UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Laboratorio 6: Applicazioni pratiche della FFT

Richiami: Uso della FFT per la stima della TFtc

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 $\omega_0 = \frac{\pi}{T_C}, f_C = \frac{1}{T_C}, f_M = \frac{\omega_M}{2\pi}$

Criterio di Nyquist. Ci sono tre formulazioni equivalenti del criterio:

$$\omega_0 \geq \omega_M \qquad T_C \leq \frac{\pi}{\omega_M} \qquad f_C \geq 2f_M$$

Se il criterio di Nyquist è soddisfatto non c'è aliasing. Si ha dunque:

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

Allora, se chiamiamo $Y(k)$ il k-esimo valore della TFD di $\hat{w}(n) = x(nT_C)$ con zero-padding pari a M, ritroviamo

$$Y(k) = \hat{W}\left(k\frac{2\pi}{M}\right) = \frac{1}{T_C} X\left(\frac{k}{M}\frac{2\pi}{T_C}\right)$$

Quindi **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) $\frac{2\omega_0}{M}$**

Richiami: Uso della FFT per la stima della TFtc

In Matlab, se i campioni di x sono in xCamp, bisogna usare la FFT e riscalare le ampiezze con il periodo di campionamento:

```
X = TC*fftshift(fft(xCamp, M));
```

Notare il comando `fftshift` usato per centrare i valori dello spettro sulla pulsazione nulla.
Rimane da calcolare correttamente l'asse delle pulsazioni:

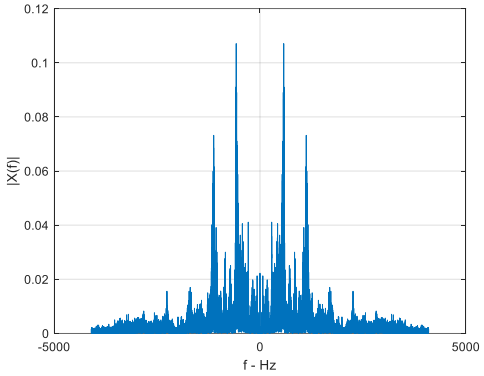
```
omega0 = pi/TC; step = (2*omega0)/M;
omega = -omega0: step : (omega0-step);
plot(omega,abs(X));
```

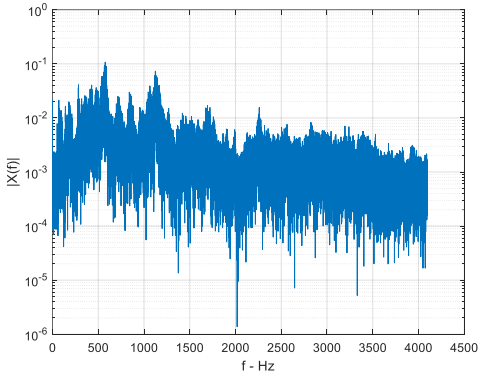
Spesso però si preferisce usare le frequenze. Abbiamo allora:

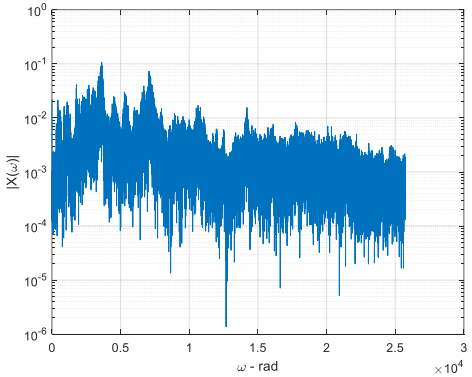
```
fC = 1/TC; step = fC/M;
F = -fC/2: step : (fC/2 - step);
plot(F,abs(X));
```

Esercizio 1

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 2π








```
clearvars; close all;

% Carica un suono campionato e la freq. camp.
load("handel.mat");
whos % Mostra le variabili caricate
%sound(y,Fs); %Rimuovere il commento per ascoltare

% Zero padding
M = 2^(nextpow2(numel(y))+2); % M>4N
TC = 1/Fs; % Periodo di campionamento
%---------------------------------
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)|');
```

Eseguite il codice illustrato

Esercizio 2: Determinare la nota di un pianoforte

I file nota1.wav, nota2.wav e nota3.wav contengono il suono di una nota di pianoforte. Lo scopo dell'esercizio è determinare quale nota è stata suonata.

A questo scopo bisogna sapere che, quando si suona una nota di pianoforte, vengono generate diverse armoniche, cioè sinusoidi a frequenza multipla di una frequenza fondamentale f_0

Bisognerà quindi determinare f_0 e trovare la nota corrispondente nella seguente tabella (frequenza f_0 in Herz approssimata all'intero più vicino)

Nota	Sol4	Sol♯4/ La♭4	La4	La♯4/ Si♭4	Si4	Do5	Do♯5/ Re♭5	Re5	Re♯5/ Mi♭5	Mi5	Fa5	Fa♯5/ Sol♭5	Sol5
f_0 [Hz]	392	415	440	466	494	523	554	587	622	659	698	740	784

Esercizio 2: Determinare la nota di un pianoforte

Per eseguire l'esercizio bisogna usare il comando `audioread` che legge da un file di tipo wav i campioni e il valore della frequenza di campionamento:

```
[nota1, Fc] = audioread('nota1.wav');
```

In seguito bisognerà determinare lo spettro di ampiezza del segnale i cui campioni sono contenuti della variable chiamata **`nota`** ed infine determinare la frequenza f_0 come quella corrispondente al massimo dello spettro di ampiezza.

Usando la tabella si potrà determinare la nota.

