

CAPITOLO 6: Implementazione Simulink

Per testare le tecniche di denoising viste precedentemente e per valutarne l'effettiva applicabilità ed efficacia nell'identificazione del processo di Hammerstein sono stati creati due schemi Simulink principali. In pratica, l'identificazione avviene in due stadi consecutivi: in un primo stadio si acquisiscono informazioni del segnale di uscita che saranno rilevanti per l'analisi spettrale, nella seconda fase avverrà l'identificazione vera e propria.

6.1 Primo modulo: analisi del segnale

Il primo modulo è stato creato per valutare alcune caratteristiche fondamentali del segnale in uscita dal processo di Hammerstein. In particolare in questa prima fase è possibile calcolare il periodo del segnale, il suo valor medio, nonché le N frequenze (con N impostato dall'utente) che presentano picchi più marcati nello spettro in potenza. Come già visto nel capitolo 5, le frequenze del segnale potrebbero essere dedotte direttamente dal periodo dello stesso, quindi il loro calcolo diretto potrebbe apparire superfluo. La stima delle frequenze viene invece mantenuta per verificare direttamente tale asserzione.

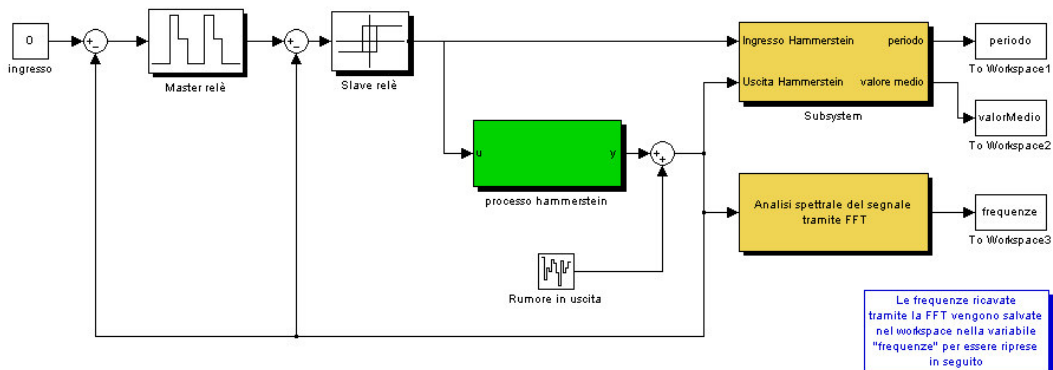


Figura 6.1. Schema Simulink del modello per l'analisi del segnale di uscita del processo di Hammerstein

Analizziamo adesso i principali blocchi dello schema.

6.1.1 Blocco Master Relè

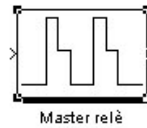


Figura 6.2. Blocco Master relè

Il blocco Master relè implementa le funzionalità del relè già descritte nel capitolo 5. All'interno della maschera del blocco è necessario tarare il relè con i valori opportuni di μ e d_1 . Si ricorda che il parametro μ permette di eliminare oscillazioni spurie dovute ai disturbi sul segnale, mentre il parametro d_1 dà l'ampiezza dei gradini in uscita dal master relè.

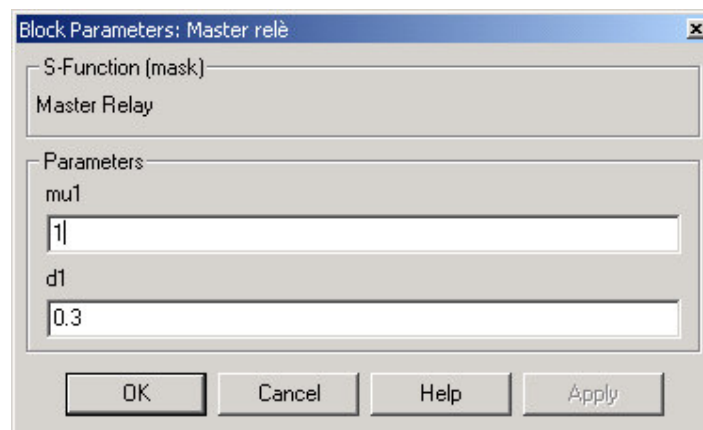


Figura 6.3. Maschera del blocco Master relè

6.1.2 Blocco Processo di Hammerstein



Figura 6.4. Blocco Processo di Hammerstein

Il blocco *Processo di Hammerstein* rappresenta il processo da identificare: per la parte riguardante la funzione di trasferimento (si ricorda che viene fatta l'ipotesi che tale funzione sia del secondo ordine più ritardo) occorre inserire i parametri a e b coefficienti del polinomio al denominatore, oltre al ritardo. Per la parte non lineare, si deve inserire la funzione che rappresenta la non linearità.

