Elaborazione Audio

Audio digitale - Trattamento

- L'audio digitale risulta più semplice da trattare rispetto all'audio analogico. Non dobbiamo preoccuparci di ulteriori distorsioni introdotte dai dispositivi di editing oltre ad avere tutta la potenza di calcolo di un elaboratore;

- Una volta noto il metodo di rappresentazione per l'audio digitale che vogliamo trattare e consapevoli del significato delle grandezze fisiche che descrivono i suoni, possiamo andare ad agire come preferiamo;

- Ad esempio se volessimo aumentare il volume di un certo fattore? Basterebbe moltiplicare il valore di ogni campione per quel fattore (stando attenti a non andare fuori range).

Audio digitale - Trattamento

Per semplicità possiamo sempre effettuare un pre-processing per riportarci a dei valori comodi per i nostri calcoli. Una scelta tipica è quella di considerare i valori d'ampiezza nell'intervallo [-1, 1]. Cioè il massimo valore nella rappresentazione originale corrisponderà ad 1, mentre il minimo a -1. Allo 0 si associa di norma il **silenzio.**

Un'altra considerazione importante riguarda la durata di un suono. Viste le frequenze di campionamento tipiche (es: 44100 Hz), anche una traccia audio relativamente breve (es: 30 secondi) sarebbe composta da milioni di campioni. Diciamo un **vettore** di milioni di elementi. Di solito di divide il vettore in pezzi più piccoli, chiamati *frame*, la cui dimensione è tipicamente una potenza di 2. Valori tipici sono **1024 o 512**. Nulla vieta di trattare i campioni tutti in una volta, ma intuitivamente capiamo, che per un essere umano, un singolo campione riprodotto sarebbe pressoché privo di significato, diverso è il discorso per i *frame*.

Audio digitale - Frame

Una volta definita la lunghezza N del *frame*, si può ricavare la sua durata. In particolare se f_c è la frequenza di campionamento dell'audio in esame, l'intervallo di tempo associato al *frame* sarà:

$$T = \frac{N}{f_c}$$

La dimensione del frame definisce anche la **risoluzione in frequenza**, cioè la frequenza più piccola rappresentabile nello **spettro** del frame. Lo spettro si ottiene applicando la **Trasformata Discreta di Fourier** al frame. La risoluzione in frequenza R vale:

$$R = \frac{f_c}{N}$$

Audio digitale - Trattamento

Le tipologie di trattamento per un segnale audio digitale potrebbero essere classificate nel seguente modo (non necessariamente):

- Ricampionamento;
- Riquantizzazione;
- Aumento/diminuzione canali;
- Modifiche dell'ampiezza;
- Filtraggio delle frequenze;
- Modifiche dell'inviluppo;
- Effetti sonori (eco ecc..);
- E molto altro…

Audio digitale - Ricampionamento

Ricampionare un segnale digitale, significa modificare il numero di campioni che fluiscono nell' unità di tempo.

- **Sovracampionamento**: se si passa da un numero di campioni più basso ad uno più alto, siamo costretti ad aggiungere nuovi elementi. Per stimare il loro valore si utilizzano di norma tecniche di **interpolazione**.

- **Sottocampionamento:** in questo caso dobbiamo necessariamente buttare via dei campioni. Non solo perdiamo informazione, ma se esageriamo rischiamo di introdurre **aliasing**!

Il ricampionamento è la stessa operazione che effettuiamo quando facciamo lo scaling di un'immagine.

Audio digitale - Riquantizzazione

Riquantizzare un segnale digitale, significa modificare il numero di parole di bit utilizzate per rappresentarne i valori

Quanto si riquantizza di solito si diminuisce la profondità in bit dell'audio originale. In questo modo si può risparmiare memoria al prezzo di un po' di precisione. Chiaramente va tenuto a mente che la riquantizzazione può essere anche non uniforme, il che ci permette di preservare meglio la qualità in molte situazioni. Vedi codifiche **Law**.

Come il ricampionamento, la riquantizzazione si utilizza anche nelle immagini per creare ad esempio delle palette di colori.

Audio digitale – Numero di canali

Per migliorare l'esperienza sonora è possibile utilizzare più flussi informativi differenti: canali. Il numero di canali può essere diminuito o aumentato.

- **Aggiunta:** purtroppo aggiungere un canale non è semplice. I canali aggiunti andrebbero stimati attraverso considerazioni non banali. A volte semplicemente si replicano per adattare l'audio ad un determinato impianto di riproduzione.

- **Rimozione:** la rimozione è chiaramente più sensata. In questo caso un canale può essere totalmente rimosso, oppure può essere «sommato» insieme ad altri per dare vita ad un singolo canale. Ad esempio un audio stereo (2 canali) può essere trasformato in audio mono (1 canale) facendo la media tra i due flussi.

Quando da RGB si passa a scala di grigi si sommano opportunamente i 3 canali.

Audio digitale – Modifica ampiezza

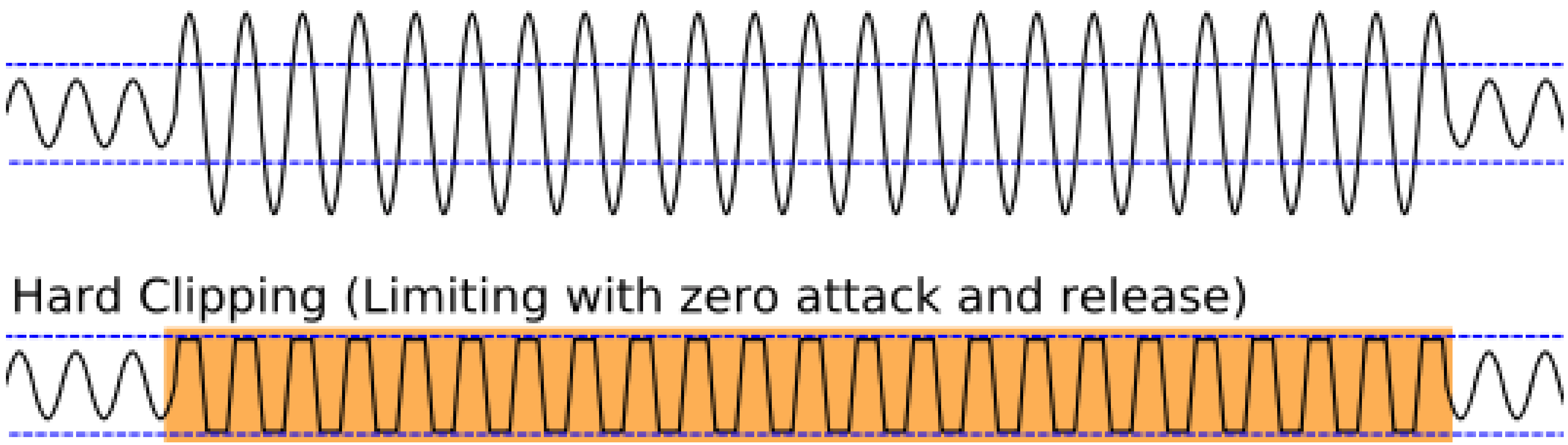
Se volessimo modificare il volume di un fattore K sappiamo che la grandezza su cui agire e l'ampiezza. In particolare ci basta moltiplicare per K il vettore dei campioni (ogni frame se abbiamo suddiviso)

Sia $\overline{x_i}$ l'i-esimo frame di un segnale audio digitale, allora il frame risultante $\overline{y_i}$ con volume aumentato/diminuito di un fattore K, si ottiene:

$$\overline{y_i} = K * \overline{x_i}$$

ATTENZIONE! Se dopo il calcolo alcuni valori finiscono oltre il range di rappresentazione, questi saranno riportati al massimo valore disponibile. Questo effetto è noto come **clipping**.

