

IMPLEMENTAZIONE SIMULINK E FUNZIONI

In questo capitolo sono esposti i modelli Simulink che implementano un relay a isteresi variabile e le funzioni create per identificare la funzione di trasferimento del processo.

3.1 Modello del Relay Variabile

Il modello del Relay Variabile è implementato attraverso Simulink di Matlab®.

Esso è composto da 3 Blocchi: Relay, Controllo Relay, Frequency Meter.

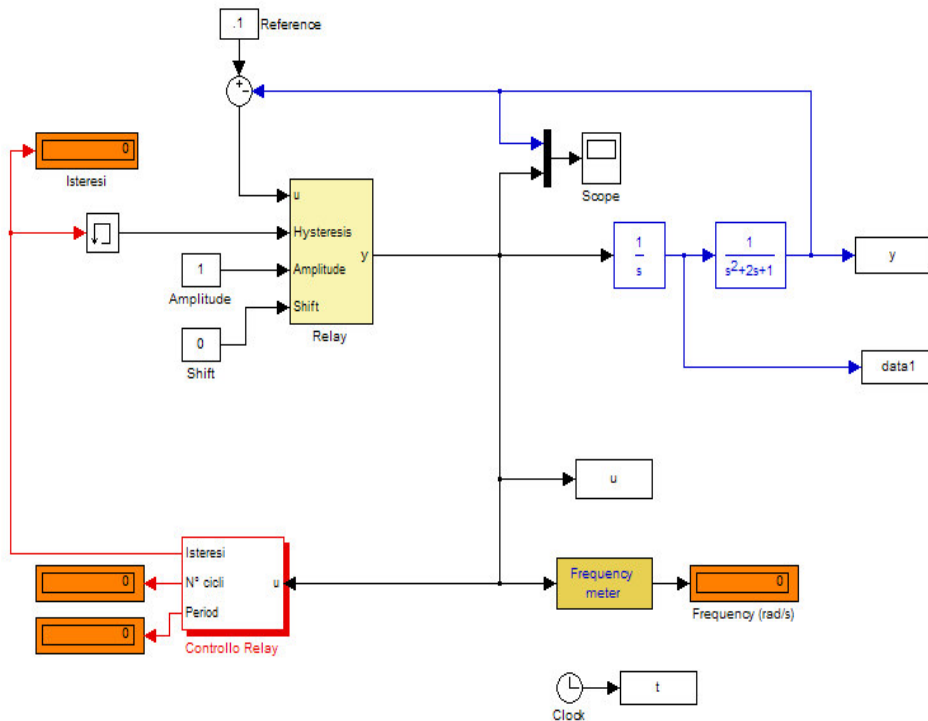


Fig. 3.1 Modello del relay variabile

3.1.1 Blocco Controllo Relay

Nel blocco **Controllo Relay** si impostano i parametri principali della nostra simulazione, quali il vettore del numero di cicli e il vettore di isteresi.

Questi parametri si settano nella maschera:

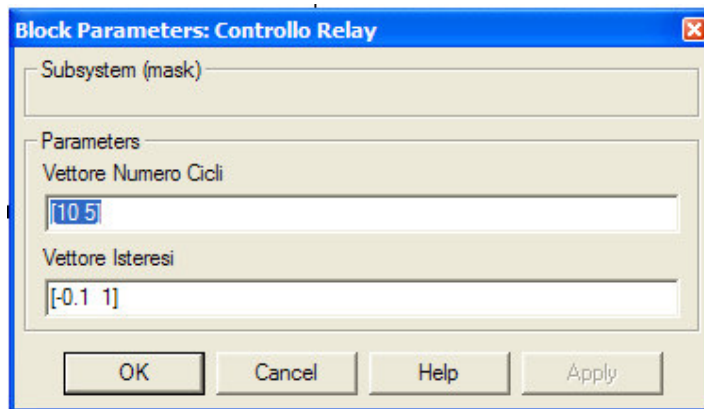


Fig 3.2

In questo esempio, verranno fatti 10 cicli con valore $h_1 = -0.1$ e 5 cicli con valore $h_2 = 1$.

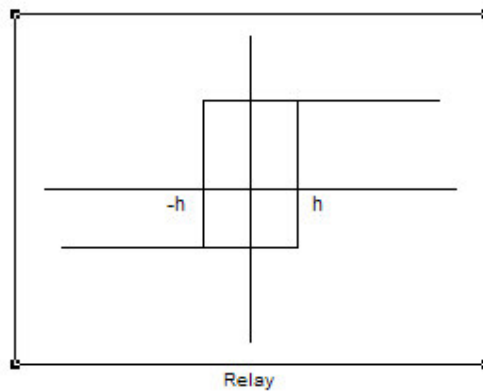


Fig 3.3

In uscita a questo blocco abbiamo il valore di isteresi h , il relativo numero di cicli parziali ed il periodo T .

