



# UNIVERSITÀ DEGLI STUDI DI PADOVA

Laboratorio 5

Legame tra le differenti trasformate di Fourier

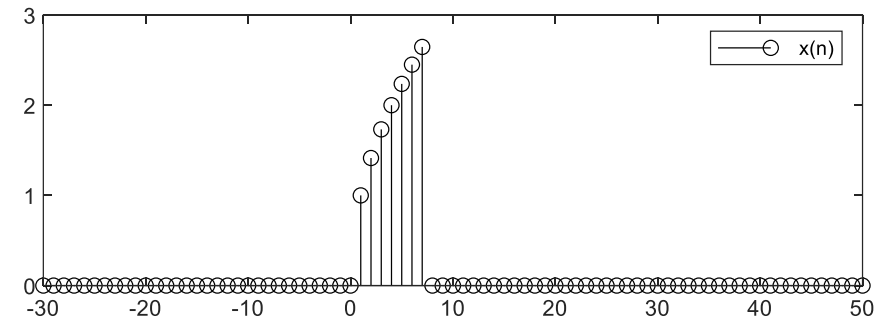


**Obiettivo:** Stimare *numericamente* la TFtd di un segnale tempo discreto usando la FFT (cioè un'implementazione rapida della TFD)

Sia  $x \in \ell^1(\mathbb{Z})$  un segnale tempo-discreto a **supporto finito**:

$$\forall n \notin \{0, 1, \dots, N-1\}, \quad x(n) = 0$$

Per esempio, in figura abbiamo tracciato  $x(n) = \begin{cases} \sqrt{n} & \text{se } 0 \leq n < N \\ 0 & \text{altrimenti} \end{cases}$

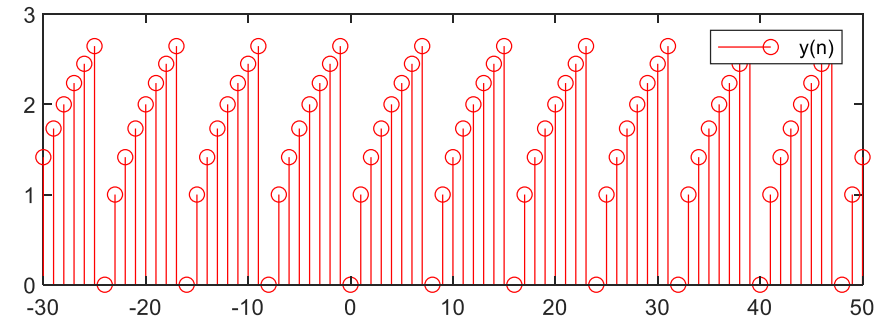


Ricordiamo che la sua TFtd è definita come segue:

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n}$$

Siccome il segnale ha supporto finito, possiamo anche scrivere:

$$X(\omega) = \sum_{n=0}^{N-1} x(n)e^{-j\omega n}$$



Sia adesso  $y$  un **nuovo segnale** tempo discreto, definito come replica periodica di periodo  $N$  dei valori di  $x$  tra  $0$  e  $N-1$   
 $y: n \in \mathbb{Z} \rightarrow x(n \bmod N)$



# Legame tra TFtd e TFD

Posto  $\omega_0 = \frac{2\pi}{N}$ , la **TFD** di  $y$ , indicata con  $Y: k \in \mathbb{Z} \rightarrow \mathbb{C}$ , si può calcolare come segue:

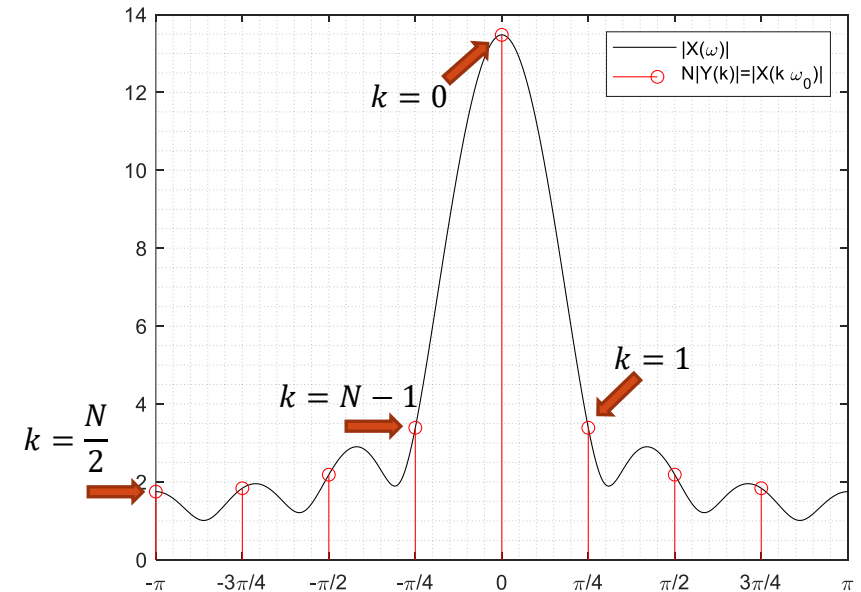
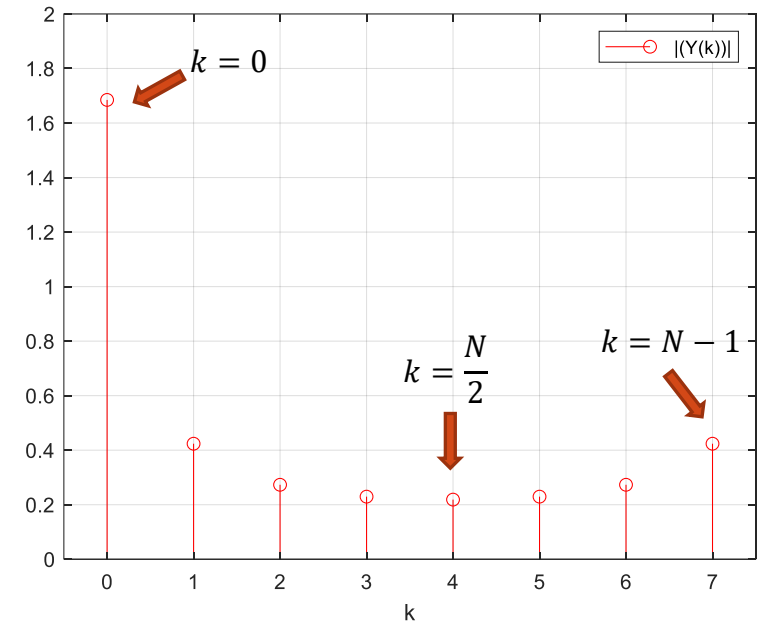
$$Y(k) = \frac{1}{N} \sum_{n=0}^{N-1} y(n) e^{-j\frac{2\pi}{N}kn} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j(k\omega_0)n} = \frac{1}{N} X(k\omega_0)$$

I valori di  $Y(k)$  sono mostrati in alto da soli ed in basso sovrapposti con l'andamento di  $X(\omega)$ , con opportuno posizionamento rispetto all'asse  $\omega$ .