Audio digitale - Clipping



Il **clipping** si verifica anche amplificando un segnale analogico oltre il massimo valore che il diffusore acustico che utilizziamo è in grado di riprodurre!

Il clipping rappresenta in effetti un limite del software e dell'hardware! L'unica cosa che possiamo fare è **massimizzare il volume**, cioè trovare il più grande K che non provochi clipping.

Audio digitale - Maximize

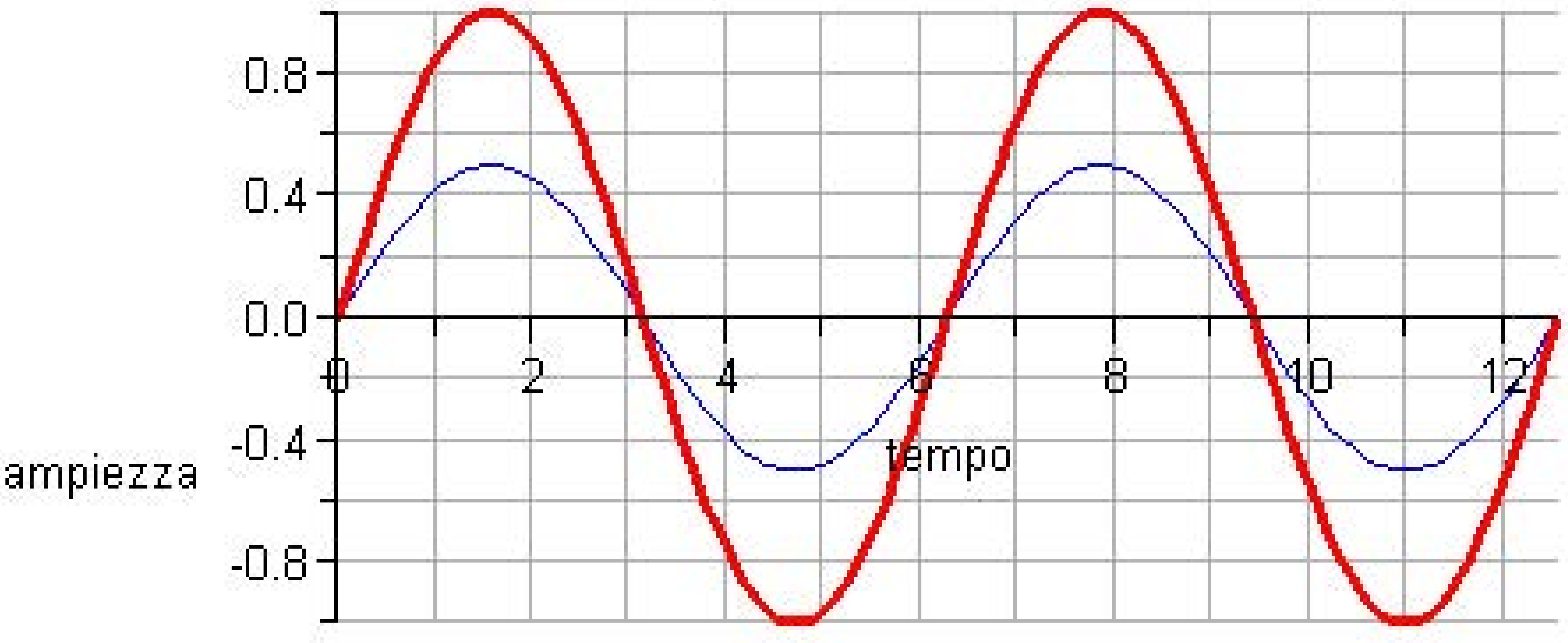
L'operazione **Maximize** permette di aumentare il volume di ogni frame al massimo, senza introdurre clipping. Si normalizzano le ampiezze facendo sì che il massimo valore di ogni frame assuma il valore della massima ampiezza rappresentabile.

Sia $\overline{x_i}$ l'i-esimo frame di un segnale audio digitale, allora il frame risultante $\overline{y_i}$ con volume massimizzato, si ottiene:

$$\overline{y_i} = \frac{M}{K_i} * \overline{x_i} \qquad\qquad K_i = \max\{|\overline{x_i}|\}$$

M è il massimo valore d'ampiezza rappresentabile. Tipicamente vale 1 oppure $2^N - 1$, dove N è il numero di bit di quantizzazione.

Maximize - Esempio



Un tono puro (in blu) la cui ampiezza viene massimizzata (in rosso). Il tutto considerando un intervallo [-1,1] di possibili ampiezze.

Audio digitale - Minimize

Se invece di aumentare volessimo diminuire al minimo il volume si può eseguire un’operazione **Minimize**.

Sia $\overline{x_i}$ l’i-esimo frame di un segnale audio digitale, allora il frame risultante $\overline{y_i}$ con volume minimizzato, si ottiene:

$$\overline{y_i} = sign(\overline{x_i}) * (|\overline{x_i}| - K_i) \qquad K_i = \min\{|\overline{x_i}|\}$$

Questa operazione porta a 0 (silenzio) quello che era il valore minimo di ampiezza in ogni frame.

Modifica inviluppo – Fade In

Il **Fade In** è l'operazione che modifica l'inviluppo di un suono, facendo sì che l'ampiezza vari in un crescendo da un frame iniziale a, ad uno finale b, di un fattore da 0 a 1.

Sia $\overline{x_i}$ l'i-esimo frame di un segnale audio digitale, $\boldsymbol{a}$ l'indice del frame da cui inizia il Fade In, $\boldsymbol{b}$ quello in cui termina, $\boldsymbol{N} = b - a$ il numero di frame interessati dal Fade In e $f(x)$ una funzione peso, allora il frame risultante $\overline{y_i}$ si ottiene:

$$\overline{y_i} = f(i - a) * \overline{x_i} \qquad\qquad a \leq i \leq b$$

ATTENZIONE! La funzione f utilizzata deve godere di tre proprietà!

