra componenti.* Un processo bianco *vettoriale* richiede anche che le singole componenti siano fra loro scorrelate: per verificare ciò si calcola la correlazione campionaria $r(\tau)$, considerando circa $1/25 \div 1/50$ dei campioni ottenuti attorno a $\tau = 0$, e si verifica che siano prossimi allo 0.

Il valore Q_1 ottimale è quello che verifica con successo entrambi i passi del test.

Nel caso in questione, poiché il modello viene linearizzato come si è visto con un approssimazione delle derivate dell'ordine di 10^{-6} , si sono cominciate le prove con guess iniziale $Q_1 = \sigma_{v_1}^2 I$, $\sigma_{v_1}^2 = 10^{-6}$, effettuando diverse stime per ottenere come valore ottimale $\sigma_{v_1}^2 = 1.5 \cdot 10^{-6}$. Il periodogramma cumulato è stato calcolato con la funzione `periodogramma`², mentre la correlazione campionaria con la routine `xcorr` (specificando l'opzione `unbiased` che effettua la normalizzazione necessaria per la validità del teorema ergodico). I grafici in figura 15-16 riportano gli esiti con tale valore.

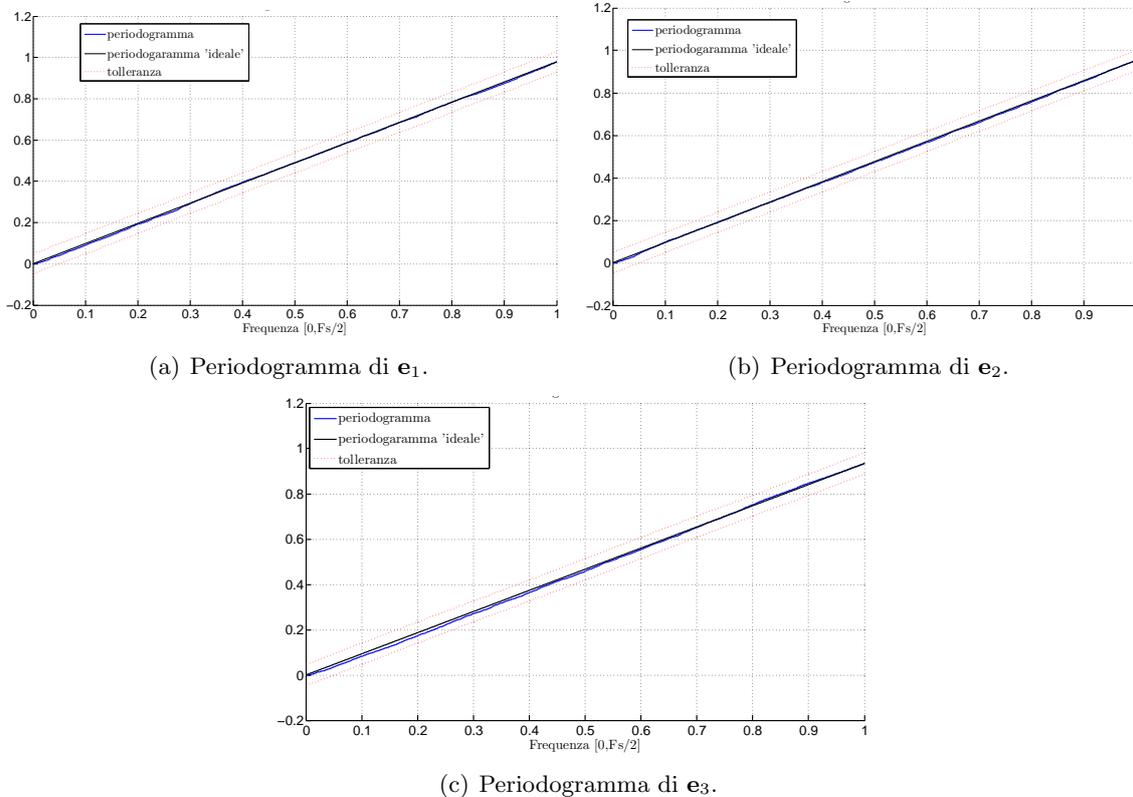


Figura 15: Periodogrammi delle componenti di \mathbf{e} , con indicazione della fascia di tolleranza al 5%.

