Corso di Segnali e Sistemi
Ingegneria Elettronica
Università degli Studi di Padova

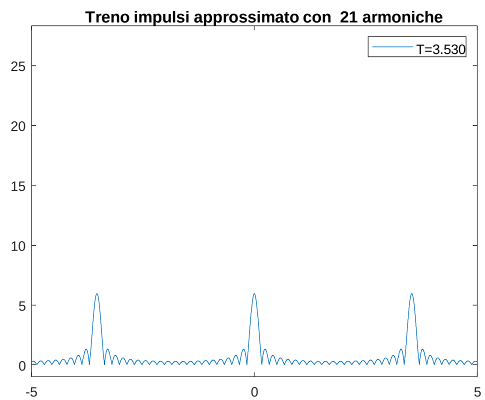
Laboratorio 04

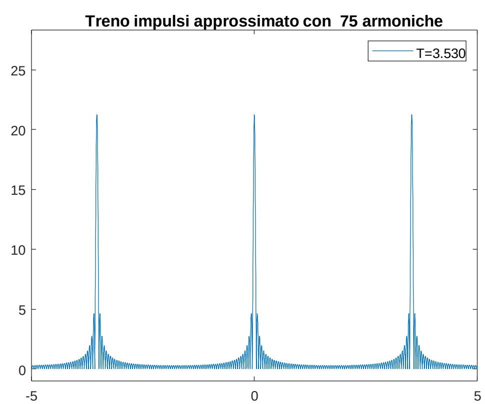
- Serie di Fourier, parte II
- Convoluzione di segnali a supporto limitato

Serie di Fourier del treno d'impulsi

Ricordiamo che il treno d'impulse $x(t) = \sum_{k\in\mathbb{Z}} \delta(t - kT)$ ammette, in senso generalizzato, la serie di Fourier seguente: $\frac{1}{T}\sum_{k\in\mathbb{Z}} e^{jk\omega_0 t}$

Esercizio: Creare uno script Matlab che calcoli le somme parziali $\frac{1}{T}\sum_{k=-M}^{M} e^{jk\omega_0 t}$ e le mostri su un grafico, verificando *empiricamente* che al crescere di M esse tendono al treno d'impulsi






```
%% Serie di Fourier del treno d'impulsi
clear vars
TMax = 5; % Estremo dell'asse dei tempi
Delta = 1e-4; % Passo di campionamento dell'asse dei tempi
t = -TMax:Delta:TMax; % Asse dei tempi
MaxAbsK = 50;% Il numero di armoniche sarà 1+2*MaxAbsK
T = TMax*rand(1); % Periodo: scegliamo un valore casuale tra 0 e 5
                  %NB per ripetibilità, sostituire con T=3 (o altro numero)
w0 = 2*pi/T; % Pulsazione
% Inizializzazione somme parziali con il termine k=0
p = 1/T*ones(size(t));
figure(1);
plot(t,p);
title('Treno impulsi approssimato con 1 armonica')
axis([min(t) max(t) -1 MaxAbsK*(2/T)]); % Per fissare gli assi
for k = 1:MaxAbsK
    % Creiamo le armoniche di indice +k e -k
    armonicaKpos = [redacted]
    armonicaKneg = [redacted]
    % Aggiungiamo le armoniche alla somma parziale
    p = p + [redacted]
    % Aggiornamento figura
    figure(1);
    plot(t,abs(p));
    axis([min(t) max(t) -1 MaxAbsK*(2/T)]); % Per fissare gli assi
    title(sprintf('Treno impulsi approssimato con %3d armoniche', 2*k+1));
    legend(sprintf('T=%4.3f',T));
    pause(0.05);
end
```

Convoluzione

Convoluzione a tempo discreto

Consideriamo due segnali x[n] e y[n] a tempo discreto.
Dalla teoria sappiamo che la loro convoluzione è definita come:

$$x * y(n) = \sum_{k=-\infty}^{+\infty} y(k)x(n-k)$$

Nella pratica non si può eseguire la sommatoria da $-\infty$ a $+\infty$ ma è anche vero che nella pratica tutti i segnali sono a supporto limitato!

