

ELABORAZIONE DEL SUONO I

Parte Seconda

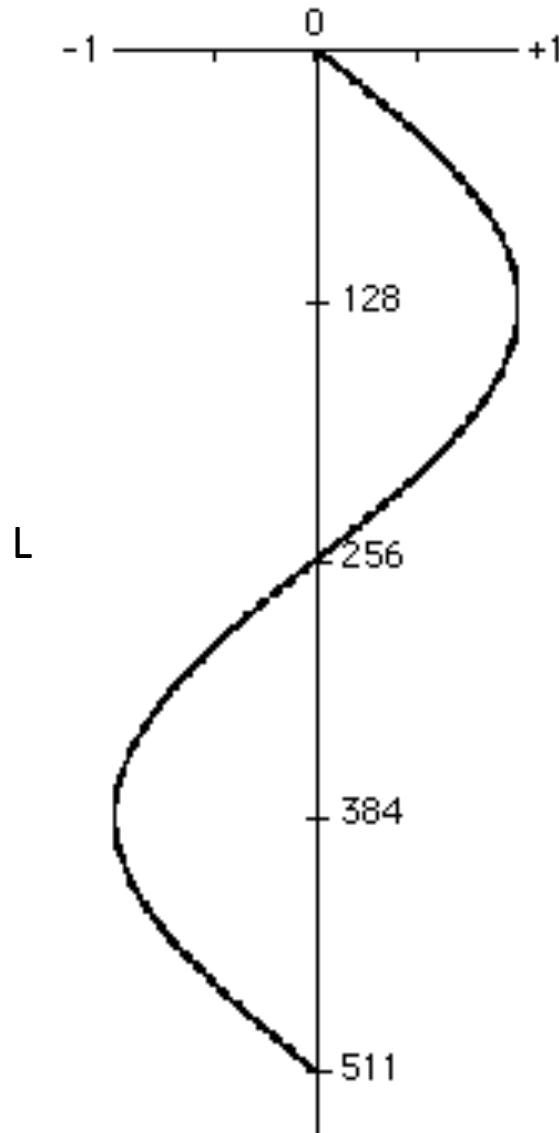
1/2

E. Giordani

L'oscillatore digitale

Indirizzo Dato o valore

0	0.0000
1	0.0123
2	0.0245
⋮	⋮
127	0.9999
128	1.0000
129	0.9999
	⋮
256	0.0000
257	- 0.0123
383	- 0.9999
384	- 1.0000
385	- 0.9999
	⋮
511	- 0.0123



Rappresentazione tabellare di un periodo completo di una forma d'onda sinusoidale composta da 512 valori.

$$F_n = sr / L$$

F_n = frequenza naturale
 sr = sampling rate o frequenza di campionamento
 L = lunghezza tabella

Frequenza dell' oscillatore digitale

Ad esempio per $sr = 44100$ e $L = 512$ si ha:

$$F_n = 44100/512 = 86,1328125 \text{ Hz}$$

La frequenza è generata attraverso un algoritmo di incremento progressivo e recursivo della fase:

$$\text{Fase} = \text{Fase} + \text{Inc}$$

dove Inc è il "passo di lettura della tabella". La frequenza generica può allora essere espressa nel modo seguente:

$$F = \text{Inc} (sr / L)$$

La fase corrente rappresenta quindi l'indirizzo a cui accedere in tabella per estrarre il campione.

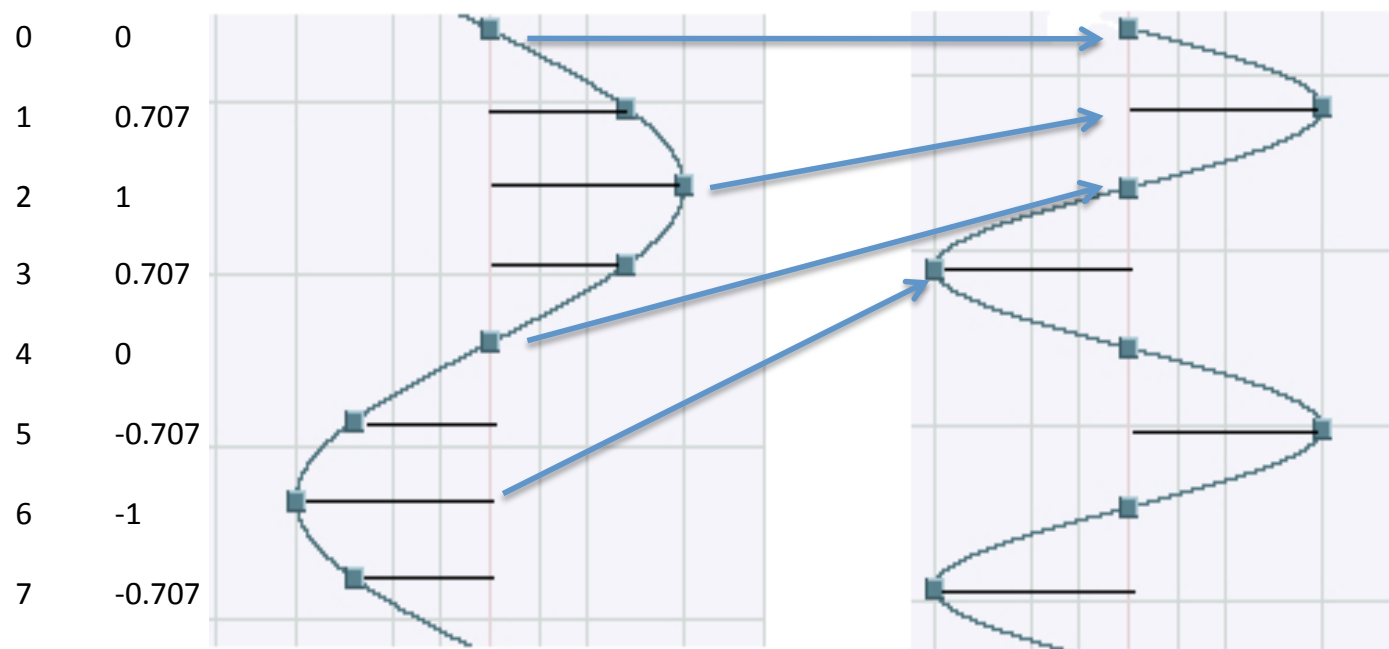
$\text{Inc} = 1$ F coincide con la frequenza naturale F_n

