UNIVERSITÀ DEGLI STUDI DI PADOVA

Laboratorio 3
Legame tra le differenti trasformate di Fourier

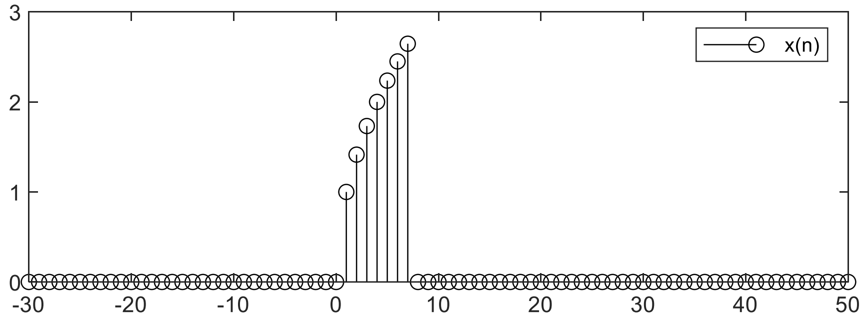
Legame tra TFtd e TFD

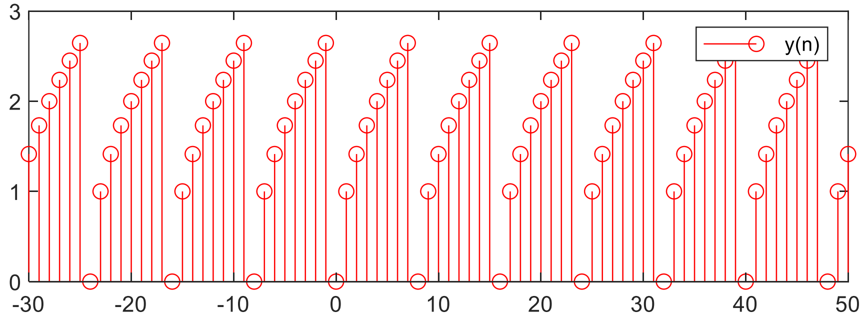
Obiettivo: Stimare *numericamente* la TFtd di un segnale tempo discreto usando la FFT (cioè un'implementazione rapida della TFD)

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

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

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





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 $N-1$

$$y: n \in \mathbb{Z} \to x(n \bmod N)$$

Legame tra TFtd e TFD

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

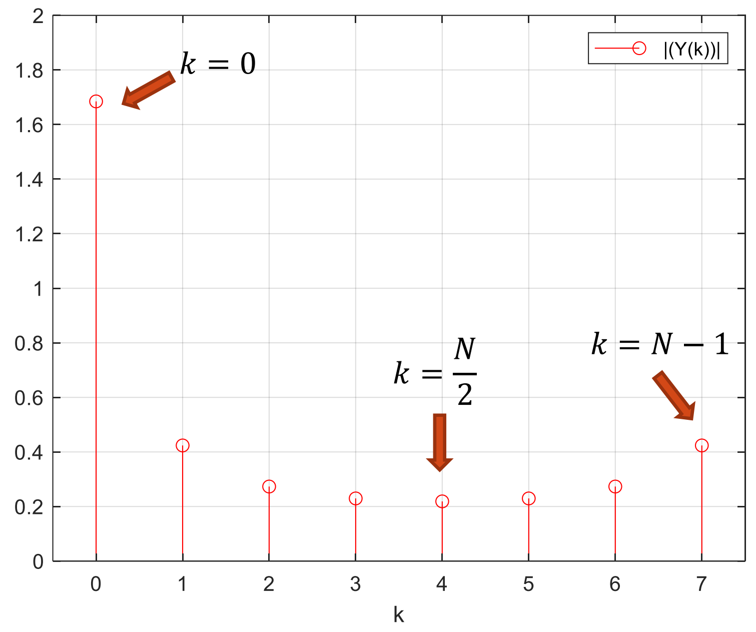
$$Y(k) = \frac{1}{N}\sum_{n=0}^{N-1} y(n) e^{-j\frac{2\pi}{N}kn} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\, e^{-j(k\omega_0)n} = \frac{1}{N} X(k\omega_0)$$