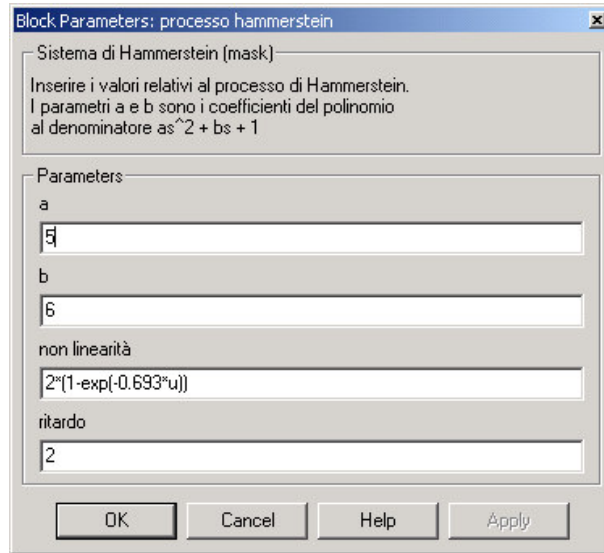


Figura 6.5. Maschera del blocco *Processo di Hammerstein*

6.1.3 Blocco Periodo e valore medio

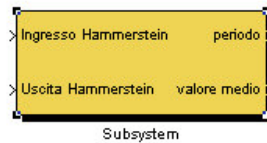


Figura 6.6. Blocco *Periodo e valore medio*

Il blocco *Periodo e valore medio* calcola il periodo ed il valore medio del segnale. Il periodo del segnale di uscita è lo stesso di quello in ingresso, facilmente calcolabile perchè tale segnale è somma di due segnali rettangolari (quello fornito dal master relè e quello fornito dallo slave relè). Una volta calcolato il periodo, il blocco interno *Periodo del segnale* (si tratta di una S-function) pone ad 1 una variabile di sincronizzazione altrimenti mantenuta sullo 0. Il blocco *Valore medio del segnale*, anch'esso S-function, una volta ricevuta l'abilitazione calcola il valore medio di tutti i campioni prelevati nel frattempo, fornendo il risultato in uscita.

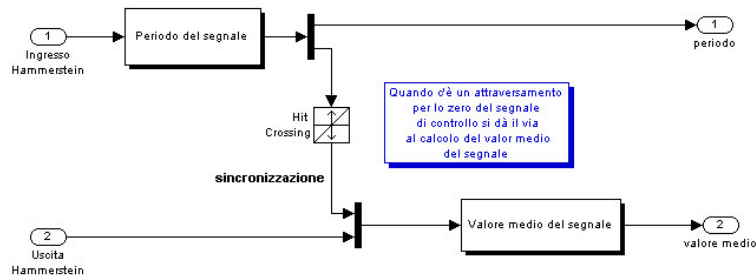


Figura 6.7. Schema interno del blocco *Periodo e valore medio*

6.1.4 Blocco Analisi spettrale



Figura 6.8. Blocco *Analisi spettrale*

Il blocco Analisi spettrale è inserito per effettuare il calcolo delle frequenze del segnale. Le frequenze del segnale in uscita sono infinite, tuttavia se ne possono riconoscere un numero limitato che approssimano molto bene il segnale stesso. Sperimentalmente per il segnale in uscita dal processo di Hammerstein con cascade relè si osserva che 8 frequenze sono sufficienti per avere una stima praticamente indistinguibile dal segnale reale. Il parametro N che indica quante frequenze considerare viene impostato dall'utente attraverso la maschera. Oltre ad N , la maschera richiede il periodo di campionamento del segnale e l'intervallo di tempo sul quale si desidera effettuare la stima frequenziale.



Figura 6.9. Maschera del blocco *Analisi spettrale*

Lo schema interno del blocco è costituito oltre che dalla parte di campionamento e memorizzazione del segnale, dalla S-function che effettua la stima frequenziale mediante FFT.

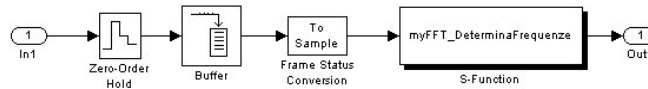


Figura 6.10. Schema interno del blocco *Analisi spettrale*

Riepilogando, questo primo schema Simulink fornisce:

1. il periodo
2. il valor medio
3. le N frequenze più rilevanti

del segnale in uscita dal processo di Hammerstein.

Le 3 variabili vengono salvate nel workspace di Matlab per essere utilizzate nello stadio successivo.

6.2 Secondo modulo: identificazione

Una volta conosciuti i parametri caratteristici del segnale, si può procedere con la fase di identificazione del processo di Hammerstein.

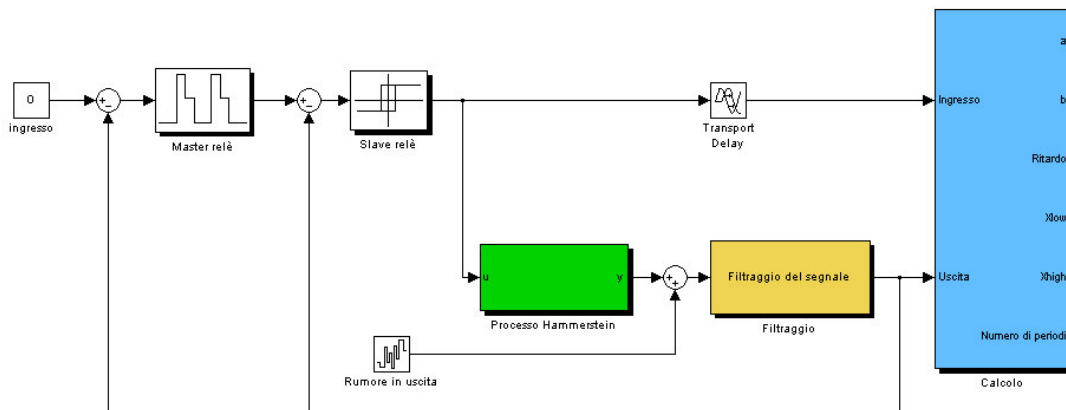


Figura 6.11. Stadio di identificazione del processo di Hammerstein

Lo stadio di identificazione è costituito essenzialmente dai due relè master e slave, dal processo da identificare, dal blocco di filtraggio del segnale e dal blocco di identificazione vero e proprio. Sono inoltre presenti il blocco per sommare rumore all'uscita e il blocco *Transport Delay* per ritardare il segnale che perviene dallo slave relè verso il blocco di calcolo (il motivo di questo ritardo verrà chiarito nel seguito).

