

CAPITOLO 4

Implementazione degli algoritmi.

4.1. Introduzione.

In questo capitolo sarà discussa l'implementazione software per l'ambiente MATLAB 6.1 che è stata fatta degli algoritmi di identificazione presentati nel Capitolo 3.

Saranno presentate, sia per l'algoritmo di base che per quello che usa la tecnica delle funzioni modulanti, le funzioni utilizzate per l'implementazione e saranno spiegate in dettaglio le procedure da seguire per identificare i sistemi.

4.2. Algoritmo di base.

4.2.1. Presentazione delle funzioni.

In questo paragrafo saranno descritte le funzioni che sono state usate per implementare l'algoritmo di identificazione di base. Per ciascuna sarà spiegato il significato dei parametri di ingresso e di uscita ed il suo funzionamento. Per il codice completo delle funzioni si veda l'Appendice A.

- ***step1:***

- *Parametri di ingresso.*

1. y : vettore contenente le uscite del sistema campionate con periodo $h/2$
2. m : grado dell'approssimazione polinomiale della $f^{-1}(y)$
3. n : ordine della $G(z)$

- *Parametri di uscita.*

1. $Dphi$: è la matrice $\Delta\phi_1$ di (3.23)
2. Dy : è il vettore Δy di (3.23)
3. $THETA$: è il vettore soluzione del sistema (3.23).

- *Funzionamento.*

La funzione *step1* implementa il primo passo dell'algoritmo di identificazione, cioè l'identificazione della non linearità di uscita e del denominatore della funzione di trasferimento lineare. La funzione costruisce la matrice $\Delta\phi_1$ ed il vettore Δy (che vengono anche restituite) come spiegato nel paragrafo 3.3, usando anche la funzione *buildphi*, e risolve il sistema lineare (3.23). Il sistema viene risolto applicando a $\Delta\phi_1$ e Δy l'operatore “\” che, nel caso di sistema sovradeterminato, lo risolve nel senso dei minimi quadrati.

- ***buildphi:***

- *Parametri di ingresso.*

1. y : vettore contenente le misure dell'uscita, con periodo di campionamento h , per $n+1$ periodi di campionamento consecutivi e in ordine cronologico inverso; se il primo elemento è $y[kh]$, allora l'ultimo elemento dovrà essere $y[kh-nh]$.
 2. m : grado dell'approssimazione polinomiale della $f^{-1}(y)$
 3. n : ordine della $G(z)$
- Parametri di uscita.
1. ϕ : se il primo elemento di y è $y[kh]$, ϕ è il vettore riga $\phi_1[kh]$.
- Funzionamento.
- La funziona *buildphi* costruisce il vettore $\phi_1[kh]$, come descritto in (3.15), a partire dalle misure dell'uscita y su un opportuno intervallo temporale.
- ***find_r_alpha:***
- Parametri di ingresso.
1. θ : vettore soluzione del sistema (3.23)
 2. m : grado dell'approssimazione polinomiale della $f^{-1}(y)$
 3. n : ordine della $G(z)$
- Parametri di uscita.
1. r : vettore contenente una stima dei coefficienti polinomiali dell'inversa della non linearità di uscita ordinati da quelli di grado 1 a quello di grado massimo (il coefficiente di grado 0 è sempre nullo)
 2. α : vettore contenente una stima dei coefficienti del denominatore della $\tilde{G}(z)$ in ordine crescente.
- Funzionamento.
- A partire dal vettore θ_1 , soluzione del sistema (3.23), ricava i coefficienti polinomiali della non linearità di uscita ed il denominatore della $\tilde{G}(z)$ utilizzando le relazioni espresse in (3.31).
- ***step2:***

□ Parametri di ingresso.

1. y_2 : vettore contenente le uscite del sistema campionate con periodo $h/2$
2. n : ordine della $G(z)$
3. p : vettore contenente i coefficienti polinomiali dell'inversa della non linearità di uscita (in un formato compatibile con la funzione *polyval* di MATLAB)

□ Parametri di uscita.

1. θ_2 : vettore soluzione del sistema lineare derivato dalla (3.42).

□ Funzionamento.

