UNIVERSITÀ DEGLI STUDI DI PADOVA

Laboratorio 4: 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 $\widehat{w}(n)$ la successione dei campioni presi a passo T_C: $\widehat{w}(n) = x(nT_C)$, per $n = 0,1, \dots N-1$

Poniamo $\omega_0 = \frac{\pi}{T_C}$

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:

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

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

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

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) $\frac{2\omega_0}{M} = \frac{2\pi}{M\,T_C}$**

Attenzione, nella (1) si considera $Y(k)$ periodico di periodo M, e se M è pari (caso più comune), l'indice k varia in $\left\{-\frac{M}{2}, -\frac{M}{2}+1, \dots, -1, 0, 1, \dots, \frac{M}{2}-1\right\}$

Richiami: Uso della FFT per la stima della TFtc

Volendo usare lo spettro in frequenza, $X_{\text{fr}}(f) = X(\omega)|_{\omega=2\pi f}$, si ha che $Y(k) = \frac{1}{T_C} X\left(\frac{k}{M}\frac{2\pi}{T_C}\right)$ e la pulsazione $\omega = \frac{k}{M}\frac{2\pi}{T_C}$ corrisponde alla frequenza $f = \frac{\omega}{2\pi} = \frac{k}{M}\frac{1}{T_C} = k\frac{f_C}{M}$ e quindi $Y(k) = \frac{1}{T_C} X_{\text{fr}}\left(k\frac{f_C}{M}\right)$ cioè:

$$\boxed{X_{\text{fr}}\left(k\frac{f_C}{M}\right) = T_C Y(k)}$$

Quindi **calcolando la TFD sui campioni di x con zero-padding, si ottengono i campioni di $X_{\text{fr}}(f)$**
L'intervallo di frequenze si ottiene dividendo per 2π quello delle pulsazioni:

$$f_0 = \frac{\omega_0}{2\pi} = \frac{\pi}{T_c}\frac{1}{2\pi} = \frac{f_C}{2}$$

$$f_{\text{step}} = \frac{2\omega_0}{M}\frac{1}{2\pi} = \frac{2\pi}{MT_c}\frac{1}{2\pi} = \frac{f_C}{M}$$

L'intervallo di frequenze è quindi $\boxed{\left(-\frac{f_C}{2}, \frac{f_C}{2} - \frac{f_C}{M}\right) \text{ campionato a passo } \frac{f_C}{M}}$

Università degli Studi di Padova

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, che supponiamo memorizzato nella variabile **TC**

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

Notare il comando `fftshift` usato per centrare i valori dello spettro sulla frequenza nulla.

Rimane da calcolare correttamente l'asse delle ascisse.

```
fC = 1/TC; step = fC/M;
freq = -fC/2: step : (fC/2 - step);
```

Il grafico dello spettro di ampiezza in frequenza si ottiene con:

```
plot(freq,abs(X)); title('|X_FR (f)|');
```

Se si vuole lo spettro di ampiezza in pulsazione, basta riscalare l'asse delle ascisse:

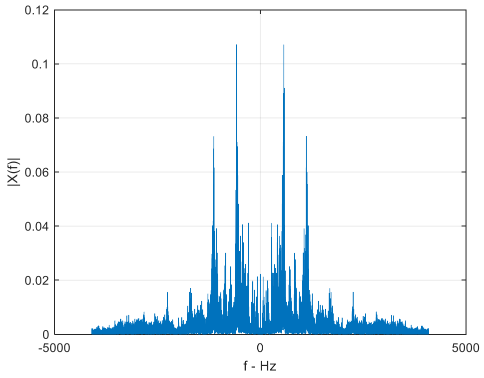
```
plot(2*pi*freq,abs(X)); title('|X (\omega)|');
```

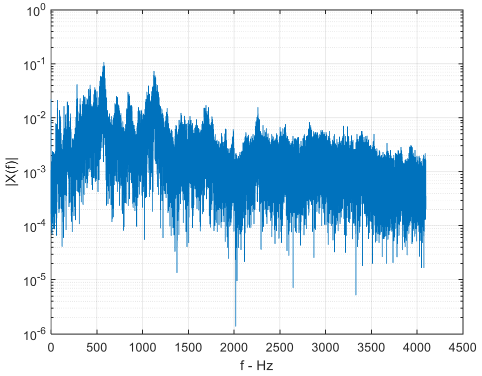
Esercizio 1

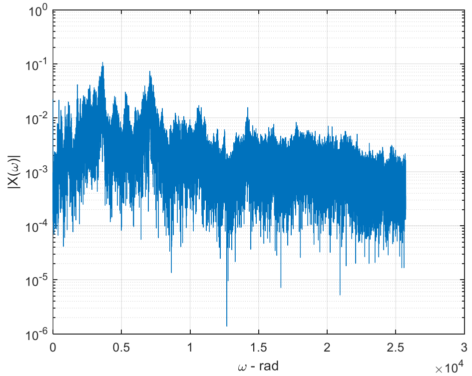
Nel file **handel.mat** si trova un segnale audio campionato. I campioni si trovano nella variabile **y**, mentre la frequenza di campionamento è in **Fs**.
Usando la traccia fornita, scrivere uno script che , riproduce il suono e mostra lo spettro in frequenza del segnale tempo continuo da cui sono stati tratti i campioni. Siccome il segnale da analizzare è reale, il suo spettro di ampiezza è pari, quindi se ne può visualizzare anche solo la parte corrispondente alle frequenza positive.

