

MATLAB-SIMULINK

Simulink

Analisi spettrali

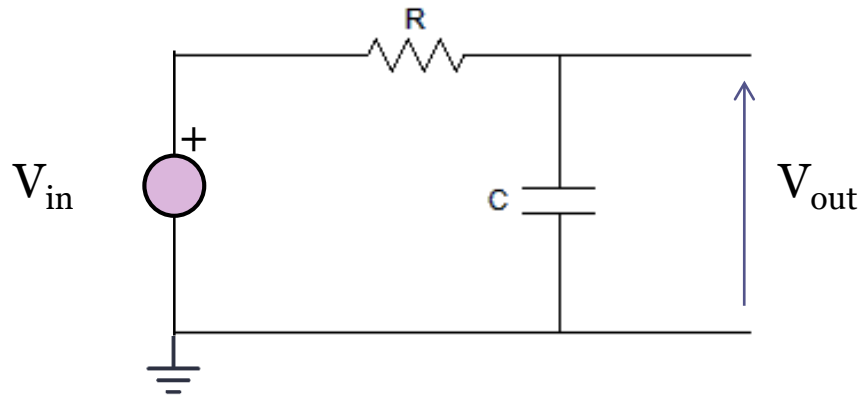
Ing. Alessandro Pisano

`pisano@diee.unica.it`

Vediamo come realizzare **analisi spettrali** di segnali campionati generati da un modello Simulink

Le procedure descritte saranno ovviamente applicabili anche per effettuare analisi spettrali di segnali campionati generati direttamente all'interno di uno script o acquisiti da file

Sviluppiamo l'esempio con riferimento ad un **filtro passa basso RC**



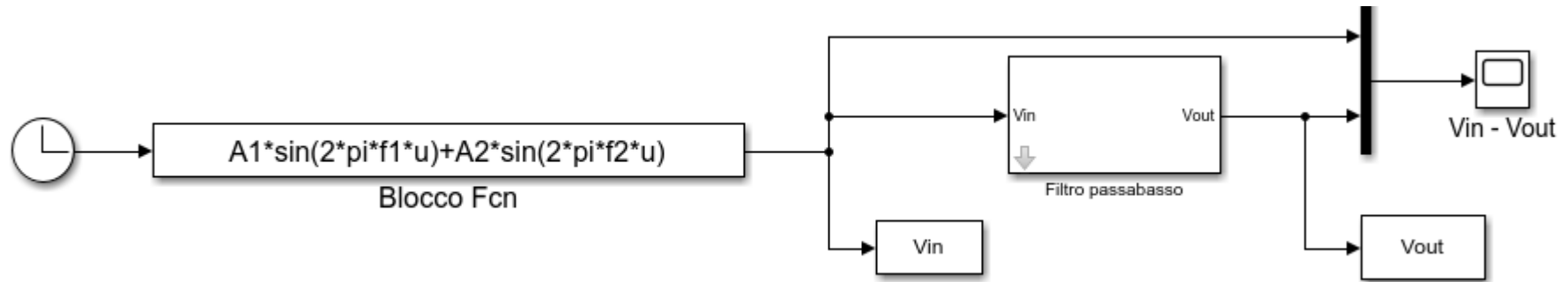
Equazione differenziale

$$RC \dot{V}_{out}(t) + V_{out}(t) = V_{in}(t)$$

Equazione differenziale esplicitata rispetto alla derivata di ordine più elevato

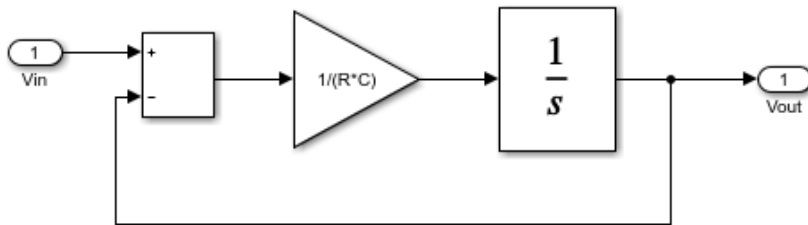
$$\dot{V}_{out}(t) = \frac{1}{RC} (V_{in}(t) - V_{out}(t))$$

Modello Simulink.