Un'ultima considerazione va fatta sulle condizioni iniziali \bar{x}_0 , P_0 : data la natura particolare del problema, è ovvio scegliere $\bar{x}_0 = 0$ (il modello rappresenta la dinamica attorno a \bar{x} , quindi è ragionevole pensare che $\mathbf{x}(0) - \bar{x}$ abbia media nulla, e tenendo conto anche del fatto che \mathbf{F} è supposto a media nulla), e per P_0 una matrice diagonale con bassi valori nei primi N elementi della diagonale (buona certezza sul valore iniziale, si è ritenuto anche da prove sperimentali ottimale un valore pari a 0.1 per ciascuna di queste N componenti), e con $\sigma_{v_F}^2$ sull' $(N + 1)$ -esimo elemento (varianza dell'ultima componente dello stato aumentato).

Una volta tarate tutte le grandezze necessarie, è stato fatto girare l'algoritmo di stima con i dati illustrati per quanto riguarda matrici, c.i., ingressi e misure: le stime della prima

²Scaricabile al sito http://www.dei.unipd.it/~picci/Controllo_dei_Processi.html.

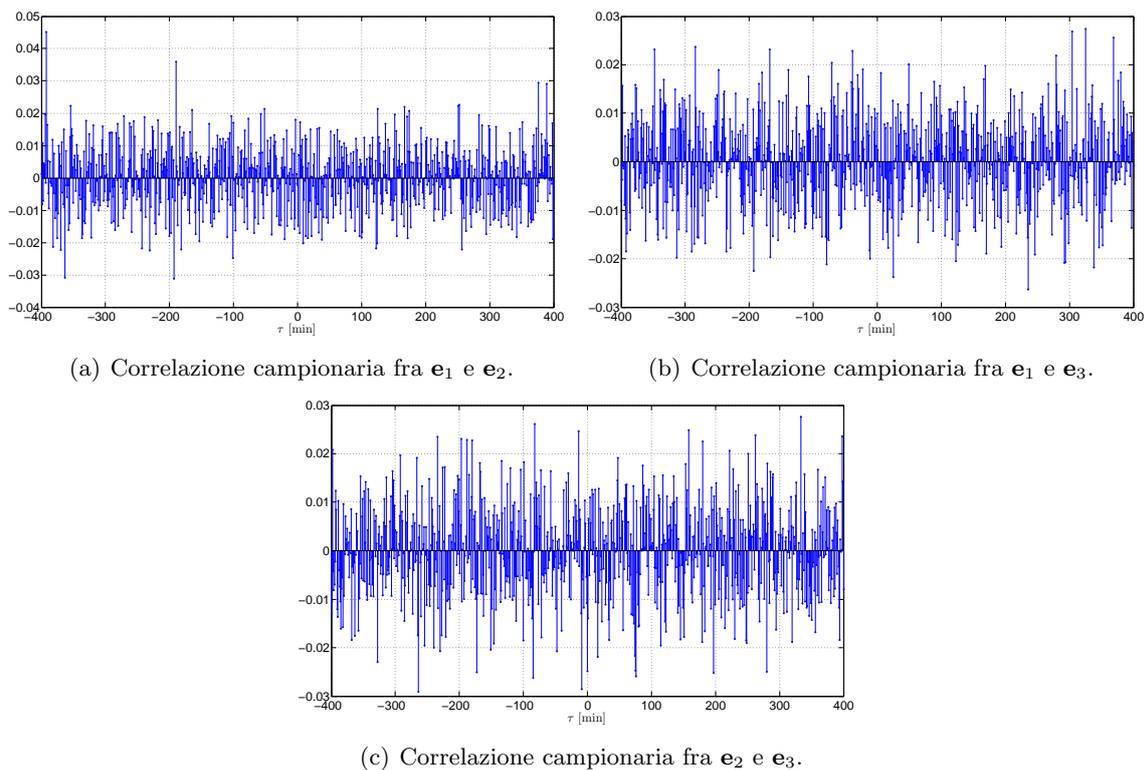


Figura 16: Andamento delle correlazioni campionarie fra le componenti di \mathbf{e} su 1/25 dei campioni.