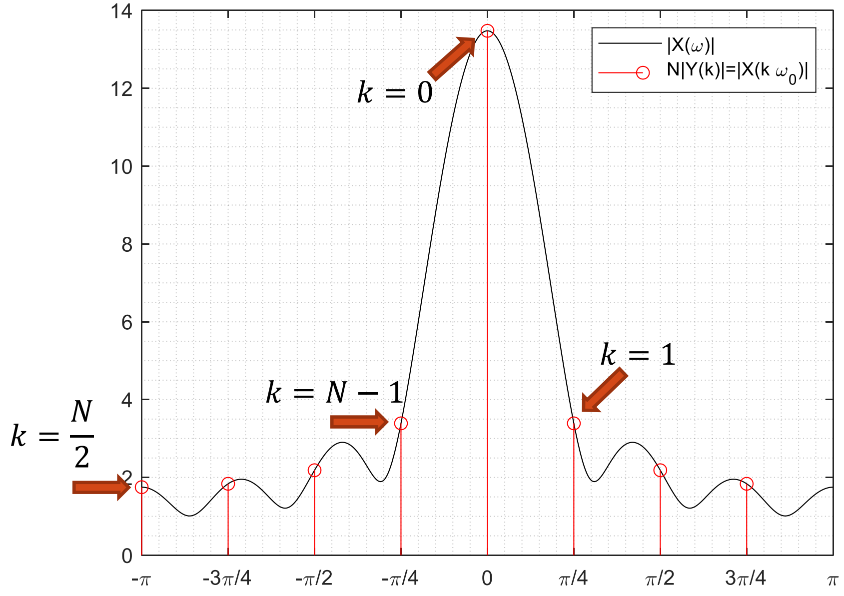
I valori di $N \cdot Y(k)$ sono mostrati in alto da soli ed in basso sovrapposti con l'andamento di $X(\omega)$, con opportuno posizionamento rispetto all'asse ω.

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 $Y(k)$ nella seconda figura 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 π

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





Università degli Studi di Padova

Lo zero-padding

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

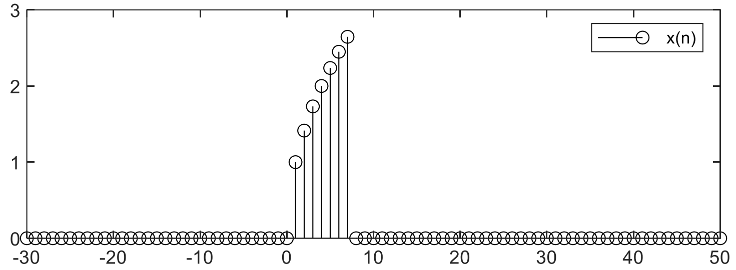
Sia $M > N$ e $z(n) = x(n \bmod M)$
Cioè z è la periodizzazione di x che però **include anche $M - N$ zeri in coda** al supporto. Per questo si parla di *zero-padding*

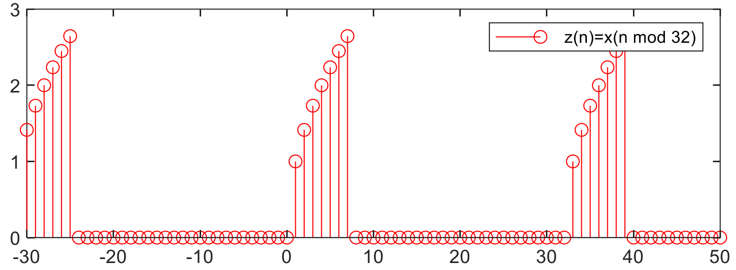
Posto $\omega_1 = \frac{2\pi}{M}$ la TFD di z vale:

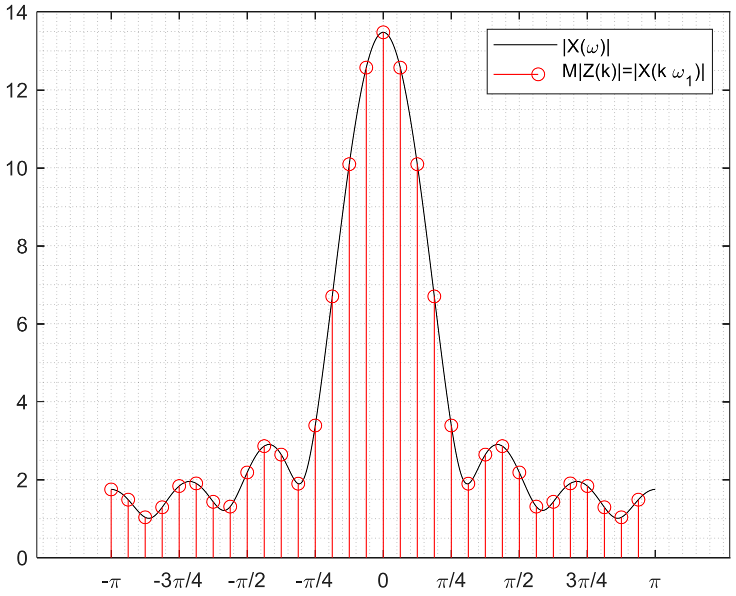
$$Z(k) = \frac{1}{M}\sum_{n=0}^{M-1} z(n)e^{-j\frac{2\pi}{M}kn} = \frac{1}{M}\sum_{n=0}^{N-1} x(n)e^{-j(k\omega_1)n} = \frac{1}{M}X(k\omega_1)$$

Quindi con lo zero-padding si può *aumentare arbitrariamento 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 $M = 2048$







Legame tra TFtd e TFD: Matlab

Come effettuare queste operazioni in un ambiente di programmazione come Matlab

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

`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

```
Zmatlab = fft(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

`ZMatlab = [`$M \cdot Z(0)\ \ M \cdot Z(1), \dots , M \cdot Z(M-1)$`] =`
`[`$X(0)\ \ X(\omega_1)\ \ X(2\omega_1), \dots X\big((M-1)\omega_1\big)$`];`

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

```
omega1 = 2*pi/M;
w = 0:omega1: (M-1)*omega1
plot(w,abs(ZMatlab));
```

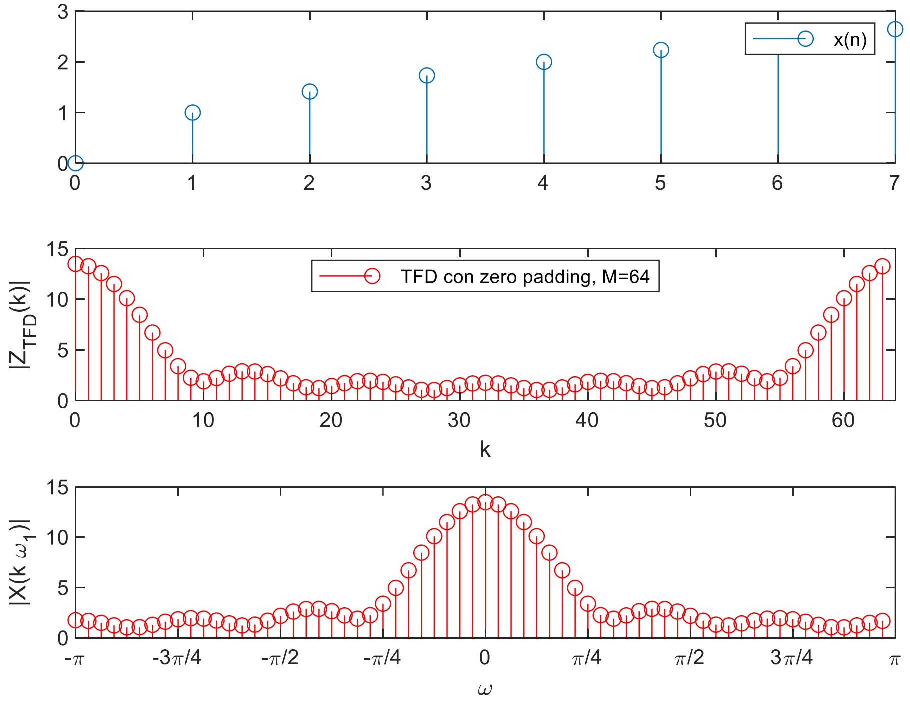
Università degli Studi di Padova

Esercizio 1

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');
% [...]
```