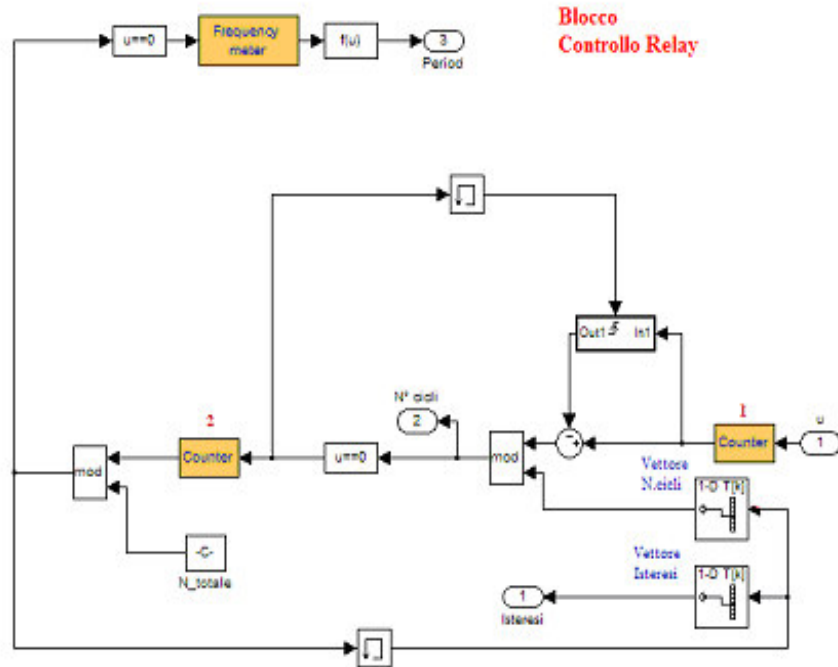


Fig 3.4 Blocco Controllo Relay

Adesso esaminiamo in dettaglio il blocco: come si può vedere è composto da 2 sottosistemi Counter, il primo conta il numero di cicli totali fatti dal mio sistema, il secondo contatore invece serve ad incrementare il valore dell'indice che scorre i due vettori.

Quando parte il sistema, con i vettori che abbiamo precedentemente settato, avremo $i=0$ ("i" indice che scorre i due vettori), $n[0]=10$, $h[0]=-0.1$, $N_totale=2$. Le azioni che vengono fatte sono:

1. Il primo contatore inizia a contare il numero di cicli;

2. Blocco Somma, se $i=0$, il valore che si sottrae al numero di cicli è 0, altrimenti si sottrae al numero di cicli il valore $\sum_{k=0}^{i-1} n[k]$, questo serve ad avere sempre il numero di cicli parziale;
3. Nel blocco Mod, viene fatto il confronto tra il numero di cicli fatti $n[i]$ e il numero di cicli totale $n[i]$ relativi al valore di $h[i]$. In uscita a questo modulo abbiamo i rimanenti numeri di cicli;
4. Si aggiorna l'uscita N° cicli;
5. Se N° cicli è uguale a 0 si attiva il trigger ed il secondo contatore aggiorna il valore "i". Possiamo notare come prima del trigger vi sia un elemento di Memory Delay, questo elemento campiona e mantiene il valore, che verrà usato nel ciclo successivo nel blocco somma; (P.to 2)
6. Il secondo blocco mod confronta il nuovo valore di "i" con la costante N_{totale} . Se l'uscita è diversa da 0 si scorrono solamente i due vettori. Altrimenti si calcola il periodo grazie al sottosistema Frequency Meter.

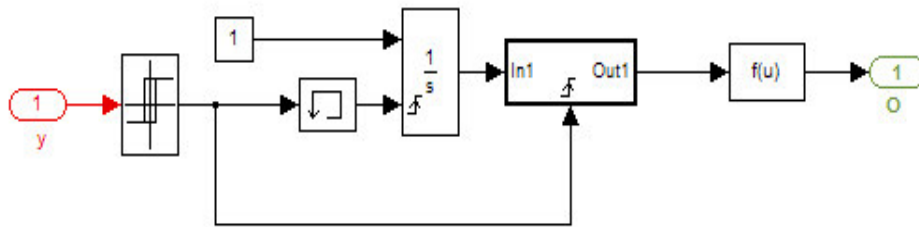


Fig 3.5

3.1.2 Blocco Frequency Meter

Il blocco **Frequency Meter** utilizzato anche nello schema principale serve a calcolare il periodo del segnale; questo viene calcolato nel blocco Controllo relay prima di cambiare il valore di isteresi.

