

Analisi Multitono di Elaborazioni digitali

Proseguendo con la trattazione dell'elaborazione di segnali audio viene presentata una tecnica di misura delle prestazioni di elaboratori digitali del suono. Tale tecnica è derivata dall'analisi multitono utilizzata da tempo dalla rivista *Audio Review* (impiegando apparecchiature professionali da laboratorio).

Il vantaggio principale della nostra versione è che può essere completamente realizzata su di un PC bello robusto senza la necessità di strumenti dedicati rimanendo quindi alla portata del lettore medio.

È doveroso sottolineare subito che questa misura, come ogni altra prova strumentale, si affianca e non può sostituire le prove di ascolto

Introduzione

Già da tempo l'informatica "domestica" e l'elaborazione audio si sono incontrate e capita molto spesso di trattare segnali musicali attraverso programmi software che girano su normali computer di basso costo. Si pensi per esempio ai riduttori di rumore ai compressori **MP3** e **WMA** e ai convertitori di formato audio.

Sorge subito il problema di misurare le prestazioni di questi dispositivi e confrontarli tra loro. La soluzione più semplicistica e più usata consiste nell'affermare che il dispositivo Tali Tali conserva la "qualità del CD Audio". Una simile affermazione è molto diffusa, specialmente nella pubblicità, ed è solo parzialmente vera. Lo scopo del presente articolo è di permettere ai lettori di MC di fare personalmente verifiche strumentali.

Le tecniche descritte in questo articolo sono ispirate da numerosi articoli apparsi su **Audio Review** e sono in gran parte frutto di discussioni con Fabrizio Montanucci. I lettori più interessati agli aspetti strettamente Audio sono consigliati di consultare gli articoli in bibliografia; qui ci dedicheremo agli aspetti più tecnici della implementazione in *Mathematica* dei programmi di generazione ed analisi.

La prima cosa da notare è che i trattamenti di compressione e di riduzione di rumore ed in genere tutti le operazioni non banali sul suono sono trasformazioni **non lineari**. Questo fatto toglie la validità teorica alle misure classiche (risposta all'impulso, all'onda quadra) che si basano appunto sulla teoria dei sistemi lineari.

La tecnica di analisi proposta consiste nel generare via soft-

ware dei file **WAV**, elaborarli (per esempio con un processo di compressione e decompressione **MP3**) e analizzare i file **WAV** ottenuti dopo la cura. La parte innovativa della analisi sono i segnali multitono: una serie di segnali sinusoidali puri che nella analisi spettrale hanno la forma di un pettine. Se le frequenze e la lunghezza della finestra di analisi sono scelte opportunamente non vengono prodotti artefatti e tutto lo sporco eventualmente presente è dovuto alla elaborazione. Noi usiamo due diversi gruppi di toni con due diverse finestre di analisi (una lunga $2^{14}=16384$ e una lunga $67 \times 250=16750$) allo scopo di evitare che eventuali interazioni tra la finestra di analisi e l'eventuale finestra utilizzata all'interno del processore sotto esame tolgano validità ai risultati.

Tutte le analisi vengono presentate in un unico grafico allo scopo di facilitarne la leggibilità

Generazione dei file test

I file di prova da generare contengono i seguenti segnali con cui provare il processore sotto esame:

- 58 segnali a frequenze equispaziate di 44100/128 Hz. La scelta delle frequenze permette di fare un'analisi spettrale su 16384 campioni senza produrre artefatti.
- 60 segnali a frequenze equispaziate di 44100/134 Hz. La scelta delle frequenze permette di fare un'analisi spettrale su 16750 campioni senza produrre artefatti.
- Un segnale consistente in 39999 campioni nulli e un campione (in posizione centrale) di valore massimo. Questa analisi permette di rilevare la risposta impulsiva del sistema. Anche in presenza di non linearità se la risposta all'impulso non permette di caratterizzare completamente il comportamento del compressore, ma fornisce comunque interessanti informazioni.
- Alcuni secondi di rumore bianco presi da un disco test.
- Alcuni secondi di rumore rosa presi da un disco test (rumore bianco filtrato) presi da un disco test.

Vediamo come i segnali vengono messi insieme in tre file

WAV (*Mathematica* genera e legge indifferentemente file **WAV** o **AIFF**. Per chi lavora con il Mac conviene generare questi ultimi).