Notiamo che la **TFD** di  $y$  (cioè  $Y(k)$ ) permette di *campionare* la TFtd di  $x$  (cioè  $X(\omega)$ ) con passo  $\omega_0 = \frac{2\pi}{N}$

Nota 1. Il riposizionamento dei campioni di  $Y(k)$  si ottiene osservando che  $Y(0)$  corrisponde alla pulsazione  $\omega = 0$ , che gli altri campioni sono separati di  $\omega_0 = \frac{2\pi}{N}$ , e che per periodicità, quando  $k > \frac{N}{2}$  possiamo riposizionare i campioni a partire da  $\pi$

Nota 2. Gli  $N$  campioni di  $X(\omega)$  non sono sufficienti per ottenere una visualizzazione dettagliata dell'andamento della TFtd (per esempio, non sono visibili i «lobi laterali») anche se ne contengono tutta l'informazione (infatti da essi si può calcolare  $y$ , da cui  $x$  e da cui  $X$ ). Ci chiediamo allora se sia possibile campionare  $X(\omega)$  con un passo più fine per avere una migliore visualizzazione della TFtd, che può essere utile nelle applicazioni





# Lo zero-padding

La soluzione si ottiene con il cosiddetto *zero-padding*.

Sia  $M > N$  e  $z(n) = x(n \bmod M)$

Cioè  $z$  è la periodizzazione di  $x$  che però **include anche  $M - N$  zeri in coda** al supporto. Per questo si parla di *zero-padding*

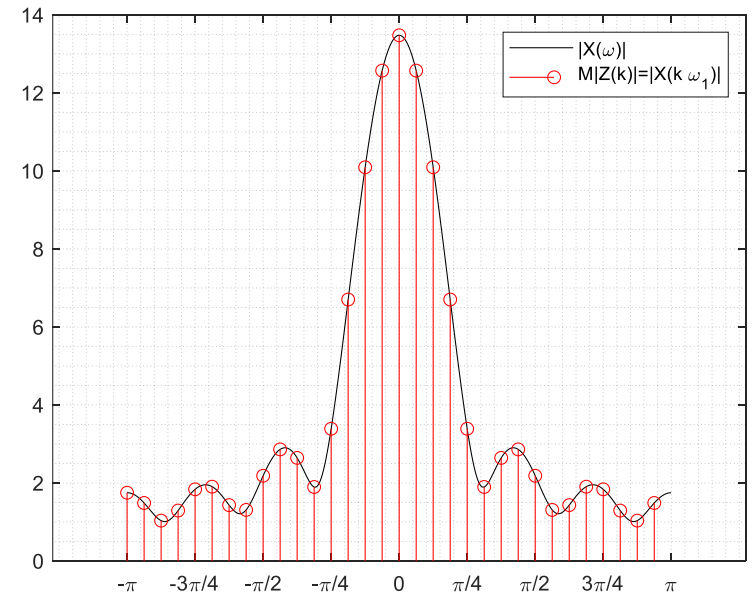
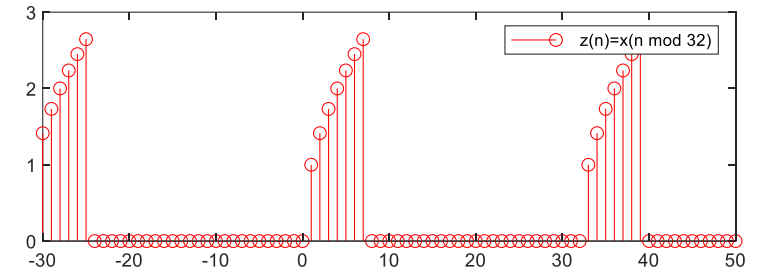
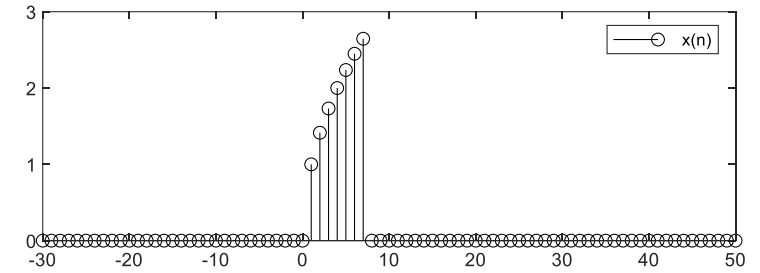
Posto  $\omega_1 = \frac{2\pi}{M}$  la TFD di  $z$  vale:

$$Z(k) = \frac{1}{M} \sum_{n=0}^{M-1} z(n) e^{-j\frac{2\pi}{M}kn} = \frac{1}{M} \sum_{n=0}^{N-1} x(n) e^{-j(k\omega_1)n} = \frac{1}{M} X(k\omega_1)$$

Quindi con lo zero-padding si può *aumentare arbitrariamente la precisione (risoluzione) del campionamento di  $X(\omega)$*

Adesso, i campioni  $Z(k)$  danno già un andamento più preciso di  $X(\omega)$ .

Siccome  $M$  si può prendere grande a piacere, diventa quindi possibile tracciare  $X(\omega)$  con precisione arbitrariamente alta. In effetti, la curva nera è stata ottenuta in Matlab scegliendo  $M = 2048$





## Come effettuare queste operazioni in un ambiente di programmazione come Matlab

Sia `xMatlab` il vettore di dati contenente i valori non nulli di  $x$ :

$$\text{xMatlab} = [x(0) \ x(1) \ x(2), \dots, x(N-1)];$$

Matlab consente di effettuare con un unico comando lo zero-padding e la TFD. Assegnato un opportuno valore di  $M$  calcoliamo

$$\text{Zmatlab} = \mathbf{fft}(\text{xMatlab}, M);$$

Il comando `fft` calcola la TFD di `xMatlab` aggiungendo  $M-N$  zeri alla fine di `xMatlab` in modo da ottenere implicitamente il segnale  $z$ , di cui viene poi calcolata la TFD. L'algoritmo è particolarmente veloce se si sceglie una potenza di 2 come valore di  $M$ .