3.1.3 Blocco Relay

Il blocco **Relay** :

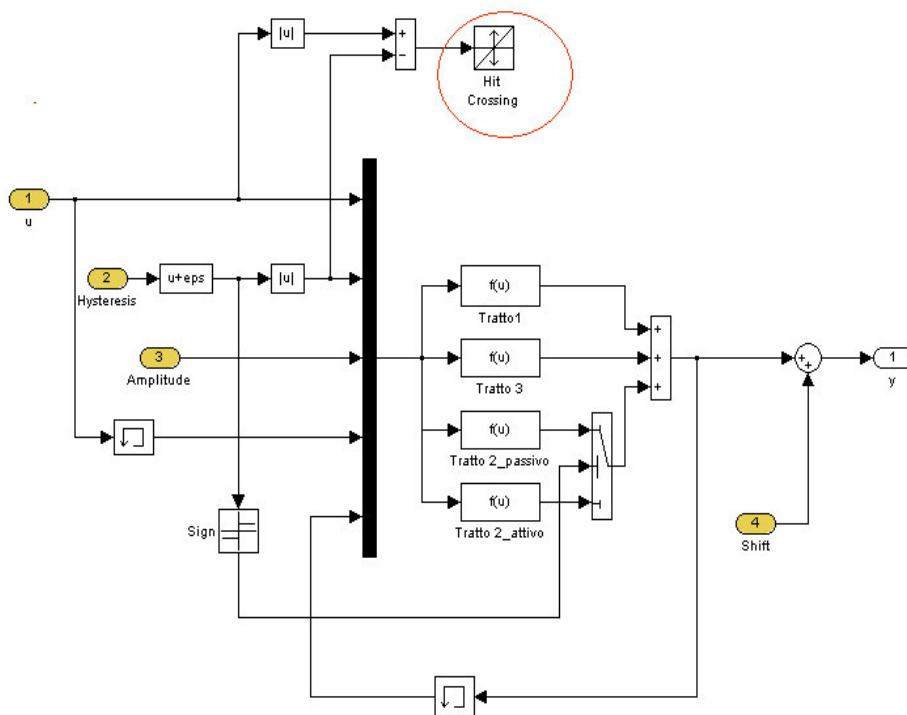


Figura 3.6

Gli ingressi di questo blocco sono **u** il segnale in ingresso, **Hysteresis** il valore di isteresi h, **Amplitude** l'ampiezza del relay A e **Shift** che permette di realizzare un relay traslato verticalmente.

Il relay può essere passivo o attivo:

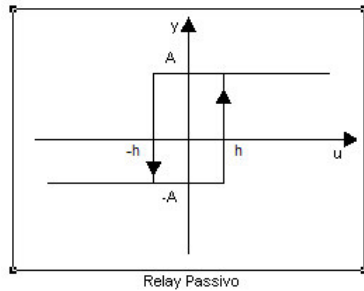


Fig 3.7 a

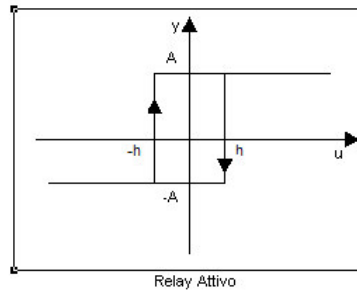


Fig 3.7 b

Nel primo caso l'uscita e le rispettive funzioni del blocco sono:

$$y(k) = \begin{cases} -A & \text{se } u < -\text{hyst} & \text{Tratto 1} \\ y(k-1) & \text{se } -\text{hyst} < u < \text{hyst} & \text{Tratto 2 pass.} \\ A & \text{se } u > \text{hyst} & \text{Tratto 3} \end{cases}$$

Nel secondo caso sono:

$$y(k) = \begin{cases} -A & \text{se } u < -\text{hyst} & \text{Tratto 1} \\ +A & \text{se } -\text{hyst} < u < \text{hyst e } u(k-1) < -\text{hyst} & \text{Tratto 2_att.} \\ -A & \text{se } -\text{hyst} < u < \text{hyst e } u(k-1) > \text{hyst} & \text{Tratto 2_att.} \\ y(k-1) & \text{se } -\text{hyst} < u < \text{hyst} & \text{Tratto 2_att.} \\ A & \text{se } u > \text{hyst} & \text{Tratto 3} \end{cases}$$

Il blocco Hit Crossing evidenziato in rosso garantisce la commutazione esattamente in corrispondenza dell'isteresi.