La funzione *step2* implementa il secondo passo dell'algoritmo di identificazione, cioè l'identificazione del numeratore della parte lineare del sistema Hammerstein-Wiener. Vengono costruite tante equazioni, come descritto in (3.42), quante ne permette la durata della misurazione dell'uscita (usando la funzione *buildphi2* per calcolare i vari vettori $\phi_2[k]$) e viene risolto, nel senso dei minimi quadrati, il sistema così ottenuto.

• ***buildphi2*:**

□ Parametri di ingresso.

1. x_{cap} : vettore contenente le stime del segnale x campionate con periodo $h/2$
2. k : numero della riga della matrice ϕ_2 che si vuole generare
3. n : ordine della $G(z)$

□ Parametri di uscita.

1. ϕ_i : è il vettore riga $\phi_2[k]$, k -esima riga della matrice ϕ_2 del sistema lineare che viene costruito nella funzione *step2*
2. x_c : vettore contenente una stima di $x[kT+T/2]$

□ Funzionamento.

La funzione *buildphi2* genera un'equazione lineare come in (3.42), calcolando il vettore riga $\phi_2[k]$ ed il termine noto associato $x[kT+T/2]$, a partire da una stima del segnale x .

- **step3:**

- Parametri di ingresso.

1. $y2$: vettore contenente le uscite del sistema campionate con periodo $h/2$
2. w : vettore contenente gli ingressi del sistema campionate con periodo $h/2$
3. n : ordine della $G(z)$
4. p : vettore contenente i coefficienti polinomiali dell'inversa della non linearità di uscita (in un formato compatibile con la funzione *polyval* di MATLAB)
5. $alpha$: vettore contenente i coefficienti del denominatore della $G(z)$ ordinati per potenze crescenti di z^{-1} (e con $alpha(1) = 1$)
6. $beta$: vettore contenente i coefficienti del numeratore della $G(z)$ ordinati per potenze crescenti di z^{-1} (con $beta(1) = 0$ e $beta(2) = 1$)
7. $betasegnato$: vettore contenente i coefficienti del numeratore della $\overline{G}(z)$ ordinati per potenze crescenti di z^{-1}

- Parametri di uscita.

1. $w2$: vettore contenente i valori del segnale di ingresso w campionati con periodo T
2. $ucap$: vettore contenente stime delle $u[kT]$; ogni coppia $(w2(i), ucap(i))$ è formata dal valore di w e di u nel medesimo istante e costituisce una stima della $g(w)$

- Funzionamento.

La funzione *step3* implementa il terzo passo dell'algoritmo di identificazione, cioè l'identificazione della non linearità all'ingresso del sistema. La funzione verifica se almeno una delle due tra $G(z)$ e $\overline{G}(z)$ è a fase minima, ed in questo caso stima le $u[kT]$ in accordo con la (3.46) o con la (3.47). Se nessuna delle due è a fase minima allora la funzione termina riferendo che si è verificato un errore.

- **inverti:**

- Parametri di ingresso.
 1. *v*: un generico vettore
- Parametri di uscita.
 1. *res*: vettore risultato della funzione
- Funzionamento.

Restituisce un vettore che ha le stesse componenti del vettore *v*, ma in ordine inverso.

4.2.2. Procedura di identificazione.

Per identificare un sistema Hammerstein-Wiener con l'algoritmo proposto occorre usare le funzioni descritte al paragrafo 4.2.1, ed in particolare le funzioni *step1*, *step2* e *step3*. E' stata sviluppata, mediante uno *script* MATLAB (il cui codice è riportato in Appendice A), una procedura che, quando avviata, identifica automaticamente il sistema se nello workspace corrente sono presenti determinate variabili. In Figura 4.1 sono schematizzate le azioni che vengono eseguite dalla procedura, mettendo in particolare evidenza il flusso dei dati nel corso della procedura stessa.