$\text{Inc} > 1$ $F > F_n$

$\text{Inc} < 1$ $F < F_n$

Incremento non unitario

Per una tabella a 8 valori (non è un caso realistico ma utile per l'esempio) se $\text{Inc} = 2$ si avrà una sequenza di indirizzi pari a:
0, 2, 4, 6, 0, 2, 4, 6, 0, 2,etc.



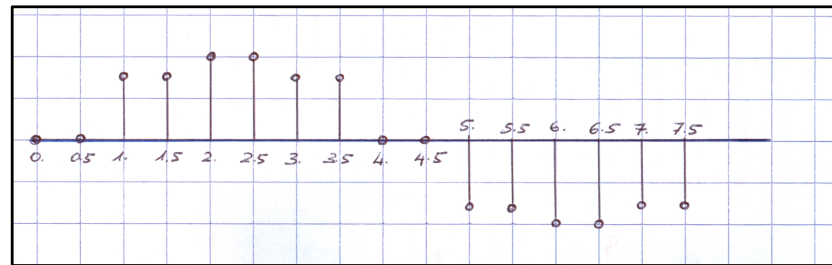
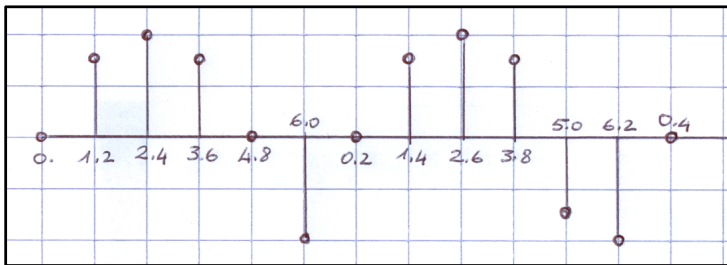
Si ottiene così una frequenza doppia e una forma d'onda virtualmente indistorta.

Incremento frazionario

Per incrementi frazionari (> 1 o < 1) si può sintetizzare qualsiasi frequenza, ma poiché l'indirizzo dei valori è un numero intero occorre fare una scelta quando la fase corrente è un numero frazionario. Ad esempio se $Inc = 1.2$ si ha la sequenza di indirizzi e valori:

0	1.2	2.4	3.6	4.8	6.0	0.2	1.4	2.6	3.8	5.0	6.2	0.4
0.	0.707	1.	0.707	0.	-1.	0.	0.707	1.	0.707	-0.707	-1.	0.

La scelta più brutale consiste nel troncare il risultato della fase e scegliere solo la parte intera. In questo modo si vede che l'aumento di frequenza consiste di fatto nel "saltare" di tanto in tanto un campione (nel primo ciclo il campione 5, nel secondo il campione 4 e così via). In tale modo l'approssimazione è troppo grossolana e ciò peggiora il S/N.



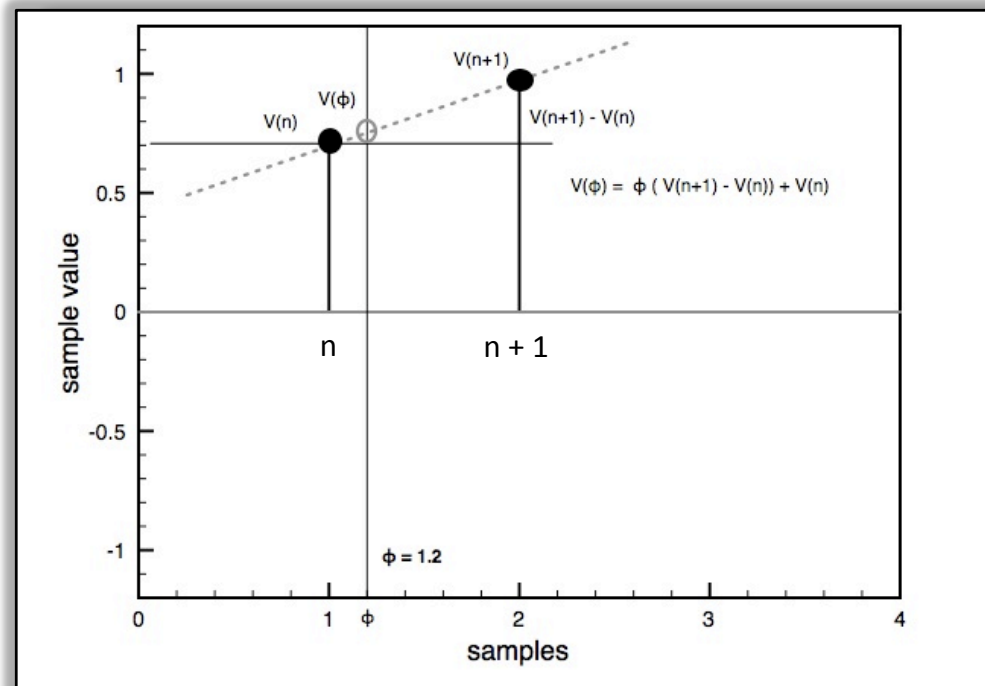
Con $Inc < 1$ (es. 0.5) si ottiene una sequenza dove ogni campione viene letto due volte consecutive con evidente distorsione della forma d'onda risultante.