Il blocco Sign pilota lo switch che differenzia il comportamento del relay (attivo o passivo).

3.2 Funzioni

Per ricostruire la funzione di trasferimento del processo “G” (blocco blu nello schema in figura 5.8) sono state implementate due funzioni: *four.m* e *calcolafdt*. Queste due funzioni (M-file) sono state sviluppate in ambiente Matlab®.

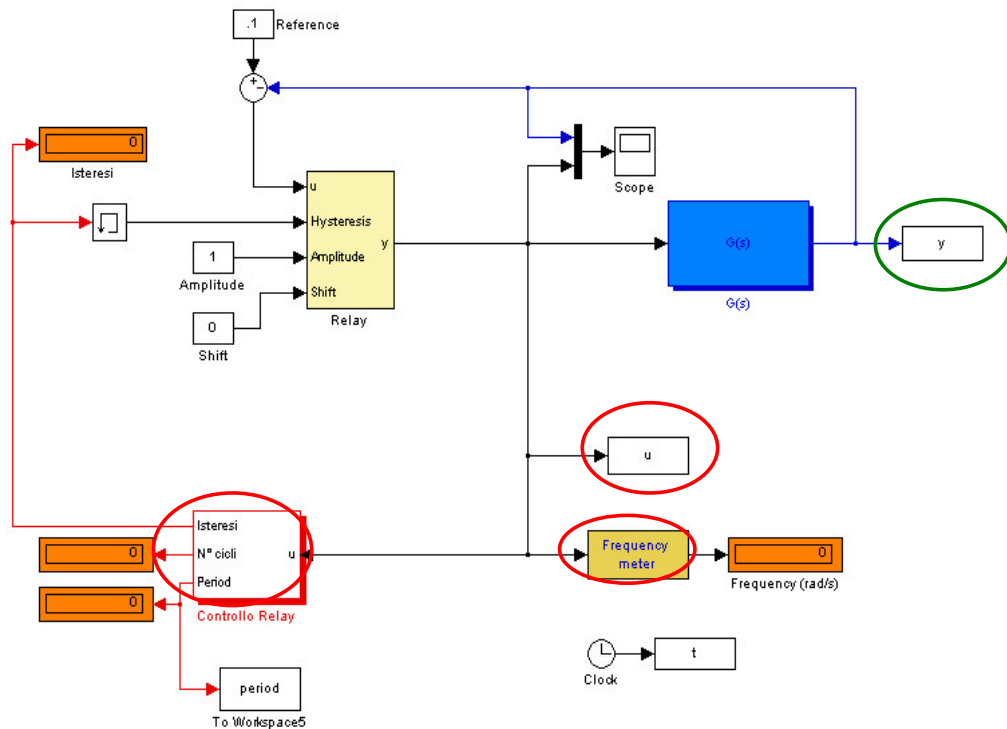


Fig 3.8

3.2.1 *Four.m*

Questa funzione ha il compito di costruire la trasformata di Fourier del segnale “u” in uscita del relay (segnale in ingresso al processo G). Il segnale “u” come si può vedere dalla figura è un segnale periodico di forma rettangolare