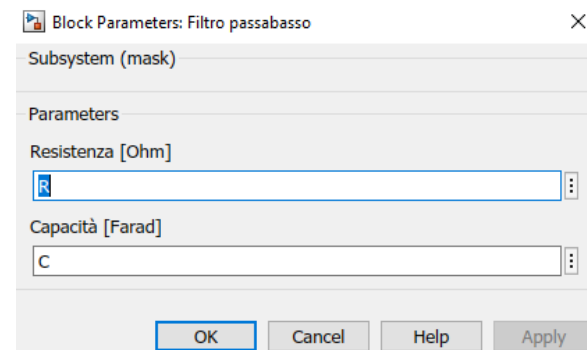


File: FiltroLP_AnalisiSpettrali.slx

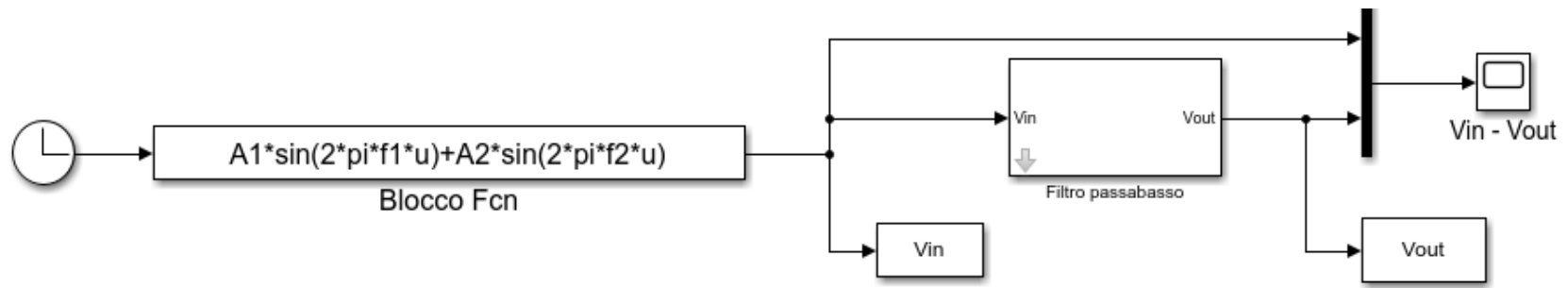
Contenuto del Subsystem



Finestra di parametrizzazione del Subsystem mascherato



Modello Simulink.

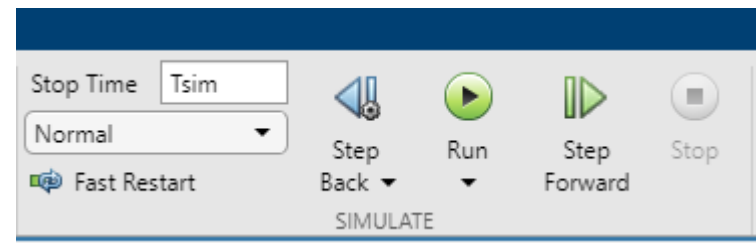


Il modello è predisposto per l'applicazione di una tensione di ingresso $V_{in}(t)$ del tipo $<$

$$V_{in}(t) = A_1 \sin(2\pi f_1) + A_2 \sin(2\pi f_2)$$

Debbono essere salvati nel workspace i parametri R e C (resistenza e capacità del filtro) oltre ai parametri di ampiezza e frequenza delle due armoniche presenti nel segnale di ingresso $V_{in}(t)$.

E' inoltre da assegnarsi anche la durata della simulazione definendo nel workspace la variabile T_{sim}



```
clear all
clc
close all
```