Definiamo dapprima alcune costanti: la frequenza di campionamento (quella usata nello standard CD), il tempo di analisi (5 secondi) e la lunghezza in campioni dei file multitono.

```
In[1]:=
RATE = 44100;
Tanal = 5;
LEN = Tanal*44100;
```

Poi leggiamo i file dei rumori (già estratti da un CD di segnali test ed accorciati con un editor musicale).

```
In[2]:=
snd1 = Import["pink5", "AIFF"];
pink = Take[snd1[[1, 1]], LEN];
snd1 = Import["white5", "AIFF"];
white = Take[snd1[[1, 1]], LEN];
```

Ecco la funzione che genera i multitono, il parametro **gamma** vale **128** o **134** a seconda del file che si vuole generare. Il numero **k** di segnali (**58** o **60**) viene calcolato automaticamente in modo da restare nella banda audio.

```
In[3]:=
genTest[gamma_, time_] := Module[{k, fl},
  k = Floor[20000. gamma/RATE];
  fl = Compile[x,
    Evaluate[N[Sum[Cos[2.π i RATE/gamma
  x], {i, k}]]]];
  N[fl[#/RATE] & /@ Range[Floor[RATE
time]]]]
```

Il primo file stereo contiene **58** toni sul canale destro e rumore rosa sul sinistro.

```
In[4]:=
smp1 = genTest[128, Tanal];
Length[smp1]
snd = Sound[SampledSoundList[{smp1/Max[Abs[smp1]],
pink}, {44100, 16}]];
Export["test128.WAV", snd, "WAV"];
```

Il secondo file stereo (generato in modo analogo) contiene **60** toni sul canale destro e rumore bianco sul sinistro. Il terzo file è dual-mono e contiene l'impulso.

```
In[5]:=
len = 40000;
sample = Range[len]*0;
sample[[Floor[len/2]]] = 1;
snd = Sound[SampledSoundList[{sample, sample}, {44100, 16}]];
Export["testimp.WAV", snd, "WAV"]
```

Analisi dei risultati

L'analisi dei risultati viene effettuata attraverso la Trasformata di Fourier. Lo spettro di potenza viene stimato prendendo una metà della trasformata complessa calcolandone il quadrato del valore assoluto e moltiplicando per opportuni valori di normalizzazione.

Il trattamento della risposta all'impulso effettua una sola

trasformata sui **32768** campioni centrati sul massimo valore assoluto. Il risultato viene messo in un vettore **TF3** di **16384** componenti.

```
In[1]:=
analphuse[file_] := (
  snd1 = Import[file, "WAV"];
  sss1 = snd1[[1, 1]];
  m = Position[Abs[sss1],
  Max[Abs[sss1]]][[1, 1]];
  sss1 = Take[sss1, {m - 16384, m + 16383}];
  n = Length[sss1];
  Tim = n/RATE;
  TF3 = Chop[2 Abs[Take[Fourier[sss1],
  n/2]^2/n]];
  TF3[[1]] = TF3[[1]]/2);
```

L'analisi di un canale è più complicata: vengono buttate via una testa e una coda di **10000** campioni, poi viene fatta la trasformata "a fette" di **n** campioni e viene reso un vettore di **n/2** medie spettrali.