I blocchi Master Relè e Processo di Hammerstein sono stati esaminati nel paragrafo precedente, quindi, per evitare inutili ripetizioni, non vengono qui riesaminati.

6.2.1 Blocco Filtraggio

Il blocco *Filtraggio* rappresenta una delle componenti principali di questa tesi: all'interno di tale blocco, infatti, sono implementate tutte quelle tecniche di stima frequenziale e denoising già evidenziate nei capitoli precedenti: FFT, wavelet, MUSIC.

Ciascuno di questi algoritmi necessita di un numero variabile di parametri di ingresso, inseribili da parte dell'utente mediante la maschera del blocco. La maschera è realizzata dinamicamente, ossia rimangono visibili solo i parametri di effettivo interesse.

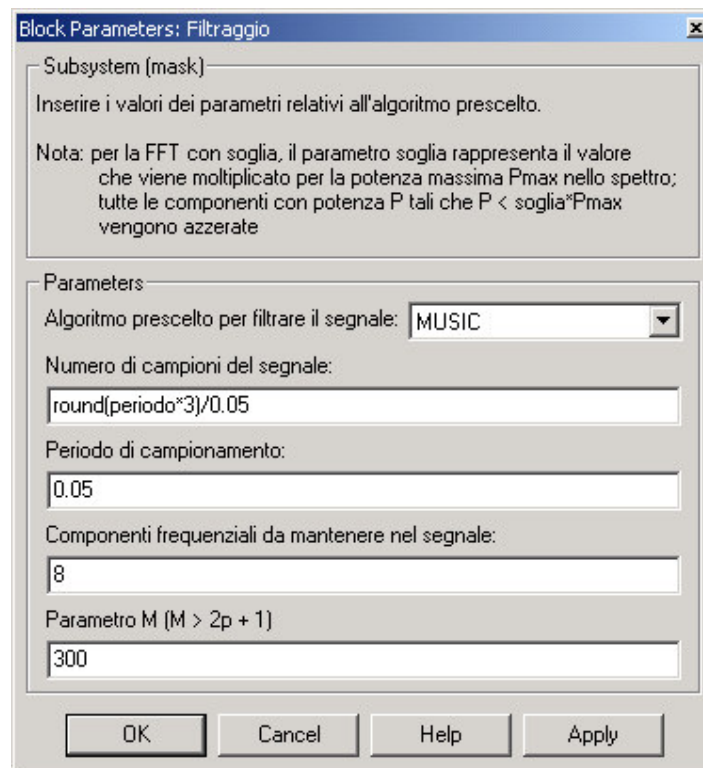


Figura 6.12. Maschera del blocco *Filtraggio*

Ciò significa che se l'algoritmo scelto è MUSIC, solo i parametri di tale algoritmo verranno richiesti.

Per capire come funzioni internamente il blocco *Filtraggio* è opportuno rifarsi allo schema interno. Lo schema è stato implementato in modo da poter risultare comprensibile a prima vista e, soprattutto, scalabile. Ciò significa che se ci fosse la necessità di introdurre un nuovo algoritmo, la procedura sarebbe possibile ed estremamente semplice.

Vediamo come si seleziona internamente un particolare algoritmo.

Supponiamo che l'utente abbia scelto quale algoritmo utilizzare per effettuare la ricostruzione del segnale; la maschera, quindi, ha memorizzato nella variabile *algoritmo* la scelta effettuata sotto forma di intero: tale intero (che risiede nell'intervallo tra il valore 1 ed il numero degli algoritmi effettivamente presenti) è ordinato con lo stesso ordine con cui si trovava l'algoritmo all'interno della casella popup della maschera (in altre parole, se viene selezionato *FFT ad N frequenze*, e questa è la prima stringa nel menu a scomparsa, allora la variabile *algoritmo* vale 1).

Per selezionare il blocco che implementa l'algoritmo prescelto le cose sono semplici: ciascun blocco è infatti un *sottosistema con abilitazione*, perciò si va ad abilitare solo il blocco desiderato ponendo come segnale di abilitazione il risultato dell'operazione di confronto tra la variabile *algoritmo* ed il valore che tale variabile avrebbe se fosse selezionato quello specifico blocco. Per selezionare l'uscita si è usato un *multiport switch* comandato dalla variabile *algoritmo*.