0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
0.	0.	0.707	0.707	1.	1.	0.707	0.707	0.	0.	-0.707	-0.707	-1.

Interpolazione lineare

Per migliorare sensibilmente la qualità della sintesi i campioni vengono riprodotti attraverso un processo d'interpolazione che al minimo grado è di tipo lineare. Il metodo consiste nel leggere due campioni $V(n)$ e $V(n+1)$ consecutivi e ipotizzare che tali campioni appartengano ad una retta che li congiunge. Detta $\phi_f = \phi - \phi_n$ la parte frazionaria della fase corrente, il campione interpolato $v(\phi_f)$ si ottiene dalla relazione:

$$V(\phi) = \phi_f [V(n+1) - V(n)] + V(n)$$



Interpolazione lineare (esempio)

Esempio. Nella figura precedente la parte frazionaria della fase corrente è $\phi_f = 0.2$ (essendo $\phi = 1.2$). In questo caso il valore cade tra due campioni successivi, precisamente tra i campioni situati all'indirizzo 1 e 2 e i cui valori sono nell'ordine 0.707 e 1.

Applicando $V(\phi_f) = \phi_f [V(n+1) - V(n)] + V(n)$ si ottiene:

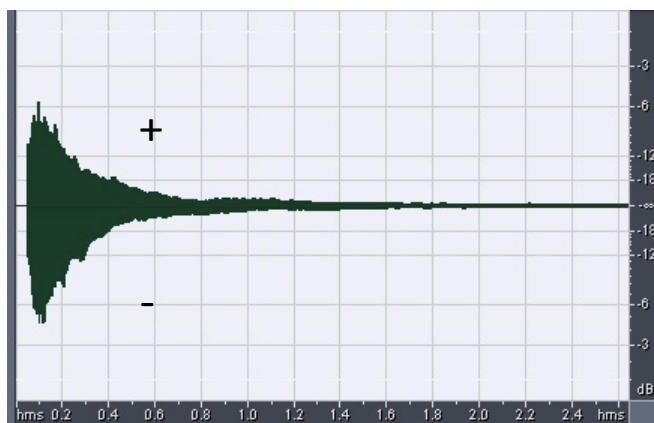
$$V(0.2) = 0.2 (1 - 0.707) + 0.707 = 0.7656$$

Come si vede il valore ottenuto è diverso (lievemente maggiore) da 0.707 e da 1 come è verosimile che sia.

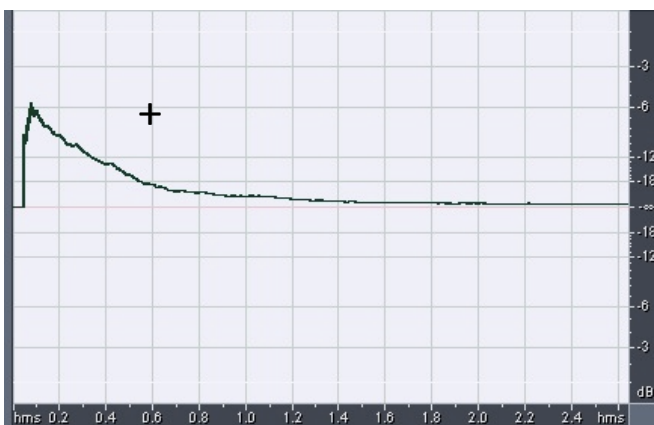
NOTA: Risultati ancora migliori si ottengono con interpolazioni di ordine superiore (quadratico e cubico). In questo caso il tempo di calcolo del campione aumenta per effetto del maggior numero di accessi in memoria campioni e per la complessità dell'algoritmo.

Profilo dinamico e inviluppo

I suoni naturali, ed in particolare i suoni prodotti dagli strumenti musicali, presentano sempre un profilo dinamico (energetico e quindi di ampiezza) variabile nel tempo. I due grafici sotto rappresentano rispettivamente la forma d'onda di una nota di pianoforte (La4 440 Hz) e il suo profilo dinamico (o inviluppo) estratto con procedura matematica



Nota di pianoforte
La4 (440 Hz)

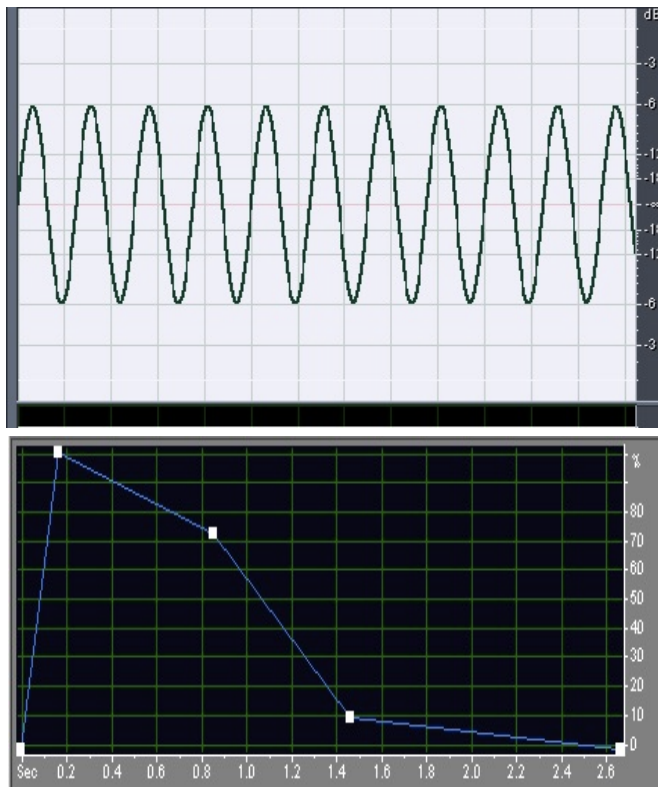


Inviluppo

Come si vede, il segnale che rappresenta la nota presenta variazioni di ampiezza sia positive che negative (bipolari) mentre il suo **inviluppo** è costituito da valori solo positivi (unipolari)