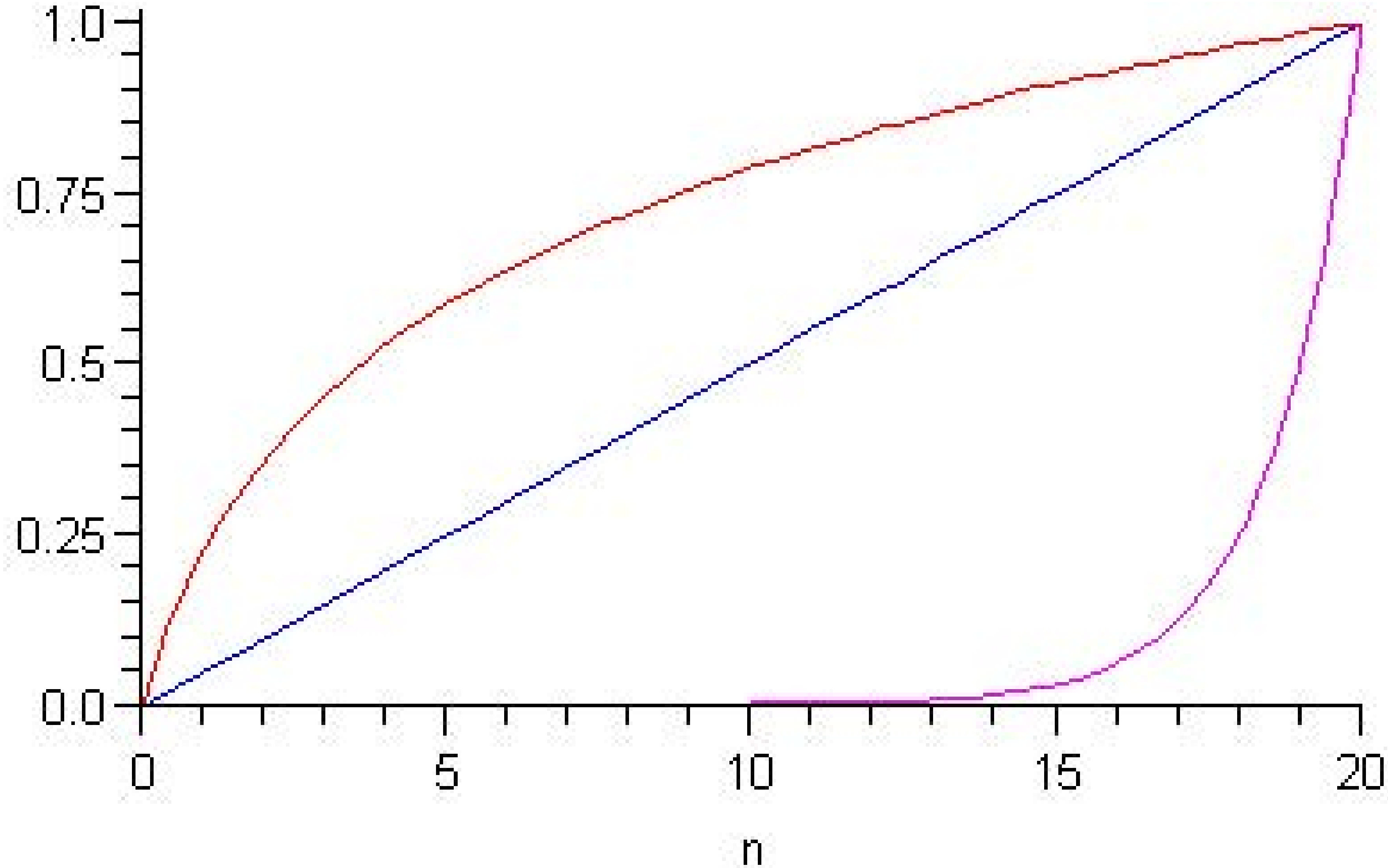
- $f: [0, N] \rightarrow [0,1]$
- f monotona crescente.
- $f(0) = 0$ e $f(N) = 1$

Audio analogico – Funzioni Fade In

Alcune possibili scelte per le funzioni f sono le seguenti:

- **Lineare:** $$f(n) = \frac{n}{N}$$
- **Logaritmica:** $$f(n) = \log_{1+N}(1+n)$$
- **Esponenziale:** $$f(n) = \begin{cases} 0 & n = 0 \\ \beta^{n-N} & con\ \beta > 1 \end{cases}$$

Esempio funzioni peso



Un esempio di funzioni per $N = b - a = 20$. In rosso la funzione logaritmo, in blu la funzione lineare e in viola una funzione esponenziale.

Modifica inviluppo – Fade Out

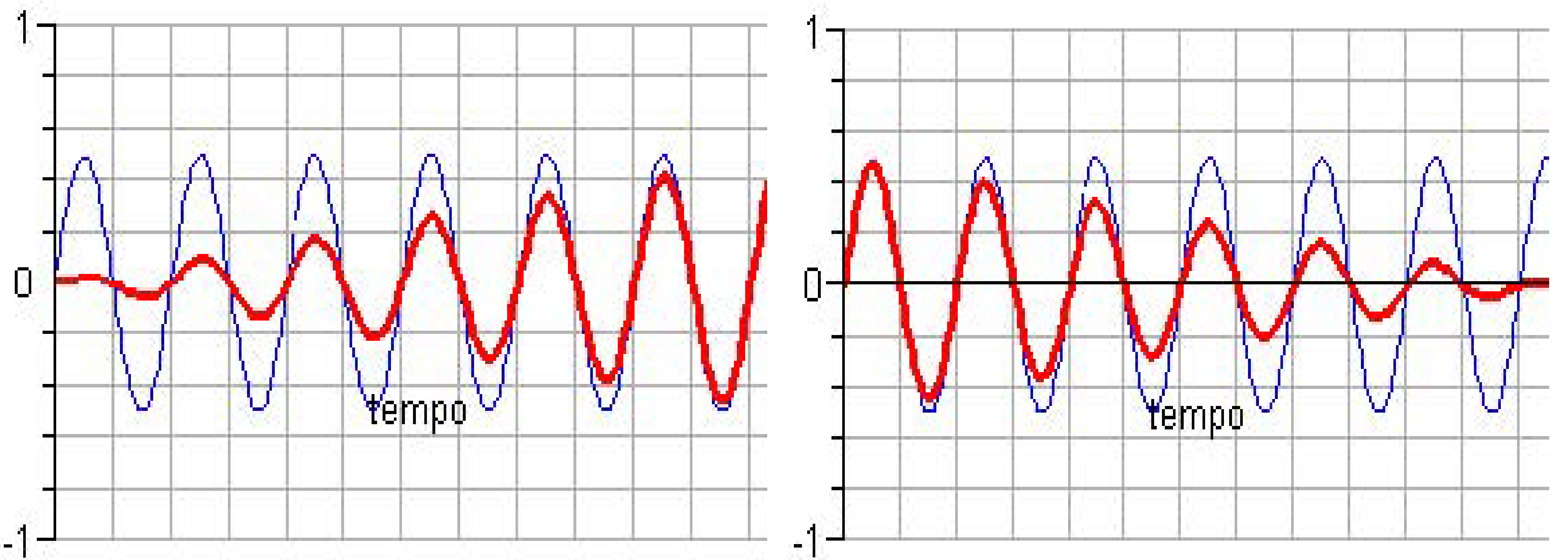
Esattamente speculare al Fade In, troviamo il **Fade Out.** Il trattamento modifica l'inviluppo di un suono, facendo sì che l'ampiezza vari descrescendo da un frame iniziale a, ad uno finale b, di un fattore da 1 a 0.

Nelle stesse identiche condizioni del Fade In:

$$\overline{y_i} = (1 - f(i - a)) * \overline{x_i} \qquad\qquad a \leq i \leq b$$

Come si può osservare non si fa altro che sottrarre ad 1 la funzione vista prima. In questo modo anziché crescere da 0 a 1, il fattore peso di ogni frame $\overline{x_i}$ descrescerà da 1 a 0.

Fade In/Out – Esempio



A sinistra un esempio di Fade In e a destra uno di Fade Out.