Provare anche ad ascoltare il suono usando `soundsc`:

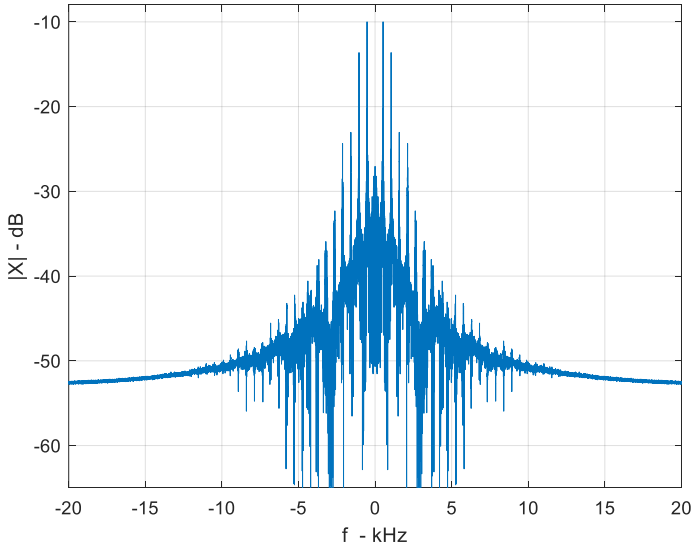
```
soundsc(nota1,Fc);
```

Oppure scaricate i file `wav` su un qualsiasi dispositivo (smartphone, tablet, …) ed ascoltatelo da quest'ultimo

Esercizio 2: Soluzione

```
clearvars; close all;

% Lettura dati: campioni e freq. di camp.
[nota, Fc] = audioread('nota1.wav');
TC = 1/Fc; % Periodo di campionamento

% Riproduzione suono
soundsc(nota,Fc);

% Analisi in frequenza
N = numel(nota); % nr di campioni del segnale
M = 2^(nextpow2(N)+3); % Zero-padding
step = Fc/M; fStart = -Fc/2;
fStop = Fc/2 - step;
F = fStart:step:fStop;

% Calcolo campioni della TFtc
X = TC*fft(nota,M);

% Calcolo spettro di ampiezza
X_amp = fftshift(abs(X));
```

```
% Visualizzazione: freq in kHz, ampiezze in decibel
figure(1); plot(F/1000,10*log10(X_amp));  grid;
% Traccia solo la parte d'interesse del grafico
axis([-20 20 -65 -8]);
xlabel('f - kHz'); ylabel('|X| - dB');

% Determinazione della nota
[XM, ind] = max(X_amp); f0 = abs(F(ind));
fprintf('L''armonica principale è a %3d Hz\n', round(f0));
```



Esercizio 3

In questo esercizio si considera un suono più complesso, presente nel file `due_note.wav`

Si richiede d'individuare non solo le due note ma anche l'istante in cui si comincia a suonare ognuna di esse.

Iniziate con l'ascoltare il suono in Matlab o scaricando il file su di un dispositivo

È possibile individuare la collocazione delle note in tempo ed in frequenza con gli strumenti disponibili?

La TF in quanto tale non è adeguata a risolvere questo tipo di *analisi tempo-frequenza,* perché l'informazione sulla localizzazione temporale delle sinusoidi è difficile da estrarre. Provate comunque ad usare il codice proposto per l'esercizio precedente: che conclusioni traete?

Lo strumento che permette di risolvere il problema si basa sulla **Trasformata di Fourier a corto termine**

La trasformata di Fourier a corto termine (Short-Time Fourier Transform, STFT)

La Trasformata di Fourier a corto termine (TFct), o Short-Time Fourier Transform (STFT) è uno strumento molto utile nell'analisi ed il trattamento dei segnali

Non possiede le proprietà algebriche della TF, ma permette un'"analisi locale" del contenuto spettrale di un segnale.

Per TFct s'intende **la TF del segnale d'interesse x moltiplicato per una *finestra***, ovvero un altro segnale w a supporto finito e opportunamente ritardato

Si noti che questo approccio si può utilizzare tanto per i segnali a tempo continuo quanto per quelli a tempo discreto. In questo laboratorio consideriamo il caso *td*

TFct

Più precisamente

$$\mathrm{TFct}[x](n,\nu) = \mathrm{TFtd}(x \cdot \mathcal{U}_n[w])(\nu)$$

x è il segnale da analizzare con supporto N_x (cioè sono stati acquisiti N_x campioni)

w è una *finestra* cioè un segnale con supporto $N_w \ll N_x$

Tale finestra viene ritardata di n e, con la moltiplicazione per x, permette di "analizzare" il contenuto frequenziale di una porzione del segnale che comincia in n e dura fino a $n + N_w - 1$

Il risultato è un segnale che dipende da due variabili:

- Il ritardo n cioè l'istante in cui "inizia" l'analisi
- La frequenza numerica $\nu \in \left(-\frac{1}{2}, \frac{1}{2}\right)$ della TFtd

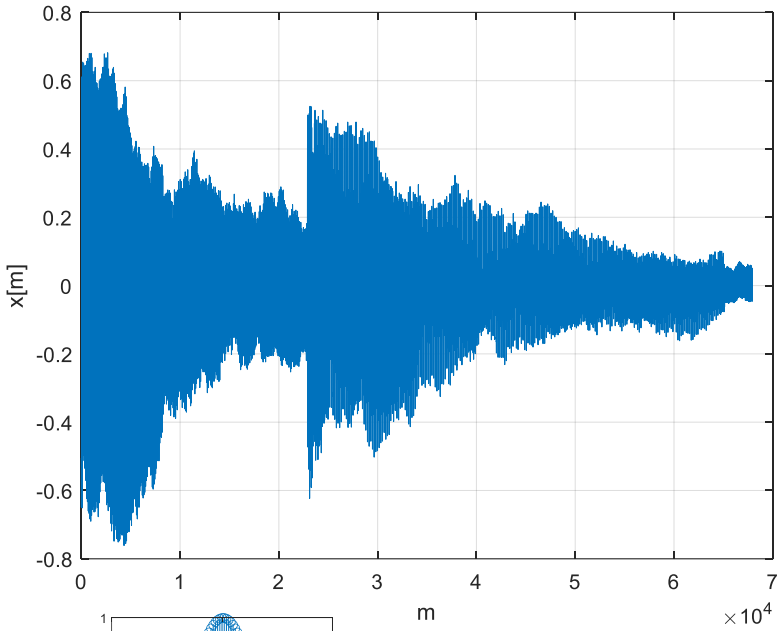
Lo Spettrogramma

Nella pratica, invece di considerare la TFtd se ne considera la versione campionata, cioè calcolata con TFD e zero-padding. Il modulo quadro di tale segnale, espresso in dB, è detto *spettrogramma* $\mathcal{S}_x(k,n)$