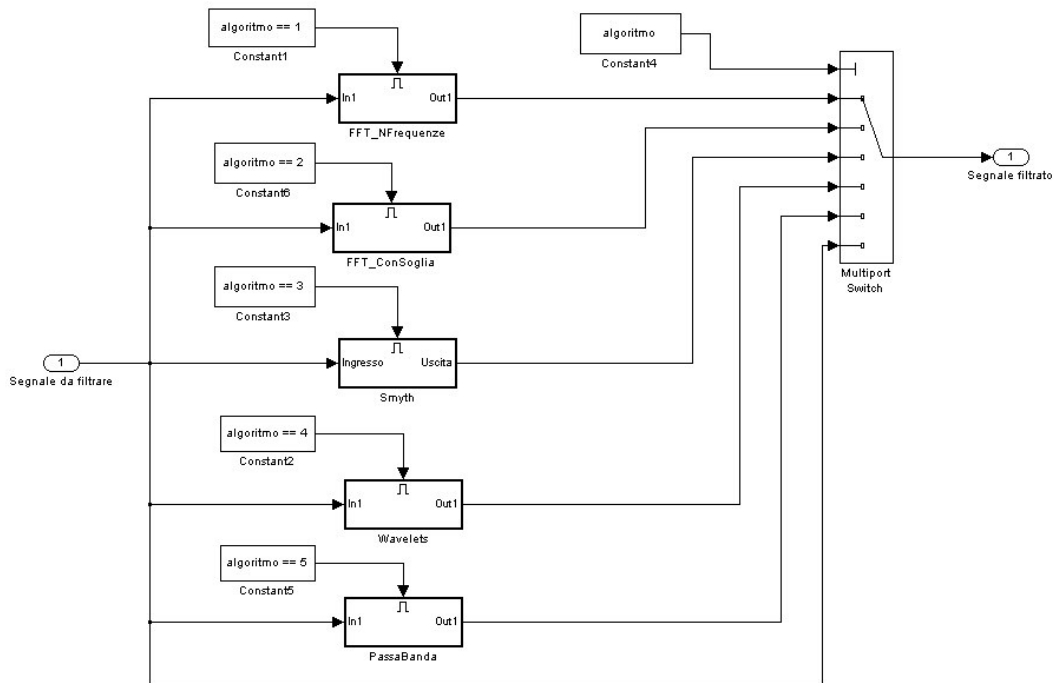


Figura 6.13. Schema interno del blocco *Filtraggio*

Ciascuna tecnica di filtraggio digitale prevede che all'avvio della simulazione vengano acquisiti un certo numero di campioni da processare. Tali campioni, quindi, non arrivano direttamente al blocco di calcolo, ma rimangono per un certo intervallo di tempo all'interno di qualche buffer; la conseguenza sarebbe che, senza il ritardo a cui si è accennato in precedenza, il valore del ritardo percepito dal blocco di calcolo sarebbe pari al ritardo reale del processo di Hammerstein sommato con il periodo di tempo in cui i campioni rimangono all'interno del buffer relativo ad un blocco di filtraggio. Inserire un ritardo tra il segnale in uscita dallo slave relè e lo stesso segnale in ingresso al blocco di calcolo evita che tutto ciò accada (il ritardo da introdurre artificialmente sarà quindi pari al tempo di transito del segnale dal blocco di filtraggio, ossia pari al numero di campioni del segnale moltiplicato per il periodo di campionamento).

6.2.2 Blocco Identificazione

Il blocco di identificazione implementa la tecnica di identificazione già illustrata nel capitolo precedente. Alcune considerazioni sono però necessarie.

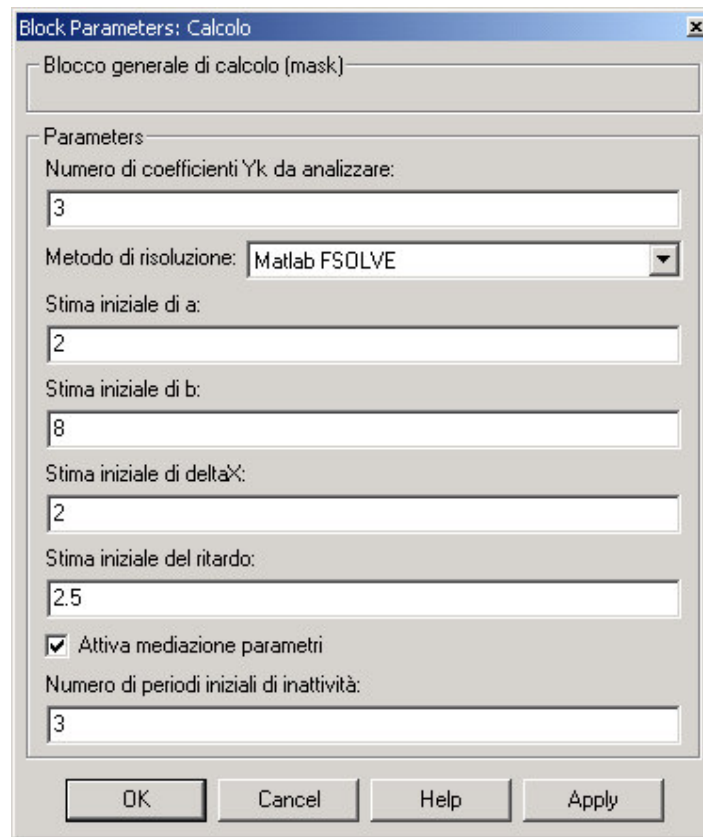


Figura 6.14. Maschera del blocco Calcolo

Anzitutto, anche questa maschera è dinamica, quindi in figura si vede solo una possibile schematizzazione (quella relativa al metodo di risoluzione *Matlab FSOLVE*).

Nella casella *Numero di coefficienti Yk da analizzare* si inserisce il numero di componenti armoniche da elaborare, componenti riferite ovviamente al segnale in ingresso al blocco di calcolo proveniente dal blocco di filtraggio.

I metodi di risoluzione proposti per il sistema non lineare presentato in (5.2.13) sono due:

1. mediante la funzione Matlab *fsolve*
2. cercando di linearizzare il sistema e risolvendo ai minimi quadrati.

Nel secondo caso le caselle relative alle stime dei parametri a , b , ritardo e ΔX si riferiscono quindi al caso di soluzione con *fsolve*; nel caso di soluzione con linearizzazione le caselle evidenziate sono relative al ritardo massimo e minimo da considerare ed al relativo intervallo di quantizzazione desiderato.

Il numero di periodi iniziali di attività è una quantità (maggiore o uguale a zero) che stabilisce dopo quanti periodi del segnale di uscita si debba iniziare ad effettuare l'elaborazione. Se tale valore è N , i periodi del segnale da 0 a $N-1$ non verranno elaborati: questo appare solo nel caso sia flaggata la casella *Attiva mediazione parametri* (il cui significato sarà chiaro nel seguito).