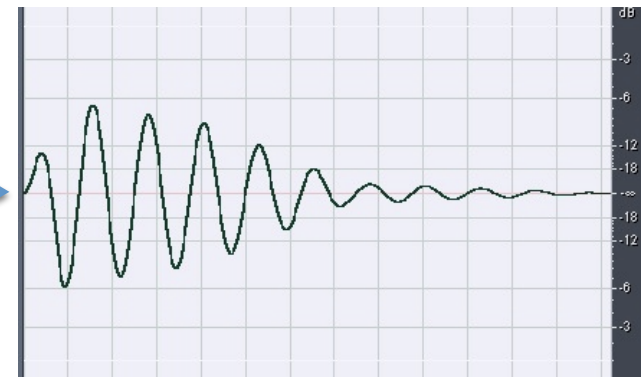
Sintesi e Inviluppo

Nella sintesi del suono, il profilo dinamico viene imposto successivamente alla generazione inarticolata del suono. Da un punti di vista operativo questa operazione consiste nella moltiplicazione algebrica tra un segnale audio e un segnale di controllo.



Suono sinusoidale
stazionario
inarticolato

Inviluppo

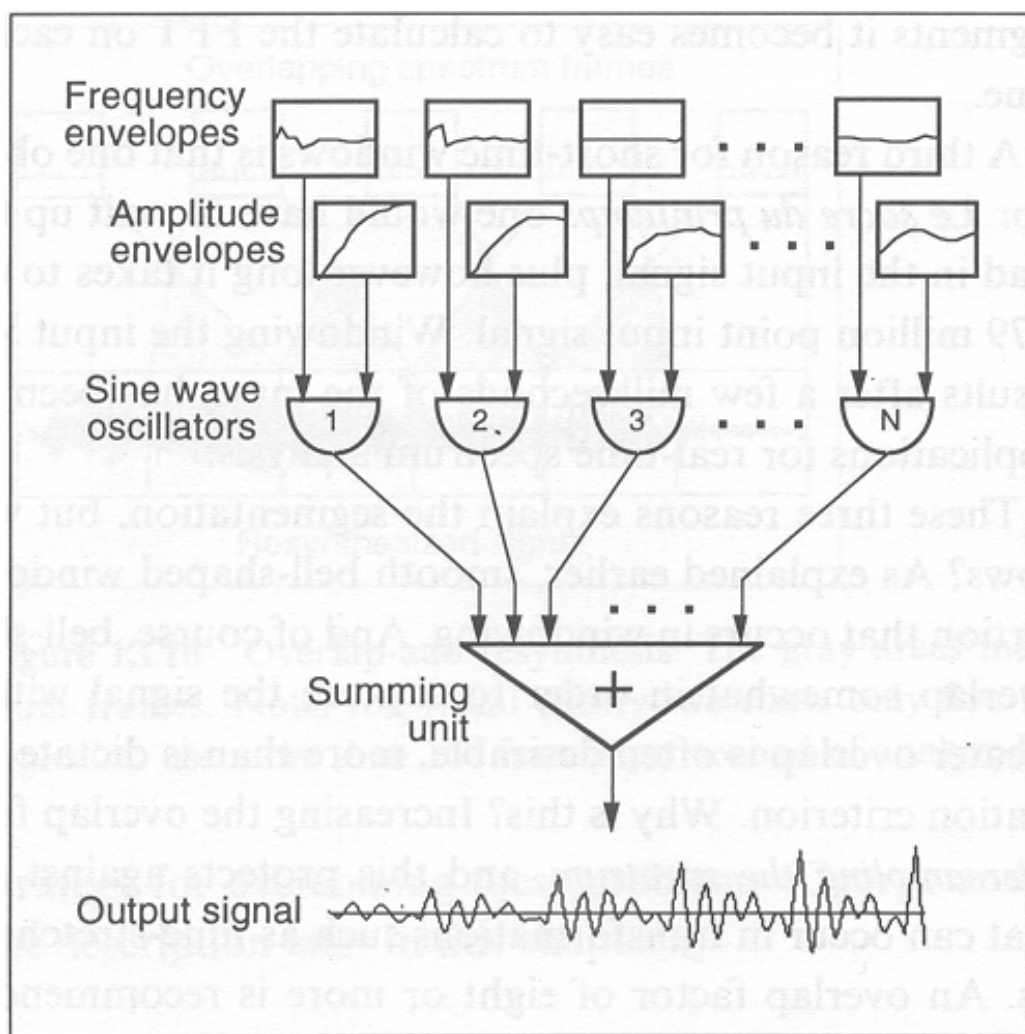


Segnale risultante dal
prodotto algebrico

Tecniche di Sintesi Audio

- Le tecniche di sintesi audio si riferiscono al processo di generazione digitale del suono attraverso algoritmi la cui gestione dipenda da un insieme di variabili di controllo che possono essere gestite individualmente, strutturalmente, manualmente o automaticamente.
- Ogni tipologia di sintesi si basa su un modello matematico che ne definisce il funzionamento e la gestione.
- Le tecniche di sintesi possono suddividersi principalmente in **Sintesi Lineari** e **Non Lineari**. Nella prima le componenti elementari sono predefinite a priori mentre nella seconda il numero e la loro relazione dipende fortemente dai parametri di controllo.
- Ad esempio la sintesi additiva e sottrattiva sono sintesi lineari poiché si conosce in anticipo tutte le componenti mentre la FM (Modulazione di frequenza) e Waveshaping (Distorsione non Lineare) appartengono alla seconda categoria.
- Le varie sintesi possono avvenire in **tempo differito** o in **tempo reale**