Si noti che Matlab usa una definizione leggermente diversa di TFD: non c'è la divisione per  $M$ . Quindi si ha che

$$\begin{aligned} \text{ZMatlab} = [M \cdot Z(0) \ M \cdot Z(1), \dots, M \cdot Z(M-1)] = \\ [X(0) \ X(\omega_1) \ X(2\omega_1), \dots, X((M-1)\omega_1)]; \end{aligned}$$

Per tracciare il grafico di  $Z(k) = X(k\omega_1)$  rimane da determinare il campionamento dell'asse  $\omega$ :

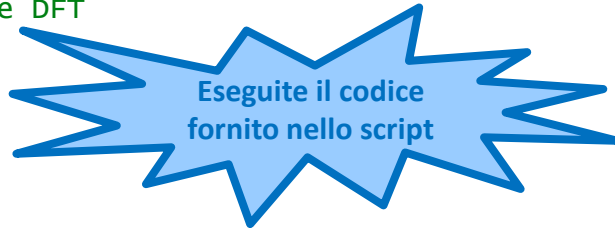
$$\text{omega1} = 2 * \text{pi} / M;$$

$$\text{w} = 0 : \text{omega1} : (M-1) * \text{omega1}$$

$$\text{plot}(\text{w}, \text{abs}(\text{ZMatlab}));$$

Proviamo in Matlab quanto detto: aprire il file `zero_padding.m`

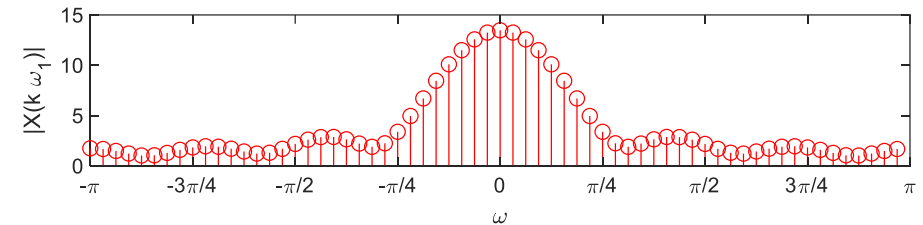
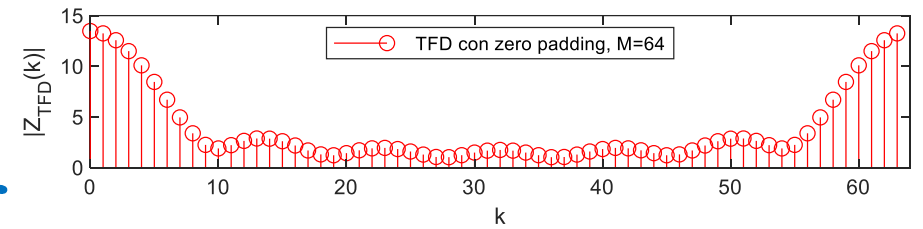
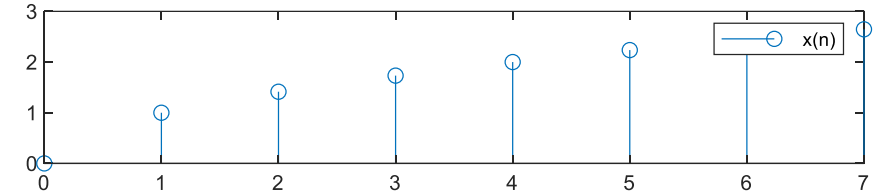
```
clc; clearvars; close all;
% Un segnale a tempo discreto e supporto 0 ... N-1
N=8; n = 0:(N-1); xMatlab = sqrt(n);
figure; subplot(311); stem(n,xMatlab); legend('x(n)')
% TFD con zero-padding
M=2^6;
ZMatlab = fft(xMatlab,M); % Effettua zero-padding e DFT
subplot(312); stem(0:M-1,abs(ZMatlab),'r');
% [...]
subplot(313);
omega1 = 2*pi/M; w = -pi:omega1: pi-omega1;
stem(w, fftshift(abs(ZMatlab)),'r');
% [...]
```



Abbiamo riportato qui la parte principale dello script. Notare come si passa dalla TFD alla TFtd:

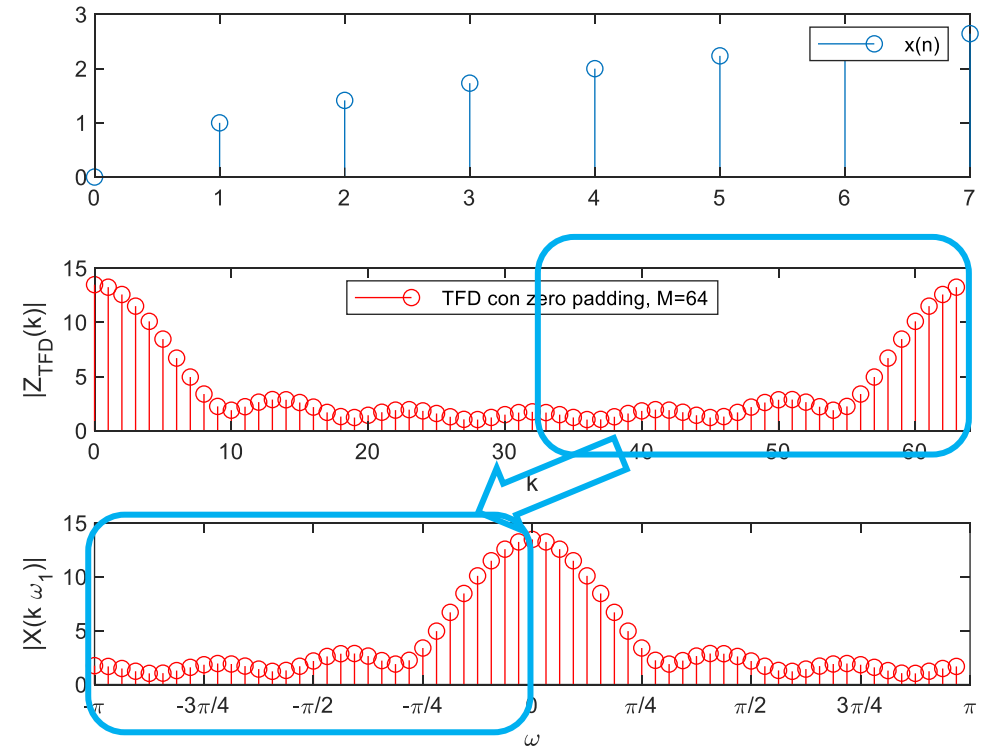
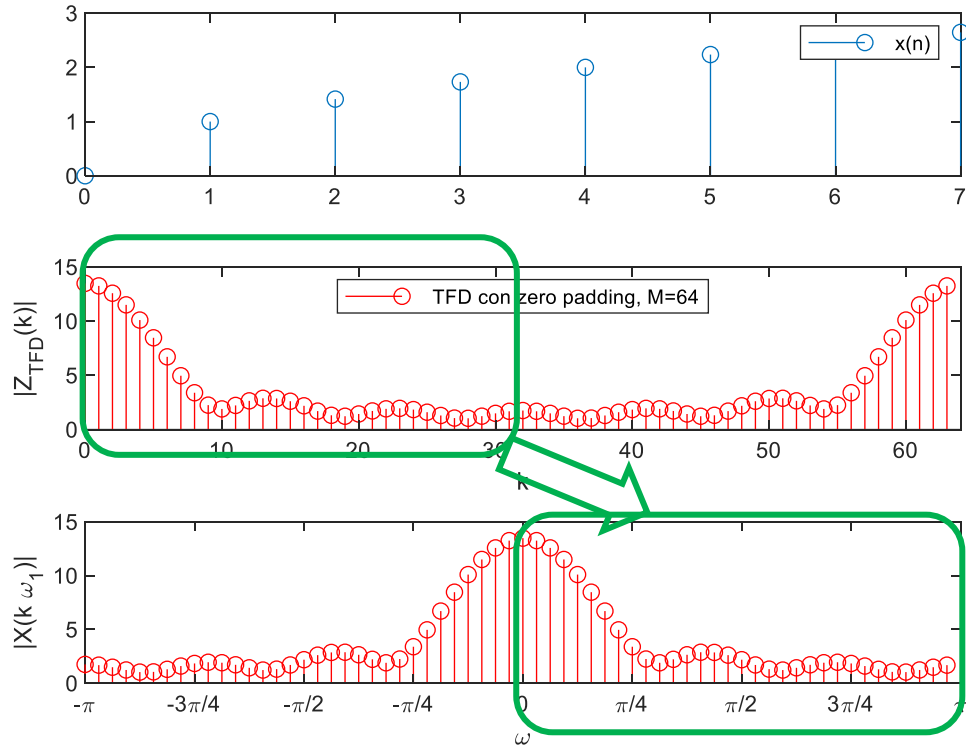
- 1) Riposizionamento dell'asse delle ascisse. I valori di  $\omega$  coprono l'intervallo  $(-\pi, \pi)$  campionandolo in  $M$  punti. Con il comando `omega1 = 2*pi/M` definiamo il passo di campionamento delle pulsazioni. Poi creiamo un vettore di  $M$  punti: `w = -pi:omega1: pi-omega1`; Notiamo che l'ultimo punto dell'intervallo deve essere  $\pi - \omega_1$  e non  $\pi$  altrimenti il vettore generato avrebbe  $M + 1$  punti (fare la prova in Matlab)
- 2) Con il comando `fftshift` si spostano i valori di  $|Z|$  (si mette la metà destra del vettore a sinistra e viceversa)