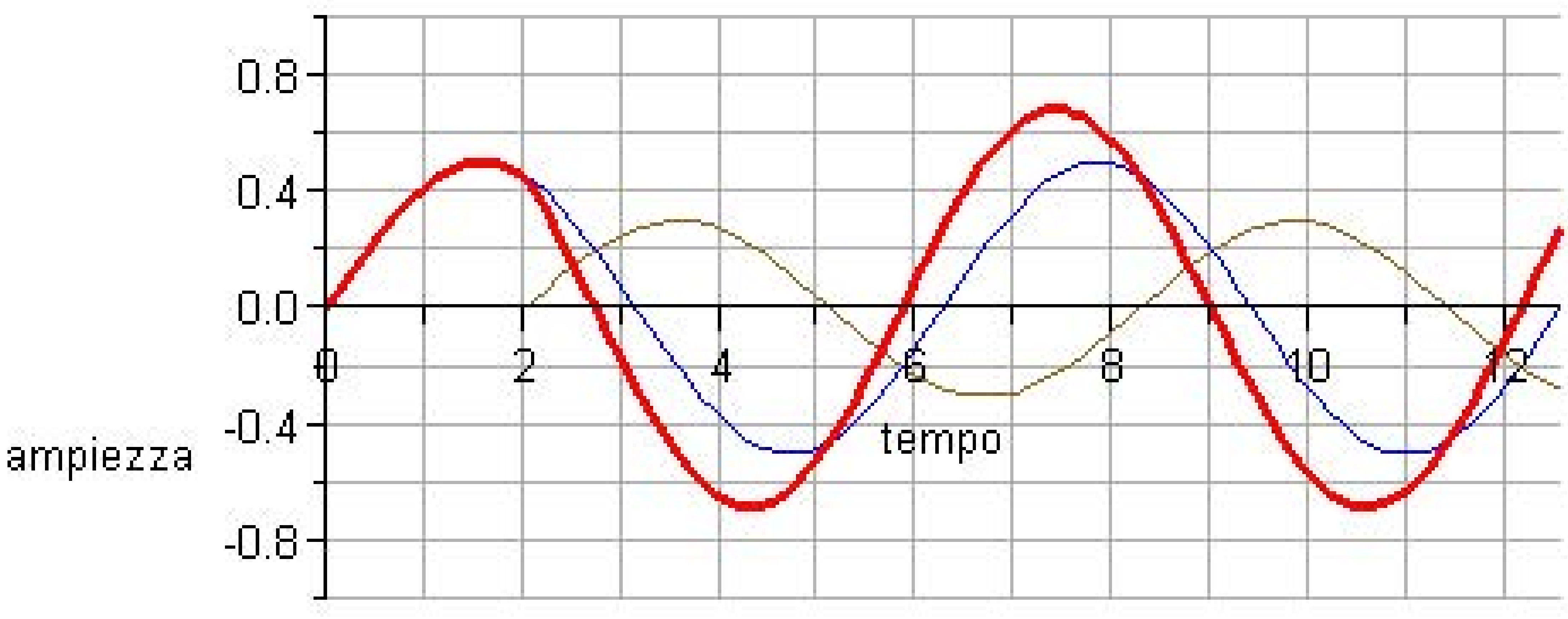
Audio digitale – Eco

Se volessimo simulare l'effetto dell'eco cosa dovremmo fare? Ci basta sommare ad un frame, uno dei precedenti opportunamente pesato (per simulare la perdita di energia).

Sia $\bar{x_i}$ l'i-esimo frame di un segnale audio digitale, $\boldsymbol{r}$ il numero di ripetizioni dell'eco, $\boldsymbol{d}$ un intero positivo che indichi il ritardo in frame, $\boldsymbol{D}$ un fattore di decadimento per pesare sempre meno i frame più lontani nel tempo, allora il frame risultante $\bar{y_i}$ si ottiene:

$$\bar{y_i} = \bar{x_i} + \sum_{j=1}^{r} (D^j * \bar{x}_{i-j*d}) \qquad 0 < D \leq 1$$

Esempio Eco



Eco (rosso) su onda sinusoidale (blu). L'onda marrone è l'onda originale replicata con un ritardo $d = 2$ e decadimento $D = \mathbf{0.6}$.

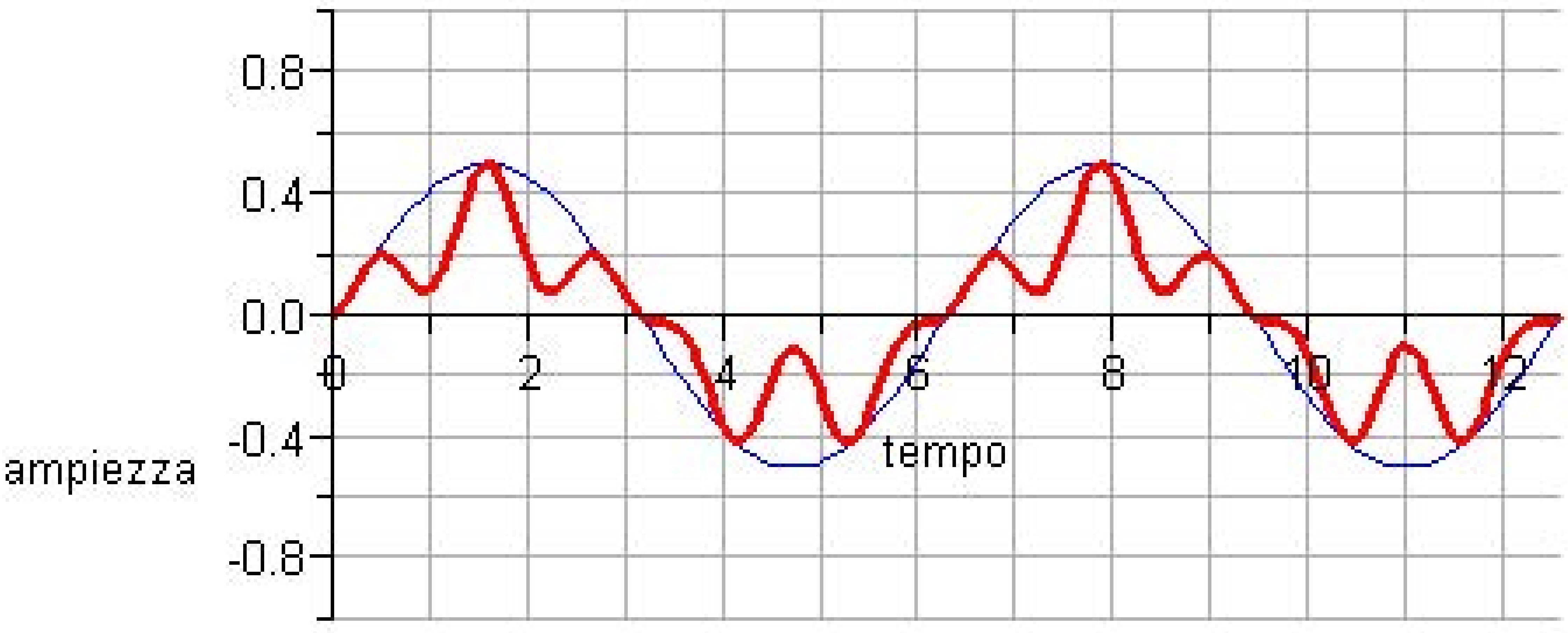
Audio digitale – Tremolo

E' possibile far variare l'ampiezza in maniera sinusoidale attorno al valore originale, introducendo una sorta di effetto di vibrazione chiamato **Tremolo**.