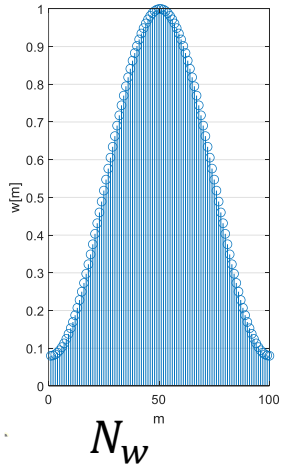
$$\mathcal{S}_x(k,n) = 10\log_{10}\left|\mathrm{TFD}_M\left[x \cdot \mathcal{U}_n[w]\right](k)\right|^2$$
$$= 10\log_{10}\left|\sum_m x(m)w(m-n)e^{-j\frac{2\pi}{M}km}\right|^2$$

Tale segnale è visualizzato in 2 dimensioni e rappresenta l'evoluzione nel tempo del contenuto spettrale della finestra di durata N_w del segnale x

Lo Spettrogramma

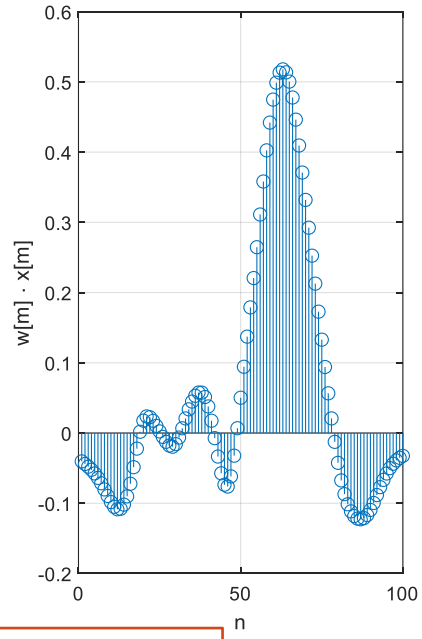


$x[m]$



$w[m]$

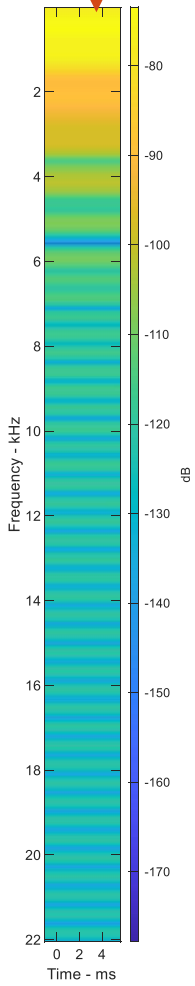
N_w



$x[m] \cdot w[m]$

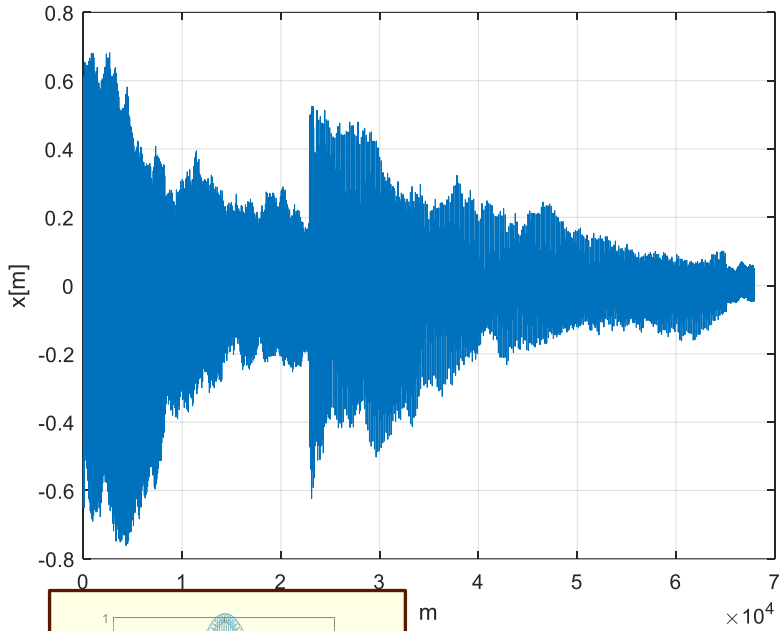
FFT

M

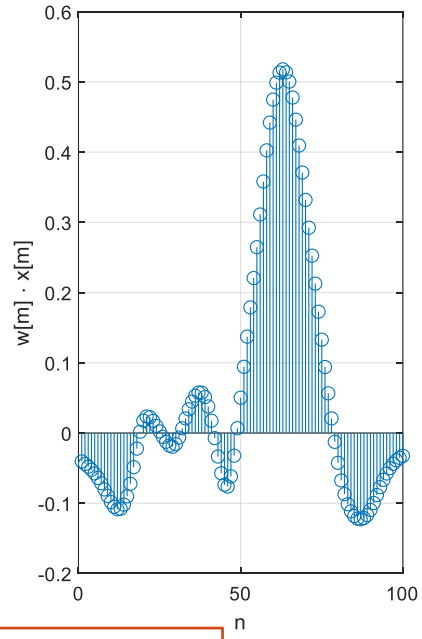


Calcolo prima colonna dello spettrogramma, cioè analisi in frequenza del primo blocco del segnale x