Ricordiamo che se il supporto di x è N (per esempio: $x(n) = 0 \ \ \forall n < 0$ e $\forall n > N-1$) e il supporto di y è M , allora la convoluzione ha supporto $N + M - 1$

Convoluzione

Convoluzione a tempo continuo

Consideriamo i segnali a tempo continuo f(t) e g(t).

La convoluzione a tempo continuo è definita da:

$$z(t) = v * w(t) = \int_{-\infty}^{+\infty} v(t-\tau)w(\tau)d\tau$$

Attenzione: in Matlab si può solo approssimare un segnale a tempo continuo scegliendo il passo di campionamento sufficientemente piccolo.
Scegliendo il passo di campionamento **T** sufficientemente piccolo, possiamo scrivere

$$z(t) = \int_{-\infty}^{+\infty} v(t-\tau)w(\tau)d\tau \approx T \sum_{k=-\infty}^{+\infty} v(t-kT)w(kT)$$

Se usiamo lo stesso passo T per campionare anche z si ottiene:

$$z(nT) \approx T \sum_{k=-\infty}^{+\infty} v(nT-kT)w(kT)$$

Se associamo ai tre segnali t.c. z, v, w le successioni dei loro campioni: $\zeta(n) = z(nT), \nu(n) = v(nT), \omega(n) = w(nT)$, si ottiene $\zeta \approx T\, \nu * \omega$ (conv. discreta)

Esercizio 2

Consideriamo due segnali a[n] e b[n] a tempo discreto a supporto limitato.

$$a[n] = \begin{cases} -1 & n = 0 \\ 3 & n = 1 \\ -5 & n = 2 \\ 2 & n = 3 \\ 0 & altrove \end{cases} \qquad b[n] = \begin{cases} 1 & n = 0 \\ 2 & n = 1 \\ -1 & n = 2 \\ 0 & altrove \end{cases}$$

- Calcolare **a mano** la convoluzione discreta $c[n] = (a * b)[n]$
- Verificare il risultato ottenuto con Matlab: per calcolare la convoluzione discreta usare il comando *conv*

Command Window

```
>> conv([-1 3 -5 2] , [1 2 -1])
ans =
    -1     1     2   -11     9    -2
fx >>
```

