***Ghazouline Abdessamad***

# Relazione sull'Analisi della Distorsione Audio tramite Sistemi Non Lineari

## Introduzione

In questa esperienza analizzeremo l'effetto della distorsione in sistemi non lineari.
La distorsione introduce componenti in frequenza non presenti nel segnale originale e altera le componenti esistenti, influenzando la qualità del suono.
Per evidenziare gli effetti, utilizzeremo un sistema composto da:

- Un ingresso audio $x(t)$
- Un filtro passa-basso con banda programmabile $B$
- Un sistema non lineare descritto dalla funzione $g(y)$
- Un'uscita distorta $z(t) = g(y(t))$

L'analisi verrà svolta mediante Matlab, che elaborerà i dati in ingresso e scriverà i campioni elaborati su un file audio per l'ascolto.

## 1. Lettura del Segnale Audio

Il segnale audio di ingresso è un file in formato WAV con una frequenza di campionamento di 44.1 kHz. Viene letto ed elaborato in MATLAB, escludendo i primi 0.1 secondi per rimuovere eventuali transitori presenti nel file.

Di seguito il codice utilizzato per leggere il file audio e selezionare il canale sinistro per l'elaborazione:

```
filename = 'audio/Handel.wav'; % Nome del file audio
fCampionamento = 44.1e3; % Frequenza di campionamento [Hz]
tempoCampionamento = 1 / fCampionamento; % Intervallo tra due campioni
durataTransitorio = 0.1; % Eliminazione transitorio iniziale [s]
durata = 6.0; % Durata totale del segnale da elaborare [s]
numeroCampioni = durata * fCampionamento; % Numero totale di campioni da
estrarre
inizioCampioni = durataTransitorio * fCampionamento; % Offset iniziale

[xstereo, fc] = audioread(filename, [inizioCampioni+1,
inizioCampioni+1+numeroCampioni]);
x = (xstereo(:,1))'; % Selezione del canale sinistro
tempo = (0:length(x)-1) * tempoCampionamento; % Creazione vettore dei tempi
```

## 2. Filtro Passa-Basso

Per limitare la banda di frequenza del segnale in ingresso, applichiamo un filtro passa-basso con frequenza di taglio 5 kHz. Il filtro è realizzato con una convoluzione tra il segnale e la funzione sinc opportunamente troncata e moltiplicata per una finestra rettangolare.

```
B = 5000; % Banda del filtro passa-basso [Hz]
T = 20 / B; % Tempo di troncamento [s]
tempoFiltro = 0:tempoCampionamento:T; % Creazione asse temporale per il filtro
h = 2 * B * sinc(2 * B * (tempoFiltro - T/2)) .* rectpuls((tempoFiltro - T/2) /
T);
```

## 3. Filtraggio del Segnale

L'operazione di filtraggio viene eseguita tramite convoluzione discreta tra il segnale e la risposta impulsiva del filtro. Successivamente, viene eliminato il transitorio iniziale introdotto dalla convoluzione.

```
y = conv(x, h) * tempoCampionamento; % Convoluzione tra segnale e filtro
y = y(length(h):length(y)); % Eliminazione transitorio iniziale
tempoY = tempo(1:length(y)) + T/2; % Correzione del ritardo di fase introdotto
dal filtro
```

## 4. Introduzione della Non Linearità (Clipping)

Il blocco non lineare introduce un effetto di saturazione, che limita l'ampiezza del segnale a una soglia prestabilita $y_M = 0,10$. Questo effetto causa una distorsione introducendo armoniche non presenti nel segnale originale.

```
yM = 0.10; % Valore massimo di saturazione
z = y; % Copia del segnale filtrato
z(y > yM) = yM; % Saturazione positiva
z(y < -yM) = -yM; % Saturazione negativa
```

## 5. Analisi in Frequenza

Per analizzare l'effetto della distorsione, calcoliamo la trasformata di Fourier del segnale filtrato e del segnale distorto. La rappresentazione spettrale permette di osservare la nascita di armoniche aggiuntive dovute alla saturazione.

```
lunghezzaFft = 2^nextpow2(length(y));
Y = fft(y, lunghezzaFft) * tempoCampionamento;
Y = [Y(lunghezzaFft/2+1:end), Y(1:lunghezzaFft/2)];
Z = fft(z, lunghezzaFft) * tempoCampionamento;
Z = [Z(lunghezzaFft/2+1:end), Z(1:lunghezzaFft/2)];
frequenza = fCampionamento * linspace(-0.5, 0.5, lunghezzaFft);
```

## 6. Coefficienti di Distorsione

Per quantificare la distorsione generata dal blocco non lineare, calcoliamo i coefficienti di distorsione come rapporto tra le ampiezze delle armoniche spurie e l'armonica fondamentale.

```
figure;
plot(frequenza(1:length(Y)) / 1e3, 20 * log10(abs(Y) ./ max(abs(Y))), 'c',
'LineWidth', 1.5);
hold on;
plot(frequenza(1:length(Z)) / 1e3, 20 * log10(abs(Z) ./ max(abs(Y))), 'k',
'LineWidth', 1.5);
xlabel('Frequenza (kHz)'); ylabel('Ampiezza (dB)');
grid on;
legend('|Y(f)|', '|Z(f)|');
```

## Conclusioni

L'analisi ha evidenziato come la distorsione dovuta alla saturazione si manifesti con la generazione di armoniche spurie. L'effetto è più marcato per segnali di ampiezza maggiore, mentre per segnali a basso livello l'effetto è minore.