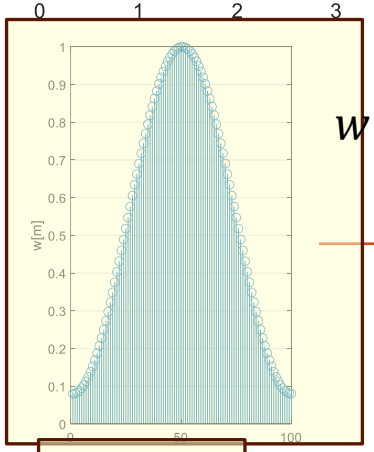
Lo Spettrogramma: parametri



$x[m]$



$x[m] \cdot w[m]$

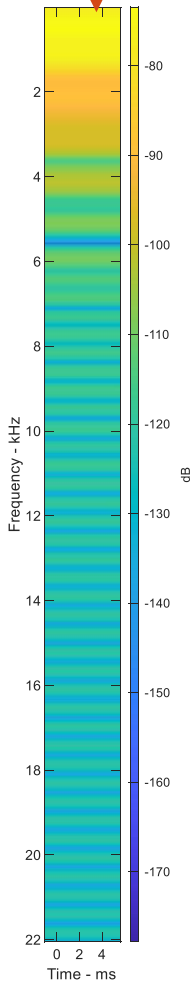


$w[m]$

N_w

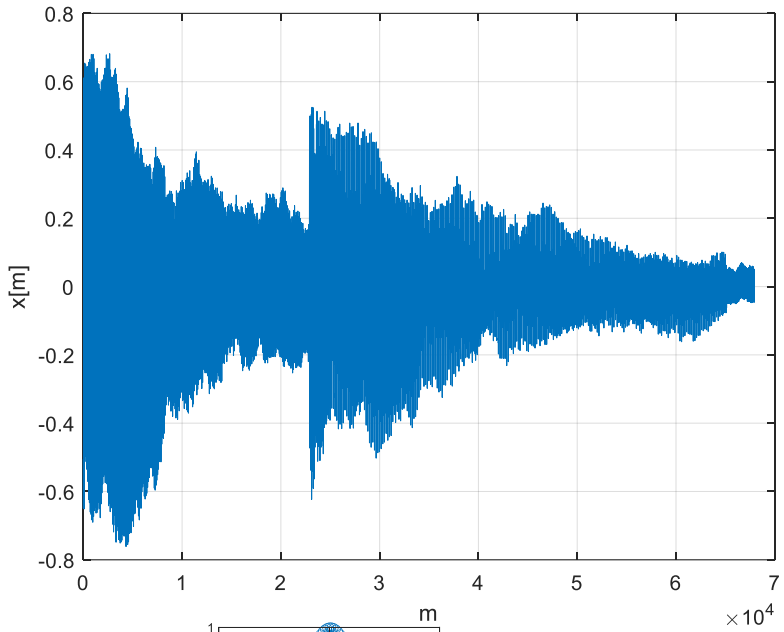
FFT

M

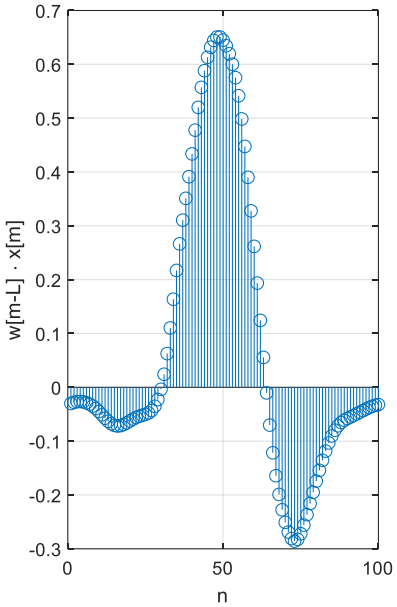


Parametri: forma e durata della finestra, ordine M della TFD

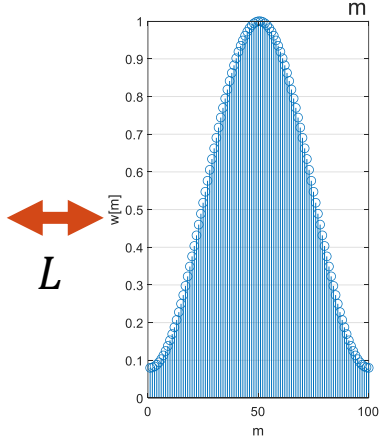
Lo Spettrogramma



$x[m]$



$x[m] \cdot w[m-L]$

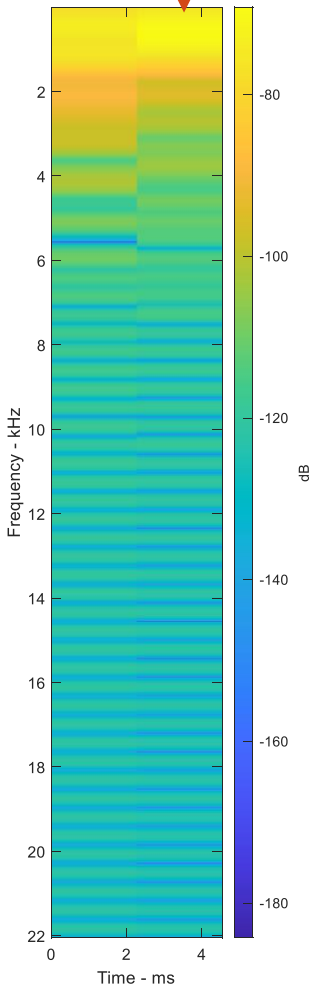


$w[m-L]$

L

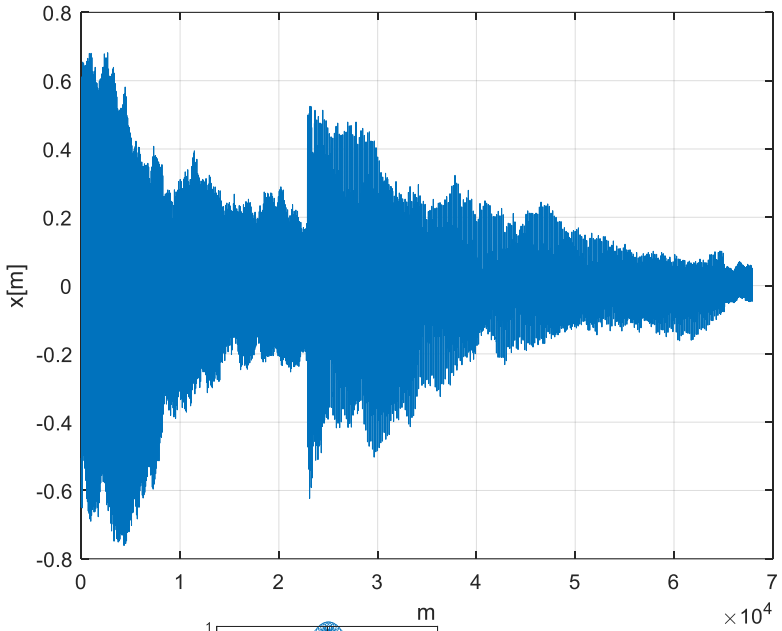
FFT

M

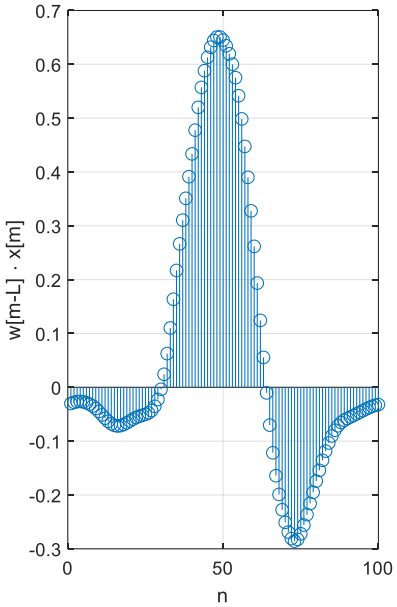


Calcolo seconda colonna dello spettrogramma, cioè analisi in frequenza del secondo blocco del segnale x, ottenuto spostando la finestra di L campioni in avanti