Sia $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$ l'i-esimo frame di dimensione, $K, \alpha\,,\ \beta$ dei fattori costanti, f_c la frequenza di campionamento del segnale audio, il frame risultante $\bar{y}_i$ si ottiene:

$$\bar{y}_i = \bar{x}_i * \left(\alpha + \beta \sin\left(\mathrm{K}\frac{i\pi}{f_c}\right)\right) \qquad 0 < \beta \leq \alpha \leq 1$$

Esempio Tremolo



In blu un tono puro e in rosso lo stesso tono dopo aver applicato l'effetto tremolo.

Elaborazioni del range dinamico

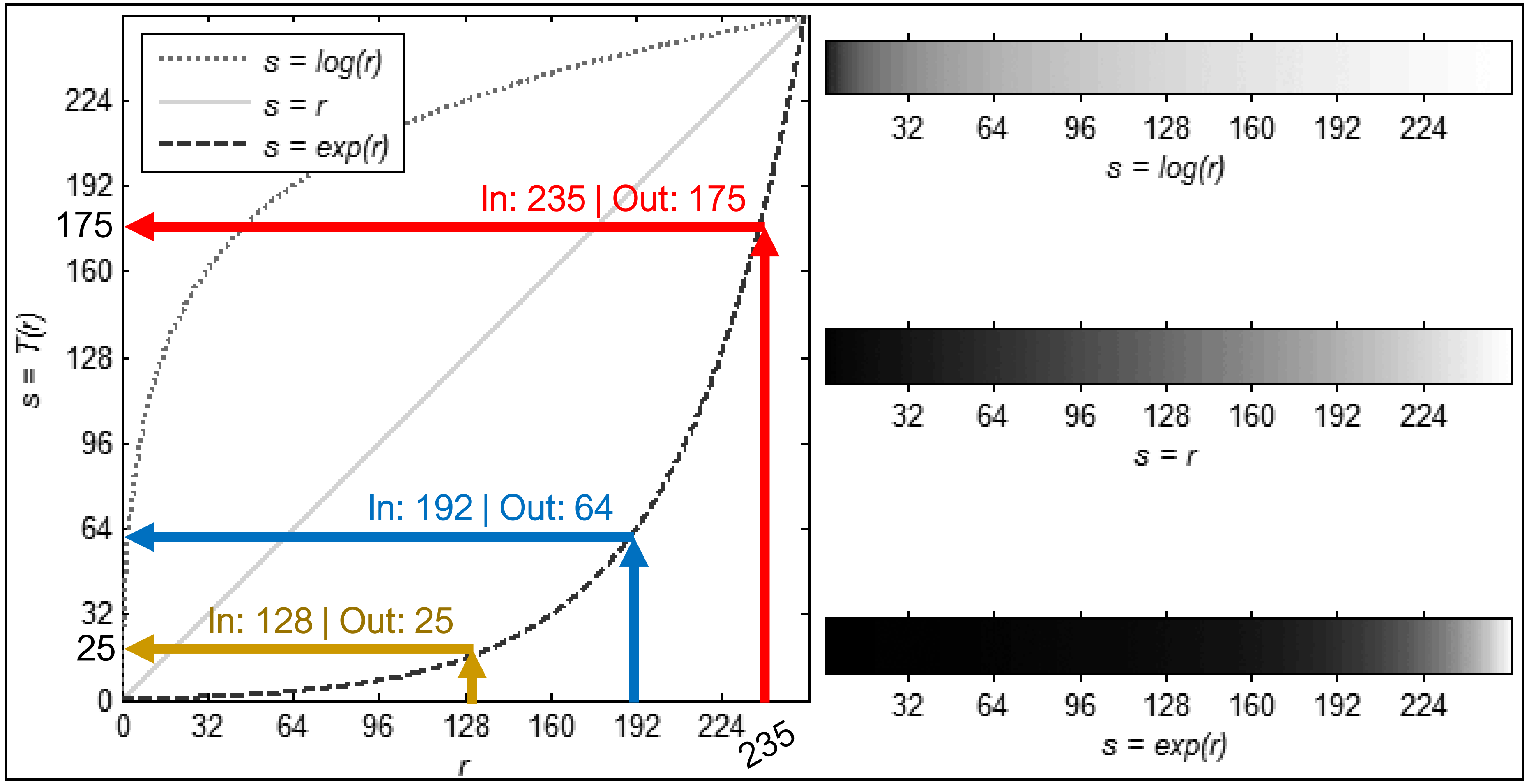
Si tratta di elaborazioni che alterano la rapidità con cui i valori del segnale audio variano all'interno del proprio range, imponendo anche dei minimi e dei massimi differenti da quelli originali.

Nelle operazioni che vedremo non servirà lavorare per frame. Ogni campione elaborato dipenderà solo da se stesso. Si può quindi definire una funzione di mapping descrivibile con una Look-up-table (**LUT**).

Tipicamente non si lavora con i valori espressi in codifica **PCM**, bensì esprimendo tali valori in **dBFS** (decibels relative to full scale).

Look-up-table

La LUT è un costrutto per associare a un certo valore di input un determinato valore di output. Nell'esempio vediamo la rappresentazione grafica di due LUT che descrivono due operazioni di alterazione dei livelli di grigio per le immagini. Noi comunque ci concentreremo sull'audio.



Decibels relative to full scale

I **dBFS** permettono di rappresentare i valori di ampiezza di un segnale audio in funzione del massimo valore disponibile nella codifica PCM.

ATTENZIONE! I segni non vengono rappresentati, poiché i dBMS sono una misura dell'energia, per cui se si vuole tornare al valore PCM originale bisogna «ricordare» i segni.

- V_{PCM} è il valore in decimale della codifica PCM a n bit che vogliamo esprimere in dBFS.

$$E_{dBFS} = 20\log_{10}\frac{|V_{PCM}|}{2^{n-1}}$$

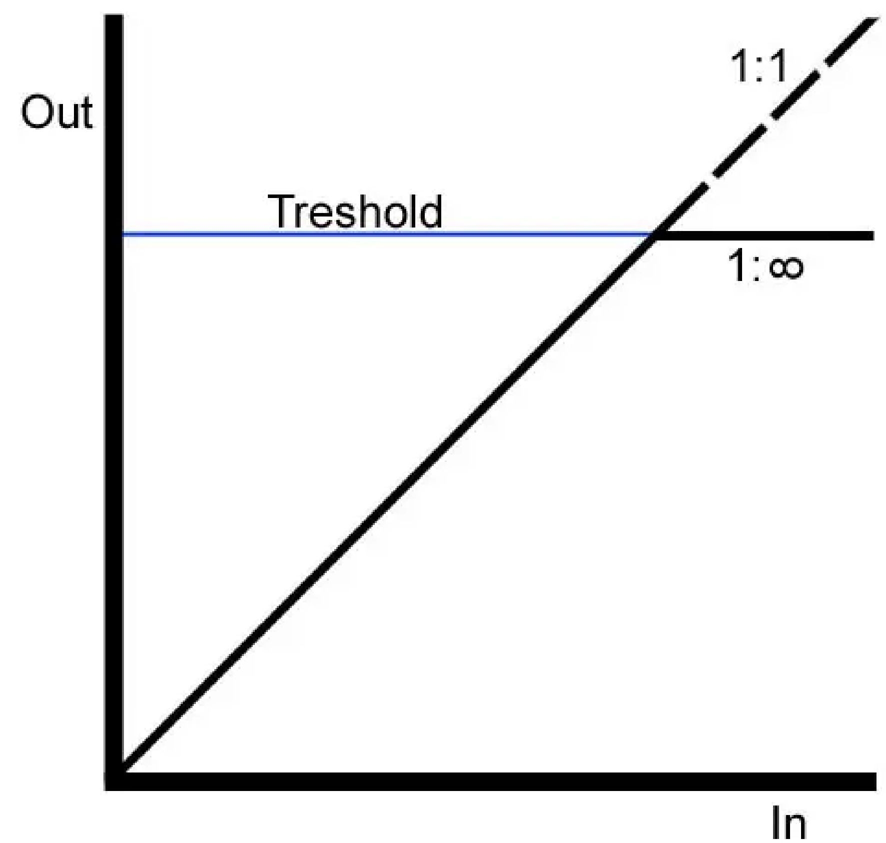
- Il denominatore non è altro che il valore massimo possibile per la codifica PCM a *n* bit.
- Il massimo valore di E_{dBFS} è quindi 0.