Sintesi Additiva

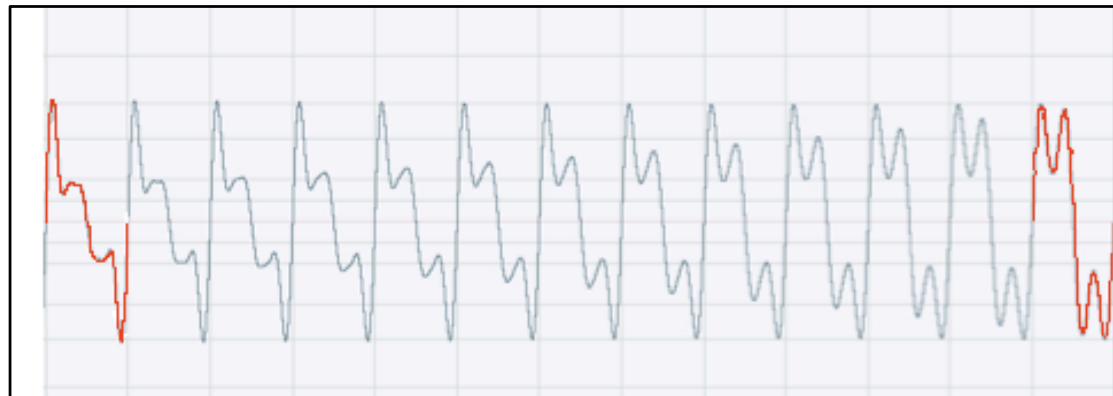


Consiste nel produrre suoni attraverso la somma di un numero rilevante di componenti sinusoidali controllate in **ampiezza** e **frequenza**.

Se i rapporti di frequenza sono di tipo armonico la sintesi produrrà forme d'onda stazionarie e con pitch identificabile.

Sintesi a forma d'onda fissa

- La sintesi a forma d'onda fissa è un tipo di sintesi molto semplificato che utilizza forme d'onda periodizzate e memorizzate in tabelle di memoria.
- Nella forma più elementare il suono prodotto è assolutamente periodico e le forme d'onda possono o essere pre-calcolate matematicamente o eventualmente estrapolate da un segnale campionato in cui si utilizza un singolo periodo.
- Un migliore risultato si può ottenere attraverso una trasformazione graduale tra diverse forme d'onda (cross-fade) a due a due (vedi figura).



Forma d'onda iniziale

X-fade

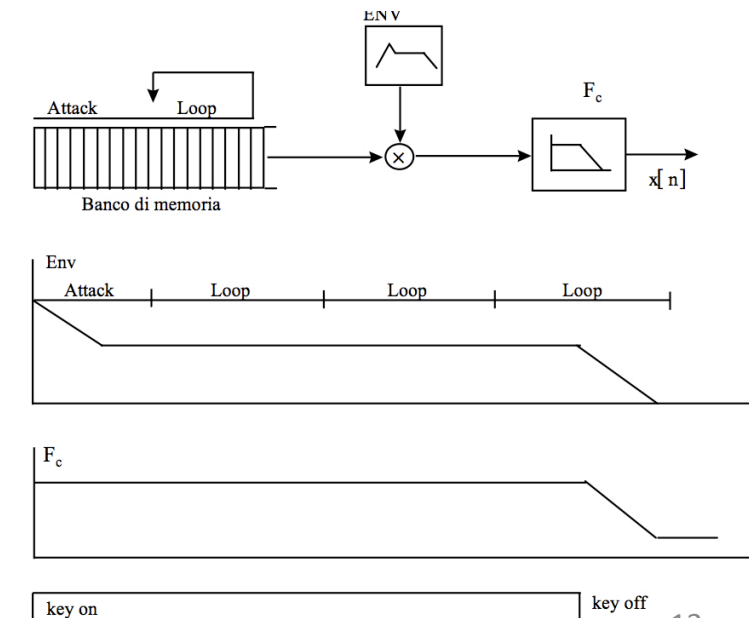
Forma d'onda finale

Sintesi per campionamento (PCM)

Le tecniche PCM fanno ricorso a tabelle di ricerca (look-up) nelle quali sono stati precedentemente memorizzati i campioni dei suoni che si vogliono riprodurre. Tali tecniche non costituiscono dei veri metodi di sintesi, in quanto non fanno ricorso ad un sistema di calcolo per la generazione del segnale. Tuttavia, i risultati che si ottengono in termini di fedeltà nella riproduzione dei suoni di strumenti tradizionali d'orchestra sono superiori a quelli ottenuti con le tecniche propriamente di sintesi.

Un problema legato alla riproduzione dei suoni PCM è dovuto al fatto che in molti casi si vuole scindere la durata di un suono presente in memoria da quella imposta dall'esecutore in fase di riproduzione. Il modo per ottenere una durata indefinita per il suono consiste nel leggere ripetutamente una parte della forma d'onda memorizzata. Il banco di memoria che contiene i relativi campioni viene diviso in tre segmenti: -

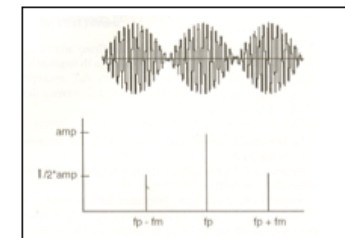
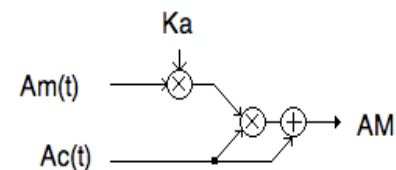
1. segmento che viene riprodotto alla fase di attack (o PCM) –
2. segmento che viene riprodotto alla fase di sustain (Loop) –
3. segmento che viene riprodotto alla fase di release. (Release PCM)



Modulazioni AM-RM

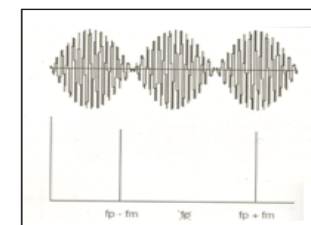
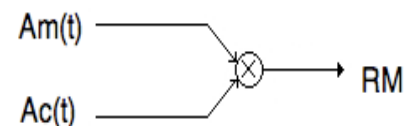
La AM (*Modulazione d'Ampiezza*) e la RM (*Ring-Modulazione o Modulazione ad Anello*, un caso particolare di AM) appartengono alla categoria delle sintesi non-lineari assieme alla FM (*Modulazione di Frequenza*) e alla WAVESHAPING (*Distorsione Non Lineare*).