componente, la concentrazione di componente volatile liquida nel condensatore, e dell'ultima, nel ribollitore, sono riportate in figura 17-18, dove è evidente come l'inseguimento dopo un breve transitorio iniziale sia soddisfacente, soprattutto osservando $\hat{\mathbf{x}}_{21}(t|t)$ dato che x_{21} è fra le due la componente che risente maggiormente dell'effetto nella variazione di F (cambiamento più marcato).

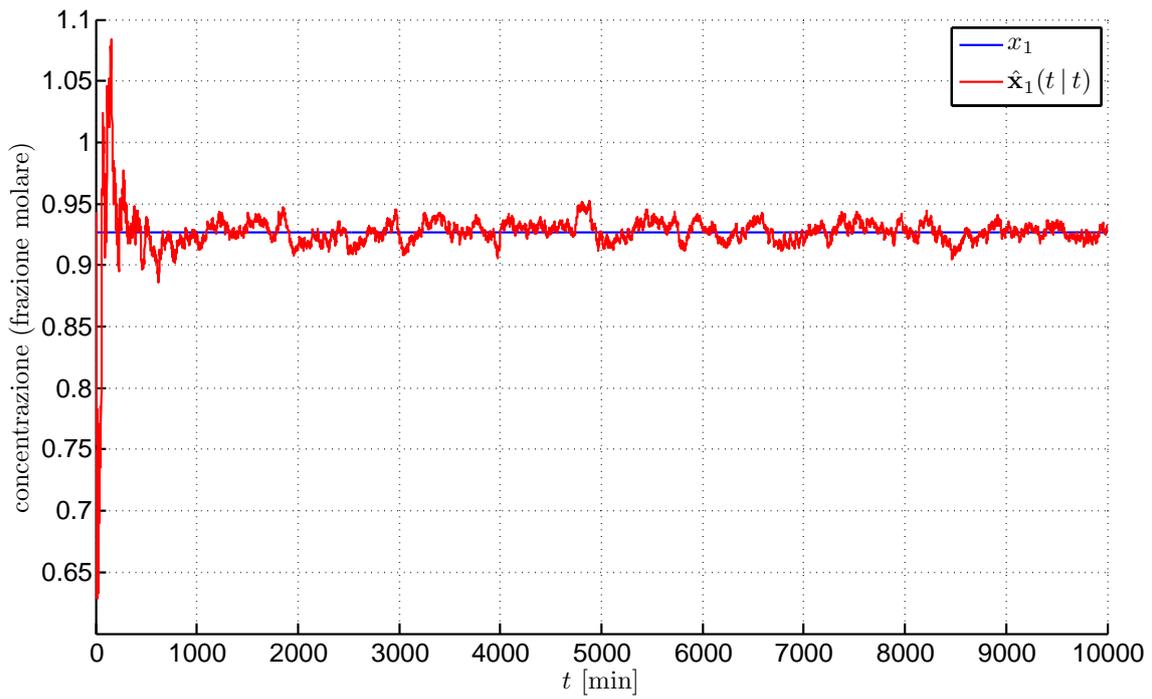


Figura 17: Confronto fra andamento di x_1 (condensatore) e la sua stima.