Si riportano adesso più nel dettaglio alcune considerazioni appena introdotte.

6.2.2.1 Soluzione mediante *fsolve*

Si può decidere di far eseguire il calcolo della soluzione del sistema non lineare direttamente con la funzione *fsolve*, che automaticamente sceglie l'algoritmo più adatto. Il prototipo di tale funzione è:

```
risultato = fsolve(@fun, parametri_iniziali, opzioni, parametri_ingresso)
```

dove:

- *fun* è la funzione che restituisce il sistema non lineare di cui calcolare la soluzione
- *parametri_iniziali* rappresenta il vettore dei valori iniziali per le N variabili definite in *fun*
- *opzioni* è la variabile che permette di specificare, tra le altre cose, il metodo di risoluzione del sistema
- *parametri_ingresso* rappresenta l'insieme di ulteriori variabili passate a *fun*
- *risultato* è il vettore di soluzioni

La possibilità di inizializzare a piacimento il guess iniziale per il sistema non lineare è importante perchè se la stima iniziale è lontana dai parametri reali, i parametri in uscita possono non essere consistenti: ciò accade soprattutto in presenza di forte rumore sul segnale.

6.2.2.2 Soluzione mediante linearizzazione

La soluzione mediante linearizzazione del sistema è stata implementata per valutare se fosse possibile migliorare la soluzione fornita dalla funzione *fsolve*, soprattutto nel caso di presenza di forte rumore sul segnale.

Sarebbe auspicabile riuscire a porre il sistema in una diversa forma per cercare strategie di calcolo più semplici ed in cui la soluzione finale non dipenda troppo dal guess iniziale (cosa che invece

accade con la *fsolve*). E' stato fatto il tentativo di portare il sistema in forma lineare per poi utilizzare un approccio ai minimi quadrati. Il problema però è apparso non risolvibile, per la presenza di variabili come argomenti di funzioni sinusoidali di ordine crescente con il crescere della frequenza di riferimento.

Il sistema originale:

$$\begin{cases} P_1 \cdot \omega_1^2 \cdot a - j \cdot P_1 \cdot \omega_1 + \Delta x \cdot e^{-j \cdot \text{Delay} \cdot \omega_1} - P_1 = 0 \\ P_2 \cdot \omega_2^2 \cdot a - j \cdot P_2 \cdot \omega_2 + \Delta x \cdot e^{-j \cdot \text{Delay} \cdot \omega_2} - P_2 = 0 \\ P_3 \cdot \omega_3^2 \cdot a - j \cdot P_3 \cdot \omega_3 + \Delta x \cdot e^{-j \cdot \text{Delay} \cdot \omega_3} - P_3 = 0 \end{cases}$$

è stato portato nella forma :

$$\begin{cases} P_1 \cdot \omega_1^2 \cdot a + \Delta x \cdot \cos(\text{Delay} \cdot \omega_1) - P_1 = 0 \\ P_1 \cdot \omega_1 \cdot b + \Delta x \cdot \sin(\text{Delay} \cdot \omega_1) = 0 \\ P_2 \cdot \omega_2^2 \cdot a + \Delta x \cdot \cos(\text{Delay} \cdot \omega_2) - P_2 = 0 \\ P_2 \cdot \omega_2 \cdot b + \Delta x \cdot \sin(\text{Delay} \cdot \omega_2) = 0 \\ P_3 \cdot \omega_3^2 \cdot a + \Delta x \cdot \cos(\text{Delay} \cdot \omega_3) - P_3 = 0 \\ P_3 \cdot \omega_3 \cdot b + \Delta x \cdot \sin(\text{Delay} \cdot \omega_3) = 0 \end{cases}$$

semplicemente decomponendo l'esponenziale presente in ognuna della 3 equazioni:

$$e^{-j \cdot \text{Delay} \cdot \omega_k} = \cos(\text{Delay} \cdot \omega_k) - j \cdot \sin(\text{Delay} \cdot \omega_k)$$

Consideriamo:

$$A_k = \Delta x \cdot \cos(\text{Delay} \cdot \omega_k)$$

$$B_k = \Delta x \cdot \sin(\text{Delay} \cdot \omega_k)$$

Se fosse possibile esprimere A_k e B_k , con $k \neq 1$, come combinazione lineare di A_1 e B_1 , allora sarebbe stato possibile scrivere il sistema come:

$$\begin{cases} P_1 \cdot \omega_1 \cdot a + A - P_1 = 0 \\ P_1 \cdot \omega_1 \cdot b + B = 0 \\ P_2 \cdot \omega_2 \cdot a + \alpha_1 A + \beta_1 B - P_2 = 0 \\ P_2 \cdot \omega_2 \cdot b + \alpha_2 A + \beta_2 B = 0 \\ P_3 \cdot \omega_3 \cdot a + \alpha_3 A + \beta_3 B - P_3 = 0 \\ P_3 \cdot \omega_3 \cdot b + \alpha_4 A + \beta_4 B = 0 \end{cases}$$

In forma più compatta:

$$\begin{bmatrix} P_1 \cdot \omega_1 & 0 & 1 & 0 \\ 0 & P_1 \cdot \omega_1 & 0 & 1 \\ P_2 \cdot \omega_2 & 0 & \alpha_1 & \beta_1 \\ 0 & P_2 \cdot \omega_2 & \alpha_2 & \beta_2 \\ P_3 \cdot \omega_3 & 0 & \alpha_3 & \beta_3 \\ 0 & P_3 \cdot \omega_3 & \alpha_4 & \beta_4 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ A \\ B \end{bmatrix} = \begin{bmatrix} P_1 \\ 0 \\ P_2 \\ 0 \\ P_3 \\ 0 \end{bmatrix}$$

Si noti la struttura lineare del sistema.