Per lo zero padding, prendere come valore di M la più piccola potenza di 2 maggiore o uguale a $8N$, dove N è il numero di campioni.

```
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))+3); % M>=8N
TC = ??? Determinare il per. camp. dalla freq. camp.
%-----------------------------------
Y =  ??? Usare TC, la fft e fftshift ???
F = ??? Freq iniziale : step : freq finale ???
%-----------------------------------
figure(1);
plot(F,abs(Y)); grid minor
xlabel('f - Hz'); ylabel('|X(f)|');
% Per un segnale reale basta tracciare |X|
% per frequenze positive e prendere la 2nda metà di Y
Fpos = 0:Fs/M:(Fs/2-Fs/M); Ypos = Y(M/2+1:end);
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)|');
```







COMPLETARE

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 (prima armonica o armonica principale) e trovare la nota corrispondente nella seguente tabella, dove la frequenza f_0 è espressa in Herz ed approssimata all'intero più vicino.

Nota	Sol4	**Sol♯4/ Lab4**	La4	**La♯4/ Sib4**	Si4	Do5	**Do♯5/ Reb5**	Re5	**Re♯5/ Mib5**	Mi5	Fa5	**Fa♯5/ Solb5**	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:

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

In seguito bisognerà determinare lo spettro di ampiezza del segnale i cui campioni sono contenuti della variable chiamata **`nota`**

Per fare questo, effettuare lo zero padding con $M \geq 8N$ e calcolare lo spettro di ampiezza sull'intervallo $\left(-\frac{f_C}{2}, \frac{f_C}{2} - \frac{f_C}{M}\right)$

Infine determinare la frequenza f_0 come quella corrispondente al massimo dello spettro di ampiezza ed usare la tabella per trovare la nota.

Per visualizzare la parte dello spettro d'interesse, usare il comando `axis([0 2500 -50 0])`

che restringe il grafico alle frequenze tra 0 e 2500 Hz e le ampiezze tra -50dB e 0dB.

Per *ascoltare* il suono in Matlab si può usare `soundsc(nota,Fc);` oppure si possono scaricare i file `wav` su un qualsiasi dispositivo (smartphone, tablet, …) ed ascoltarlo da quest'ultimo

UNIVERSITÀ DEGLI STUDI DI PADOVA

Esercizio 3

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

Si richiede d'individuare non solo le due note ma anche l'istante in cui si comincia a suonare ognuna di esse, detto «onset»[1]

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?

Usate il codice della soluzione dell'esercizio 2 con il nuovo file: ci si accorge che la non è adeguata a risolvere questo tipo di *analisi tempo-frequenza,* perché l'informazione sulla localizzazione temporale delle sinusoidi è difficile da estrarre.

---

[1] Il problema di determinare l'onset è piuttosto difficile in generale, vedere ad esempio Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., Sandler, M.B. (2005) "A Tutorial on Onset Detection in Music Signals", IEEE Transactions on Speech and Audio Processing 13(5), pp 1035-1047

La trasformata di Fourier a corto termine – TFCT (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

La TFCT 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 considereremo la TFCT a tempo discreto applicata a segnali campionati.

TFCT

Più precisamente, definiamo la TFCT di un segnale x a tempo discreto (eventulamente risultante da un campionamento):

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

Cioè, è la TFtd del segnale x moltiplicato per una finestra w ritardata di n campioni

Tipicamente si usa la frequenza numerica ν invece della pulsazione

Sia N_x il numero di campioni del segnale x

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

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 si usa la TFD calcolata con l'algoritmo FFT. Ciò equivale a campionare la TFtd in M punti per periodo (con lo zero padding). Il modulo quadro di tale TFD, espresso in dB, è detto *spettrogramma* $\mathcal{S}_x(k,n)$

$$\mathcal{S}_x(k,n) = 10\log_{10} \left|\mathrm{TFD}_M\left[x \cdot \mathcal{U}_n[w]\right](k)\right|^2$$