Eseguite il codice fornito nello script



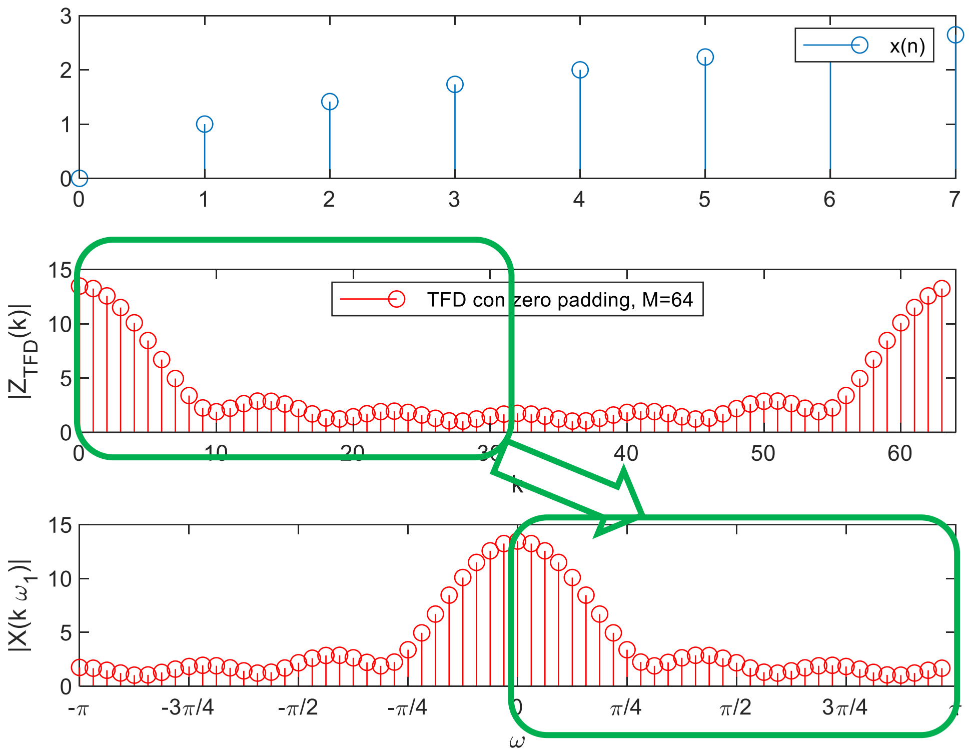
Abbiamo riportato qui la parte principale dello script. Notare come si passa dalla TFD alla TFtd:

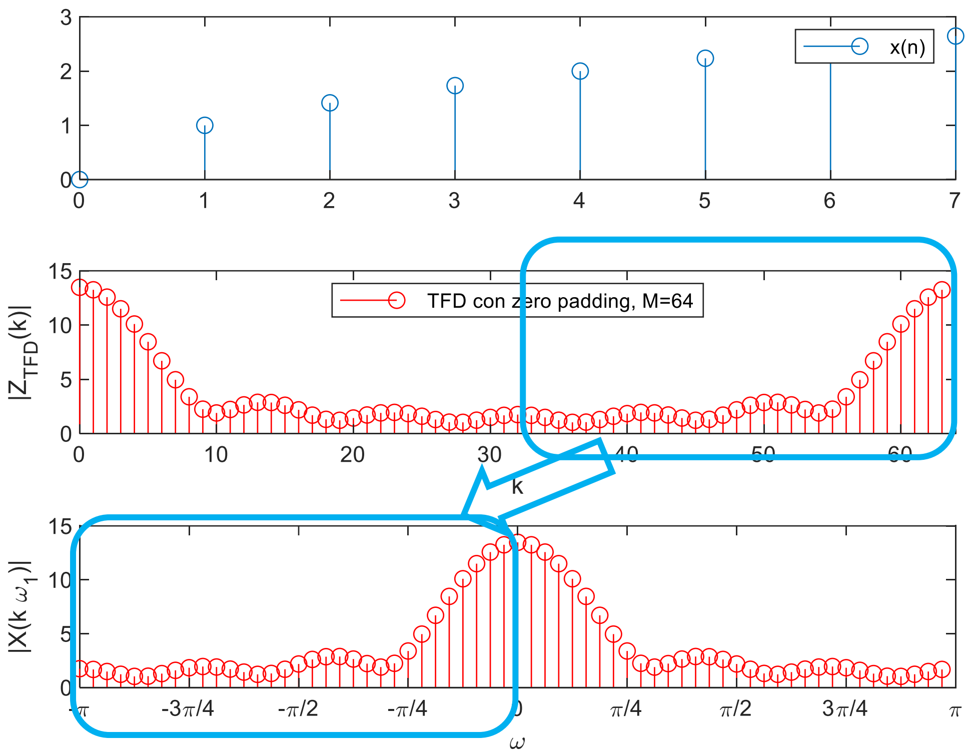
1) Riposizionamento dell'asse delle ascisse. I valori di ω coprono l'intervallo $(-\pi, \pi)$ campionandolo in M punti. Con il comando `omega1 = 2*pi/M` definiamo il passo di campionamento delle pulsazioni. Poi creiamo un vettore di M punti: `w = -pi:omega1: pi-omega1;` Notiamo che l'ultimo punto dell'intervallo deve essere $\pi - \omega_1$ e non π 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 $\boldsymbol{N}$ e $\boldsymbol{M}$