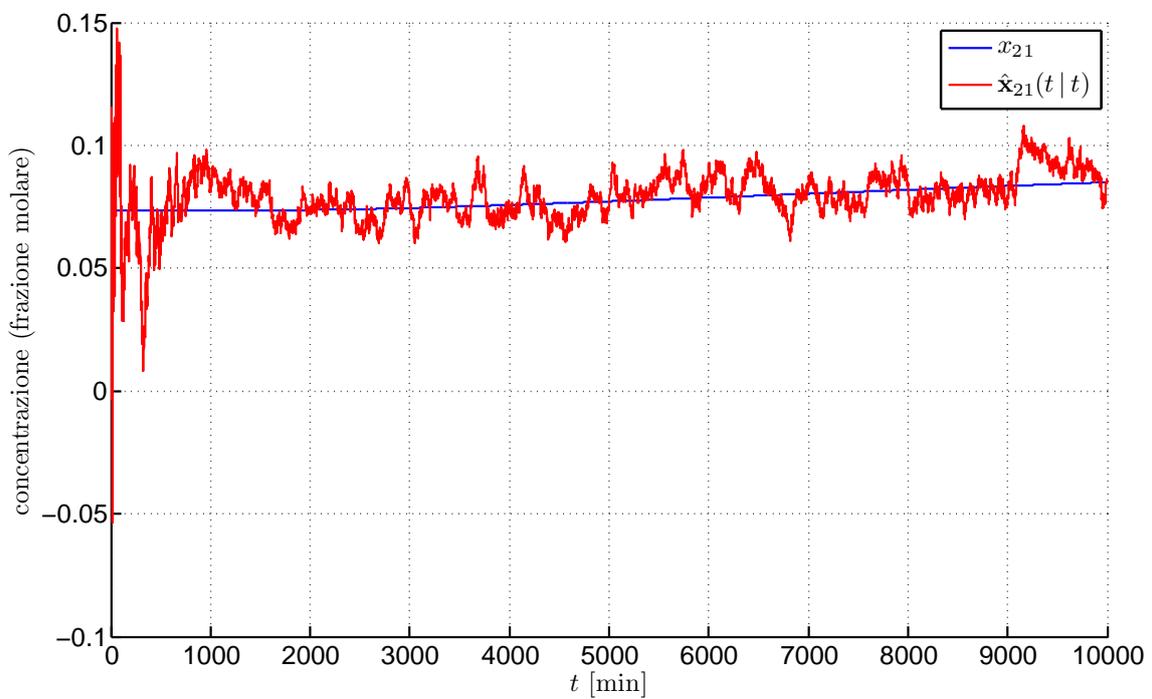


Figura 18: Confronto fra andamento di x_{21} (ribollitore) e la sua stima.

Appendici

A Codice MATLAB

Sono riportati i listati dei programmi implementati più importanti: le restanti funzioni, compreso lo script per produrre i grafici (*Tesina.m*) sono nell'archivio allegato.