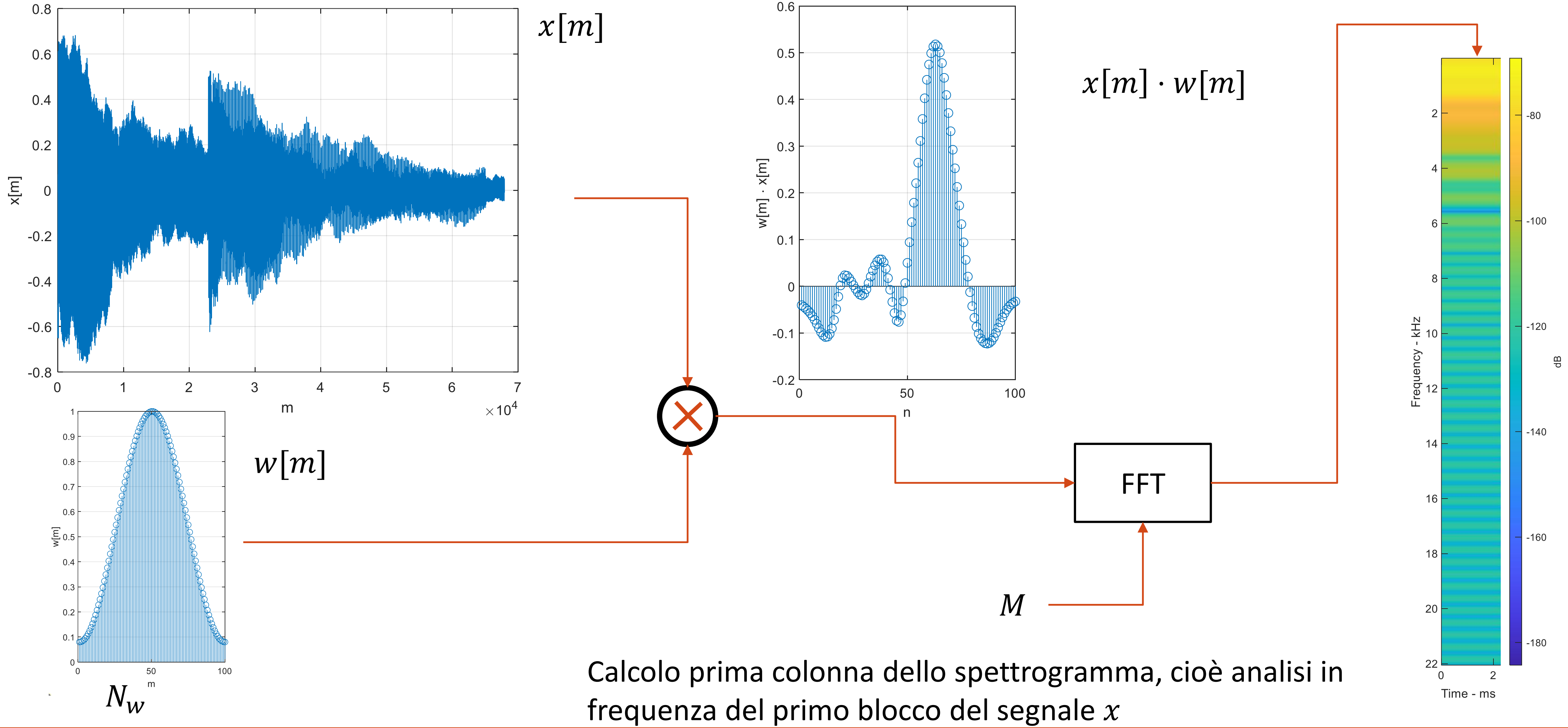
$$= 10\log_{10}\left|\sum_m x(m)w(m-n)e^{-j\frac{2\pi}{M}km}\right|^2$$

k è l'indice del coefficiente TFD, e corrisponde alla frequenza numerica $\frac{k}{M}$. Si considerano solo frequenze positive, per cui $k \in \left\{0, 1, \dots, \frac{M}{2} - 1\right\}$. L'indice k corrisponde alla frequenza t.c. $f_k = k\frac{f_C}{M}$

n è l'indice temporale e corrisponde all'instante d'inizio della finestra di analisi