**Esercizio 1. Testare lo script `zero_padding.m` (che contiene il codice mostrato in questa slide) provando diversi valori di  $N$  e  $M$**





# Effetto di fftshift





### ESERCIZIO 2: Stima della frequenza di un segnale esponenziale immaginario puro

Supponiamo di poter acquisire  $N$  campioni di un segnale esponenziale immaginario puro:  $s(n) = e^{j\omega_1 n}$

*Perché non consideriamo il caso più realistico di  $s(n) = \cos \omega_1 n$ ? Perché usando l'esponenziale si semplificano alcune parti del problema, senza modificarne gli elementi principali*

L'obiettivo di questo esercizio è quello di stimare il valore di  $\omega_1$  usando la **TFtd**

È fondamentale notare che non possiamo calcolare direttamente la TFtd di  $s(n)$  perché non abbiamo a disposizione tutti i campioni, ma soltanto un numero finito. Questo è equivalente a moltiplicare il segnale  $s(n)$  per la funzione indicatrice di  $\{0, 1, \dots, N - 1\}$ .

Sia allora  $x(n) = w(n)s(n)$ , dove  $w(n) = \begin{cases} 1 & \text{se } 0 \leq n < N \\ 0 & \text{altrimenti} \end{cases}$  ( $w$  richiama il termine *window*, la finestra di osservazione del segnale)

Considereremo il problema di stimare  $\omega_1$  a partire dal segnale  $x(n)$ . I campioni di  $x$  sono memorizzati nella variabile `xCamp` contenuta nel file





## Domande:

2.1 Calcolare la TFD di  $x$  in funzione della TFD di  $w$  e mostrare che  $X(\omega) = W(\omega - \omega_1)$

2.2 Posto  $y_K(n) = \begin{cases} \frac{1}{2K+1} & \text{se } |n| \leq K \\ 0 & \text{altrimenti} \end{cases}$ , abbiamo visto in classe che  $Y_K(\omega) = \frac{\sin\left(\frac{2K+1}{2}\omega\right)}{\sin\frac{\omega}{2}}$ . Sia ora  $N$  **dispari**.

Sfruttare tale risultato per mostrare che  $|W(\omega)| = N \frac{|\sin\left(\frac{N}{2}\omega\right)|}{|\sin\frac{\omega}{2}|}$ . A tal fine si osservi che  $w(n) = y_{\frac{N-1}{2}}\left(n - \frac{N-1}{2}\right)$

**NB.**  $|W(\omega)| = N \frac{|\sin\left(\frac{N}{2}\omega\right)|}{|\sin\frac{\omega}{2}|}$  anche se  $N$  è pari (senza dim.)

2.3 Tracciare  $|W(\omega)|$  in Matlab. Qual è l'ampiezza del lobo principale?

2.4 Nel file `lab5_ex2.mat` i campioni di  $x$  sono memorizzati nella variabile `xCamp`. Inoltre il file contiene il valore di  $N$  in `N` ed il valore di  $\omega_1$  in `omega1`

Caricare i dati dal file e individuare  $\omega_1$  come la pulsazione corrispondente al valore massimo di  $|X(\omega)|$ .

Come influenza la precisione del risultato la scelta dei parametri  $N$  ed  $M$ ?



### Domande:

2.1 Calcolare la Tftd di  $x$  in funzione della Tftd di  $w$  e mostrare che  $X(\omega) = W(\omega - \omega_1)$

Usando la proprietà di modulazione,  $x(n) = w(n)e^{j\omega_1 n} \Rightarrow X(\omega) = W(\omega - \omega_1)$

2.2 Mostrare che  $|W(\omega)| = N \frac{|\sin(\frac{N}{2}\omega)|}{|\sin\frac{\omega}{2}|}$ . A tal fine si osservi che  $w(n) = y_{\frac{N-1}{2}}\left(n - \frac{N-1}{2}\right)$

In effetti,  $y_{\frac{N-1}{2}}(n)$  è non nullo sugli  $N$  campioni da  $-\frac{N-1}{2}$  a  $\frac{N-1}{2}$ . Per cui, ritardando tale segnale di  $\frac{N-1}{2}$  si ottiene un segnale non nullo tra 0 e  $N - 1$ . Moltiplicando per  $N$  si ottiene esattamente  $w(n)$