A.1 Routine colamod_tb

```
function xprime=colamod_tb(t,X,U)
%
% Modifica dell'originale colamod di Skogestad con riordinamento top-bottom
% by Fabio Maran: i commenti originali sono stati lasciati per una miglior
% comprensione
%
% colamod_tb - This is a nonlinear model of a distillation column with
%               NT-1 theoretical stages including a reboiler (stage NT) plus a
%               total condenser ("stage" 1). The liquid flow dynamics are
%               modelled by a simple linear relationship.
%               Model assumptions: Two components (binary separation); constant
%               relative volatility; no vapor holdup; one feed and two products;
%               constant molar flows (same vapor flow on all stages);
%               total condenser
%
%               The model is based on column A in Skogestad and Postlethwaite
%               (1996). The model has 21 states.
%
% Inputs:      t    - time in [min].
%              X    - State, the first 21 states are compositions of light
%                    component A with reboiler/bottom stage as X(21) and
%                    condenser as X(1). State X(42) is holdup in reboiler/
%                    bottom stage and X(22) is hold-up in condenser.
%              U(1) - reflux L,
%              U(2) - boilup V,
%              U(3) - top or distillate product flow D,
%              U(4) - bottom product flow B,
%              U(5) - feed rate F,
%              U(6) - feed composition, zF.
%              U(7) - feed liquid fraction, qF.
%
% Outputs:     xprime - vector with time derivative of all the states
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Number of stages (including reboiler and total condenser):
%   NT=21;
% Location of feed stage (stages are counted from the bottom):
%   NF=11;
% Relative volatility
%   alpha=1.5;
% Nominal liquid holdups
%   M0(1)=32.1;           % Nominal condenser holdup (kmol)
%   i=2:NT-1; M0(i)=0.5*ones(1,NT-2); % Nominal stage (tray) holdups (kmol)
%   M0(NT)=10;          % Nominal reboiler holdup (kmol)
% Data for linearized liquid flow dynamics (does not apply to reboiler and condenser):
%   taul=0.063;         % time constant for liquid dynamics (min)
%   F0=1;              % Nominal feed rate (kmol/min)
%   qF0 = 1;          % Nominal fraction of liquid in feed
%   L0=2.706;         % Nominal reflux flow (from steady-state data)
%   L0b=L0 + qF0*F0;  % Nominal liquid flow below feed (kmol/min)
```

```

lambda=0;          % Effect of vapor flow on liquid flow ("K2-effect")
V0=3.206;V0t=V0+(1-qF0)*F0;% Nominal vapor flows - only needed if lambda is nonzero
%-----

% Splitting the states
x=X(1:NT)';          % Composizione liquida dall'alto al basso
M=X(NT+1:2*NT)';    % Massa totale dall'alto al basso

% Inputs and disturbances
LT = U(1);          % Reflux
VB = U(2);          % Boilup
D = U(3);           % Distillate
B = U(4);           % Bottoms

F = U(5);           % Feedrate
zF = U(6);          % Feed composition
qF = U(7);          % Feed liquid fraction

% THE MODEL

% Vapor-liquid equilibria
% modificati gli indici per andare dal piatto sotto il condensatore al
% ribollitore
i=2:NT;      y(i)=alpha*x(i)./(1+(alpha-1)*x(i));

% Vapor Flows assuming constant molar flows
% modificati gli indici per andare dal piatto sotto il condensatore al
% ribollitore
i=2:NT;      V(i)=VB*ones(1,NT-1);
i=2:Nf;      V(i)=V(i) + (1-qF)*F;

% Liquid flows assuming linearized tray hydraulics with time constant taul
% Also includes coefficient lambda for effect of vapor flow ("K2-effect").
% modificati gli indici per andare dal feed al piatto sopra il ribollitore
i=Nf:NT-1;    L(i) = L0b + (M(i)-M0(i))./taul + lambda.*(V(i-1)-V0);
% modificati gli indici per andare dal piatto sotto il condensatore al feed
i=2:Nf-1;    L(i) = L0 + (M(i)-M0(i))./taul + lambda.*(V(i-1)-V0t);
% modificato l'indice per tener conto del riflusso LT
L(1)=LT;

% Time derivatives from material balances for
% 1) total holdup and 2) component holdup

% Column
i=2:NT-1;
dMdt(i) = L(i-1)          - L(i)          + V(i+1)          - V(i);
dMxdt(i) = L(i-1).*x(i-1) - L(i).*x(i) + V(i+1).*y(i+1) - V(i).*y(i);

% Correction for feed at the feed stage
% The feed is assumed to be mixed into the feed stage
dMdt(Nf) = dMdt(Nf) + F;
dMxdt(Nf) = dMxdt(Nf) + F*zF;

% Reboiler (assumed to be an equilibrium stage)
% modificati gli indici per far in modo che il ribollitore sia l'ultimo
% piatto
dMdt(NT) = L(NT-1)          - V(NT)          - B;
dMxdt(NT) = L(NT-1).*x(NT-1) - V(NT).*y(NT) - B*x(NT);

% Total condenser (no equilibrium stage)
% modificati gli indici per far in modo che il condensatore sia il primo
% piatto
dMdt(1) = V(2)          - LT          - D;

```

```

dMxdt(1)= V(2)*y(2) - LT*x(1) - D*x(1);

% Compute the derivative for the mole fractions from d(Mx) = x dM + M dx
i=1:NT;
dxdt(i) = (dMxdt(i) - x(i).*dMdt(i) )./M(i);

% Output
xprime=[dxdt';dMdt'];