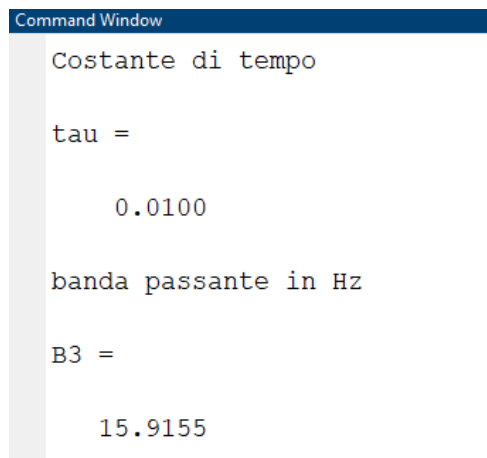
File: FiltroLP_AnalisiSpettrali_Script.m

```
R=1e3; % 1k Ohm
C=1e-5; % 10 pF
```

```
disp('Costante di tempo')
tau=R*C
disp('banda passante in Hz')
B3=1/(2*pi*tau)
```

Nella prima parte dello Script si assegna un valore i parametri R e C e si determinano successivamente la costante di tempo del filtro (in secondi) e la sua **banda passante** in Hertz

Output nella
Command Window



```
Command Window
Costante di tempo
tau =
    0.0100
banda passante in Hz
B3 =
    15.9155
```

La banda passante è pari a circa 16 Hertz. Ciò implica che le componenti armoniche del segnale di ingresso aventi frequenza inferiore a 16 Hz si propagheranno pressoché inalterate attraverso il filtro , mentre invece le componenti armoniche aventi frequenza superiore a tale valore verranno attenuate in ampiezza (e sfasate in ritardo) in misura progressivamente maggiore al crescere della frequenza.

Applichiamo il seguente segnale di tensione in ingresso

$$V_{in}(t) = \sin(4\pi) + 2 \sin(100\pi)$$

che corrisponde ai seguenti valori dei parametri

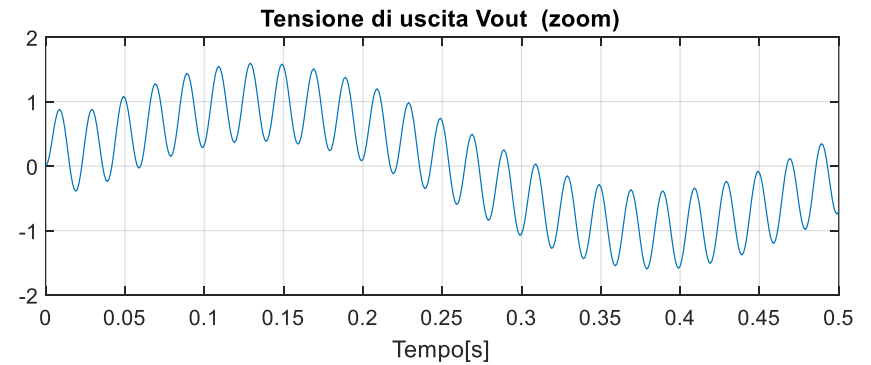
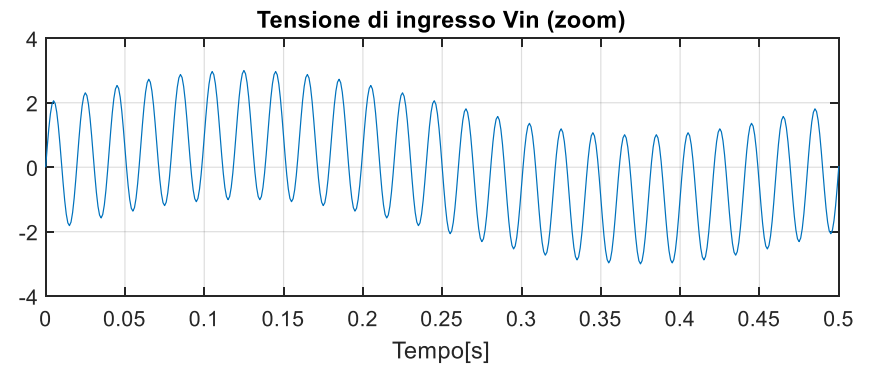
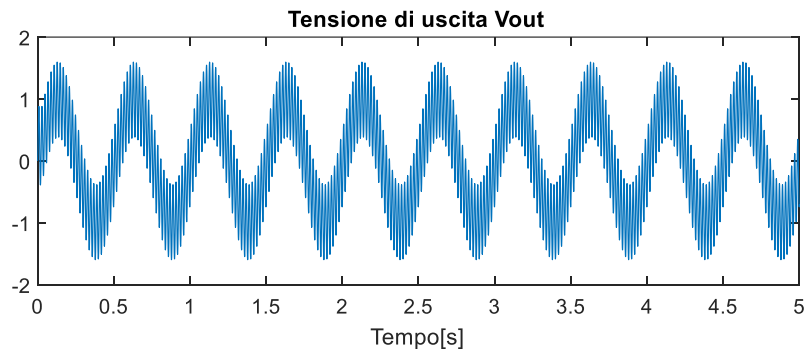
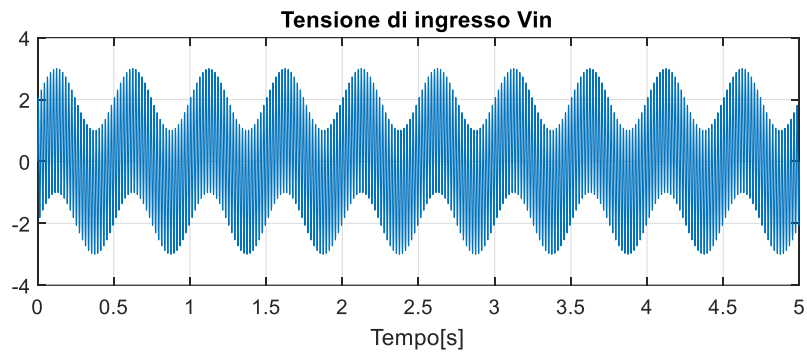
```
A1=1; %ampiezza prima armonica  
f1=2; %frequenza prima armonica [Hz]  
A2=2; %ampiezza seconda armonica  
f2=50;%frequenza seconda armonica [Hz]
```