Una volta trovati a , b , A , B adottando una soluzione ai minimi quadrati (magari aumentando il numero di equazioni), i valori di Δx e $Delay$ si potrebbero ricavare semplicemente nel seguente modo:

$$Delay = \frac{\arctan\left(\frac{B}{A}\right)}{\omega_1}$$

$$\Delta x = \frac{A}{\cos\left(\arctan\left(\frac{B}{A}\right)\right)}$$

Il problema però nasce dal fatto che non è possibile arrivare ad una tale forma delle equazioni: dal

momento che $\omega_2 = 2\omega_1$ e che $\omega_3 = 3\omega_1$, dove $\omega_1 = \frac{\omega_c}{2}$, il sistema diventa:

$$\begin{cases} P_1 \cdot \omega_1^2 \cdot a + \Delta x \cdot \cos(\text{Delay} \cdot \omega_1) - P_1 = 0 \\ P_1 \cdot \omega_1 \cdot b + \Delta x \cdot \sin(\text{Delay} \cdot \omega_1) = 0 \\ P_2 \cdot \omega_2^2 \cdot a + \Delta x \cdot \cos(2 \cdot \text{Delay} \cdot \omega_1) - P_2 = 0 \\ P_2 \cdot \omega_2 \cdot b + \Delta x \cdot \sin(2 \cdot \text{Delay} \cdot \omega_1) = 0 \\ P_3 \cdot \omega_3^2 \cdot a + \Delta x \cdot \cos(3 \cdot \text{Delay} \cdot \omega_1) - P_3 = 0 \\ P_3 \cdot \omega_3 \cdot b + \Delta x \cdot \sin(3 \cdot \text{Delay} \cdot \omega_1) = 0 \end{cases}$$

Si nota come adesso sono esplicitati i termini $\Delta x \cdot \cos(k \cdot \text{Delay} \cdot \omega_1)$ e $\Delta x \cdot \sin(k \cdot \text{Delay} \cdot \omega_1)$, che corrispondono ad A_k e B_k . Dal momento che tali termini non possono essere espressi come combinazione lineare di $\Delta x \cdot \cos(\text{Delay} \cdot \omega_1)$ e $\Delta x \cdot \sin(\text{Delay} \cdot \omega_1)$, le precedenti considerazioni non possono essere qui utilizzate, perciò il sistema rimane non lineare.

Una linearizzazione completa non è possibile; se però il ritardo fosse noto il sistema diverrebbe lineare. L'idea è allora quella di quantizzare un intervallo nel quale si è sicuri si trovi il ritardo cercato; a questo punto si risolvono tanti sistemi lineari quanti sono gli N campionamenti dell'intervallo.

Il passo successivo è quello di calcolare un indice di qualità. Nel caso più semplice il procedimento si può schematizzare in questo modo: considerando il sistema lineare

$$Ax = B$$

si può risolvere x ai minimi quadrati, ottenendo \hat{x} . Adesso sostituiamo \hat{x} in x nell'equazione sopra, e calcoliamo il vettore \hat{B} . L'indice di qualità può essere ottenuto quindi come modulo della quantità $\hat{B} - B$, cioè:

$$I_q = \|\hat{B} - B\|$$

Valutando per ognuno degli N intervalli di quantizzazione del ritardo, si ricava l'intervallo che presenta indice di qualità inferiore. Tale valore del ritardo sarà quello fornito in uscita, assieme agli altri parametri, come soluzione del sistema non lineare.

Per ottenere risultati migliori potrebbe essere opportuno inserire dei pesi nel sistema, ossia associare a ciascuna equazione del sistema un parametro che ne specifica l'importanza all'interno di tutto il

sistema. In pratica, comunque, si osserva che pesi diversi tra loro portano al degrado della soluzione; di default, quindi, i pesi utilizzati sono tutti uguali all'unità.

Effettuando un paragone tra i due metodi di soluzione proposti, si può però constatare come generalmente il calcolo della soluzione con la funzione *fsolve* dia risultati migliori che con il metodo della linearizzazione.

6.2.2.3 Numero di componenti frequenziali da analizzare

Un altro tentativo di miglioramento della soluzione è stato effettuato cercando di capire se aumentando il numero di equazioni nel sistema non lineare la soluzione si fosse irrobustita. In pratica si tratta di considerare un numero N di componenti del segnale in uscita, con N deciso dall'utente. Le simulazioni dimostrano come generalmente 3 componenti frequenziali da elaborare diano i migliori risultati; aumentando il numero di equazioni la soluzione non migliora, anzi tende a degradarsi. Ciò è dovuto al fatto che ogni equazione in più che si introduce porta comunque con sé un carico di rumore.

6.2.2.4 Mediazione dei parametri

L'ultimo approccio tentato è il risultato di semplici considerazioni sul segnale corrotto da rumore: se il rumore è a media nulla, significa che il suo contributo, dopo un tempo infinito, sarà mediamente pari a zero. Su un periodo del segnale, comunque, il contributo è generalmente diverso da zero: dal momento che i parametri da inserire come costanti nel sistema non lineare vengono calcolati su un periodo, risentiranno del rumore su quel particolare periodo del segnale. Ma se le stime vengono effettuate basandosi su parametri noti che sono la media tra i parametri calcolati su tutti i periodi precedenti, la stima dovrebbe, in linea di principio, convergere ai valori corretti. Le simulazioni effettuate dimostrano che in effetti questa convergenza esiste. Si potrebbe pensare di mediare non i parametri calcolati sui periodi, ma le stime che ogni volta vengono fornite: in questo caso, però, andremmo a mediare dei valori che sono soluzione non esatta di una particolare procedura di soluzione del sistema non lineare: il rischio è quello di calcolare il valor medio di parametri non consistenti, quindi di avere a che fare con una procedura di calcolo che fornisce risultati non corretti e che non converge ad una soluzione.