```

A.2 Routine cola_lv_tb

```

function xprime=cola_lv_tb(t,X)
%
% Modifica dell'originale cola_lv di Skogestad con riordinamento top-bottom
% by Fabio Maran: i commenti originali sono stati lasciati per una miglior
% comprensione
%
% cola_lv_tb - Subroutine for simulation with LV-configuration.
% It calls the model colamod.tb, and
% includes control of condenser and reboiler level
% using two P-controllers with the LV-configuration.
%
% Inputs are reflux (LT) and boilup (VB). Disturbances
% are feedrate and feed composition. These are set by directly
% altering 'cola_lv_tb.m'. Outputs are liquid composition and
% liquid hold up for stages 1 through NT, given in x.

% Number of stages in the column
NT=21;

% Inputs and disturbances
LT=2.70629; % Reflux
VB=3.20629; % Boilup
F=1.0 + 0.00; % Feedrate
zF=0.5; % Feed composition
qF=1.0; % Feed liquid fraction

% P-Controllers for control of reboiler and condenser hold up.
KcB=10; KcD=10; % controller gains
MDs=32.1; MBs=10; % Nominal holdups - these are rather small
Ds=0.5; Bs=0.5; % Nominal flows
% modificati gli indici per prendere i valori corretti di massa corrente al
% condensatore e al ribollitore
MB=X(2*NT); MD=X(NT+1); % Actual reboiler and condenser holdup
D=Ds+(MD-MDs)*KcD; % Distillate flow
B=Bs+(MB-MBs)*KcB; % Bottoms flow

% Store all inputs and disturbances
U(1)=LT; U(2)=VB; U(3)=D; U(4)=B; U(5)=F; U(6)=zF; U(7)=qF;

xprime=colamod.tb(t,X,U);

```

A.3 Routine distillatore

```

function xdot = distillatore(xu);
%
% Modifica dell'originale dist_dyn di Bequette per poter essere usata in
% Simulink e con variazioni su L, V , F, zF diverse da gradini, e con
% parametri memorizzati nella funzione e non in un vettore globale esterno,
% by Fabio Maran: i commenti originali sono stati lasciati per una miglior
% comprensione
%
% solve for the transient stage compositions in an ideal
% binary distillation column using odeXX.

```

```

%
% (c) 1997 B. Wayne Bequette - 24 Jan 1997
% revised 31 Dec 96
% modified for using with Simulink by Fabio Maran in 2007
%
% All flowrates are molar quantities. Stages are numbered
% from the top down. A total condenser is assumed.
% The overhead receiver is stage 1. The partial reboiler
% is stage ns (the number of equilibrium "trays" is then
% ns-1).
%
% parametri della colonna
alpha = 1.5;
ns = 21;
nf = 11;
feedi = 1;
zfeedi = 0.5;
qf = 1;
refluxi = 2.706;
vapori = 3.206;
md = 32.1;
mb = 10;
mt = 0.5;

%
% i 4 stati aggiuntivi sono in realtà i valori istantanei delle variazioni
% nel 4 parametri L, V, zF, F
% xu(ns+1) = DELTA.L
% xu(ns+2) = DELTA.V
% xu(ns+3) = DELTA.zF
% xu(ns+4) = DELTA.F
%
reflux = refluxi + xu(ns+1);

vapor = vapori + xu(ns+2);

zfeed = zfeedi + xu(ns+3);

feed = feedi + xu(ns+4);

%
% rectifying and stripping section liquid flowrates
%
lr = reflux;
ls = reflux + feed*qf;

%
% rectifying and stripping section vapor flowrates
%
vs = vapor;
vr = vs + feed*(1-qf);

%
% distillate and bottoms rates
%
dist = vr - reflux;
lbot = ls - vs;

%
if dist < 0
    disp('error in specifications, distillate flow < 0')
    return
end
if lbot < 0