Effetto di fftshift





Esercizio 2

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

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

Dopo considereremo 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 $\boldsymbol{\omega_1}$** 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 N. 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 $w(n) = \begin{cases} 1 \text{ se } 0 \leq n < N \\ 0 \quad\quad \text{altrimenti} \end{cases}$

Il nome w richiama il termine *window*, cioè la finestra di osservazione del segnale.

Considereremo il problema di stimare ω_1 a partire dal segnale $x(n)$. I campioni di x sono memorizzati nella variabile `xCamp` contenuta nel file `lab3_ex2.mat`

Esercizio 2

Domande:

2.1 Calcolare (carta e penna!) la TFtd di x in funzione della TFtd 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 (carta e penna!) che $|W(\omega)| = N \frac{\left|\sin\left(\frac{N}{2}\omega\right)\right|}{\left|\sin\frac{\omega}{2}\right|}$ (1) .

A tal fine si mostri che $w(n) = N \cdot y_{\frac{N-1}{2}}\left(n - \frac{N-1}{2}\right)$

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

2.3 Tracciare $|W(\omega)|$ in Matlab, usando la formula (1), scegliendo un valore arbitrario per N.
Qual è l'ampiezza del lobo principale (in funzione di N)? Volendo stimare ω_1 come punto massimale per $|X(\omega)|$ come tale stima è influenzata dai parametri N ed M? Come garantire che la stima di ω_1 abbia un errore non superiore ad un certo $\Delta\omega$?

2.4 Nel file `lab3_ex2.mat` i campioni di x sono memorizzati nella variabile `xCamp`. Inoltre il file contiene `N` e `omega1`
Caricare i dati dal file valutare $|X(\omega)|$ tramite FFT con opportuno zero padding. Stimare ω_1 come la pulsazione corrispondente al valore massimo di $|X(\omega)|$.

Esercizio 3

ESERCIZIO 3: Stima della frequenza di un segnale sinusoidale

Questa volta consideriamo un segnale **sinusoidale**, $s(t) = \cos \omega_1 t$, i cui campioni sono nel file `lab3_ex3.mat`

Sia allora $x(n) = w(n) \cos \omega_1 \, n$

3.1 Determinare un'espressione di $X(\omega)$ in funzione di $W(\omega)$ ed un'approssimazione di $|X(\omega)|$ per $\omega > 0$ che permetta di individuare ω_1 come punto massimale di $|X(\omega)|$

3.2 Scelto lo zero-padding M, calcolare tramite FFT $X(\omega)$ e tracciarne il modulo per $\omega \in \left(-\pi, \pi \frac{M-2}{M}\right)$

3.3 Stimare ω_1 come punto massimale di $|X(\omega)|$, come nel caso precedente

3.4 In questo caso però, i lobi secondari di $W(\omega + \omega_1)$ interferiscono con la posizione del picco di X. Infatti, la stima di ω_1 risulta meno accurata rispetto al caso precendete, indipendentemente dal valore scelto per M.

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**. Si usi la cosiddetta *finestra di Hamming*:

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

Tracciare $\widehat{w}(n)$, $\left|\widehat{W}(\omega)\right|$. Infine usare la nuova finestra per stimare ω_1 come valore massimale di $\left|\hat{X}(\omega)\right|$, dove $\hat{x}(n) = \widehat{w}(n) s(n)$