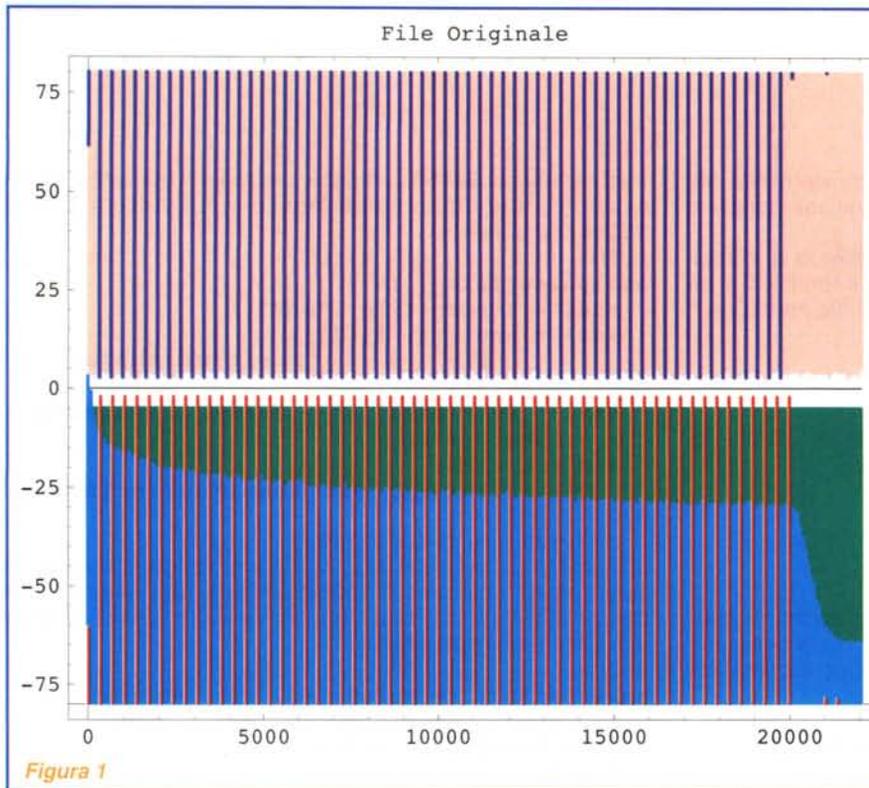
```
In[2]:=
analchan[n_] := Module[{i, tfv, Buffer, tf},
  sss = Drop[Drop[sss, 10000], -10000];
  tf = Array[0 &, n/2];
  i = 0;
  While[Length[sss] > n,
  i++;
  Buffer = Take[sss, n];
  sss = Drop[sss, n];
  tfv = 2 Abs[Take[Fourier[ Buffer,
  n/2]^2/n]];
  tfv[[1]] = tfv[[1]]/2;
  tf += tfv];
  tf/i];
```

L'analisi di un file stereo effettua separatamente l'analisi dei due canali e rende i risultati nei due vettori **TF1** e **TF2**.

```
In[3]:=
analS[file_, LBUFFER_] := (
  snd1 = Import[file, "WAV"];
  n = LBUFFER;
  T = n/44100;
  sss = snd1[[1, 1, 1]];
  TF1 = analchan[n];
  sss = snd1[[1, 1, 2]];
  TF2 = analchan[n]);
```

Presentazione grafica

L'idea della presentazione consistere nel mettere in uno stesso grafico tutte e 5 le risposte in frequenza. la figura è divisa in due partile ordinate contengono valori in dB (da -BASE a 0 quella inferiore e da 0 a BASE quella superiore). Nella parte bassa trovano posto tre grafici, sullo sfondo c'è la risposta impulsiva (in verde) con il bordo superiore a -5dB; sopra di essa c'è la risposta al rumore rosa (in colore celeste) portata a circa -20dB; Infine, nello strato più esterno, ci sono i **58** multitono (in rosso) portati a circa -2dB. Questa parte della figura è es-



```
pk = graf1[TF2, Cyan, 32, T];
analS[file2 <> ".134", 16750];
1134 = graf2[TF1, Blue, 36,
T];
wh = graf2[TF2, Pink, 38, T];
analphulse[file3 <> ".imp"];
im = graf1[TF3, Green, 82,
Tim];
Show[Graphics[{
Line[{{0, 0}, {22050, 0}}],
im, wh, pk, 1128, 1134}],
Axes -> True,
AspectRatio -> 0.8,
Frame -> True,
AxesOrigin -> {0, -BASE},
PlotLabel->Name;)]
```

I risultati

Vediamo dapprima il grafico relativo ai tre file prima della cura.

```
In[1]:=
doAnal[test128.WAV, test134.WAV,
testimp.WAV, "FileOriginale"]
```

(Vedi Figura1).

Si nota che la riduzione a 16 bit non ha effetti visibili a parte una lieve componente continua a -60dB.