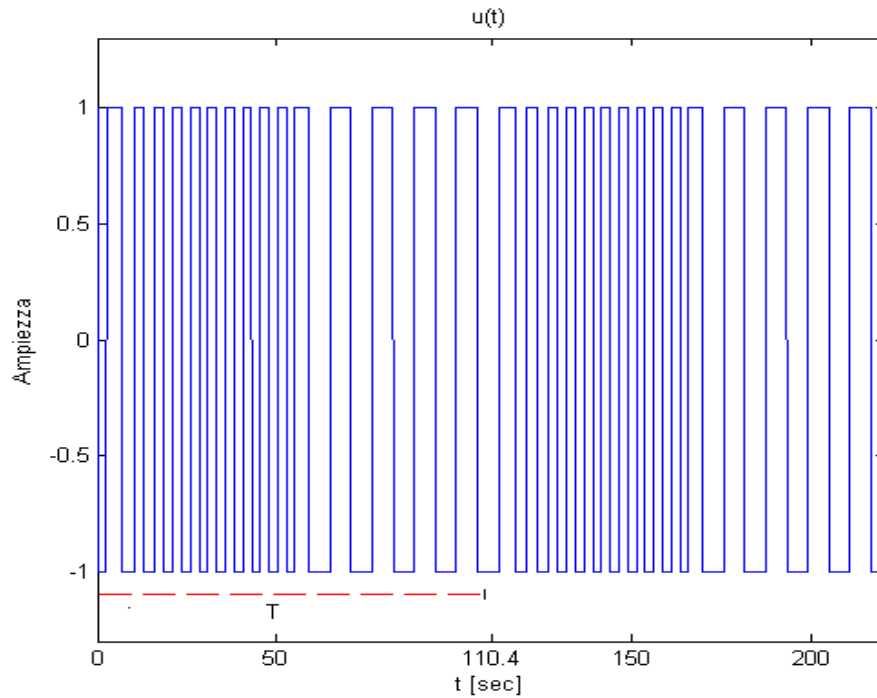


Fig 3.9

Si ha questo tipo di uscita settando la maschera del blocco Controllo Relay come spiegato nei paragrafi precedenti.. La funzione four calcola:

- la parte reale;
- la parte immaginaria ;

- il modulo;
- le armoniche fondamentali;
- il tempo di campionamento;
- il valore medio

del segnale $u(t)$.

Per calcolare i parametri la funzione ha bisogno del vettore $u(t)$, del periodo del segnale e del vettore dei tempi. Tutti questi valori vengono calcolati nel Modello Relay Variabile come è evidenziato in rosso nella figura 3.8. Questa funzione implementa la scomposizione di Fourier del segnale vista nei capitoli precedenti, la $u(t)$ è:

$$u(t) = \sum_{k=1}^{\infty} \left[\left(\sum_{i=1}^n A_{ik} \right) \cos(k\Omega t) + \left(\sum_{i=1}^n B_{ik} \right) \sin(k\Omega t) \right] + \frac{A}{T} \sum_{i=1}^n (t_i - t_{i-1})_{i_dispari} - \frac{A}{T} \sum_{i=1}^n (t_i - t_{i-1})_{i_pari}$$

dove

$$A_{ik} = -\frac{A_i}{\pi k} [-\sin(k\Omega t_i) + \sin(k\Omega t_{i-1})]$$

$$B_{ik} = -\frac{A_i}{\pi k} [-\cos(k\Omega t_i) + \cos(k\Omega t_{i-1})]$$

$$\text{con } A_i = \begin{cases} A & \text{se } i \text{ è dispari,} \\ -A & \text{se } i \text{ è pari} \end{cases}$$

A è l'ampiezza del segnale $u(t)$,

$$\Omega = \frac{2\pi}{T},$$

T periodo del segnale.

La funzione viene chiamata con il comando in linea di Matlab®:

```
[Real,Imag,f,Medio,Freq,Mod,Tc]=four(t,u,period,j)
```

L'unico parametro da settare è "j" che regola la sommatoria su k (per j=200, si riesce a ricostruire il segnale).