```

```

        disp('error in specifications, stripping section ')
        disp(' ')
        disp('liquid flowrate is negative')
        return
    end
%
% zero the function vector
%
    xdot = zeros(ns,1);
%
% calculate the equilibrium vapor compositions
%
    for i=1:ns;
        y(i)=(alpha*xu(i))/(1.+(alpha-1.)*xu(i));
    end
%
% material balances
%
% overhead receiver
%
    xdot(1)=(1/md)*(vr*y(2)-(dist+reflux)*xu(1));
%
% rectifying (top) section
%
    for i=2:nf-1;
        xdot(i)=(1/mt)*(lr*xu(i-1)+vr*y(i+1)-lr*xu(i)-vr*y(i));
    end
%
% feed stage
%
    xdot(nf) = (1/mt)*(lr*xu(nf-1)+vs*y(nf+1)-ls*xu(nf)-vr*y(nf)+feed*zfeed);
%
% stripping (bottom) section
%
    for i=nf+1:ns-1;
        xdot(i)=(1/mt)*(ls*xu(i-1)+vs*y(i+1)-ls*xu(i)-vs*y(i));
    end
%
% reboiler
%
    xdot(ns)=(1/mb)*(ls*xu(ns-1)-lbot*xu(ns)-vs*y(ns));

```

A.4 Routine linearization_lv

```

function [f,g] = linearization_lv(x,u)

% funzione che valuta il valore di f(x,u) e g(x,u), dove f e g sono le
% funzioni che descrivono la dinamica della colonna di distillazione,
% xdot=f(x,u), y=g(x,u); si usa il modello a massa costante di Bequette
% modificato (distillatore_mod.m), le uscite misurate y sono le temperature ai
% piatti 2, NF, N-1

NT = length(x);
NF = 11;

% Calcolo f
xprime = distillatore_mod(x,u);
f = xprime';

% Calcolo g
T_bL = 341.9;
T_bH = 355.4;

g = [x(2)*T_bL+T_bH*(1-x(2)), x(NF)*T_bL+T_bH*(1-x(NF)), x(NT-1)*T_bL+T_bH*(1-x(NT-1))];

```

A.5 Routine Kalman_filter

```
function stime=Kalman_filter(A,B,C,Q,R,x0,P0,u,y)
% Routine per il calcolo della stima di un passo tempo-variante a partire
% dalle osservazioni y()
%
% INPUT : matrice di sistema A
%         matrice ingresso-stato B
%         matrice stato-uscita C
%         matrice varianza Q del rumore di modello v()
%         matrice varianza R del rumore di misura w()
%         condizioni iniziali x0 (stato), P0 (varianza dell'errore)
%         vettore degli ingressi u
%         vettore delle osservazioni y
%
% OUTPUT : vettore delle stime x(t|t)
%
% ipotesi: rumori di modello e di misura scorrelati
%

[n,m]=size(A);
N = size(y,2);

% Inizializzazione del vettore di stima e delle condizioni iniziali
x=x0;
P=P0;
stime=[];

% Ciclo che realizza l' algoritmo
for i=1:N
    % calcolo preliminare delle matrici necessarie ad ogni passo
    Lambda=C*P*C'+R;
    L=P*C'*inv(Lambda);

    % aggiornamento della stima x(t|t)
    x=x+L*(y(:,i)-C*x);
    stime=[stime x];

    % aggiornamento della varianza dell'errore di stima P(t|t)
    P=(eye(n)-L*C)*P*(eye(n)-L*C)'+L*R*L';

    % aggiornamento della predizione x(t+1|t)
    x=A*x+B*u(:,i);

    % aggiornamento della varianza dell'errore di predizione P(t+1|t)
    P=A*P*A'+Q;
end
```