Adesso che  $w(n) = Ny_{\frac{N-1}{2}}\left(n - \frac{N-1}{2}\right)$ , si ha, dalla proprietà del ritardo:

$$W(\omega) = NY_{\frac{N-1}{2}}(\omega)e^{-j\frac{N-1}{2}\omega}$$

$$|W(\omega)| = N \left| Y_{\frac{N-1}{2}}(\omega) \right| = N \left| \frac{\sin\left(\frac{N}{2}\omega\right)}{\sin\frac{\omega}{2}} \right|$$



## Domande:

### 2.3 Tracciare $|W(\omega)|$ in Matlab

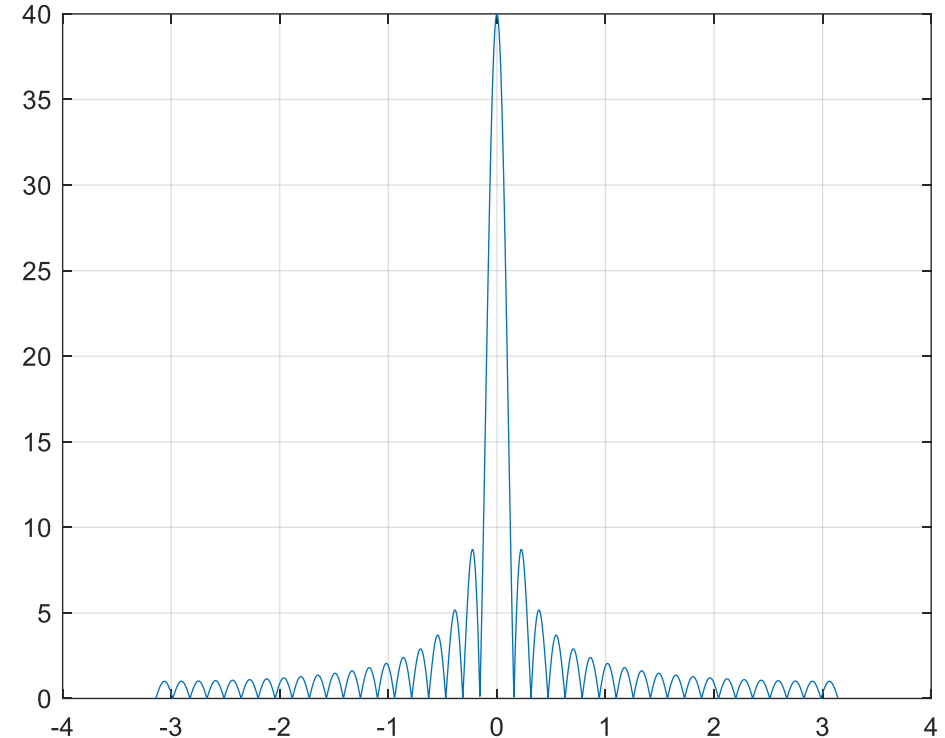
```
% Tftd dell'esponenziale imm. puro finestrato  
N=40; % scegliamo dei valori per i parametri  
omega = -pi:1e-3:pi;  
W = N * sin(N/2 * omega)./sin(omega/2);  
figure; plot(omega,abs(W)); grid;
```

L'ampiezza del lobo principale è la distanza tra i due zeri più piccoli in valore assoluto. Gli zeri di  $W$  sono dati da:

$$\frac{N}{2}\omega = k\pi, \text{ per } k \neq 0 \Leftrightarrow \omega = k \frac{2\pi}{N} \Rightarrow \Delta = \frac{4\pi}{N}$$

È chiaro che, siccome  $X(\omega) = W(\omega - \omega_1)$ , il massimo di  $X$  corrisponde a  $\omega_1$ .

A sua volta, tale massimo è all'interno del lobo principale. Se aumentiamo  $N$  il lobo principale diventa sempre più stretto. Se invece stimiamo  $X$  con la FFT, aumentando lo zero-padding ( $M$ ), campioniamo  $X$  con precisione crescente





## Soluzione

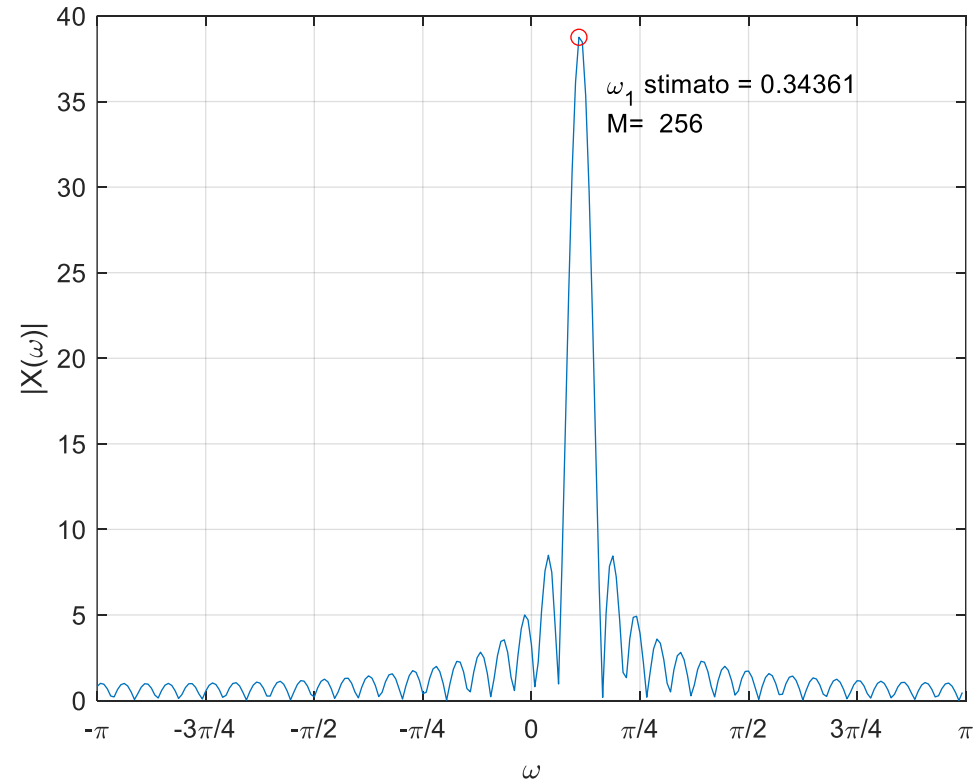
Ecco il codice per stimare  $\omega_1$ :

```
%% stima pulsazione
load lab5

% Zero padding
M=2^10;
% Asse delle pulsazioni campionato a 2pi/M e centrato in zero:
step = 2*pi/M;
omegaFFT = -pi: step: (pi - step);
% Modulo della DTFT campionata e centrata sulla pulsazione zero
X_FFT = fftshift(abs(fft(xCamp,M))) ;

% Trovare il max di |X|
[XM, k] = max(X_FFT); % k è l'indice del valore max
omega1_stimato = omegaFFT(k);
```