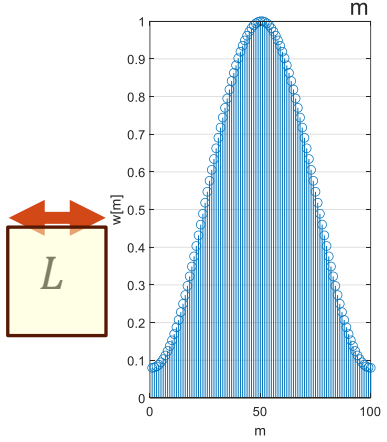
Lo Spettrogramma: parametri



$x[m]$



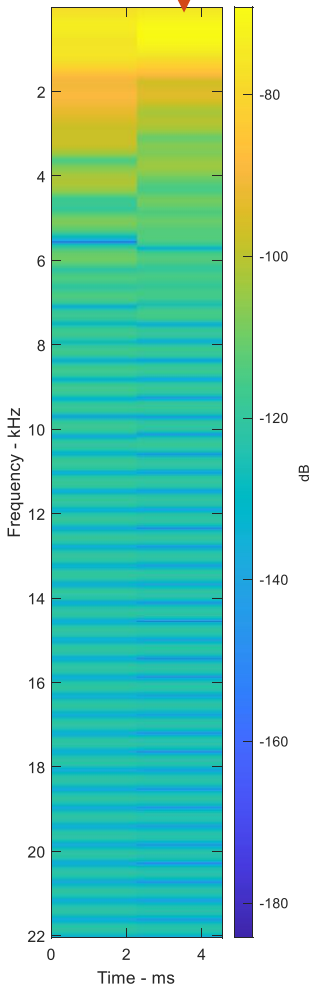
$x[m] \cdot w[m-L]$



$w[m-L]$

FFT

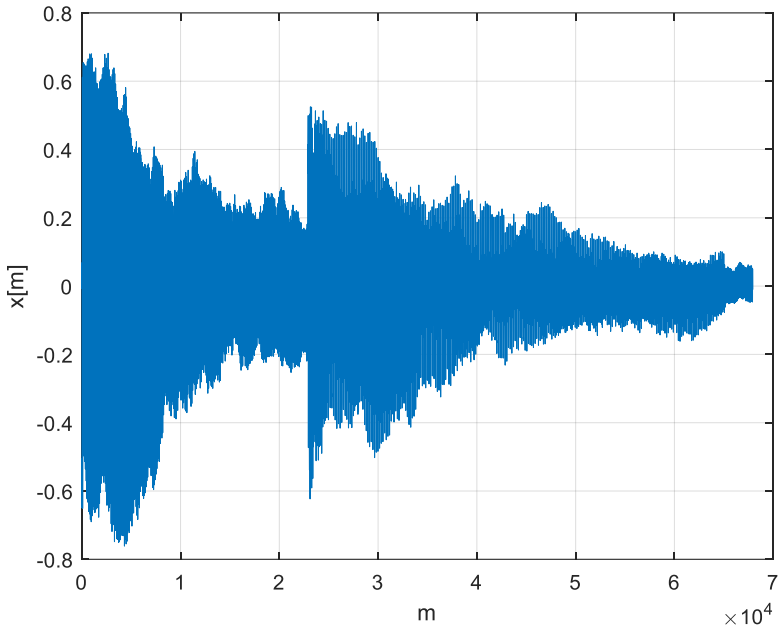
M



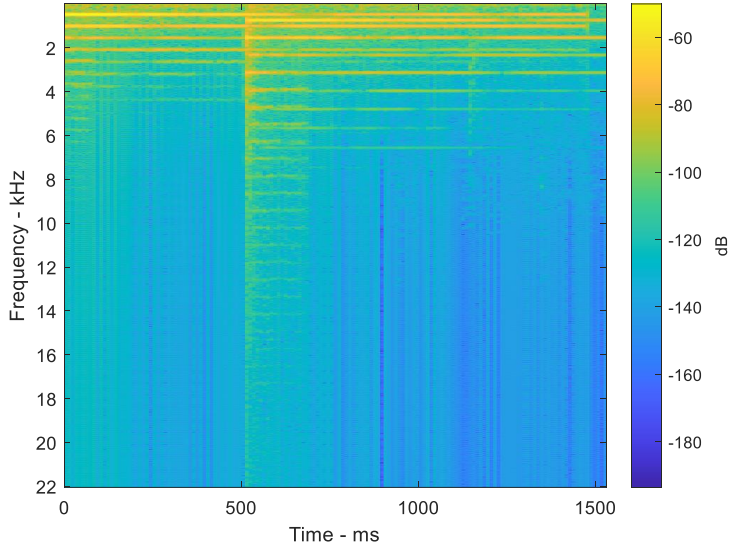
Anche il parametro L va scelto opportunamente (vd seguito)

Continuando a «spostare» w si genera tutto lo spettrogramma

Spettrogramma completo



spettrogramma
(x, Fc, Nw, M, L, tipo)



Lo Spettrogramma: parametri (1 su 2)

Nel calcolo dello spettrogramma ci sono diversi parametri da considerare:

- La durata della finestra N_w: una finestra breve permette una risoluzione temporale fine, ma la risoluzione in frequenza peggiora, perché una sinusoide genera uno spettro con lobo principale la cui ampiezza è inversamente proporzionale a N_w (vd. Lab 5)
- Lo zero-padding M permette di campionare più fittamente in frequenza la TFtd del segnale finestrato: il prezzo da pagare è un aumento del tempo di calcolo
- Il tipo di finestra w: abbiamo visto che la scelta della finestra implica un trade-off tra decadimento asintotico e ampiezza del lobo centrale

Lo Spettrogramma: parametri (2 su 2)

Infine in genere si calcola $S_x(k,n)$ non per tutti i valori di n tra 0 e $N_x - N_w$, ma con un sottocampionamento temporale per ridurre la complessità di calcolo

In altre parole, si fissa un fattore di sottocampionamento L e si calcola lo spettrogramma per valori di ritardo pari a $0, L, 2L, 3L, \ldots$

- Si può scegliere $L = N_w$: in questo caso il segnale e ripartito in segmenti non sovrapposti, ognuno analizzato separatamente dagli altri
  - In altre parole, si analizzano blocchi di $L = N_w$ campioni, ed i vari blocchi non hanno elementi in comune
- Un valore tipico è $L = \frac{N_w}{2}$. In questo caso, il segnale è suddiviso in blocchi, ma la seconda metà di ogni blocco è la prima metà del blocco seguente
- Il caso estremo è $L = 1$ cioè la finestra di analisi viene spostata di un solo campione alla volta