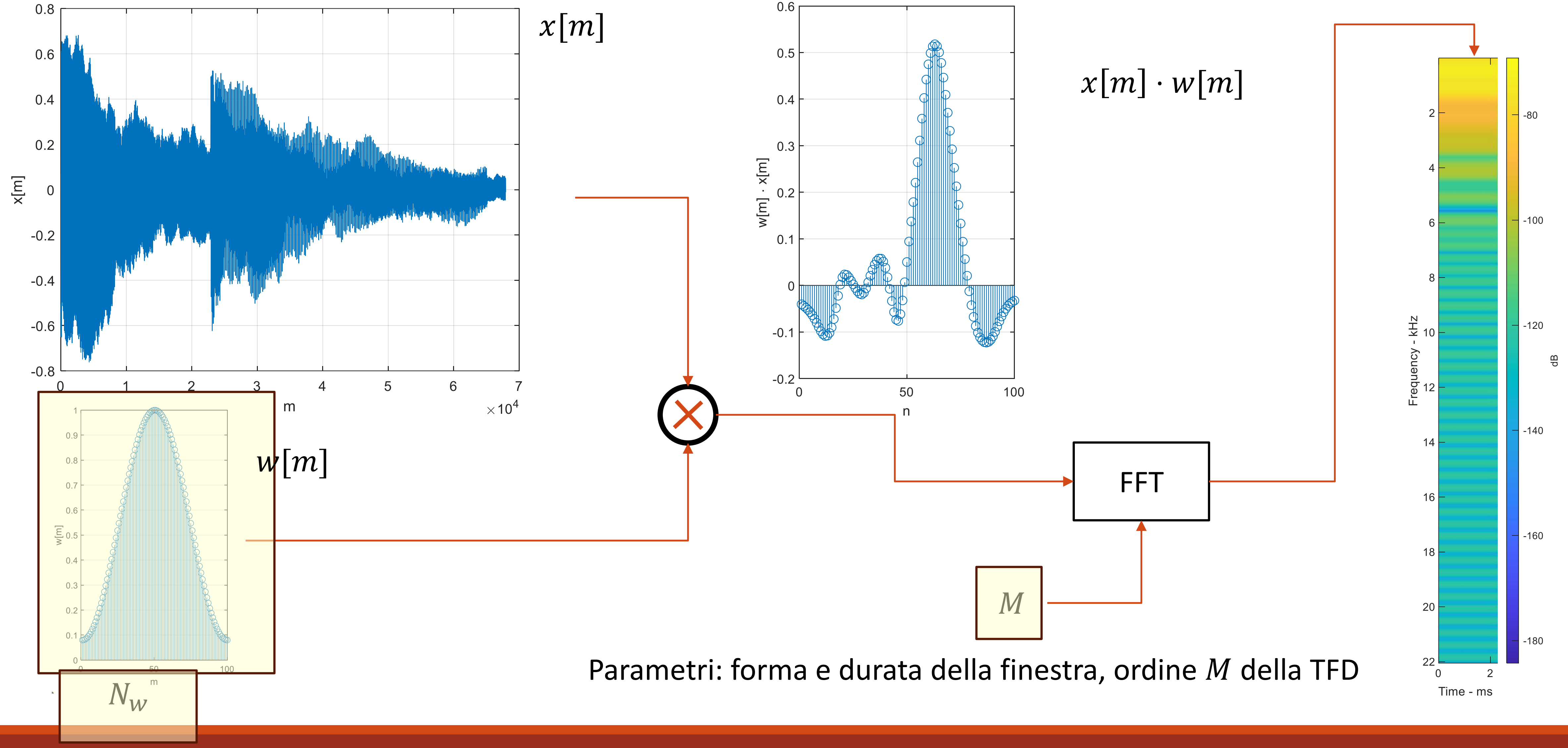
Tale segnale è visualizzato in 2 dimensioni e rappresenta l'evoluzione nel tempo del contenuto spettrale della finestra di durata N_w del segnale x

Lo Spettrogramma



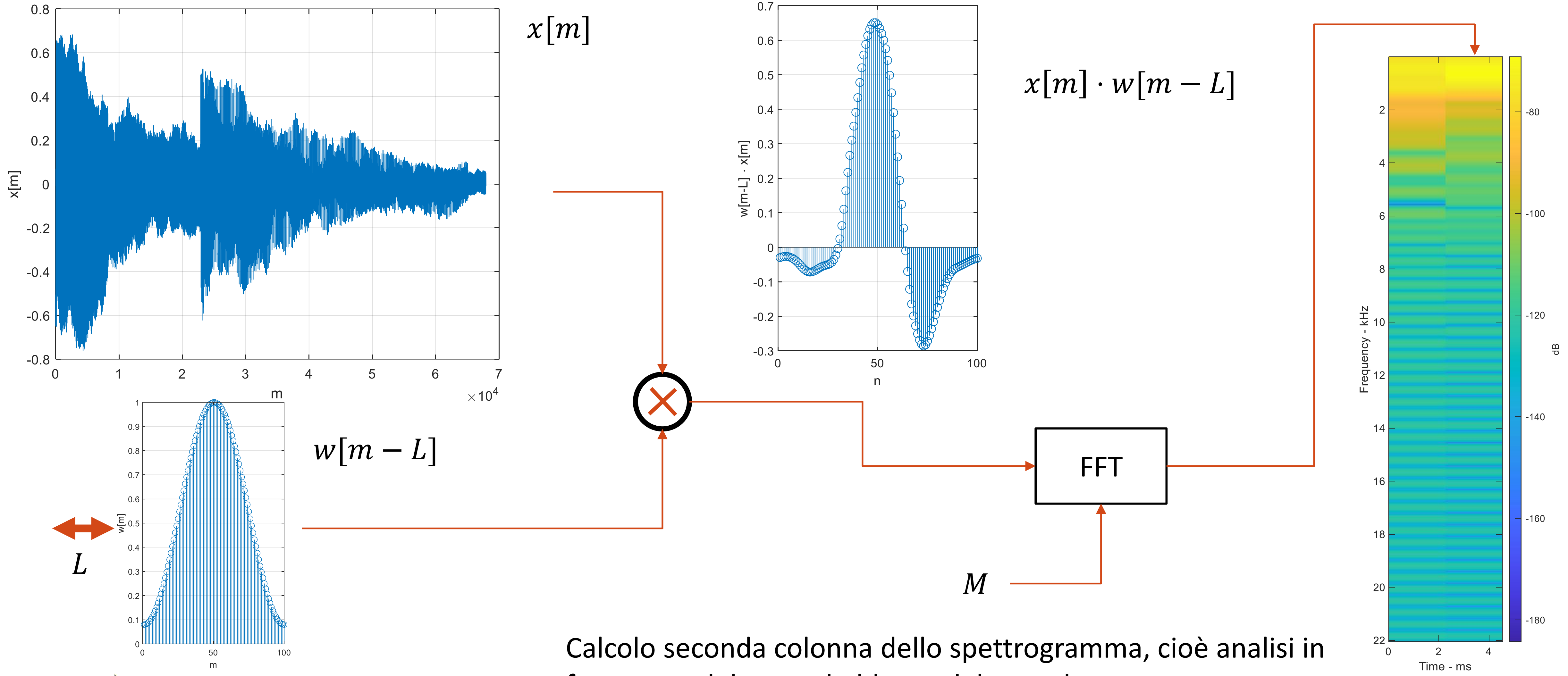
Calcolo prima colonna dello spettrogramma, cioè analisi in frequenza del primo blocco del segnale x