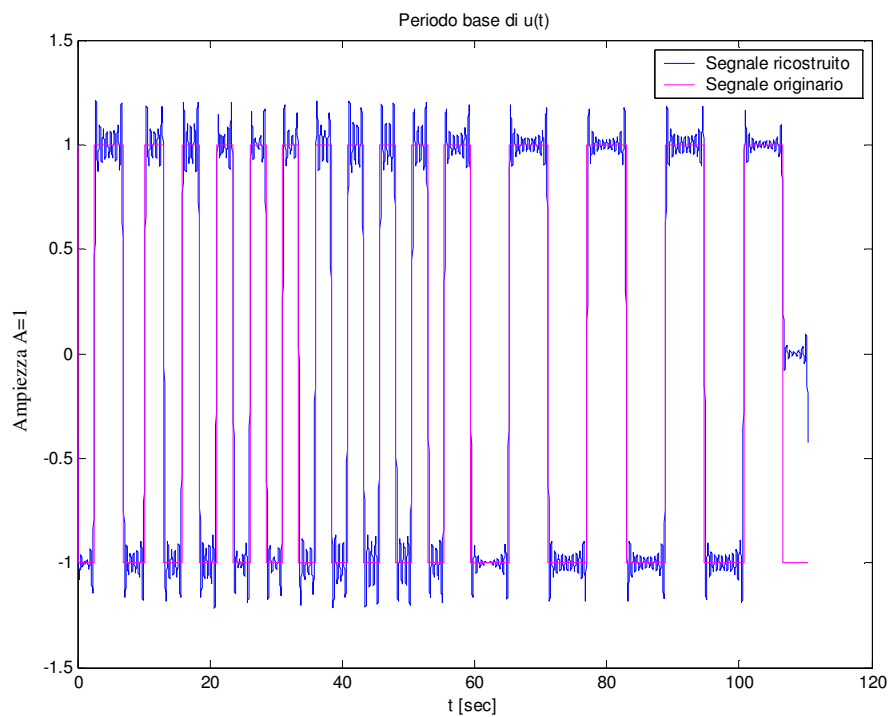


Fig. 3.10

Come si può vedere dalla fig. 3.10, il segnale è ricostruito fedelmente per j=200.

Lo spettro di ampiezza del segnale così calcolato è rappresentata nella figura 5.11.

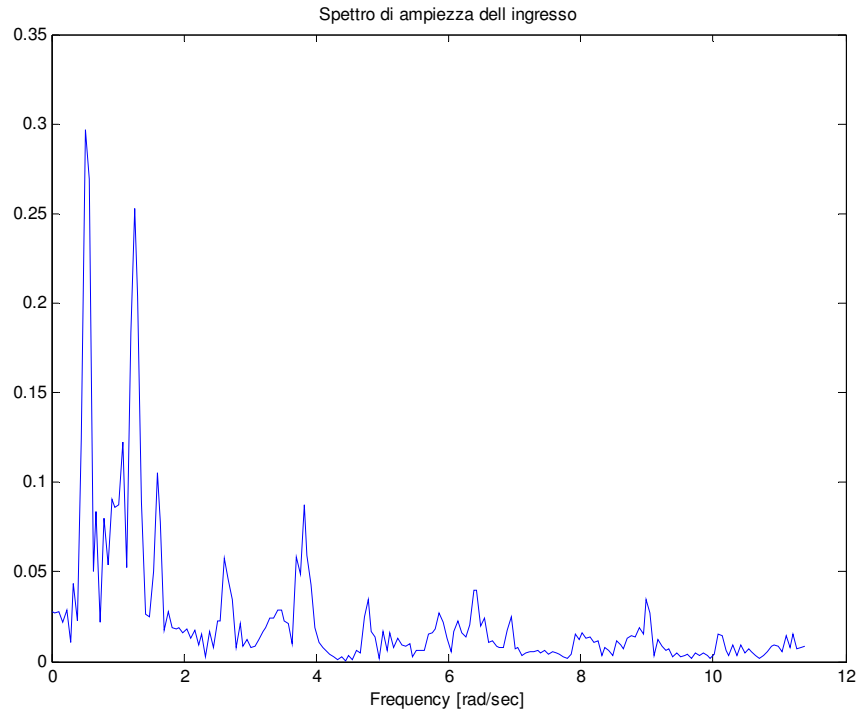


Fig. 3.11

3.2.2 *Calcolafdt.m*

La funzione *Calcolafdt.m* calcola la funzione di trasferimento del processo “G(s). La funzione ha come variabili di ingresso *Mod* e *Freq* che sono due vettori contenenti rispettivamente il modulo e le armoniche fondamentali del segnale in ingresso al processo, il vettore *y* segnale di uscita del processo, *t* vettore dei tempi e *Tc* tempo di campionamento del segnale. I vettori *Mod* e *Freq* e il tempo di campionamento sono stati calcolati nella funzione *four.m*, il vettore *y* e *t* sono dei parametri di uscita della

simulazione del Modello Relay Variabile (evidenziati il primo in verde il secondo in rosso nella figura 5.8). Le variabili di uscita della funzione sono Mod_usc , $Freq_usc$, $Modg$, $Freq$. In cui Mod_usc e $Freq_usc$ sono il modulo e le frequenze del segnale di uscita, $Modg$ e $Freq$ sono il modulo e le frequenze in cui si è calcolata la funzione di trasferimento del processo.

La funzione viene chiamata con il comando in linea di Matlab®:

```
[Mod_usc,f_usc,Modg,Freq]=calcolafdt(Mod,Freq,Tc,y,t)
```

La funzione è divisa in due blocchi, il primo che si occupa di calcolare la trasformata discreta di Fourier dell'uscita " $y(t)$ ", il secondo che calcola il modulo della funzione di trasferimento. La trasformata di Fourier dell'uscita viene fatta usando la funzione $y = \text{fft}(x,n)$ implementata da Matlab® ed esaminata nel capitolo 2.

