



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.

| | | | | | | | | | | | | | |
|-------------|------|------------------------|-----|-----------------------|-----|-----|-----------------------|-----|-----------------------|-----|-----|------------------------|------|
| | | | | | | | | | | | | | |
| <i>Nota</i> | 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



```
clearvars; close all;
% Lettura dati
file = 'nota1.wav';
% Legge i campioni e la freq. camp.
[nota, Fc] = audioread(file);
TC = 1/Fc; % Periodo di campionamento

% Riproduzione suono
soundsc(nota, Fc);

% Analisi in frequenza
N = numel(nota); % numero di campioni del segnale
M = 2^(nextpow2(N)+3); % Zero-padding: M >= N*8
step = Fc/M; fStart = -Fc/2; fStop = Fc/2 - step;
F = fStart:step:fStop;

% calcolo campioni della Tftc
X = TC*fft(nota, M);
```

```
% Visualizzazione spettro di ampiezza
X_amp = fftshift(abs(X));
% Freq in Hz, ampiezze in decibel
figure(1); plot(F, 10*log10(X_amp)); grid;
% Traccia solo la parte d'interesse del grafico
axis([0 2500 -50 0])
xlabel('f - Hz'); ylabel('|X| - dB'); title(file)

% Determinazione della nota
[XM, ind] = max(X_amp); f0 = abs(F(ind));
fprintf('L'armonica principale è a %3d Hz\n', round(f0));
hold on; plot(f0, 10*log10(XM), 'o', 'linewidth', 2);
stringa = sprintf('Armonica principale\n a %d Hz', round(f0));
text(f0+25, -10, stringa);
```



Esercizio 2: Soluzione

```
clearvars; close all;
% Lettura dati
file = 'nota1.wav';
% Legge i campioni e la freq. camp.
[nota, Fc] = audioread(file);
TC = 1/Fc; % Periodo di campionamento

% Riproduzione suono
soundsc(nota, Fc);

% Analisi in frequenza
N = numel(nota); % numero di campioni del segnale
M = 2^(nextpow2(N)+3); % Zero-padding: M >= N*8
step = Fc/M; fStart = -Fc/2; fStop = Fc/2 - step;
F = fStart:step:fStop;

% calcolo campioni della Tftc
X = TC*fft(nota, M);
```

Nella prima parte della soluzione c'è da osservare:

1. L'uso del comando `nextpow` per generare il valore di M (leggere la documentazione di `nextpow`)
2. Il calcolo dell'asse delle frequenze: si parte da $-\frac{F_c}{2}$, e a passi di $\frac{F_c}{M}$ si arriva a $\frac{F_c}{2} - \frac{F_c}{M}$
3. Il calcolo dei campioni della T_{ftc} tramite FFT, zero-padding, riscalatura (cioè moltiplicazione per TC)



Nella seconda parte osserviamo:

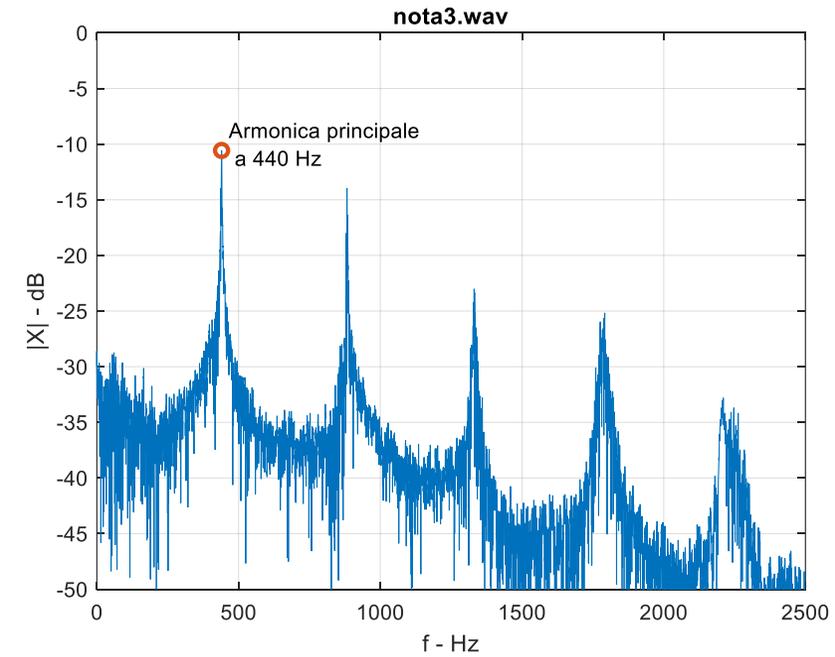
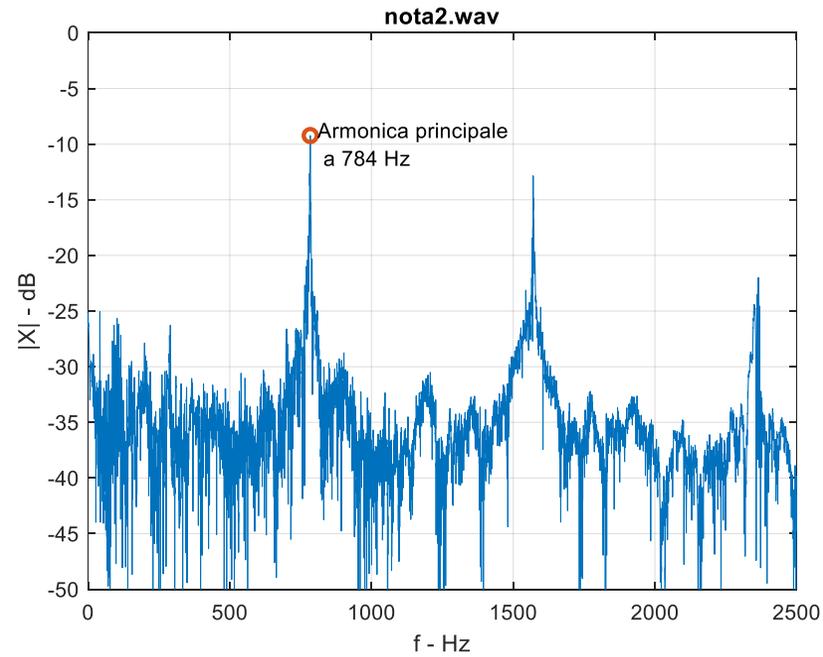
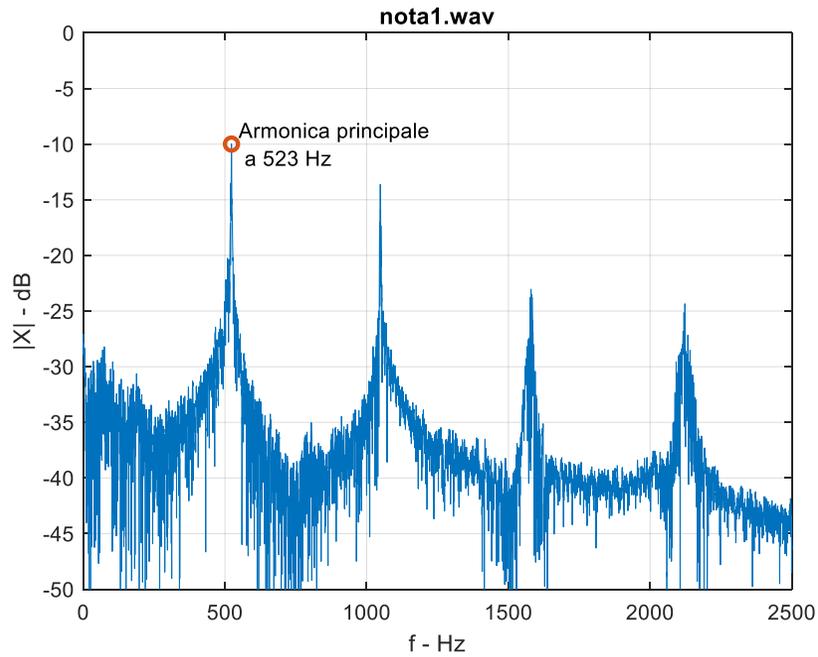
1. L'uso di `fftshift` per centrare i valori dello spettro di ampiezze intorno alla frequenza zero
2. L'uso di una scala logaritmica per lo spettro di ampiezza
3. L'uso del comando `axis` per visualizzare solo le frequenze relativamente basse, dove presumibilmente troveremo la prima armonica (armonica principale)
4. L'armonica principale viene individuata tramite il valore massimo dello spettro di ampiezza
5. Notare che la funzione `max` di Matlab restituisce la posizione del *primo* massimo assoluto dell'ingresso. Lo spettro di ampiezza è pari, quindi ha due massimi uguali a frequenze $-f_0$ e $+f_0$. Siccome abbiamo effettuato `fftshift` il primo massimo sarà $-f_0$. Ecco perché c'è bisogno di `abs`
6. Il comando `hold on` mantiene nella figura corrente lo spettro di ampiezza e permette di aggiungere un cerchio nella posizione del massimo. Senza `hold on`, il secondo `plot` cancellerebbe il grafico precedente
7. Il comando `text` posiziona una stringa su di un grafico

```
% Visualizzazione spettro di ampiezza
X_amp = fftshift(abs(X));
% Freq in Hz, ampiezze in decibel
figure(1); plot(F,10*log10(X_amp)); grid;
% Traccia solo la parte d'interesse del grafico
axis([0 2500 -50 0])
xlabel('f - Hz'); ylabel('|X| - dB'); title(file)

% Determinazione della nota
[XM, ind] = max(X_amp); f0 = abs(F(ind));
fprintf('L'armonica principale è a %3d Hz\n', round(f0));
hold on; plot(f0,10*log10(XM), 'o', 'linewidth',2);
stringa = sprintf('Armonica principale\n a %d Hz', round(f0));
text(f0+25, -10, stringa);
```



Esercizio 2: Soluzione



Si notino negli spettri di ampiezza le armoniche secondarie a frequenze multiple della principale