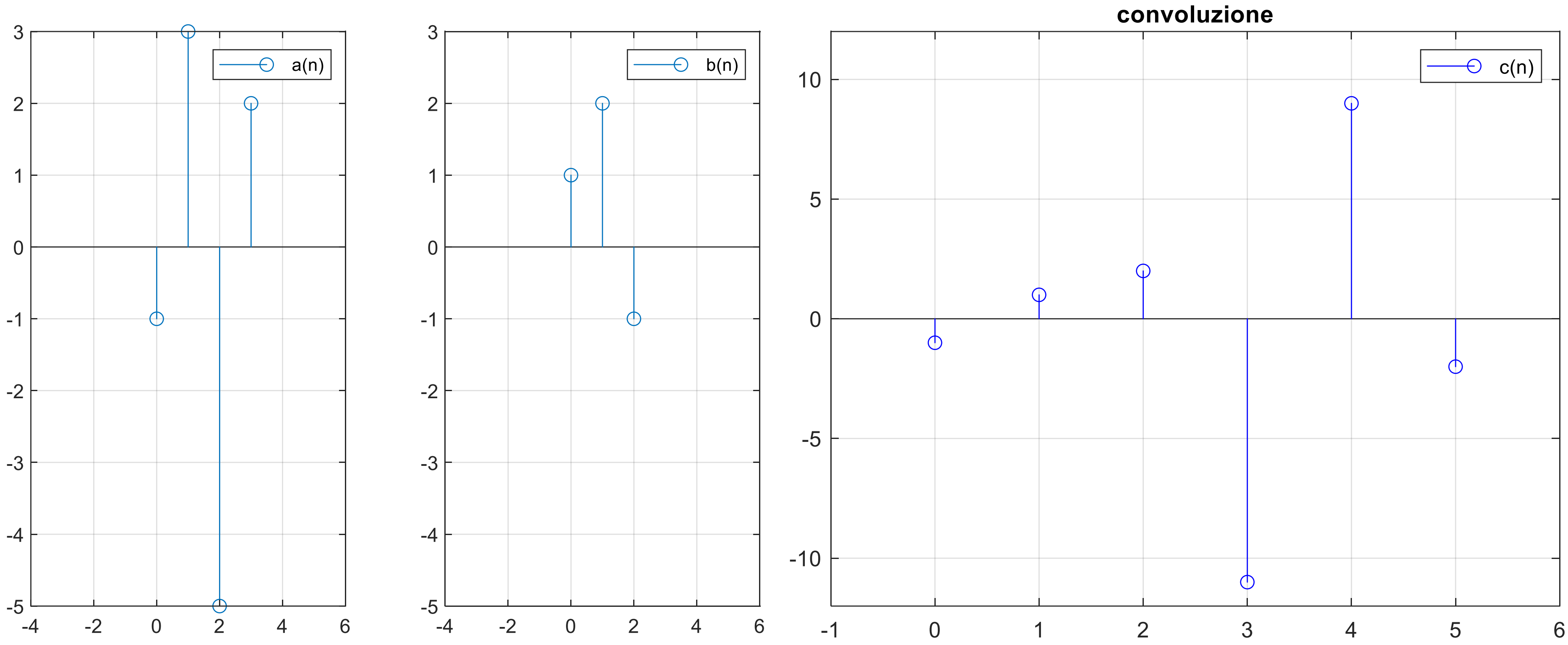
Soluzione

```
close all % chiude le finestre figura aperte
clc % pulisce la command window
% Segnali a e b
a=[-1 3 -5 2];
n_a=0:3;
b=[1 2 -1];
n_b=0:2;
%% Grafici dei segnali di interesse
figure , title ('Segnali di interesse')
subplot (121)
stem (n_a ,a)
axis ([ -4 6 -5 3 ])
legend ('a(n)')
grid on;
subplot (122)
stem (n_b ,b)
axis ([ -4 6 -5 3 ])
legend ('b(n)')
grid on;
%% convoluzione
c = conv (a,b);
%% Grafico della convoluzione c = a*b
figure
n_c=(n_a(1)+n_b(1)):(n_a(end)+n_b(end))
stem (n_c,c,'b')
axis ([ -1 6 -12 12 ])
title ('convoluzione')
legend('c(n)')
grid on;
```

`n_a` e `n_b` sono i supporti di `a` e `b` rispettivamente

Ricordare che, se `n_a` e `n_b` sono i supporti di `a` e `b` il supporto della loro convoluzione si ottiene sommando gli estremi dei due supporti

Soluzione



Esercizio 3

Parte 1: **carta e penna**

- Si consideri un LTI tempo continuo di risposta impulsiva $h(t) = u(t)e^{-at}$, con $a > 0$
- Verificare, applicando la definizione, che la risposta in frequenza del sistema è:

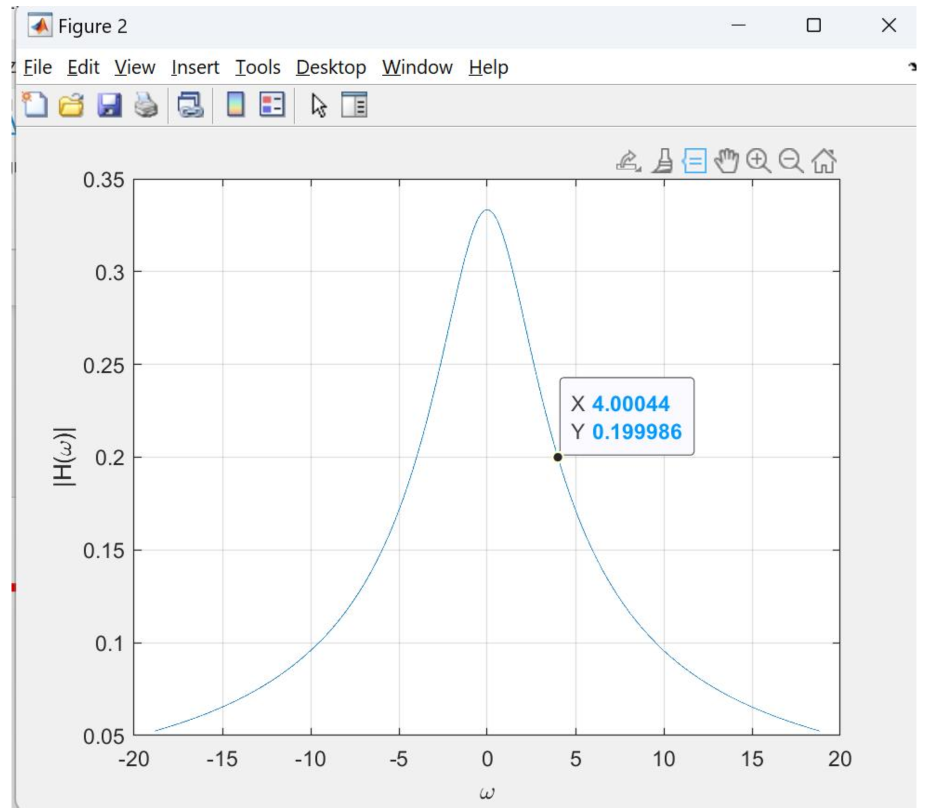
$$H(\omega) = \frac{1}{a + j\omega}$$

- Calcolare $|H(\omega)|$ e tracciarne un grafico qualitativo
  - Il sistema è passa-alto, passa-basso o nessuno dei due?
  - Giustificare la risposta
- Per $a = 3$, calcolare la risposta del sistema a $x(t) = \cos 4t$ NB. Porre $\phi = \angle H(4)$ Non è necessario valutare esplicitamente ϕ. Invece è necessario calcolare $|H(4)|$

Esercizio 3

Parte 2: **Matlab**

- Tracciare $h(t) = u(t)e^{-at}$, con $a = 3$ per $t \in (0{,}10)$
- Per $\omega \in (-6\pi, 6\pi)$, tracciare $|H(\omega)| = \left|\frac{1}{a+j\omega}\right|$
- Usando i Data Tips (menu Tools), verificare che $|H(4)| = 0.2$
- Calcolare per $t \in (0{,}10)$ la convoluzione $y = h * x$ dove $x(t) = \cos 4t \;\; u(t)$
  - *Qual è l'ampiezza di y? Perché?*



NB: utilizzare solo la parte di y della stessa durata di x, per evitare che la convoluzione assuma incorrettamente che x è nullo per $t > 10$

Esercizio 3

```matlab
%% Esercizio 3
%Inizializzazioni
clear vars
T = 1e-3;
t= 0:T:10;
a=3;
%% Risposta impulsiva
h = exp(-a*t);
figure(1);
plot(t,h);
grid; xlabel('t'); ylabel('h(t)')
%% Risposta in frequenza
w = -6*pi:T:6*pi;
H = 1./abs(a+1i*w);
figure(2);
plot(w,H); xlabel('\omega'); ylabel('|H(\omega)|')
grid;
%% Risposta a ingresso sinusoidale
x = cos(4*t); % Attenzione: il vettore ha lunghezza finita
              % Quindi è come se usassimo cos(4t)*rect((t-5)/10)
y = T*conv(h,x);
y=y(1:numel(x)); % USARE SOLO LA PARTE DI Y DELLA STESSA DURATA DI X
figure(3);
plot(t,x,t,y);
xlabel('t'); legend('x(t)=u(t) cos 4t', 'y(t) = h*x (t)')
grid
```