- Il costo computazionale è inversamente proporzionale a L

Funzione spettrogramma

La funzione spettrogramma è fornita nell'omonimo file e mostrata qui →

Dopo aver creato la finestra, il segnale è analizzato a blocchi di lunghezza Nw presi a passo L

Infine lo spettrogramma è rappresentato come un immagine, con un opportuna scala di colori che rappresentano i valori dello spettro di ampiezza in dB

```
function SX = spettrogramma(x, Fc, Nw, L, M, tipo)
%SX = spettrogramma(x, Fc, Nw, L, M, tipo)
%  Spettrogramma del segnale x
% Fc frequenza di campionamento di x
% Nw durata della finestra di analisi
% L intervallo tra due finestre consecutive, in numero di campioni
% M parametro di zero padding
% tipo nome della finestra: 'rect' o 'hamming'

% Inizializzazioni
Tc = 1/Fc;  winPeriod = L*Tc; Nx =numel(x);
% Creazione della finestra
switch lower(tipo)
    case 'rect'
        w = ones(1,Nw);
    case 'hamming'
        w =  0.54- 0.46*cos(2*pi*(1:Nw)/Nw);
    otherwise % Default: rect
        w = ones(1,Nw);
end
nFreq = M/2; % nr campioni frequenza
nTimes = round((Nx-Nw)/L); % nr campioni temp
SX = zeros(nFreq,nTimes);
% Ciclo sulle diverse finestre temporali
for n = 0: L : Nx-Nw
    y = x(n+1:n+Nw) .* w(:); % "Finestratura" del segnale
    Y = Tc * fft(y,M);
    SX(1:M/2, n/L+1 ) = 10*log10(abs(Y(1:M/2)).^2);
end
% Comandi per la corretta visualizzazione
timeAxis = [winPeriod/2 winPeriod*(nTimes+1/2)]  * 1000;
freqAxis = [Fc/(2*M),  Fc/2 - Fc/M]/1000;
% Creazione figura
figure;  imagesc(timeAxis,freqAxis, SX);
c = colorbar; c.Label.String = 'dB';
xlabel('Time - ms'); ylabel('Frequency - kHz');
```

Applicazione dello spettrogramma

Esercizio 4: Utilizzare lo spettrogramma per determinare le note contenute nel file due_note.wav ed il loro istante di attacco con risoluzione temporale di 10 ms

Scegliere opportunamente i parametri nel codice seguente:

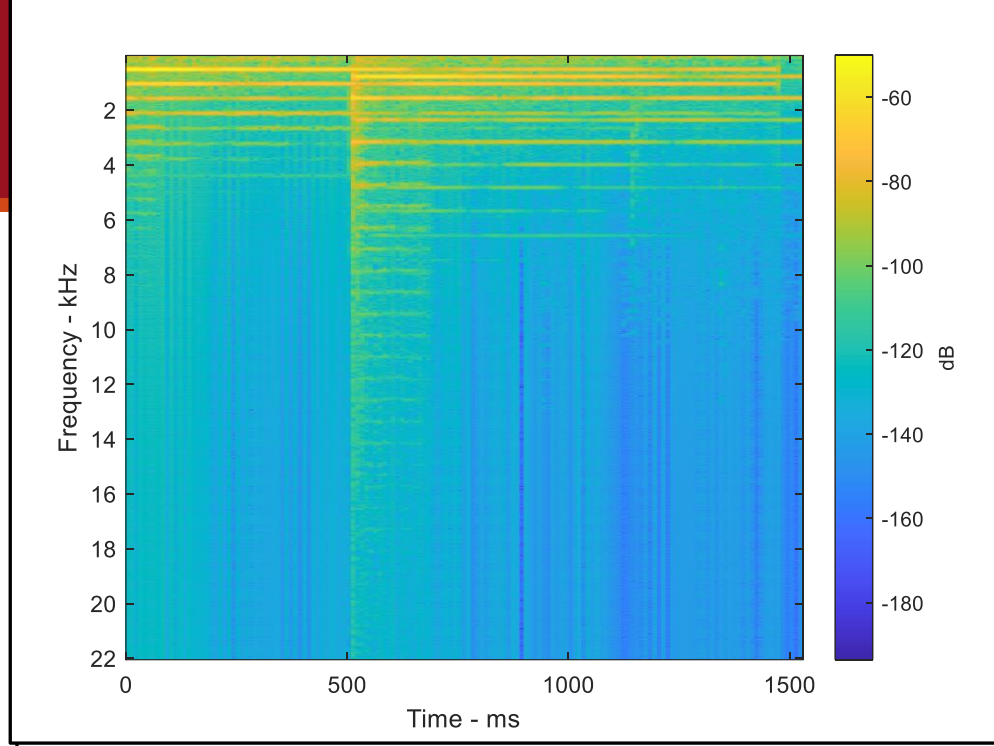
```
% Lettura dati
[suono, Fc] = audioread('due_note.wav'); % Legge i campioni e la freq. camp.
TC = 1/Fc; % Periodo di campionamento


% Parametri dello spettrogramma
durataFinestra = 20e-3; % Durata in secondi della finestra di analisi
Nw = round(durataFinestra *Fc); % Durata in numero di campioni
% Prendiamo L = Nw/2
L = Nw/2; % In questo modo, analizziamo blocchi separati da 10 ms
% Per lo zero-padding, prendiamo M >= 8 Nw
M = 2^(nextpow2(Nw) + 3);
tipo = 'hamming';
SX = spettrogramma(suono, Fc, Nw, L, M, tipo);
```

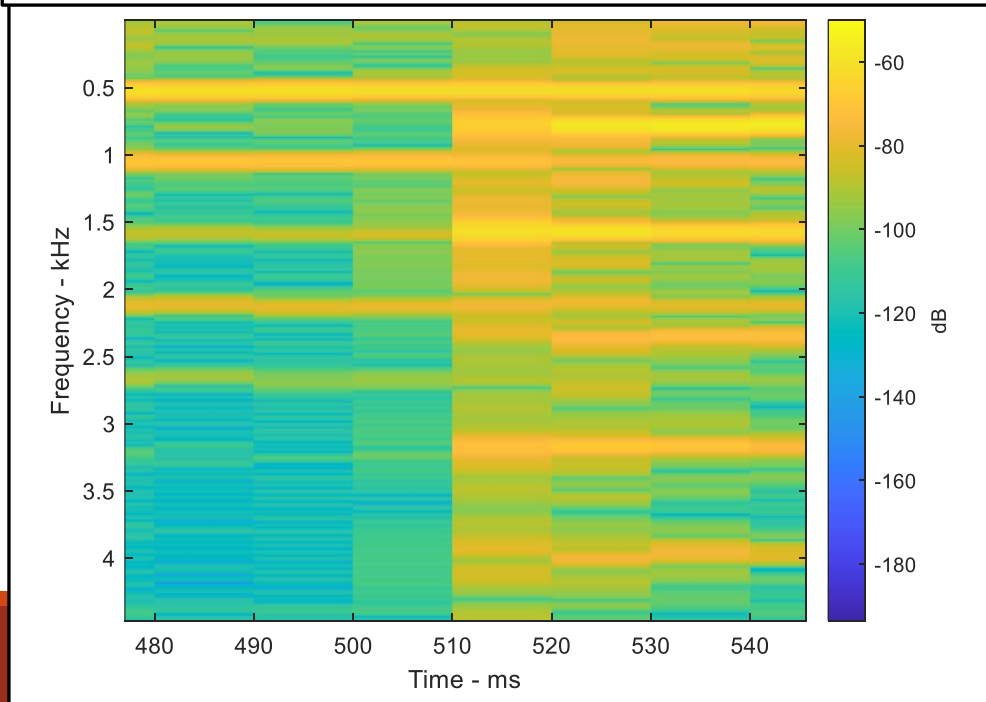
Applicazione dello spettrogramma

Dopo aver creato la figura dello spettrogramma (immagine in alto) è utile fare uno zoom per analizzare le regioni d'interesse del piano tempo-frequenza (figura in basso)

Come si vede dallo spettrogramma, la seconda nota appare intorno ai 510ms.