Nella AM il suono si ottiene attraverso una combinazione tra somma e prodotto tra due segnali. Per semplicità supponiamo che i due segnali siano entrambi sinusoidali di frequenza rispettivamente f_m e f_c (freq. Modulante e freq. Portante).



AM

Nella RM (più utilizzata nella sintesi) il suono si ottiene attraverso un semplice prodotto tra due segnali. Per semplicità supponiamo che i due segnali siano entrambi sinusoidali di frequenza rispettivamente f_m e f_c (freq. Modulante e freq. Portante).



RM

Modulazione AM

In generale il processo di modulazione è esprimibile matematicamente nel seguente modo: supponendo $v_m(t)$ e $v_c(t)$ due segnali sinusoidali del tipo $a_m(t) = A_m \cos \omega_m t$ e $a_c(t) = A_c \cos \omega_c t$ con $\omega_c \gg \omega_m$ si ha:

$$a(t) = [(A_c + K_a A_m \cos \omega_m t) \cos \omega_c t]$$

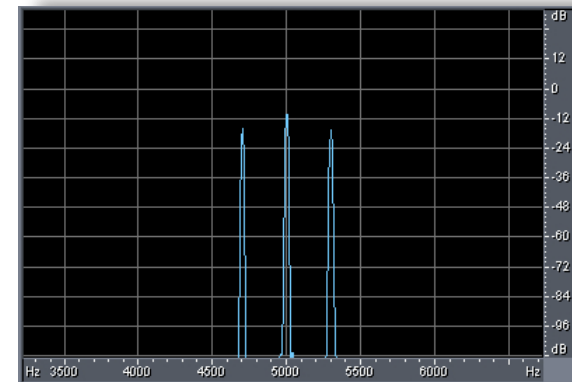
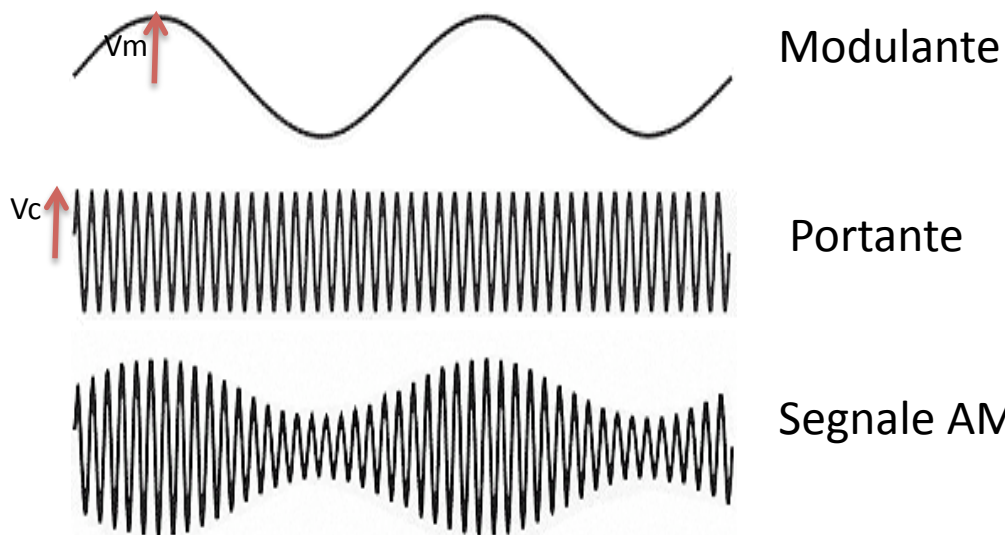
$$a(t) = A_c [(1 + K_a A_m / A_c \cos \omega_m t) \cos \omega_c t]$$

dove $v(t)$ = segnale AM

V_c = ampiezza di picco del segnale portante

V_m = " " " modulante

$K_a A_m / A_c = m$ = indice di modulazione d'ampiezza



Spettro AM –

Si noti la presenza della portante oltre alle righe laterali.

L'ampiezza della righe (bande) laterali vale $m A_c / 2$

Modulazione RM

Nella modulazione RM, supponendo sempre $v_m(t)$ e $v_c(t)$ due segnali sinusoidali del tipo

$$a_m(t) = A_m \cos \omega_m t \quad \text{e} \quad a_c(t) = A_c \cos \omega_c t \quad \text{con} \quad \omega_c \gg \omega_m \quad \text{si ha:}$$