Lo Spettrogramma: parametri



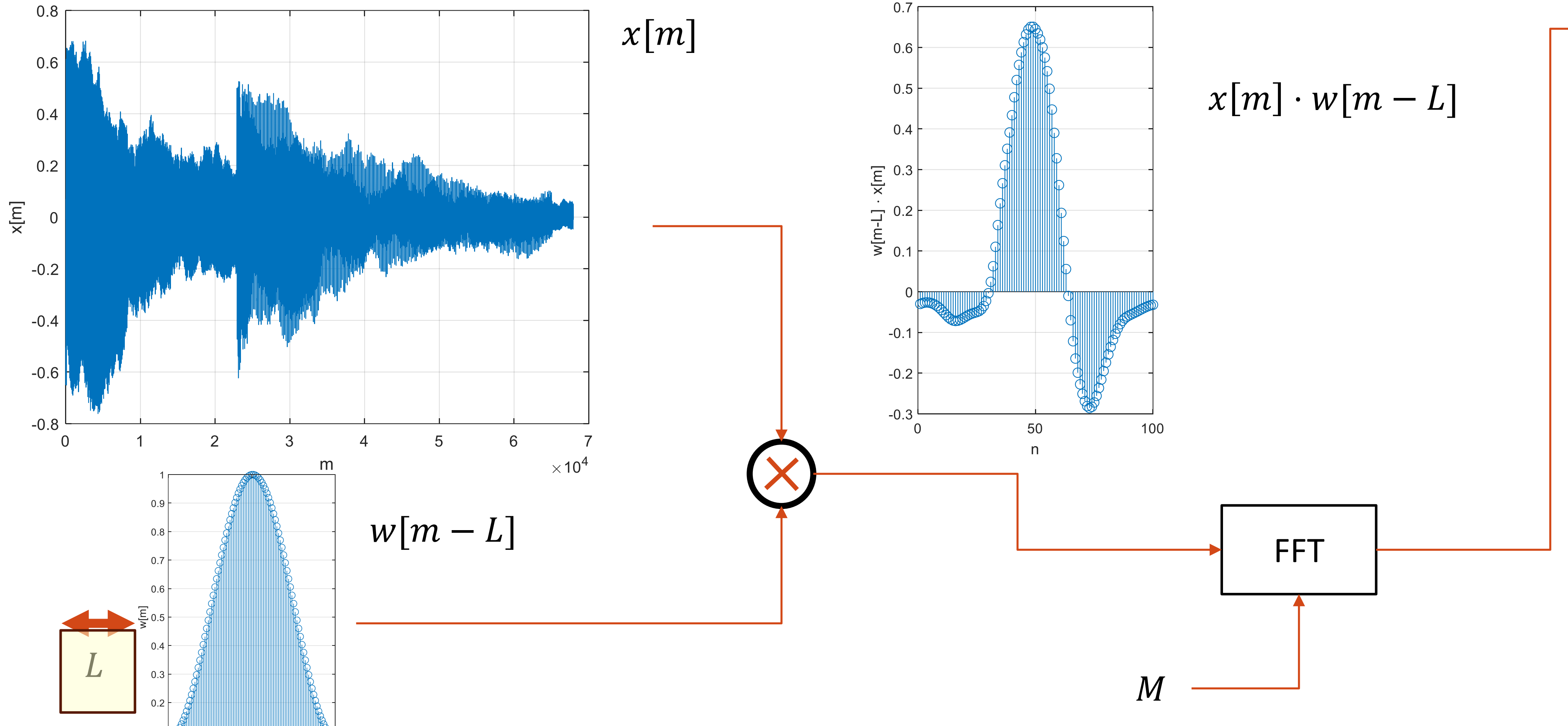
Parametri: forma e durata della finestra, ordine M della TFD