Se il segnale in ingresso al sistema di Hammerstein può essere considerato non affetto da rumore, gli unici parametri che ne risentono sono i coefficienti di Fourier Y_k del segnale filtrato. Periodo

dopo periodo, si ottiene un Y_k che è la media di tutti gli Y_k dei periodi precedenti. Si deve però porre attenzione a calcolare la media solo per quei parametri che corrispondono a periodi del segnale di ingresso che danno oscillazioni stabili in uscita, ossia valutare l'uscita solo quando la sua forma d'onda è divenuta periodica da un istante temporale in poi.

6.2.2.5 Schema interno del blocco di identificazione

Vediamo ora di dettagliare lo schema del blocco di identificazione.

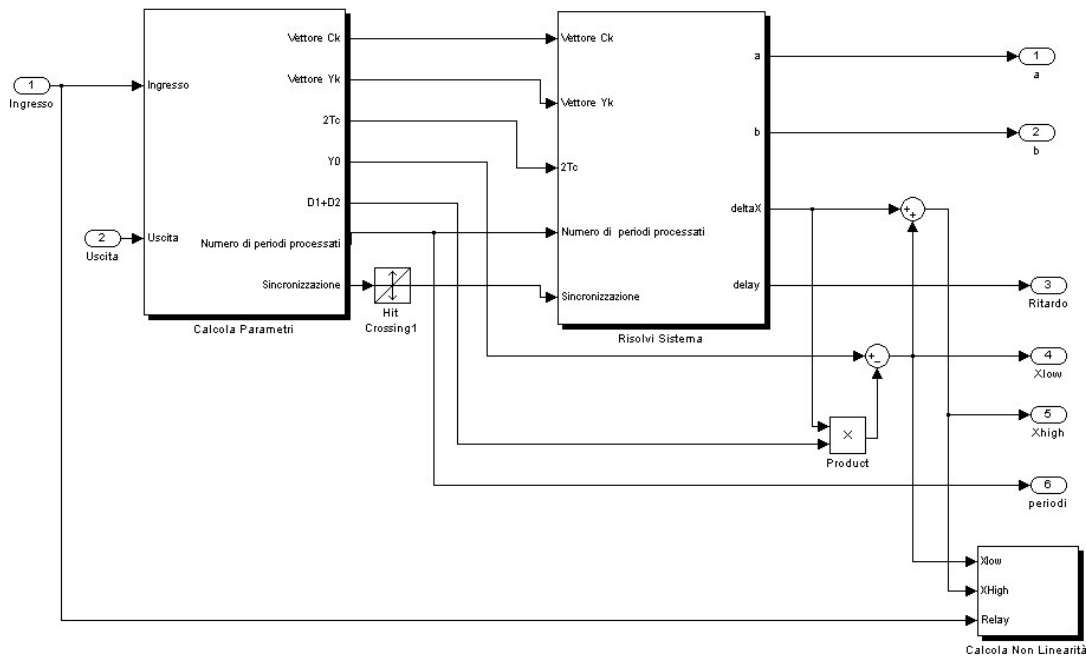


Figura 6.15. Schema interno del blocco di identificazione

Dalla figura si vede come vi siano 3 sottoblocchi degni di nota: il blocco *Calcola parametri*, il blocco *Risolvi sistema* ed il blocco *Calcola non linearità*.

Il blocco *Calcola parametri* prende in ingresso due segnali:

1. il segnale *Ingresso* proveniente dallo slave relè (che è uguale al segnale in ingresso al processo di Hammerstein, a meno del ritardo introdotto)
2. il segnale *Uscita* proveniente dal blocco di filtraggio.

In uscita produce 7 parametri:

1. il vettore dei coefficienti di Fourier C_k del segnale *Ingresso*, organizzati secondo lo schema seguente:

1° elemento	2° elemento	3° elemento	4° elemento	...
$\text{Re}(C_1)$	$\text{Imm}(C_1)$	$\text{Re}(C_2)$	$\text{Imm}(C_2)$...

2. il vettore dei coefficienti di Fourier Y_k del segnale *Uscita*, organizzati secondo lo stesso schema
3. il periodo del segnale filtrato
4. il valor medio del segnale filtrato
5. il valore D_1+D_2 , cioè la somma dei duty cycle dei due segnali rettangolari che sommati costituiscono il segnale *Ingresso*
6. il numero dei periodi del segnale *Uscita* attualmente elaborati
7. la variabile di sincronizzazione