Se aumentiamo  $N$  andiamo a cambiare la forma di  $X(\omega)$  perché restringiamo il lobo principale  
Se però il numero di campioni  $N$  è assegnato, possiamo comunque migliorare la precisione della stima aumentando  $M$ , perché riduciamo il passo di campionamento di  $X$





## ESERCIZIO 3: Stima della frequenza di un segnale sinusoidale

Scaricare dal sito di e-learning il file `lab05_ex3.mat`

Adesso il file contiene i campioni di un segnale **sinusoidale**,  $s(t) = \cos \omega_1 t$

Quindi questa volta il segnale a supporto finito  $x$  è:  $x(n) = w(n) \cos \omega_1 n$

3.1 Determinare un'espressione di  $X(\omega)$  in funzione di  $W(\omega)$  e un'espressione approssimata di  $|X(\omega)|$

3.2 Scelto il valore di zero-padding  $M$ , calcolare e tracciare la stima di  $|X(\omega)|$  nell'intervallo  $\left(-\pi, \pi \frac{M-2}{M}\right)$

3.3 Stimare  $\omega_1$  in corrispondenza del massimo di  $|X(\omega)|$ , come nel caso precedente

3.4 In questo caso, i lobi secondari di  $W(\omega + \omega_1)$  interferiscono con la posizione del picco di  $X$ . Un modo per risolvere il problema consiste nel cambiare la forma della finestra  $w(n)$ . Invece di prendere una finestra rettangolare, possiamo prendere una finestra la cui trasformata  $\widehat{W}(\omega)$  abbia un decadimento più rapido.

Provare per esempio ad usare la *finestra di Hamming*:

$$\widehat{w}(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right)$$

Tracciare  $\widehat{w}(n)$ ,  $|\widehat{W}(\omega)|$ . Infine usare la nuova finestra per modulare i campioni e stimare la pulsazione



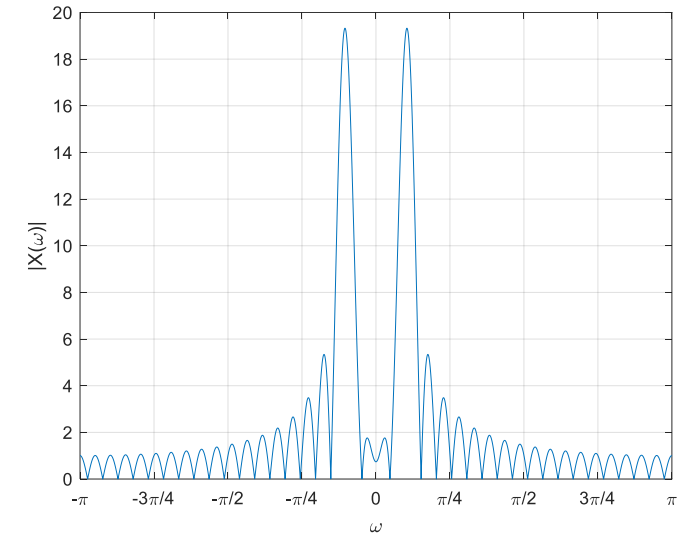
## ESERCIZIO 3: Stima della frequenza di un segnale sinusoidale

3.1 Determinare un'espressione di  $X(\omega)$  in funzione di  $W(\omega)$  e un'espressione approssimata di  $|X(\omega)|$

$$X(\omega) = \frac{1}{2} [W(\omega - \omega_1) + W(\omega + \omega_1)]$$

Se i lobi secondari di  $W$  decadono abbastanza rapidamente, possiamo dire che, per  $\omega > 0$ ,  $X(\omega) \approx W(\omega - \omega_1)$  e quindi  $|X(\omega)| \approx |W(\omega - \omega_1)|$ . Per le pulsazioni negative usiamo la simmetria (il modulo della TFtd di un segnale reale è pari)

3.2 Scelto il valore di zero-padding  $M$ , calcolare e tracciare la stima di  $|X(\omega)|$  nell'intervallo  $(-\pi, \pi \frac{M-2}{M})$



```
load lab5_ex3
```

```
% Zero padding
```

```
M=2^14;
```

```
% Asse delle pulsazioni campionato a 2pi/M e centrato in zero:
```

```
step = 2*pi/M;
```

```
omegaFFT = -pi: step: (pi - step);
```

```
% Modulo della DTFT campionata e centrata sulla pulsazione zero
```

```
X_FFT = fftshift(abs(fft(xCamp,M)));
```

```
figure(1); plot(omegaFFT,X_FFT); grid;
```



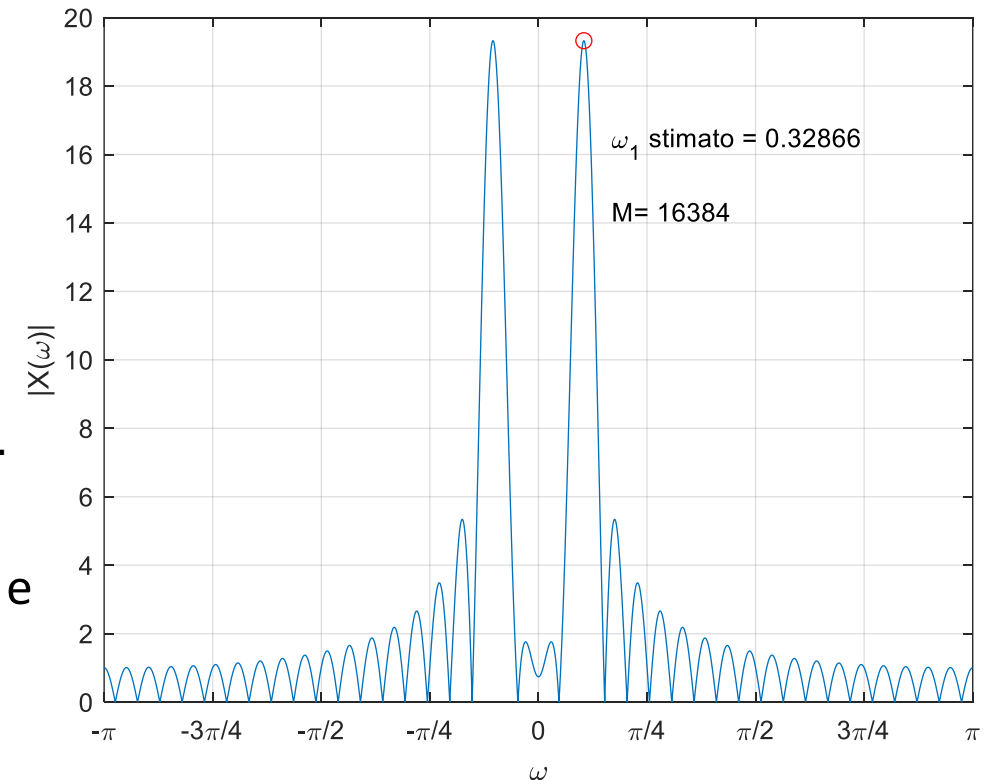
## ESERCIZIO 3: Stima della frequenza di un segnale sinusoidale