In questo modo, otteniamo lo spettro di ampiezza del segnale di uscita $y(t)$ alle frequenze calcolate dalla fft mettendo n uguale al numero di campioni del segnale. Successivamente riduciamo il numero di parametri calcolati alle frequenze del segnale in ingresso, grazie alla funzione Matlab® $y_i = \text{interp1}(x,Y,x_i)$, dopo questo comando lo spettro di ampiezza del segnale di uscita, considerando solo metà dei punti perché è simmetrico, è mostrato in Fig. 3.12.

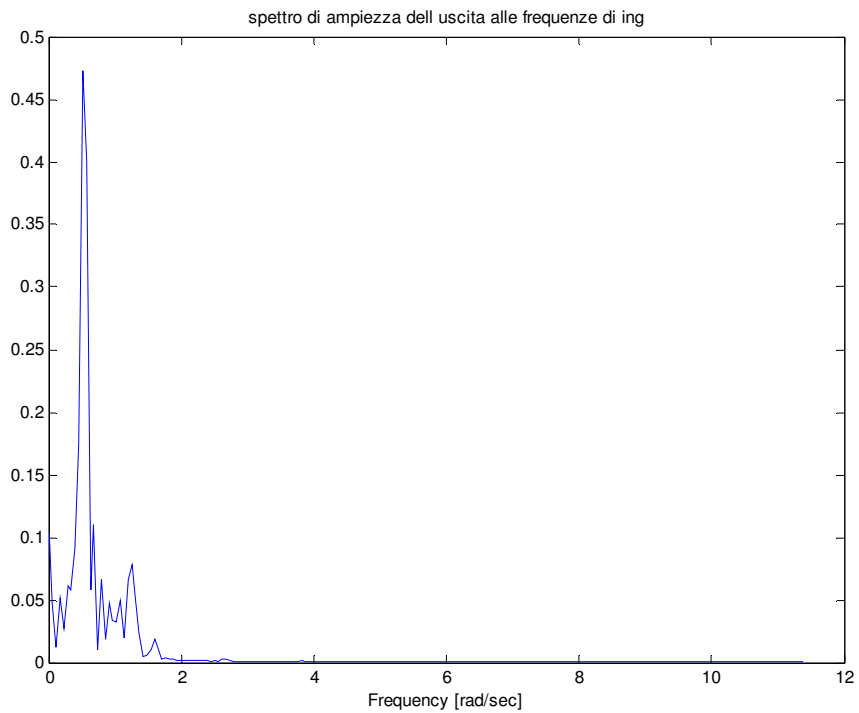


Fig. 3.12

La seconda parte della funzione si occupa di calcolare lo spettro di ampiezza della funzione di trasferimento usando questa formula

$$|G(j\omega_i)| = \frac{|Y(j\omega_i)|}{|X(j\omega_i)|}$$

dove $|Y(j\omega_i)|$ è stata calcolato con la funzione di Matlab® *fft.m* mentre $|X(j\omega_i)|$ con la funzione *four.m*. Lo spettro di ampiezza della funzione di trasferimento è mostrato in Fig. 3.13.