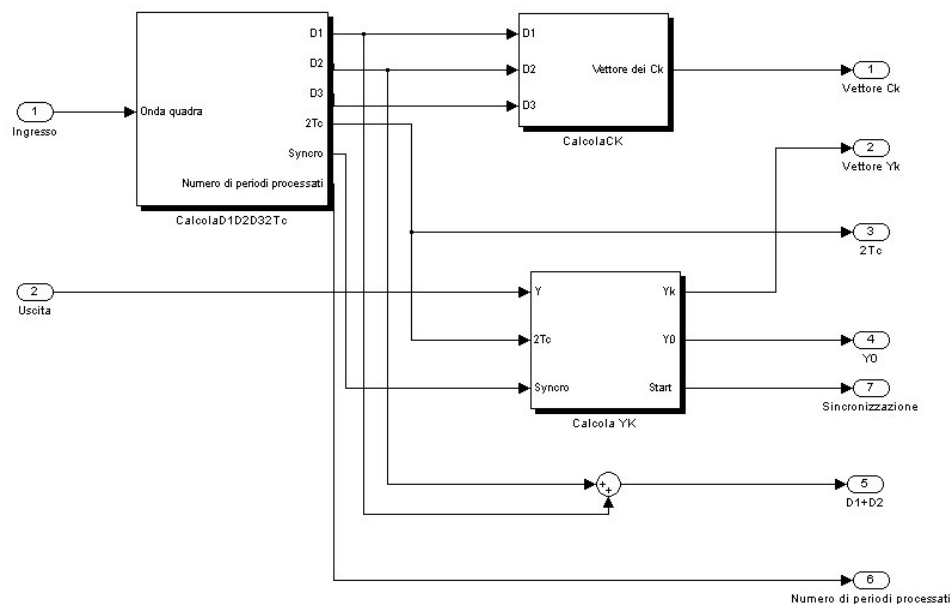


Figura 6.16. Schema interno del blocco *Calcola parametri*

I valori C_k e Y_k , il periodo, il numero di periodi processati e la variabile di sincronizzazione vengono poi passati al blocco *Risolvi sistema* che si occupa di ricavare i parametri a , b , ΔX e ritardo, operando la prima identificazione del sistema. Il numero di periodi processati serve comunque solo in caso di mediazione dei parametri; la variabile di sincronizzazione consente invece di avvertire il

blocco quando un nuovo periodo è stato processato e quindi è opportuno ricalcolare i parametri stimati. Gli altri parametri in ingresso al blocco vengono utilizzati direttamente nel calcolo della soluzione.

Il parametro ΔX in uscita dal blocco deve essere ancora utilizzato per calcolare i valori X_{high} e X_{Low} , valori assunti dalla funzione non lineare ricostruita per punti attraverso il blocco *Non linearità*.

Notare che il periodo ed il valor medio del segnale *Uscita* sono già stati calcolati nel primo stadio; il motivo per cui si reintroducono nello stadio di identificazione consiste nel prevedere, per il futuro, un unico stadio in cui tutte le informazioni sul segnale vengano raccolte automaticamente ed una sola volta.

Lo schema del blocco *Risolvi sistema* è il seguente:

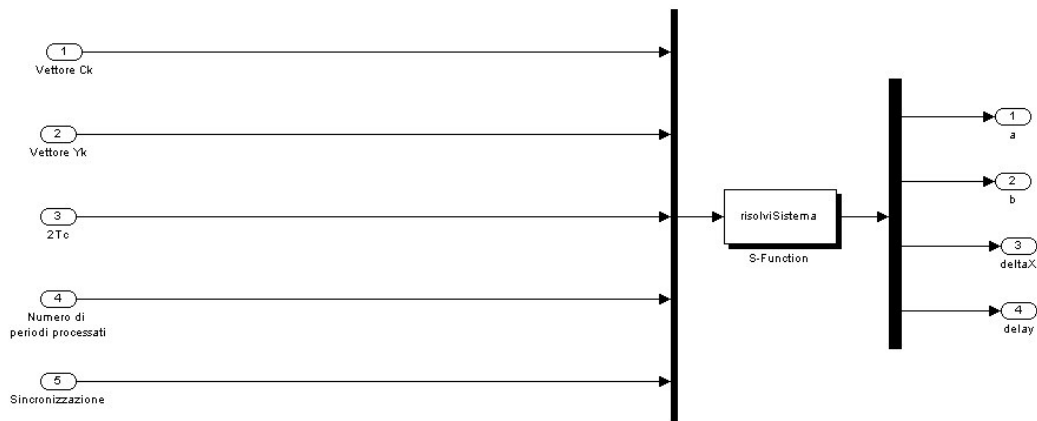


Figura 6.17. Schema interno del blocco *Risolvi sistema*

In questo caso, l'ingresso per la S-function *risolviSistema* è così costituito:

1. Vettore C_k dei coefficienti di Fourier del segnale *Ingresso*
2. Vettore Y_k dei coefficienti di Fourier del segnale *Uscita*
3. Periodo del segnale *Uscita*
4. Numero di periodo attualmente elaborati
5. Variabile di sincronizzazione che avverte quando un periodo del segnale è stato acquisito

La S-function *risolviSistema*, a partire da tale ingresso, produce in uscita:

1. i parametri a e b coefficienti del polinomio al denominatore
2. il ritardo del sistema
3. il valore ΔX che passato al blocco *Non linearità*, permetterà di calcolare due punti della funzione non lineare

Infine, lo schema interno del blocco *Calcola non linearità* è il seguente:

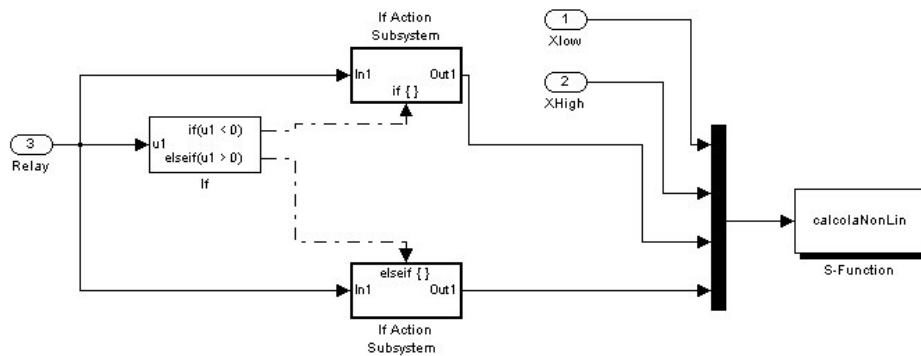


Figura 6.18. Schema interno del blocco *Calcola non linearità*