Avviamo un test simulativo di durata pari a **5 secondi**, e grafichiamo le tensioni di ingresso e di uscita

```
%% TEST 1 & Grafici Vin e Vout
Tsim=5; %Durata del Test 1
sim('FiltroLP_AnalisiSpettrali');
t=Vin.time;
vin=Vin.data;
vout=Vout.Data;

figure(1),
subplot(2,1,1),
plot(t,vin),grid,
title('Tensione di ingresso Vin'),
xlabel('Tempo[s]'),
subplot(2,1,2)
plot(t,vout),
title('Tensione di uscita Vout'),
xlabel('Tempo[s]')

figure(2),
subplot(2,1,1),
plot(t,vin),grid
title('Tensione di ingresso Vin (zoom)'),
xlabel('Tempo[s]'), axis([0 0.5 -4 4])
subplot(2,1,2)
plot(t,vout),grid
title('Tensione di uscita Vout (zoom)'),
xlabel('Tempo[s]'), axis([0 0.5 -2 2])
```



Per realizzare analisi spettrali, inseriamo in fondo allo Script la seguente funzione, in modo che sia possibile utilizzarla all'interno dello script

```
function [freq data]=spettro(t,x)
% determinazione del periodo di campionamento
% NB il segnale deve essere campionato con periodo uniforme
Tc=t(2)-t(1);

% calcolo del vettore delle frequenze
f=0:1/t(length(t)):1/Tc;
f=f';

% calcolo della Fast Fourier Transform
Y=fft(x);

% calcolo della densità spettrale di potenza normalizzata
P=2*abs(Y)/length(Y);

freq=f(1:ceil(length(f)/2));
data=P(1:ceil(length(P)/2));
end
```

La funzione `spettro.m` riceve in ingresso il vettore equispaziato degli istanti di campionamento ed il vettore che contiene i campioni del segnale (i due vettori che, passati come argomento di ingresso alla funzione `plot`, consentono di crearne il grafico).

La funzione `spettro.m` restituisce il vettore delle frequenze (nell'intervallo $[0, 1/T_c]$, in cui T_c è il periodo di campionamento) ed i relativi valori dello spettro di potenza normalizzato.