PCM	**128**	**-64**	**1**	**-1**	**10**
dBFS	**0**	**-6.0**	**-42.1**	**-42.1**	**-22.1**

Operazioni: Limitatore (Limiter)

Impone un clipping. Se i valori superano una certa soglia T vengono riportati al valore di T. Quelli inferiori alla soglia rimangono inalterati. La funzione è la seguente

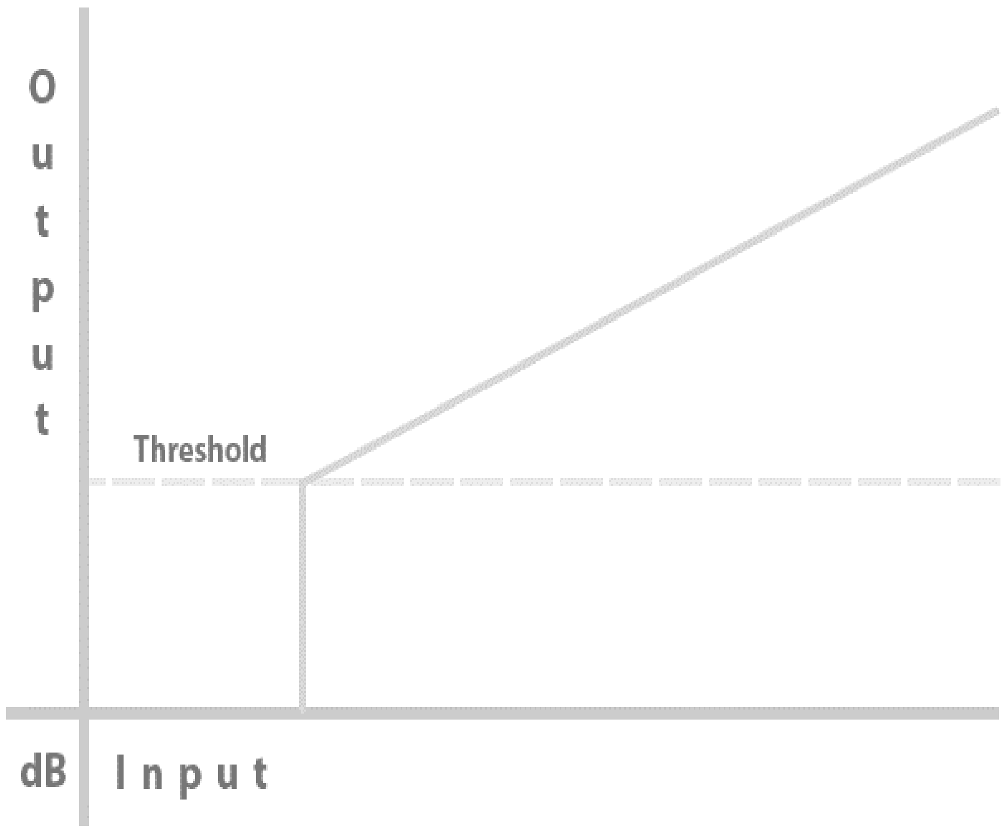
$$f(x) = \begin{cases} x & se\ x < T \\ T & se\ x \geq T \end{cases}$$



Operazioni: Noise Gate

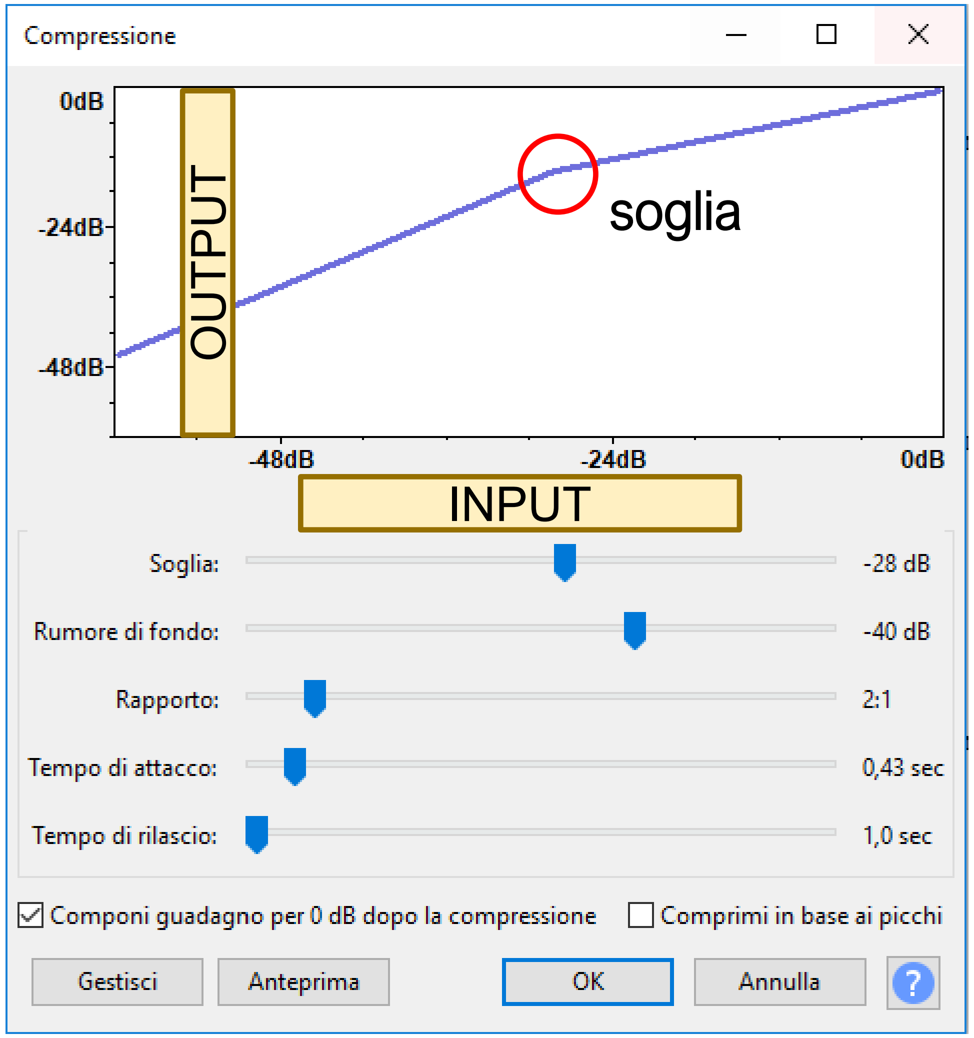
Ha un comportamento opposto al limitatore. Tutti i valori sotto la soglia T vengono portati al **minimo possibile** (0 nel caso di rappresentazione in PCM), ossia alla non udibilità.