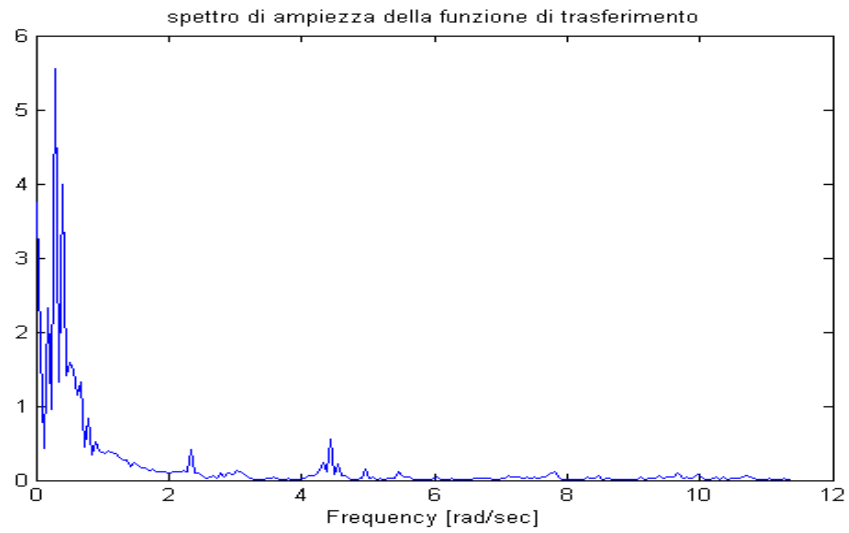


Fig. 3.13

Lo spettro di ampiezza originale è visibile in rosso nella figura 3.14

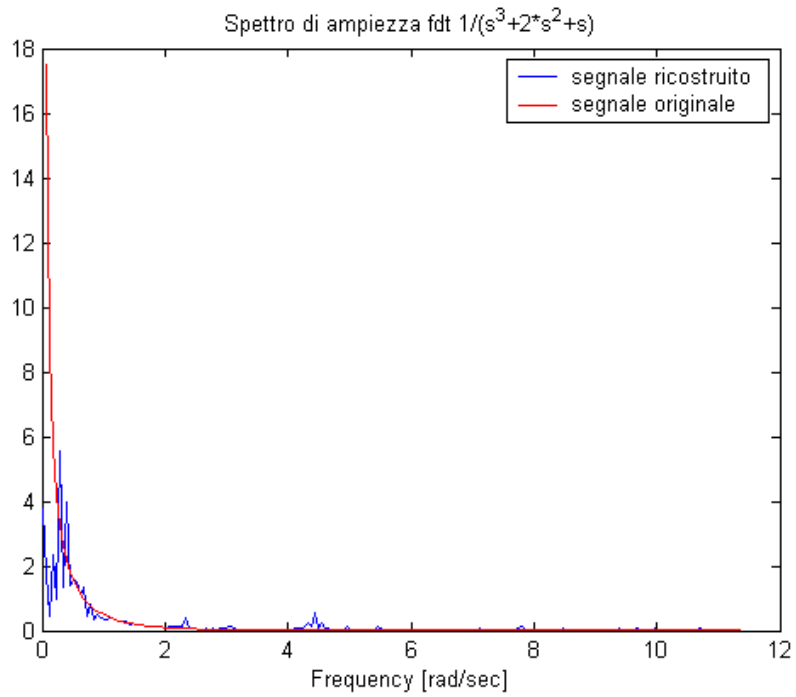
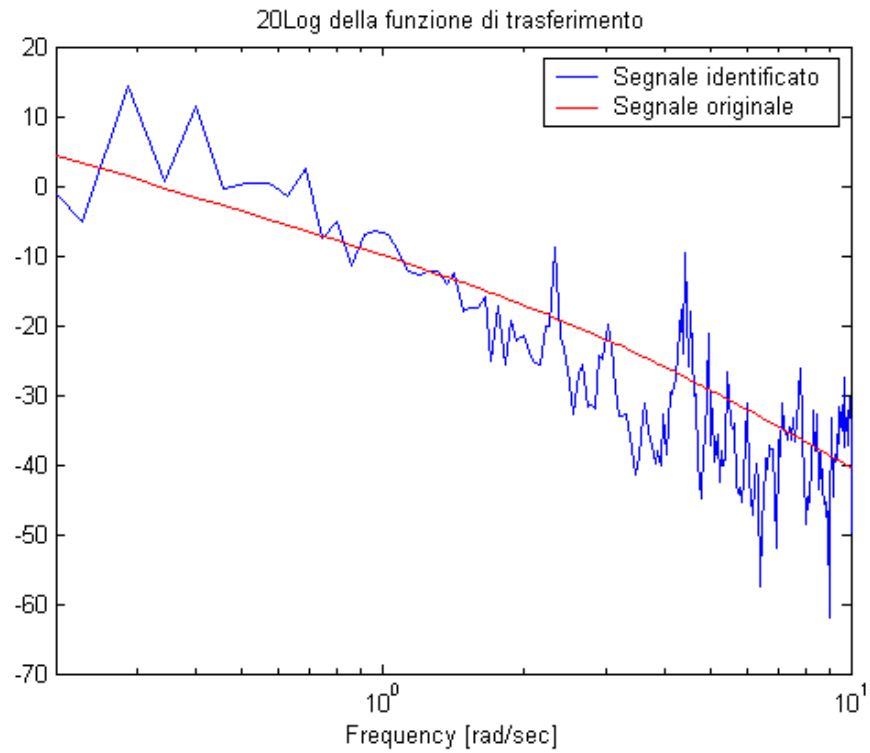


Fig. 3.14

Mentre il diagramma di Bode del modulo avendo eliminato in un passo precedente i valori negativi è:



Questa simulazione è stata fatta con $j=200$, per avere un maggior numero di frequenze basta variare questo parametro.