Questo è il grafico di riferimento da tenere presente quando si guardano i

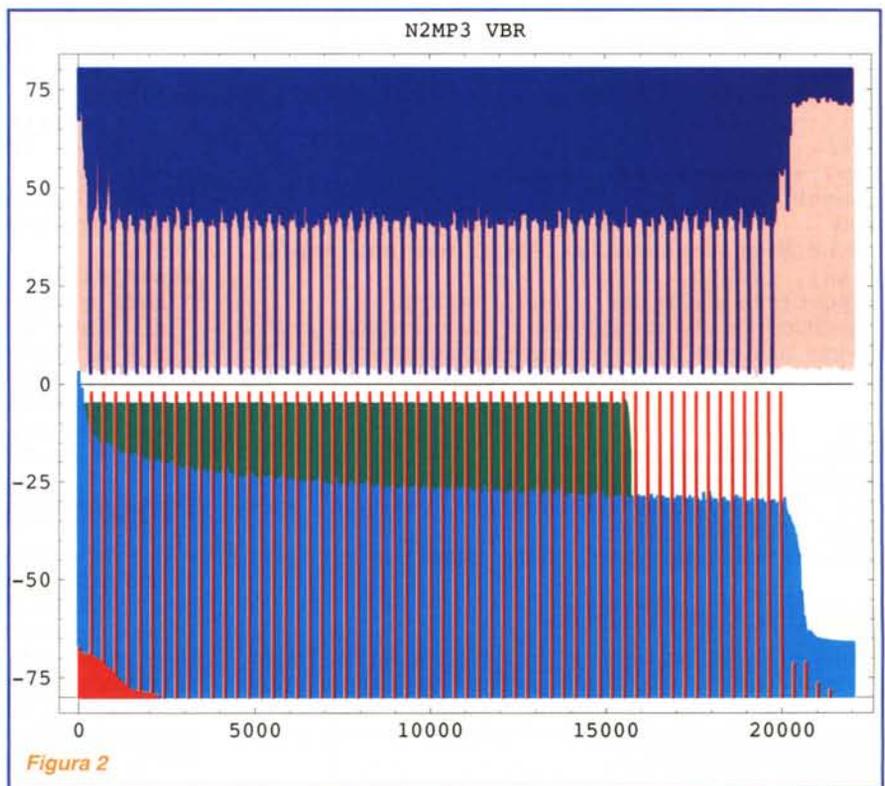
senzialmente quella già pubblicata in **Audio Review** n 201. Nella parte alta sono collocati gli ultimi 2 grafici: sullo sfondo la risposta al rumore bianco (in colore rosa) portata a circa -5dB e nello strato più esterno, ci sono i 60 multitono (in blu) portati a circa -2dB.

La funzione **graf1** disegna i grafici della parte bassa (esiste anche una **graf2** per la parte alta con 2 segni cambiati).

```
In[1]:=
graf1[tf_, col_, off_, t_] := (
vals = Transpose[{N[
Range[n/2]-1]/t,
10Log[10., Max[#, 10^-
(BASE/10)] &/e
(10^(off/10)*tf)]];
{Flatten[{
col,
Thickness[0.004],
Map[Line[{{#[[1]], -BASE},
{#[[1]], #[[2]]}]&,
vals]}]}];
```

Infine **doAnal** effettua l'intera analisi, leggendo i tre file elaborati e mettendo tutto insieme.

```
In[2]:=
doAnal[file1_, file2_, file3_,
Name_] := (
analS[file1 <> ".128", 16384];
1128 = graf1[TF1, Red, 36, T];
```



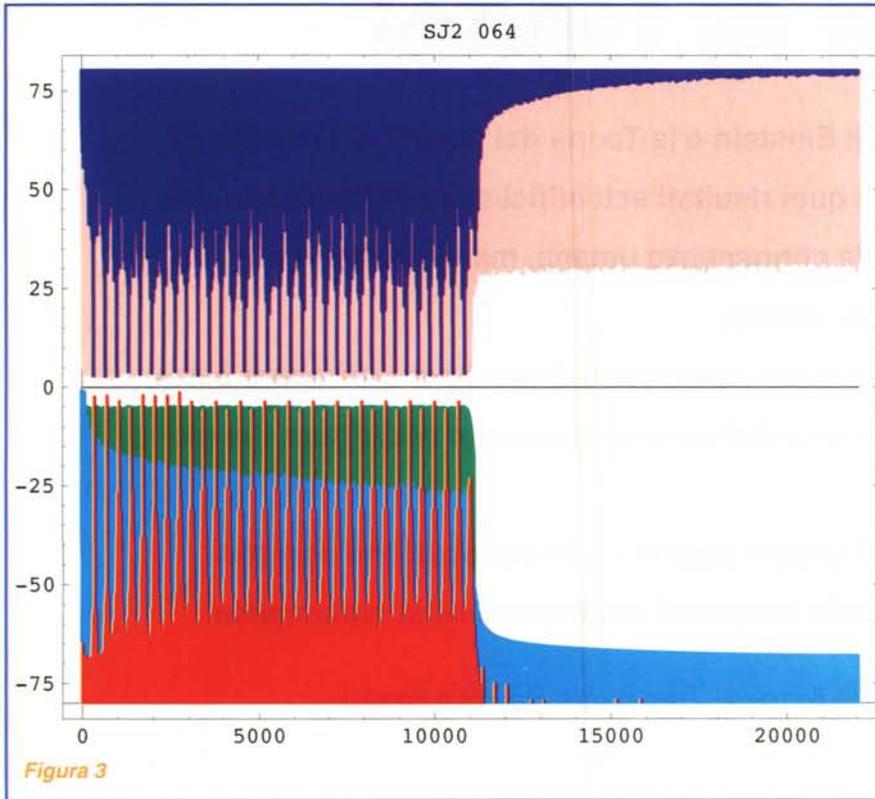


Figura 3

risultati di una elaborazione.