Tuttavia la risoluzione in frequenza è piuttosto grossolana e permette solo indicativamente di trovare le frequenze delle note, intorno ai 500 e 750 Hz





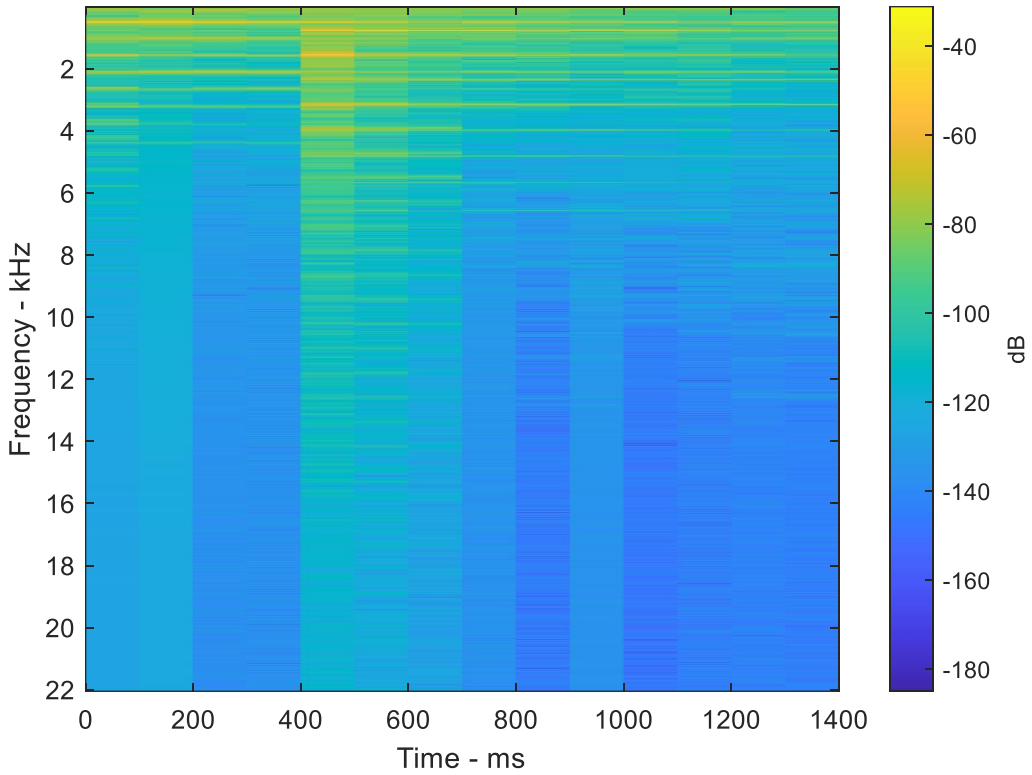
Applicazione dello spettrogramma

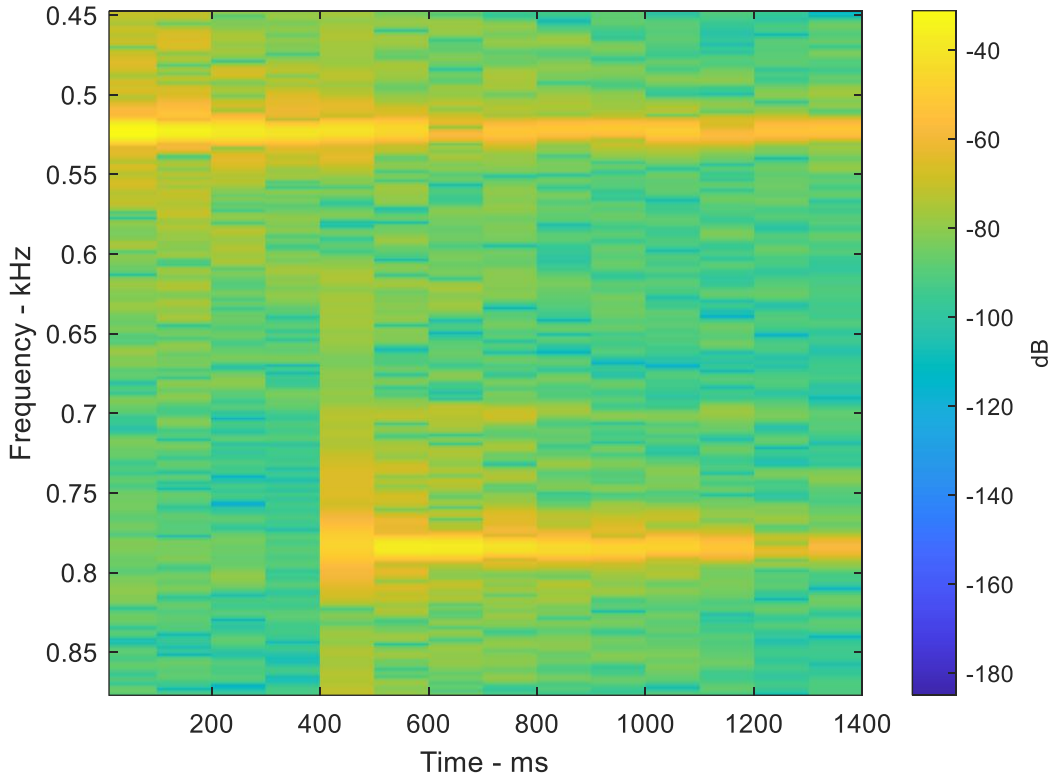
Per avere una migliore risoluzione in frequenza bisogna incrementare la durata della finestra.

Proviamo con finestre da 200 ms e campionamento dello spettrogramma a 100 ms:

```
durataFinestra = 200e-3; % Passiamo a 200 millisecondi
Nw = round(durataFinestra *Fc); % Durata in numero di campioni
L = Nw/2;
M = 2^(nextpow2(Nw) + 3);
tipo = 'hamming';
SX = spettrogramma(suono, Fc, Nw, L, M, tipo);
```

Applicazione dello spettrogramma





Adesso la risoluzione temporale è piuttosto grossolana, ma si riescono ad individuare più facilmente le frequenze del Do5 e del Sol5