3.3 Stimare  $\omega_1$  in corrispondenza del massimo di  $|X(\omega)|$ , come nel caso precedente

```
% Trovare il max di |X|  
step = 2*pi/M;  
omegaFFT = -pi: step: (pi - step);  
X_FFT = fftshift(abs(fft(xCamp,M))) ;  
  
[XM, k] = max(X_FFT); % k è l'indice del valore max  
omega1_stimato = abs(omegaFFT(k));  
  
fprintf('w1 = %1.5f\nw1 stimato = %1.5f\nErrore = %5.4f rad\n', ...  
        omega1, omega1_stimato, abs(omega1-omega1_stimato));
```

In questo caso, i lobi secondari di  $W(\omega + \omega_1)$  non sono trascurabili e possono indurre un errore sulla posizione del picco di  $X$ , che non corrisponde più esattamente a  $\omega_1$

In questo caso, il vero valore di  $\omega_1$  è 0.31623





## ESERCIZIO 3: Stima della frequenza di un segnale sinusoidale

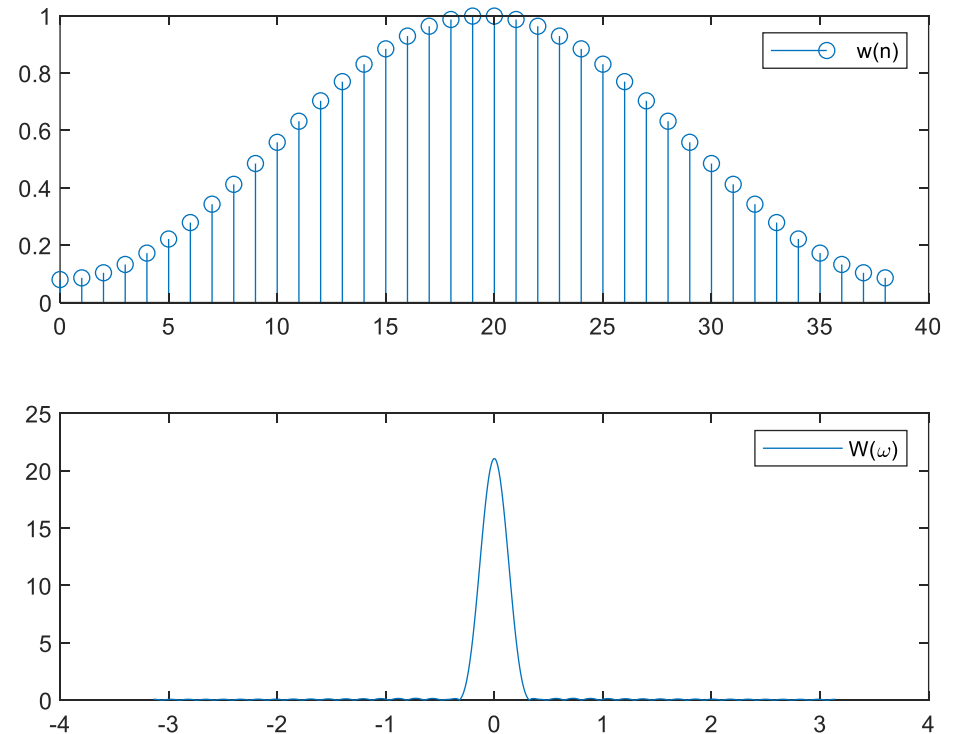
3.4 Data la *finestra di Hamming*:

$$\hat{w}(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right)$$

Tracciare  $\hat{w}(n)$ ,  $|\hat{W}(\omega)|$ .

```
%% Hamming
N=numel(xCamp);
n = 0:N-1; M=2^12;
wHamming = 0.54- 0.46*cos(2*pi*n/N);
figure;
subplot(211); stem(n,wHamming); legend('w(n)')
W = fft(wHamming, M);
step = 2*pi/M;
omegaFFT = -pi: step: (pi - step);
subplot(212), plot(omegaFFT, fftshift(abs(W)));
legend('W(\omega)')
```

Si osserva che il lobo principale di  $\hat{W}$  è più largo rispetto al caso di finestra rettangolare, ma i lobi secondari sono molto più attenuati







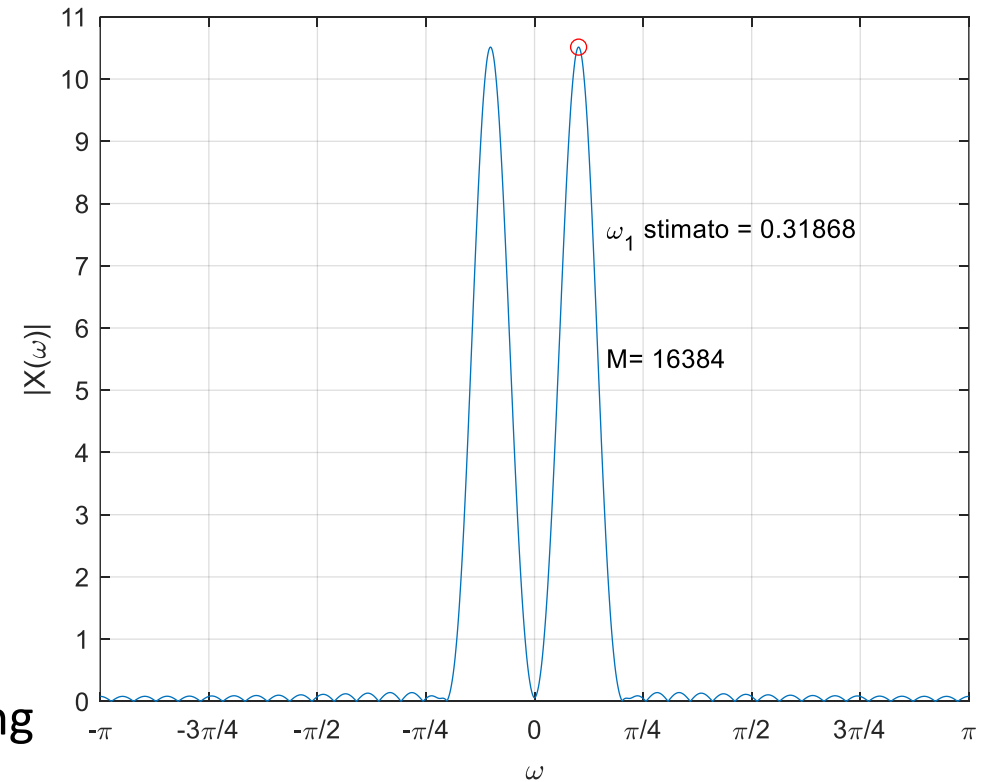
## ESERCIZIO 3: Stima della frequenza di un segnale sinusoidale