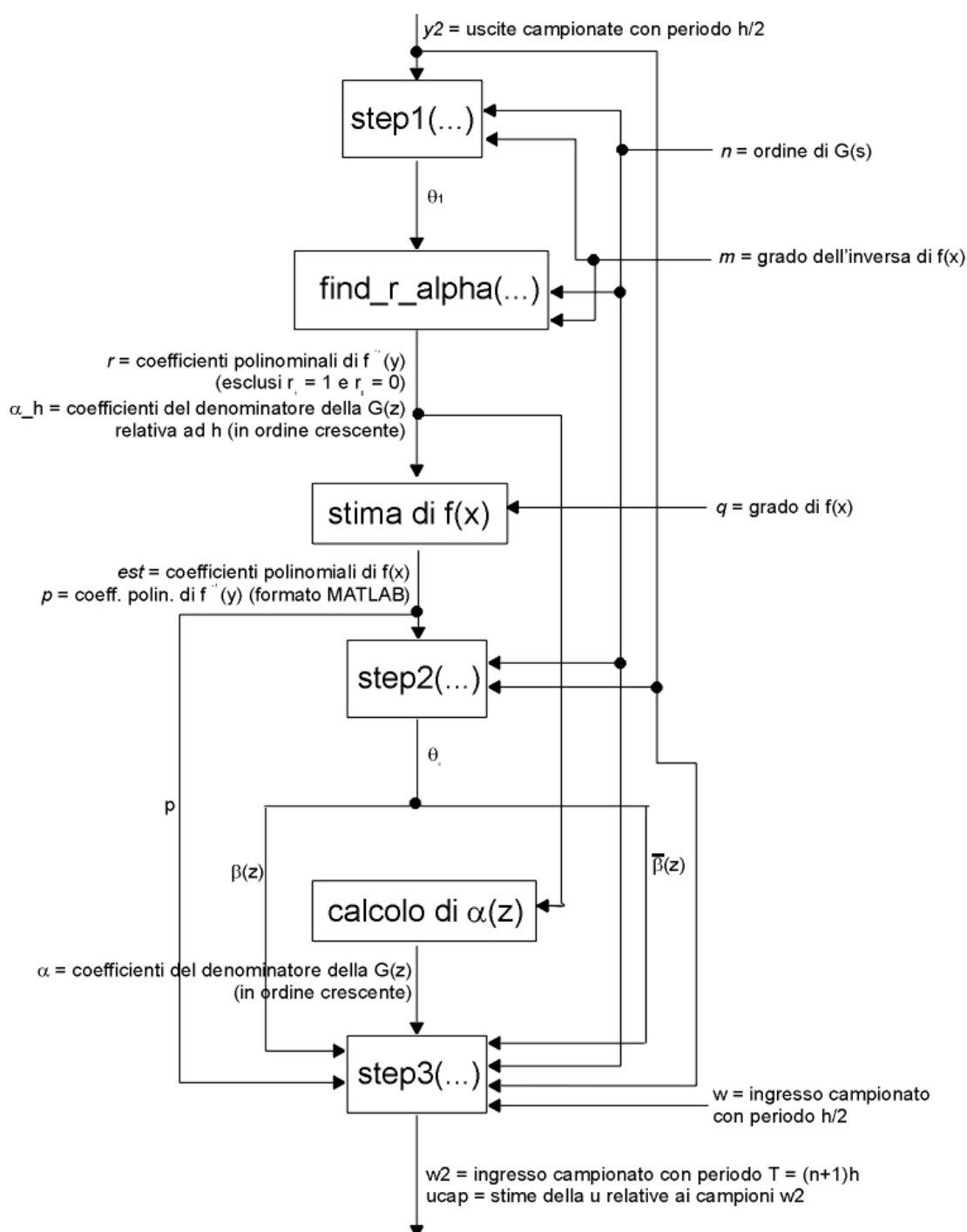


Figura 4.1. Procedura di identificazione

Nel momento in cui la procedura viene attivata nello workspace corrente di MATLAB devono essere presenti le seguenti variabili:

1. $y2$: vettore contenente le uscite del sistema campionate con periodo $h/2$.
Questo periodo di campionamento è stato scelto per fare in modo che in una unica misurazione dell'uscita comparissero i valori che assume l'uscita stessa sia negli istanti kh , sia negli istanti $kT+T/2$.
2. w : vettore contenente gli ingressi del sistema campionate con periodo $h/2$.
3. m : grado dell'approssimazione polinomiale della $f^{-1}(y)$
4. n : ordine della $G(z)$
5. q : grado della $f(x)$
6. xx, yy (*facoltative*): vettori contenenti coppie di punti $\{x, y\}$ (gli x in xx , gli y in yy) che caratterizzano la non linearità di uscita del sistema; queste variabili devono essere presenti solo se la n.l. di uscita è nota e si vuole che lo *script* faccia un confronto grafico con quella stimata.
7. T (*facoltativa*): periodo di campionamento per l'ingresso, consente alla procedura di creare nello workspace una variabile contenente la f.d.t. T.D. stimata nel corso dell'identificazione.