Un volta eseguita la funzione con la seguente sintassi

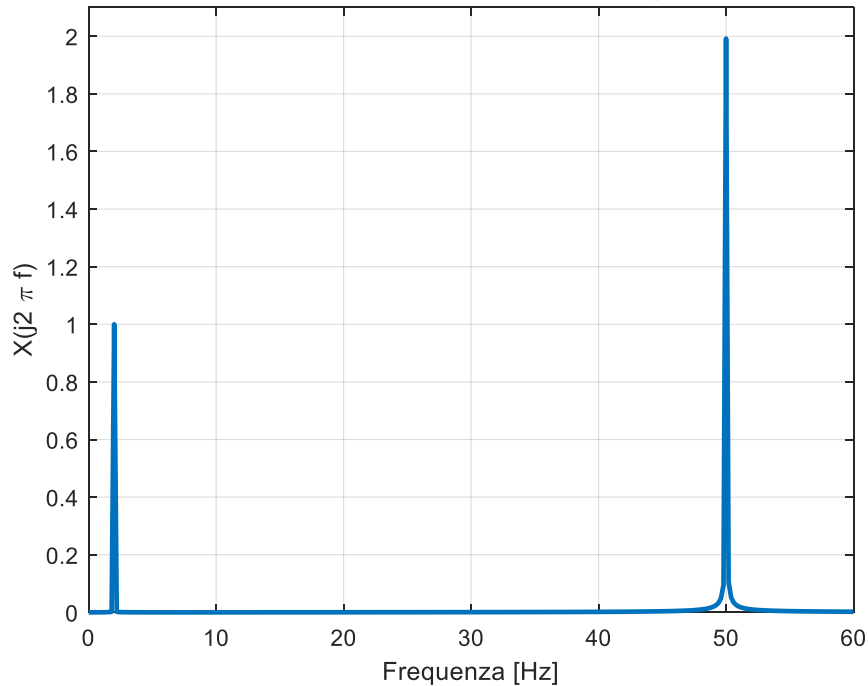
```
[F,X]=spettro(vettore_tempi, vettore_campioni);
```

Si può realizzare il grafico dello spettro normalizzato passando in ingresso i vettori `F` ed `X` alla funzione `plot`

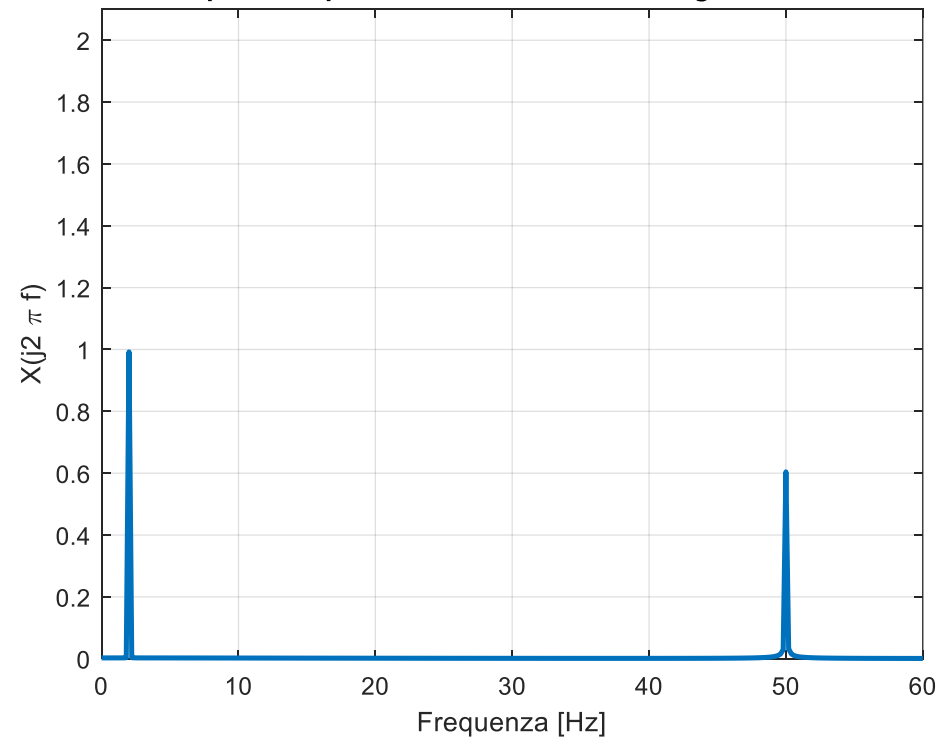
```
%% Analisi spettrale TEST 1
[F,X]=spettro(Vin.time,Vin.data);
figure(3),
plot(F,X,'LineWidth',2),grid
xlabel('Frequenza [Hz]')
ylabel('X(j2 \pi f)')
title('Spettro di potenza normalizzato del segnale Vin')
axis([0 60 0 2.1])

[F,X]=spettro(Vout.time,Vout.data);
figure(4),
plot(F,X,'LineWidth',2),grid
xlabel('Frequenza [Hz]')
ylabel('X(j2 \pi f)')
title('Spettro di potenza normalizzato del segnale Vout')
axis([0 60 0 2.1])
```