$$f(x) = \begin{cases} \min & se\ x < T \\ x & se\ x \geq T \end{cases}$$

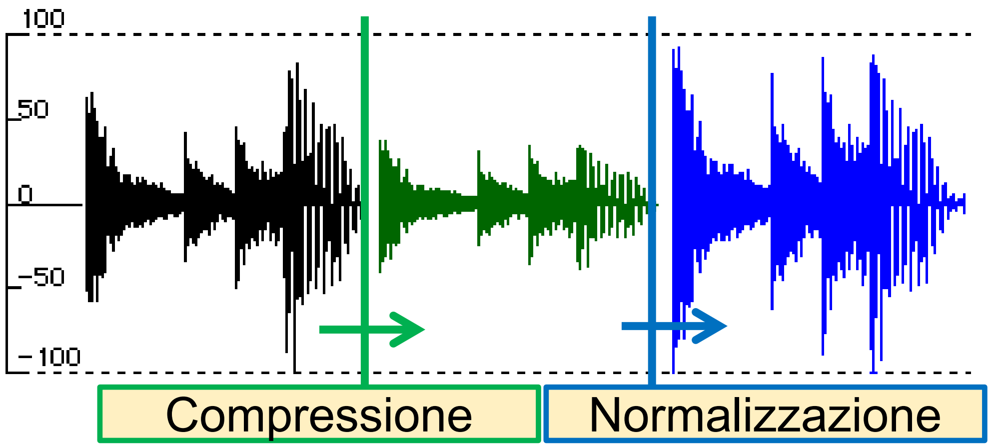


Operazioni: Compressore (compressor)

- Diminuisce le ampiezze sopra una soglia (modalità **Downward)**
- Aumenta le ampiezze sotto una soglia (modalità **Upward)**
- Quindi… riduce il range dinamico.



Tipicamente dopo questa operazione si applica una Normalizzazione (o Maximize)



Operazioni: Compressore

Sia T la soglia scelta e R il fattore di compressione, allora il compressore opera secondo le seguenti funzioni in modalità **downward** e **upward**.

$$f(x) = \begin{cases} x & se\ x < T \\ \dfrac{(x-T)}{R} + T & se\ x \geq T \end{cases}$$

downward

Le funzioni sono sostanzialmente identiche, cambia solo il fatto che il segnale viene alterato sopra la soglia (downward) o sotto la soglia (upward)

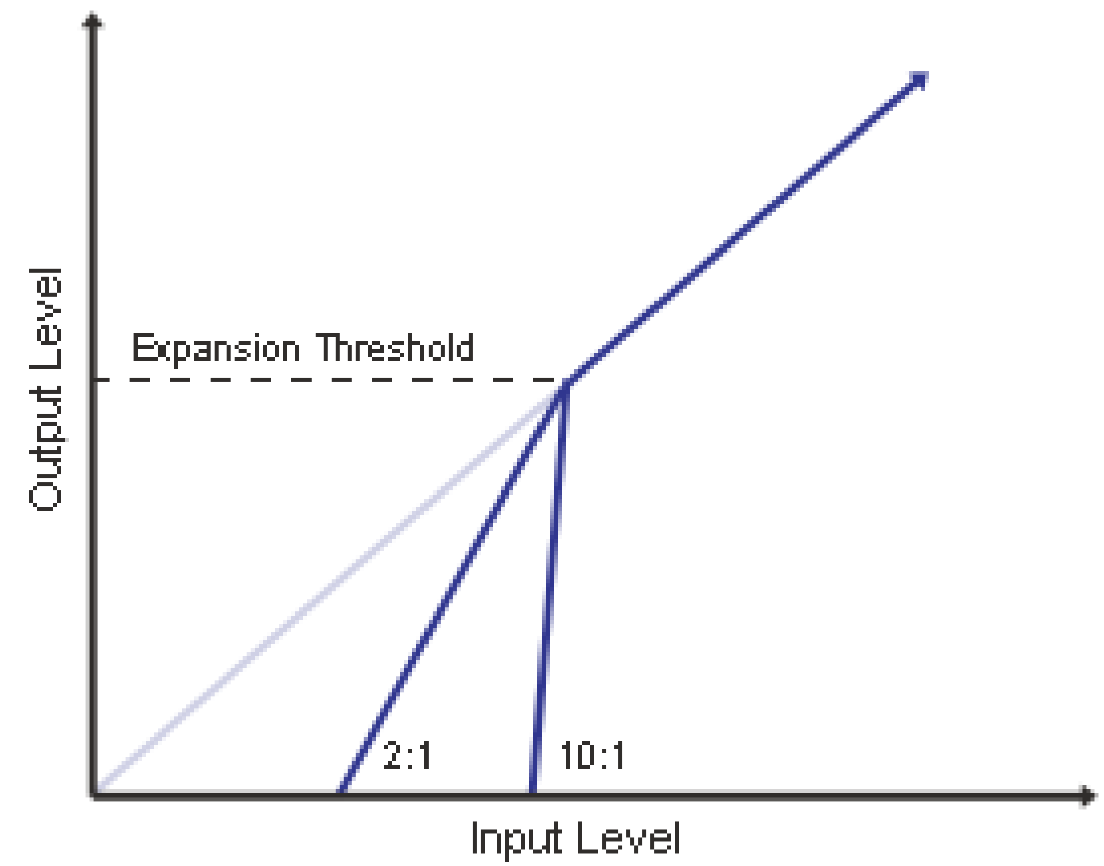
$$f(x) = \begin{cases} \dfrac{(x-T)}{R} + T & se\ x < T \\ x & se\ x \geq T \end{cases}$$

upward

Che succede se R tende a infinito?

Operazioni: Espansore (expander)

- Aumenta le ampiezze sopra una soglia (modalità **Upward)**
- Diminuisce le ampiezze sotto una soglia (modalità **Downward)**
- Quindi… aumenta il range dinamico. Comportamento opposto al compressore.



Operazioni: Espansore

Sia T la soglia scelta e R il fattore di espansione, allora l'espansore opera secondo le seguenti funzioni in modalità **downward** e **upward**.