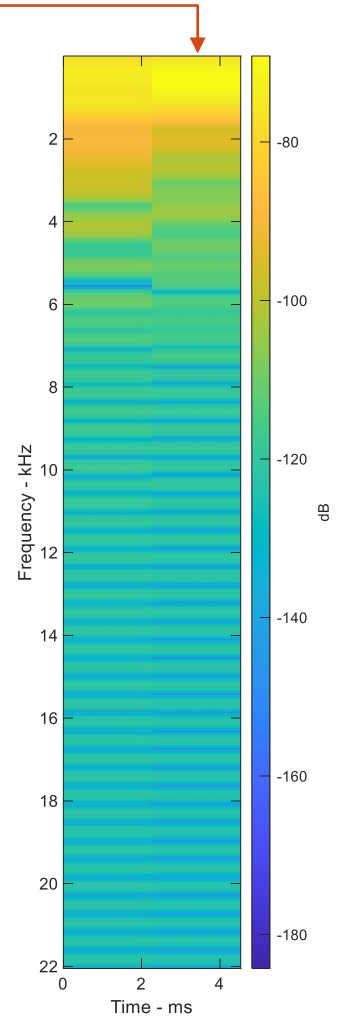
Lo Spettrogramma



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



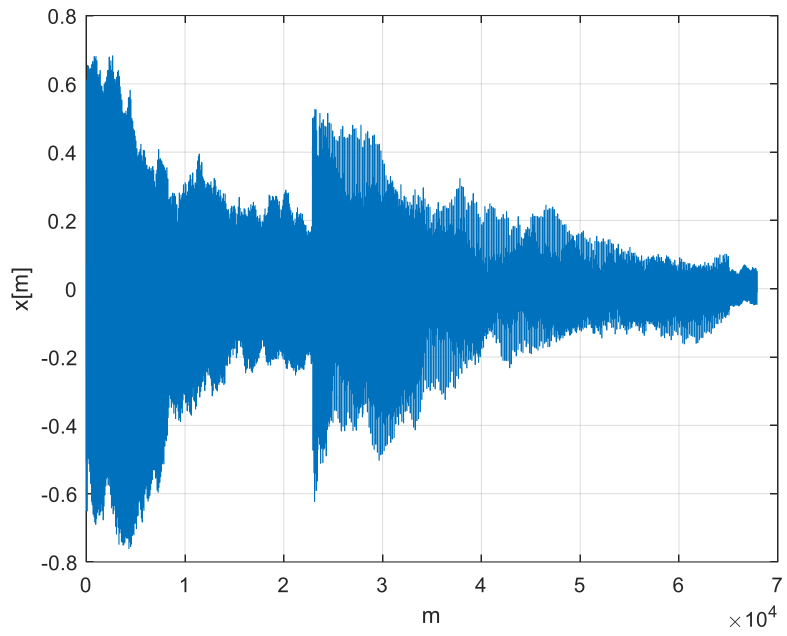


Anche il parametro L va scelto opportunamente (vd seguito)

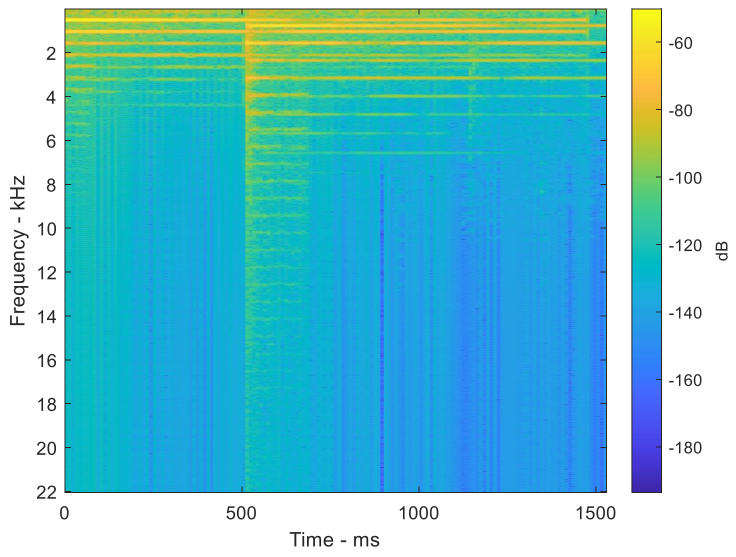
Continuando a «spostare» w si genera tutto lo spettrogramma

UNIVERSITÀ DEGLI STUDI DI PADOVA

Spettrogramma completo



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



Lo Spettrogramma: parametri (1 su 3)

Nel calcolo dello spettrogramma ci sono diversi parametri da considerare:

- La durata della finestra N_w: una finestra breve permette una risoluzione temporale fine, perché evita di analizzare insieme parti del segnale a distanza di più di N_w campioni. La risoluzione temporale è $\boldsymbol{\Delta_T} = N_w T_c = \boldsymbol{N_w / F_c}$
- Tuttavia, la risoluzione in frequenza è inversamente proporzionale alla durata della finestra, perché una sinusoide genera uno spettro con lobo principale che ha larghezza Δ_F inversamente proporzionale a N_w (vd. Lab 3): tale larghezza indica la risoluzione in frequenza della finestra. Questo perché se un segnale contiene 2 sinusoidi con differenza di frequenza minore di Δ_F, esse saranno indistinguibili nello spettro (generano un solo lobo)