Spettro di potenza normalizzato del segnale Vin

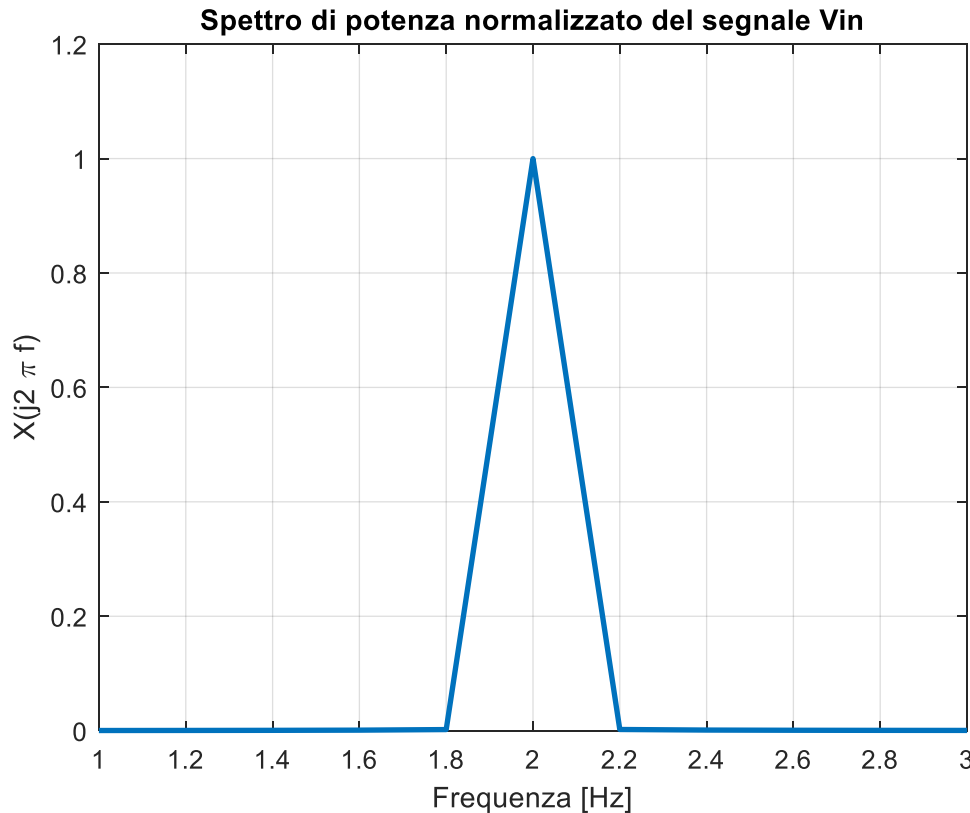


Spettro di potenza normalizzato del segnale Vout



L'analisi spettrale mette in luce le due armoniche del segnale di ingresso, di ampiezza rispettivamente pari ad 1 e 2.
Nel segnale di uscita ritroviamo inalterata l'armonica in bassa frequenza, ed attenuata del 40% l'armonica in alta frequenza