Un buon compressore (**N2MP3** per Macintosh, opzione **VBR**) lascia quasi intatte le frequenze con la prova a 58 toni mentre introduce una buona dose di intermodulazione in quella a 60 toni. (Figura 2). Ciò probabilmente è dovuto al fatto che l'elaborazione interna è fatta su finestre che sono potenze di due.

Un compressione più spinta (**Sound Jam** 64Kbit/s, Figura 3) fa invece un massacro! Per mantenere una buona qualità (da radio portatile) le frequenze alte (sopra i 12KHz) sono tagliate.

Questa è una scelta di progetto che va benissimo per ascoltare musica in treno con una cuffia da 5mila lire.

Infine vediamo come va il **WMA** (il sistema di compressione Windows) a 80Kbit/s (Figura 4). Qui, probabilmente, le finestre non sono potenze di due (l'algoritmo di compressione non è pubblico) e la distorsione compare in entrambi i grafici.

Il singolo impulso viene ucciso e il grafico verde non compare.. All'ascolto con un buon impianto le cose vanno meglio di come sembra a prima vista e il suono è di media qualità.

MC

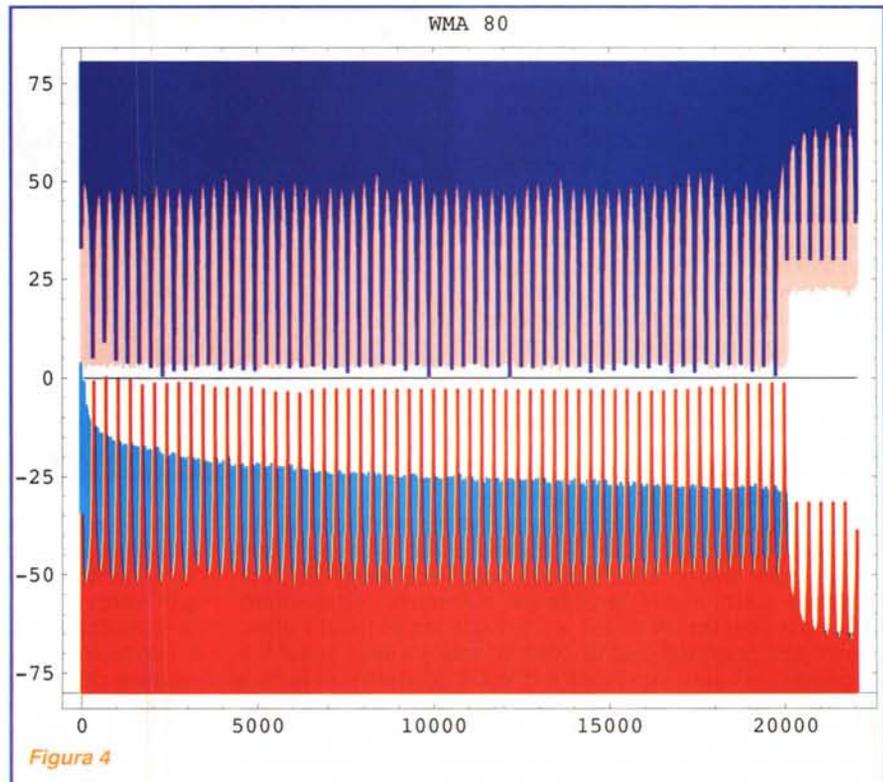


Figura 4

Bibliografia

Roberto Lucchesi, **Le prestazioni dei sistemi digitali compressi**, *Audio Review n. 198, Gennaio 2000.*

Fabrizio Montanucci, **MPEG layer 3 contro Windows Media Audio**, *Audio Review n. 198, Gennaio 2000.*

Francesco Romani, **Un test software per i compressori musicali**, *Audio Review n. 201, Aprile 2000.*