Esempi: per la finestra rettangolare, la risoluzione in pulsazione è $\Delta_\omega = \frac{4\pi}{N_w}$ (vd. Lab 3), che corrisponde alla freq. numerica $\Delta_\nu = \frac{\Delta_\omega}{2\pi} = \frac{2}{N}$ ed infine alla risoluzione in frequenza naturale pari a $\Delta_F = F_C \Delta_\nu = \boldsymbol{\frac{2F_C}{N_w}}$

Per la finestra di Hamming, $\Delta_\omega = \frac{8\pi}{N_w}$ e quindi $\boldsymbol{\Delta_F = \frac{4F_C}{N_w}}$

Esempio: risoluzione in frequenza

Illustriamo con un esempio la risoluzione in frequenza delle finestre

Si considerano due segnali

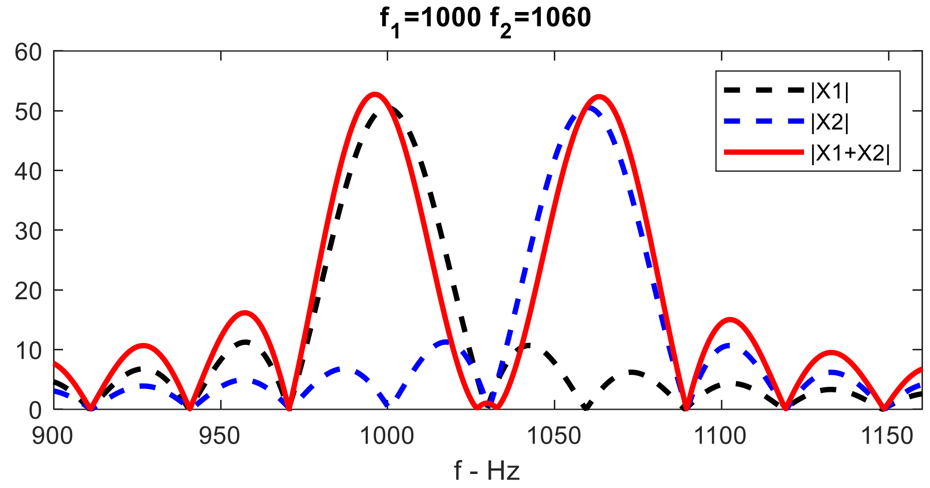
$$x_1(t) = \cos 2\pi f_1 t$$
$$x_2(t) = \cos 2\pi (f_1 + \Delta f) t$$

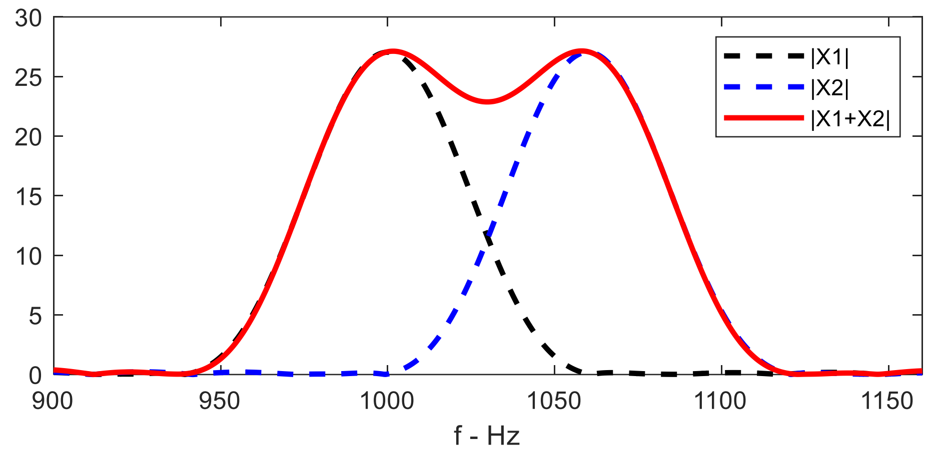
campionati con frequenza $f_C = 3\text{kHz}$.
Si usa una finestra di $N_w = 100$ campioni per l'analisi.

Se si usa una finestra rect (sopra), la risoluzione in frequenza è $\Delta_F = \frac{2F_C}{N_w} = 60\text{Hz}$ e i due lobi sono separati in frequenza

Se si usa una finestra di Hamming (sotto), la risoluzione è $\Delta_F = \frac{4F_C}{N_w} = 120\text{Hz}$ e: i due lobi si sovrappongono e si considera che le due sinusoidi non sono risolte

Potete generare questi grafici con lo script `esempi_risoluzione_frequenza.m`





Lo Spettrogramma: parametri (2 su 3)