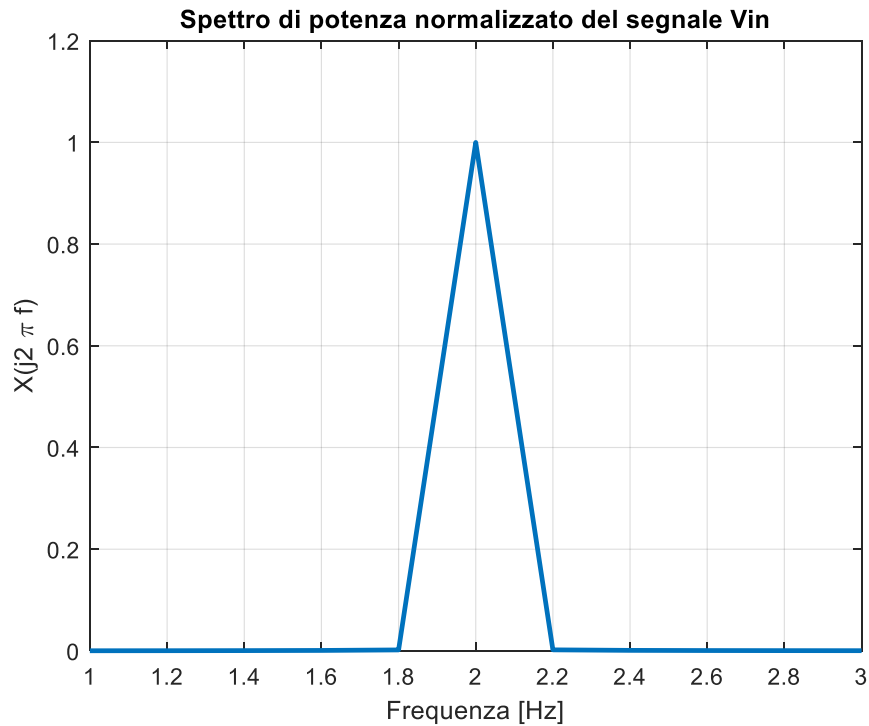
Zoomando su una delle armoniche si nota come l'analisi spettrale abbia restituito uno spettro non nullo in un intorno della frequenza centrale.



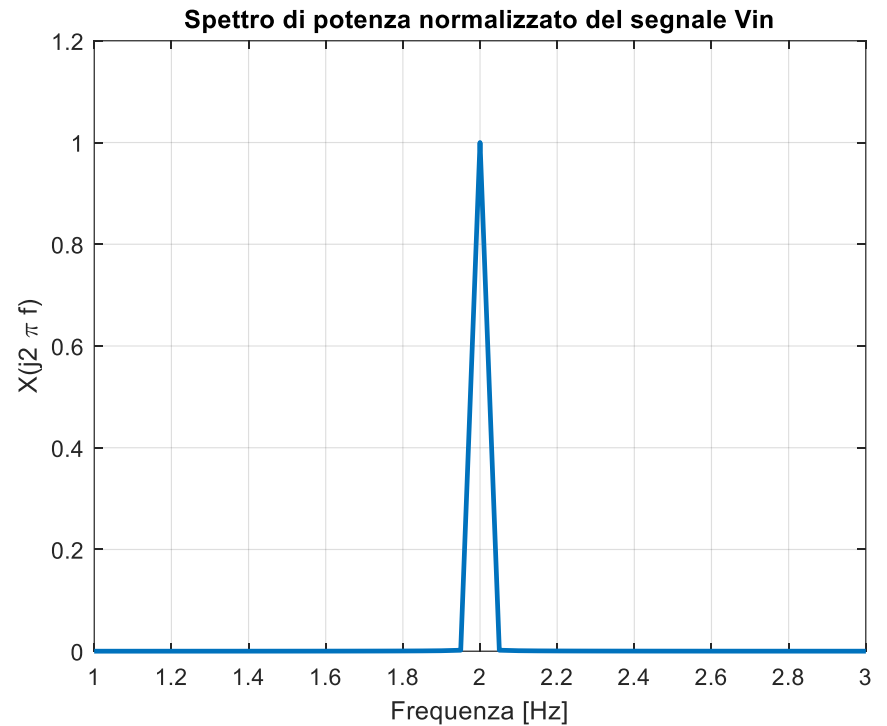
Per ottenere uno spettro più accurato bisogna passare alla funzione spettro uno stream di dati di durata maggiore.

Eseguiamo un nuovo test simulativo stavolta di durata pari a **20 secondi**.

$$T_{sim} = 5s$$



$$T_{sim} = 20s$$



Lo spettro è decisamente più stretto in corrispondenza del test di durata 20s