### 3.4 Usare la nuova finestra per modulare i campioni e stimare la pulsazione

```
% Zero padding
M=2^14;
step = 2*pi/M; omegaFFT = -pi: step: (pi - step);
X_FFT = fftshift(abs(fft(xCamp.*wHamming,M))) ;

figure; plot(omegaFFT,X_FFT);
xlabel('\omega'); ylabel('|X(\omega)|')
set(gca,'XTick', [-pi, -3*pi/4 -pi/2, -pi/4, 0, pi/4, pi/2, 3*pi/4, pi])
set(gca,'XTickLabel', {'-\pi', '-3\pi/4', '-\pi/2', '-\pi/4', '0', ...
'\pi/4', '\pi/2', '3\pi/4', '\pi'})
axis([-pi pi 0 11]); grid

% Trovare il max di |X|
[XM, k] = max(X_FFT); % k è l'indice del valore max
omega1_stimato = abs(omegaFFT(k));
```



La nuova stima è migliore grazie all'usa della finestra di Hamming



# Esercizio 4 - da fare dopo la lezione sulla formula di Poisson generalizzata

Sia  $x$  un segnale che soddisfa le ipotesi della formula di Poisson, a banda limitata in  $-\omega_M, \omega_M$

Sia  $\hat{w}(n)$  la successione dei campioni presi a passo  $T_C$ :  $\hat{w}(n) = x(nT_C)$ , per  $n = 0, 1, \dots, N - 1$

Poniamo  $\omega_0 = \frac{\pi}{T_C}$ . Se  $\omega_0 \geq \omega_M$  allora  $T_C \leq \frac{\pi}{\omega_M}$  e quindi non c'è aliasing. Si ha dunque:

$$\forall \omega \in (-\pi, \pi), \hat{W}(\omega) = \frac{1}{T_C} X\left(\frac{\omega}{T_C}\right) \Leftrightarrow \forall \omega \in (-\omega_0, \omega_0), X(\omega) = T_C \hat{W}(T_C \omega)$$

Allora, se chiamiamo  $Y(k)$  il  $k$ -esimo valore della TFD di  $\hat{w}(n) = x(nT_C)$  periodizzato, ritroviamo  $Y(k) =$

$\hat{W}\left(k \frac{2\pi}{M}\right) = \frac{1}{T_C} X\left(\frac{k}{M} \frac{2\pi}{T_C}\right)$ . In altre parole, **calcolando la TFD sui campioni di  $x$  con zero-padding, si ottengono i**

**campioni di  $X(\omega)$  in  $(-\omega_0, \omega_0)$  con passo di campionamento (della pulsazione)  $\frac{2\omega_0}{M}$**

In Matlab, se i campioni di  $x$  sono in `xCamp`

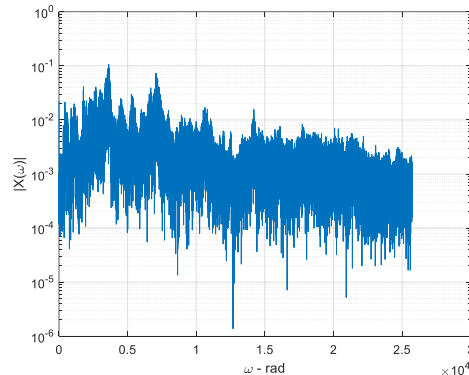
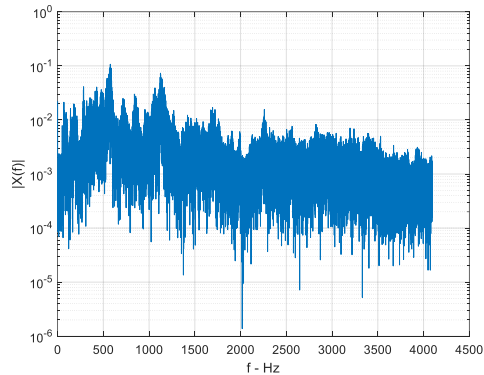
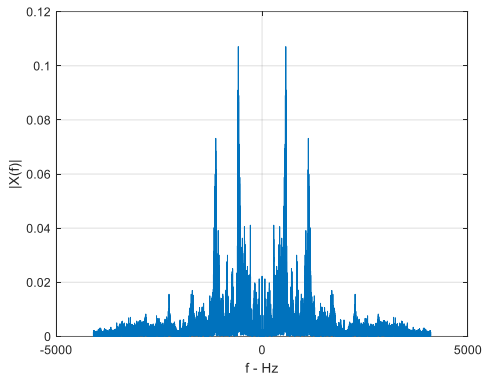
```
X = TC*fftshift(fft(xCamp, M));  
omega0 = pi/TC; step = (2*omega0)/M;  
omega = -omega0: step : (omega0-step);  
plot(omega, abs(X));  
% in frequenza:  
fc = 1/TC;  
F = -fc/2: fc/M: fc/2*(1 - 2/M);  
plot(F, abs(X));
```



# Esercizio 4

Esempio Nel file `handel.mat` si trova un segnale audio campionato. I campioni si trovano nella variabile `y`, mentre la frequenza di campionamento è in `Fs`.

Il seguente script legge i dati, riproduce il suono, e mostra lo spettro del segnale campionato usando la frequenza come asse delle ascisse. Per passare alla pulsazione, basta moltiplicare `F` per  $2\pi$



```
close all
clear variables

load("handel.mat"); % Carica un suono campionato
whos
%sound(y,Fs); % Rimuovere il commento per ascoltare
```

```
M = 2^nextpow2(numel(y)); % M>N
TC = 1/Fs; % Periodo di campionamento
% Spesso il fattore di scala delle ordinate TC
% è trascurato
%-----
Y = TC*fft(y, M);
F = -Fs/2 : Fs/M : (Fs/2-Fs/M);
%-----
```

```
figure(1);
plot(F,abs(fftshift(Y))); grid
xlabel('f - Hz'); ylabel('|X(f)|');
```

```
% Per un segnale reale basta tracciare |X|
% per frequenze positive
Fpos = F(M/2+1:end);
Ypos = Y(1:M/2);
figure(2);
% Spesso si usa una scala log sulle ordinate:
semilogy(Fpos,abs(Ypos)); grid
xlabel('f - Hz'); ylabel('|X(f)|');
```

```
% Grafico di |X(omega)|
figure(3);
semilogy(2*pi*Fpos,abs(Ypos)); grid
xlabel('\omega - rad'); ylabel('|X(\omega)|');
```