A.6 Script kalman

```
% Script che setta i valori per la stima per il modello usato

clear all;

% dati della colonna (a regime)
alpha = 1.5;           % relative volatility
ns     = 21;           % total number of stages
nf     = 11;           % feed stage
feed   = 1;           % feed flowrate
zfeed  = 0.5;          % feed composition, light comp
qf     = 1;           % feed quality (1 = sat'd liqd,
                    %                   0 = sat'd vapor)
reflux = 2.706;        % reflux flowrate
vapor  = 3.206;        % reboiler vapor flowrate
```

```

md      = 32.1;          % distillate molar hold-up
mb      = 10;           % bottoms molar hold-up
mt      = 0.5;          % stage molar hold-up

% calcolo condizioni di regime sul modello non lineare (punto di equilibrio
% attorno cui linearizzare)
sim('massa_costante');
x_in = x1(end,:);
x_in = x_in';
% calcolo delle temperature a regime
T_bL = 341.9;
T_bH = 355.4;
T_in = T_bL*x_in+(1-x_in)*T_bH;

% calcolo del comportamento del sistema per variazioni a regime del 10% del
% flusso F
sim('modello_kalman');

% linearizzazione, discretizzazione e estrazione delle matrici fondamentali
[Ac,Bc,Cc,Dc] = cola_linearize('linearization_lv',x_in,[reflux vapor zfeed feed]);
Tc = 0.001;
[A,B,C,D] = ssdata(c2d(ss(Ac,Bc,Cc,Dc),Tc,'zoh'));
E = B(:,4);
B = B(:,1:2);
D = D(:,1:2);

n = size(A,1);

sigma2w = 1;

% calcolo del rumore di misura
w = sqrt(sigma2w)*randn(3,size(x_k,1));

% ingressi del sistema linearizzato: nulli perchè si considerano le
% variazioni attorno al punto di lavro, che non ci sono su L e V
u = [zeros(1,size(x_k,1)); zeros(1,size(x_k,1))];
% misure del sistema linearizzato: si sommano dei rumori bianchi per
% simulare il rumore di misura, e si sottraggono le temperature di regime
% perchè le uscite del modello linearizzato sono le differenze rispetto
% tali valori
y = [(T_bL*x_k(:,2)+(1-x_k(:,2))*T_bH)'+w(1,:); (T_bL*x_k(:,11)+(1-x_k(:,11))*T_bH)'+w(2,:); ...
      (T_bL*x_k(:,20)+(1-x_k(:,20))*T_bH)'+w(3,:)];
y = [y(1,:)-T_in(2); y(2,:)-T_in(11); y(3,:)-T_in(20)];

% varianza di v1(t)
sigma2v1 = 1.5e-6;
Q1 = sigma2v1*eye(n);

% varianza della passeggiata aleatoria F
sigma2v = .3;

% varianza di w(t)
R = sigma2w*eye(3);

% calcolo delle matrici del sistema aumentato
A_ex = [A E; zeros(1,n) 1];
B_ex = [B; zeros(1,2)];
C_ex = [C zeros(3,1)];

% varianza di v(t)
Q = [Q1 zeros(n,1); zeros(1,n) sigma2v];

% condizioni iniziali

```

```
x0 = [zeros(1,n+1)]';  
P0 = [0.1*eye(n) zeros(n,1); zeros(1,n) sigma2v];  
  
% calcolo delle stime  
stime = Kalman_filter(A_ex,B_ex,C_ex,Q,R,x0,P0,u,y);
```

Riferimenti bibliografici

- [1] http://www.nt.ntnu.no/users/skoge/book/matlab_m/cola/cola.html, pagina con i sorgenti MATLAB di Skogestad usati.
- [2] http://www.rpi.edu/~bequeb/Process_Dynamics/MATLAB/mod_10, pagina con i sorgenti MATLAB di Bequette usati.
- [3] S. Skogestad, *Dynamics and control of distillation columns: A tutorial introduction*, 1997
- [4] Dispense dal corso di Controllo dei Processi, a.a. 2006/2007, reperibili al sito http://www.dei.unipd.it/~picci/Controllo_dei_Processi.html
- [5] G. Picci, *Filtraggio statistico (Wiener, Levinson, Kalman) e applicazioni*, Edizioni Progetto, 2004.