- Il tipo di finestra w: la scelta della finestra implica un trade-off tra *decadimento asintotico* e ampiezza del lobo centrale. Tale aspetto non è facile da valutare, e la scelta della finestra ottimale dipende dalla natura del problema e dalle nostre conoscenze a priori sui segnali. Per esempio, per risolvere due sinusoidi ad ampiezza comparabile può essere più opportuno scegliere una finestra con lobo centrale stretto (rect), mentre per risolvere due sinusoidi con ampiezze molto diverse è meglio prendere una finestra con decadimento asintotico veloce

- 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. Tuttavia la scelta di M non cambia la «forma» dello spettro

Lo Spettrogramma: parametri (3 su 3)

Infine, in genere si calcola $\mathcal{S}_x(k,n)$ non per tutti i valori di n tra 0 e $N_x - N_w$, ma con un sottocampionamento temporale per ridurre il costo computazionale

In altre parole, si fissa un fattore di "decimazione" 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 altro valore tipico è $L = \frac{N_w}{2}$. In questo caso, il segnale è suddiviso in blocchi, ma la seconda metà di ogni blocco coincide con 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, perché si eseguono circa $\frac{N_x - N_w}{L}$ calcoli di FFT di ordine M per calcolare tutto lo spettrogramma

UNIVERSITÀ DEGLI STUDI DI PADOVA

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)
% Calcola e mostra in una figura lo spettrogramma del segnale x
% Fc - frequenza di campionamento di x in Hz
% Nw - durata della finestra di analisi (numero campioni)
% L - intervallo tra due finestre consecutive (numero di campioni)
% M - parametro di zero padding
% tipo - nome della finestra (stringa): 'rect' o 'hamming'
%
% (C) Marco Cagnazzo, Università di Padova, 2024-2025


% Inizializzazioni
Tc = 1/Fc;  winPeriod = L*Tc; Nx =numel(x);
% Creazione della finestra
switch lower(tipo)
    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: solo frequenze positive
nTimes = ceil((Nx-Nw)/L); % nr campioni tempo dello spettrogramma
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); % Calcolo campioni TF
    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/(2*M)];
% Creazione figura
figure;  imagesc(timeAxis,freqAxis, SX);
c = colorbar; c.Label.String = 'dB';
xlabel('Time - ms'); ylabel('Frequency - Hz');
```

Esercizio 4

Utilizzare lo spettrogramma per determinare le note contenute nel file `dueNote.wav` ed il loro istante di onset con risoluzione temporale di 10 ms e con risoluzione in frequenza di 10 Hz

Ricordiamo che la risoluzione temporale Δ_T è legata alla durata della finestra N_w dalla relazione $\Delta_T = N_w T_c$.

La risoluzione in frequenza è invece data da $\Delta_F = \frac{4F_C}{N_w}$ nel caso di finestra di Hamming e $\Delta_F = \frac{2F_C}{N_w}$ per una finestra rettangolare.

Per semplicità, i valori di L e M sono fissati: $L = N_w$ e $M \geq 8N_w$; inoltre useremo la finestra di Hamming.

Suggerimento: usare due volte lo spettrogramma, una per ottenere i tempi di attacco ed una per individuare le frequenze delle note.

Una volta ottenuto lo spettrogramma, i parametri possono essere stimati per **ispezione visuale**

Esercizio 4

$$\Delta_T = N_w/F_c$$

$$\Delta_F = \frac{4F_C}{N_w}$$

Si fornisce una bozza della soluzione: completare ed eseguire

Nella prima parte, trovare il numero di campioni della finestra in funzione della risoluzione temporale.

Poi determinare la risoluzione in frequenza risultante.

Nella seconda parte invece il numero di campioni della finestra va determinato in funzione della risoluzione in frequenza che si vuole ottenere. Si calcola poi la risoluzione nel tempo risultante.

```
clear variables
close all
% Lettura dati
[suono, Fc] = audioread('dueNote.wav');
TC = 1/Fc; % Periodo di campionamento
tipo = 'hamming'; % Tipo di finestra
%% Prima parte: determinare l'attacco con Delta T = 10 ms
DeltaT = 10e-3 ; % Ris. temporale (durata della finestra in sec.)
Nw =  ???? ; % Durata in numero di campioni (numero intero!)
DeltaF = ??? % Risoluzione in frequenza
L = Nw; % In questo modo, analizziamo blocchi separati
% Per lo zero-padding, prendiamo M >= 8Nw
M = 2^(nextpow2(Nw) + 3);
SX = spettrogramma(suono, Fc, Nw, L, M, tipo);
title(sprintf('\\Delta_T: %d ms \\Delta_F: %d Hz', (DeltaT*1000),
DeltaF));
```

```
%% Seconda parte: determinare le note con Delta F = 10 Hz
DeltaF = 10;
% Determinare la durata della finestra
% in funzione della risoluzione in frequenza
Nw = ??? ; % deve essere un numero intero!
% Risoluzione temporale in secondi
DeltaT = ??? ;
L = Nw; %  come prima
M = 2^(nextpow2(Nw) + 3); %  come prima
SX = spettrogramma(suono, Fc, Nw, L, M, tipo); grid minor
title(sprintf('\\Delta_T: %d ms \\Delta_F: %d Hz',
DeltaT*1000, DeltaF));
```