Il primo passo che deve essere fatto dalla procedura è generare e risolvere il sistema (3.23), chiamando la funzione *step1*. Una volta risolto il sistema, ottenendo il vettore θ_1 , occorre ricavare i coefficienti polinomiali della $f^{-1}(y)$ e del denominatore della f.d.t. $\tilde{G}(z)$, relativa al periodo di campionamento h . A questo scopo viene chiamata la funzione *find_r_alpha*, che calcola i suddetti coefficienti sfruttando tutte le informazioni contenute nel vettore θ_1 . A questo punto la procedura crea un vettore contenente i coefficienti della $f^{-1}(y)$ nel formato standard di MATLAB e stima i coefficienti della $f(x)$ a partire da quelli della $f^{-1}(y)$. Per fare questo si individua l'intervallo massimo all'interno del quale varia la y con il segnale di ingresso usato, si suddivide tale intervallo in un numero sufficientemente elevato di sottointervalli e, per ciascun punto y_i che determina i sottointervalli, si calcola $x_i = f^{-1}(y_i)$; si ricava

poi il polinomio di grado q che meglio approssima le coppie $\{x_i, y_i\}$ nel senso dei minimi quadrati, costruendo la matrice di Vandermonde relativa ai punti x_i (senza la colonna per il grado 0, avendo ipotizzato che $f(x)$ passa per l'origine) e usando come vettore dei termini noti un vettore colonna contenente i vari y_i . Se nello workspace sono presenti le variabili xx e yy , viene poi effettuato un confronto grafico tra la non linearità di uscita stimata e quella reale.

La seconda fase della procedura di identificazione consiste nel ricavare i parametri della parte lineare. Il primo passo è chiamare la funzione *step2*, che ricava il vettore θ_2 come descritto nel paragrafo 3.4. Da questo si ricavano immediatamente i coefficienti dei numeratori delle funzioni di trasferimento $G(z)$ e $\bar{G}(z)$, che vengono portati in una forma compatibile con la funzione *filt* di MATLAB. Poi, a partire dai coefficienti del denominatore della $\tilde{G}(z)$, si ricavano i poli della $G(z)$ come mostrato in (3.33) e da questi i coefficienti del denominatore della $G(z)$, anch'essi riportati in un vettore in una forma compatibile con la funzione *filt* di MATLAB. Infine, se nello workspace è disponibile l'ampiezza del periodo di campionamento T , la procedura vi crea una nuova variabile contenente la $G(z)$, in forma di filtro digitale.

La terza ed ultima fase della procedura di identificazione consiste nell'identificazione della non linearità di ingresso. Si utilizza la funzione *step3* per ricavare delle coppie $\{w, u\}$ che costituiscano una stima della $g(w)$; successivamente si suddivide l'intervallo, all'interno del quale si trovano le w , in un certo numero di sottointervalli e per ciascuno si calcola la u media tra le varie u associate a w dentro il sottointervallo stesso, in modo da ridurre l'effetto di rumore ed approssimazioni. Infine si mostrano graficamente i risultati di queste ultime operazioni.