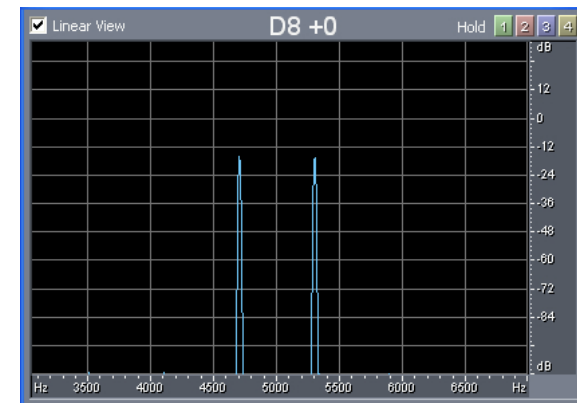
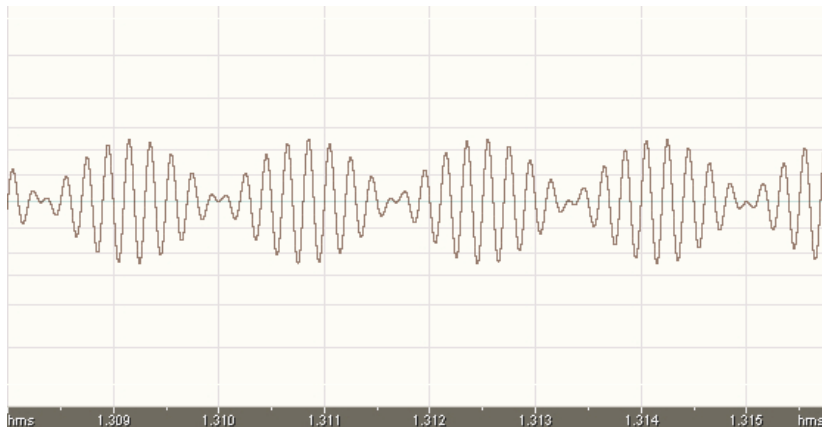
$$\text{Segnale RM} \quad \text{--->} \quad a(t) = (A_m \cos \omega_m t) (A_c \cos \omega_c t)$$

dove $v(t)$ = segnale AM

V_c = ampiezza di picco del segnale portante

V_m = " " " modulante

L'indice di modulazione è semplicemente un fattore di scala.

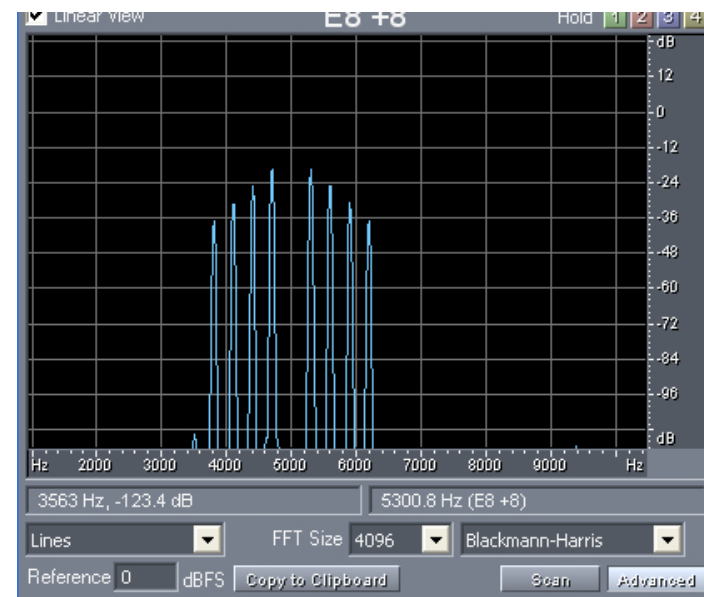
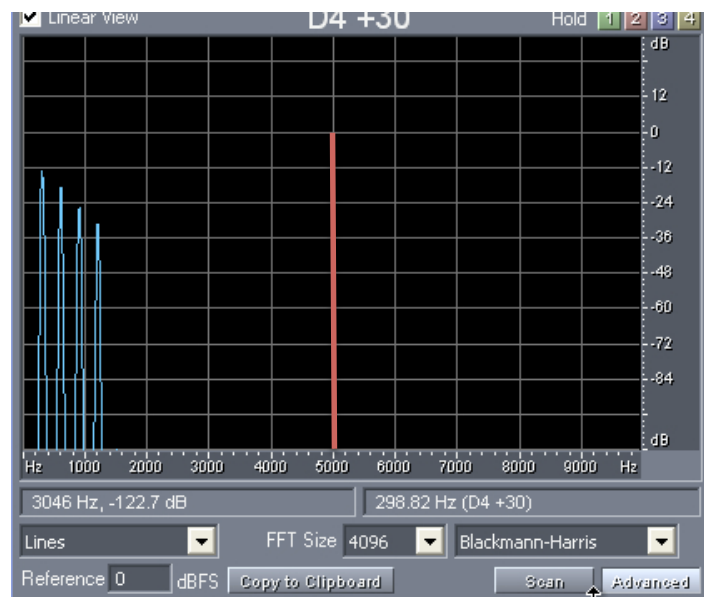


Spettro RM –
Si noti l'assenza della portante.

L'ampiezza della righe (bande) laterali
vale sempre $m A_c/2$

Modulazione RM di segnali non sinusoidali

Nel caso in cui il segnale modulante sia una forma d'onda non sinusoidale (ad esempio una somma di 4 sinusoidi come mostrato nelle figure) la relazione somma/differenza si instaura tra ogni componente della modulante con la portante.



Spettro di un segnale RM con portante sinusoidale a 5 KHz sinusoidale e modulante costituito da un segnale complesso (4 componenti armoniche) con freq. fondamentale a 300 Hz. Nel grafico di sinistra sono mostrati insieme lo spettro del modulante e della portante. Nel grafico di destra, il segnale ring-modulato. Si noti la traslazione dello spettro e la duplicazione simmetrica

Proprietà dei segnali RM

Nel caso in esame le frequenze del modulante sono:

$F_m = 300$, $2F_m = 600$, $3F_m = 900$, $4F_m = 1200$ Hz mentre la portante F_c vale 5 Khz.