$$f(x) = \begin{cases} (x - T) * R + T & se\ x < T \\ x & se\ x \geq T \end{cases}$$

downward

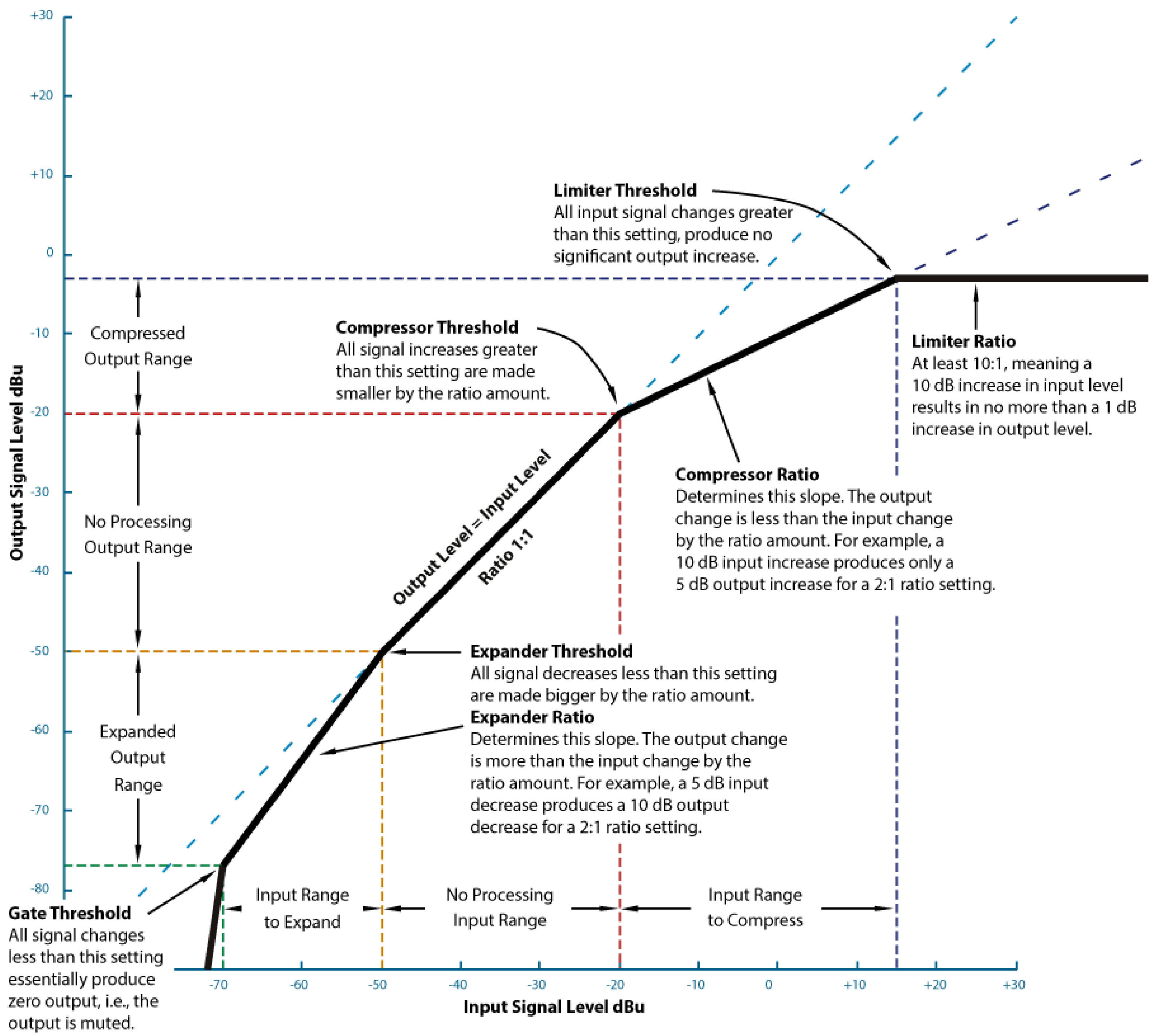
Le funzioni sono sostanzialmente identiche, cambia solo il fatto che il segnale viene alterato sopra la soglia (upward) o sotto la soglia (downward)

$$f(x) = \begin{cases} x & se\ x < T \\ (x - T) * R + T & se\ x \geq T \end{cases}$$

upward

Che succede se R tende a infinito?

Operatori su range dinamico



Sono stati presentati alcuni operatori che alterano il range dinamico e delle possibili funzioni per la loro applicazione. Nulla vieta di usare funzioni differenti, che ad esempio abbiano un andamento più graduale in corrispondenza delle soglie.

Il range dinamico può anche essere modificato utilizzando combinazioni o ulteriori variazioni degli operatori visti.

Ricordiamo sempre che se si sforano il minimo e il massimo rappresentabili si verifica comunque il fenomeno del **clipping**.

Trasformata discreta di Fourier

Forma analitica **simmetrica** della Trasformata e Antitrasformata discreta di Fourier. MATLAB di default usa una forma **asimmetrica**, ma si può scegliere anche la versione simmetrica settando un parametro.

Sia $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$ l' i-esimo frame di dimensione N, allora il vettore di coefficienti associato $\overline{\boldsymbol{X}_{\boldsymbol{j}}}$ si può ottenere nel seguente modo:

$$\overline{\boldsymbol{X}_{\boldsymbol{i}}}(\boldsymbol{u}) = \frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} \bar{\boldsymbol{x}}_{\boldsymbol{i}}(a)\, e^{-j2\pi\frac{ua}{N}}$$

Per tornare a $\bar{\boldsymbol{x}}_{i}$ partendo dai coefficienti $\overline{\boldsymbol{X}_{\boldsymbol{i}}}$ si utilizza l'antitrasformata:

$$\bar{\boldsymbol{x}}_{\boldsymbol{i}}(\boldsymbol{u}) = \frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} \overline{\boldsymbol{X}_{\boldsymbol{i}}}(a)\, e^{j2\pi\frac{ua}{N}}$$

DFT – Significato coefficienti

- Il vettore dei coefficienti $\overline{\boldsymbol{X_i}}$ relativo ad un frame, non è altro che lo **spettro** del frammento di audio rappresentato nel frame stesso!

- Ogni elemento del vettore $\overline{\boldsymbol{X_i}}$ rappresenta l'energia associata ad una certa frequenza, cioè l'ampiezza per la sinusoide a quella frequenza.

- Se R è la **risoluzione in frequenza**, all'elemento i-*esimo* del vettore $\overline{\boldsymbol{X_i}}$ corrisponderà la frequenza $j * R$.

- In questo modo al primo elemento corrisponderà la frequenza $\frac{f_C}{N}$ e all'ultimo elemento la frequenza di campionamento f_c. La gamma delle frequenze rappresentabili è così coperta.

- Ricordiamo: nulla vieta di processare l'intero audio senza suddividere in frame (come se avessimo un unico frame).

Audio digitale – Filtraggio Temporale

- Come per le immagini è possibile applicare delle trasformazioni di filtraggio, cioè di «selezione» o «eliminazione» di alcuni elementi del segnale.

- Per l'audio digitale possono essere definiti operatori puntuali o locali come per le immagini. Gli effetti finali sono pressoché identici;

- Ad esempio la **convoluzione** (ad una dimensione) con un filtro di media corrisponde ad uno smoothing del segnale, cioè un'attenuazione delle alte frequenze. Può essere usato per attenuare il **rumore bianco gaussiano**.

- Usare un **filtro mediano** permette invece di ridurre il rumore impulsivo, cioè picchi di ampiezza che si presentano con una certa probabilità.

Audio digitale – Filtraggio Frequenziale

- Un'altra possibilità per filtrare l'audio digitale è quello di agire direttamente sullo **spettro**.

- In questo modo possiamo attenuare, amplificare o rimuovere le frequenze che ci interessano.

- I filtri da applicare sono gli stessi usati per le immagini: **Filtro passa basso ideale, gaussiano, di butterworth ecc..**

- Un' applicazione tipica è quella di individuare un segnale di disturbo in una parte dell'audio in cui ci si aspetta del **silenzio.** Scoperto lo spettro del rumore si può procedere attenuando quelle frequenze.

- Chiaramente la pipeline è la seguente: **DFT->Filtraggio->IDFT**