Alla fine della procedura, nello workspace saranno presenti le seguenti variabili, che contengono i risultati ottenuti:

1. *dphi, dy, theta*: matrice, colonna dei termini noti e soluzione del sistema
(3.23)

2. *p*: coefficienti polinomiali stimati della $f^{-1}(y)$, ordinati da quello di grado massimo a quello di grado zero
3. *est*: coefficienti polinomiali stimati della $f(x)$, ordinati da quello di grado massimo a quello di grado zero
4. *theta2*: soluzione del sistema la cui struttura è descritta al paragrafo 3.4
5. *betasegnato*: coefficienti del numeratore della $\overline{G}(z)$, ordinati per potenze crescenti di z^{-1}
6. *beta*: coefficienti del numeratore della $G(z)$, ordinati per potenze crescenti di z^{-1}
7. *alpha*: coefficienti del denominatore di $G(z)$ e $\overline{G}(z)$, ordinati per potenze crescenti di z^{-1}
8. *Gz (non sempre)*: se nello workspace era presente una variabile denominata *T* e contenente il periodo di campionamento per l'ingresso, contiene la f.d.t. $G(z)$ stimata in formato *filt*
9. *wu*: matrice con due righe. Ciascuna colonna contiene una coppia $\{w, u\}$ dove *u* è all'incirca uguale a $g(w)$. Le coppie sono ordinate per valori crescenti di *w*.
10. *wu_media*: come *wu*, ma contiene solo 50 coppie di punti. Ciascuna coppia è ottenuta calcolando la *u* media all'interno di un sottointervallo pari a un cinquantesimo dell'intervallo $[wu(1,1), wu(size(wu,1),1)]$.

4.3. Uso della tecnica delle funzioni modulanti.

In questo paragrafo sarà discussa l'implementazione dell'algoritmo di identificazione che utilizza la tecnica delle funzioni modulanti per stimare i parametri della parte Hammerstein del sistema Hammerstein-Wiener, descritto al paragrafo 3.8.

La procedura di identificazione viene implementata in un'unica funzione principale, *hw_fun_mod*, il cui codice è riportato in Appendice A. Eccone una descrizione schematica:

- *hw_fun_mod*:

- Parametri di ingresso.

1. *np*: numero di poli della funzione di trasferimento T.C. $G(s)$
2. *nz*: numero di zeri della funzione di trasferimento T.C. $G(s)$
3. *go*: grado della non linearità di uscita
4. *gi*: grado della non linearità di ingresso
5. *m*: grado dell'approssimazione polinomiale della $f^{-1}(y)$
6. *y*: vettore contenente le uscite del sistema campionate con periodo $h/2$
7. *t*: vettore che raccoglie gli istanti di campionamento dei segnali di ingresso e di uscita del sistema che saranno usati per l'identificazione della parte Hammerstein con la tecnica delle funzioni modulanti
8. *w*: vettore contenente il segnale di ingresso del sistema Hammerstein-Wiener (campionato negli istanti riportati in *t*) usato per l'identificazione della parte Hammerstein con la tecnica delle funzioni modulanti
9. *y_fm*: vettore contenente il segnale di uscita del sistema Hammerstein-Wiener (campionato negli istanti riportati in *t*) usato per l'identificazione della parte Hammerstein con la tecnica delle funzioni modulanti
10. *T*: finestra di integrazione usata per l'identificazione della parte Hammerstein con la tecnica delle funzioni modulanti

- Parametri di uscita.

1. *p*: coefficienti polinomiali dell'inversa della non linearità di uscita identificata
2. *est*: coefficienti polinomiali della non linearità di uscita identificata
3. *num*: coefficienti del numeratore della $G(s)$ identificati
4. *den*: coefficienti del denominatore della $G(s)$ identificati
5. *nlc*: coefficienti polinomiali della non linearità di ingresso identificata

□ Funzionamento.

La funzione *hw_fun_mod* esegue la procedura necessaria per identificare un sistema Hammerstein-Wiener con l'algoritmo descritto al paragrafo 3.8. Per prima cosa viene chiamata la funzione *step1* per identificare la non linearità di uscita, esattamente come già spiegato nel paragrafo 4.2.2. A questo punto si ricava la stima del segnale x corrispondente al segnale y_{fm} utilizzando la stima di $f^{-1}(y)$ appena calcolata; a questo punto si identifica la parte Hammerstein usando questa approssimazione di x ed il relativo segnale di ingresso w chiamando la funzione *fun_mod_ham* descritta al paragrafo 2.7.

Per maggiori informazioni su come usare questa funzione e sull'identificazione con il metodo delle funzioni modulanti si faccia riferimento al paragrafo 3.8 ed al Capitolo 2.