Dopo la modulazione lo spettro risulta essere composto da:

Banda laterale inferiore: $F_c - F_1 = 4700$

$$F_c - 2F_1 = 4400$$

$$F_c - 3F_1 = 4100$$

$$F_c - 4F_1 = 3800$$

Banda laterale superiore: $F_c + F_1 = 5300$

$$F_c + 2F_1 = 5600$$

$$F_c + 3F_1 = 5900$$

$$F_c + 4F_1 = 6200$$

Data la particolare sequenza di frequenze risulterà uno spettro a partire da un ordine elevato di armonica. In questo caso si tratta di un segnale la cui fondamentale vale 100 Hz e in cui la prima armonica che compare è la numero 38 ($3800/100$). Il risultato acustico sarà sicuramente molto simile ad un suono totalmente inarmonico.

Frequenze riflesse

Generalizzando il caso appena visto si può dire che nella sintesi RM si generano sempre somma e differenza tra la portante e il modulante. Nel caso di entrambe le onde di tipo sinusoidale si ha una sola coppia $F_{i,s}$: (inferiore, superiore)

$$F_{i,s} = F_c \pm F_m$$

Se $F_c > F_m$ la frequenza f_i rimane > 0 .

Se $F_c < F_m$ la frequenza f_i diventa negativa e in questo caso viene riflessa simmetricamente attorno al valore 0.

Es: Se $F_c = 500$ e $F_m = 600$ si avrà:

$$F_i = 500 - 600 = -100 \rightarrow 100 \text{ Hz}$$

$$F_s = 500 + 600 = 1100$$

Quindi tutte le frequenze che diventano negative verranno riflesse intorno all'origine. Nel caso di spettro complesso del modulante le righe riflesse possono cadere sulle posizioni di quelle non riflesse andando a modificare la simmetria dello spettro risultante.

Osservazioni su AM/RM

- La modulazione AM/RM, al di là delle lievi differenze evidenziate nell'un caso o nell'altro, può essere vista come un tecnica applicabile sia a segnali sintetici o a segnali concreti.
- In generale, questo tipo di modulazione produce due diversi effetti a seconda che la velocità della modulazione sia elevata o no.
 - a) Velocità di modulazione bassa : produce effetti nell'ambito temporale (tremolo)
 - b) Velocità di modulazione più elevata (> 16 Hz) : produce effetti nell'ambito frequenziale-timbrico
- Per definizione, l'ampiezza del segnale modulato varia nel tempo

FM (Modulazione di Frequenza)

Modulazione di “angolo”

- Nella modulazione di ampiezza, l'ampiezza (istantanea) del segnale modulato è proporzionale all'ampiezza della modulante.
- Nelle modulazioni di frequenza e di fase, l'ampiezza del segnale modulato è costante; variano o la frequenza (istantanea) o la fase (istantanea) del segnale modulato. In entrambi i casi, la modulante cambia l'argomento (cioè l'angolo) della sinusoidale portante.
 - Modulazione di frequenza:
 $y(t) = \cos(2\pi f(t)t)$, con $f(t) = F(m(t))$
 - Modulazione di fase:
 $y(t) = \cos(2\pi f_c t + \varphi(t))$, con $\varphi(t) = \Phi(m(t))$

Modulazione di fase (PM) (1/2)

Il caso più semplice è la modulazione di fase (PM = phase modulation), in cui la fase è proporzionale all'ampiezza istantanea della modulante:

$$\Phi(m(t)) = k_{\varphi}m(t)$$

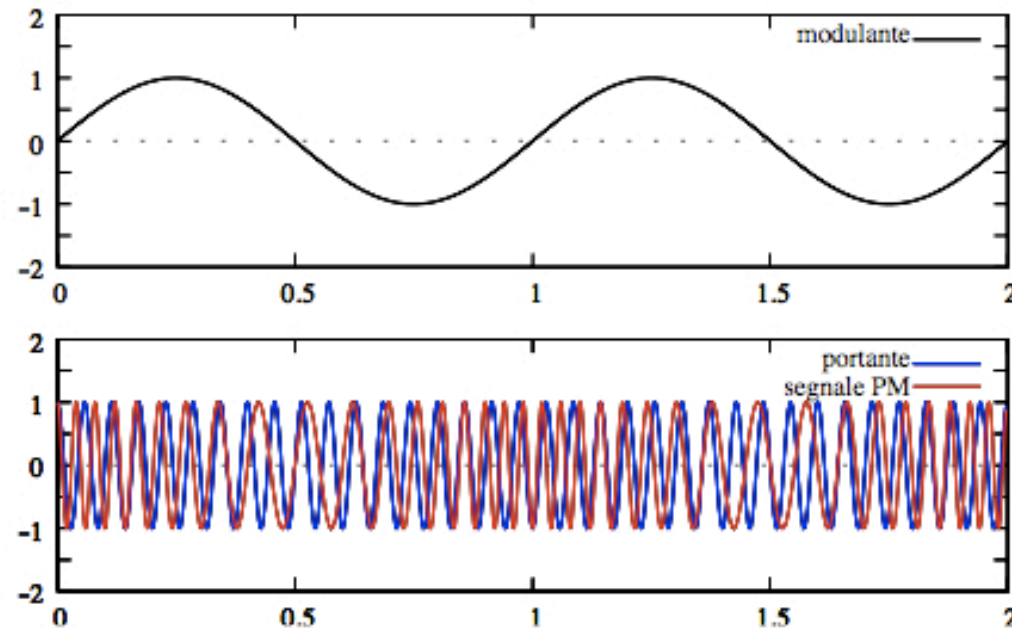
Il segnale modulato $\cos(2\pi f_c t + k_{\varphi}m(t))$ è anticipato o ritardato rispetto alla portante.

$\Phi(m(t))$ è la deviazione istantanea di fase. La deviazione massima di fase è il massimo valore assoluto che $\Phi(m(t))$ assume al variare di t .

La differenza non è solo formale poiché nell'implementazione reale dei sistemi di sintesi la FM è sempre una PM poiché permette una migliore gestione dei parametri ed in particolare consente l'automodulazione. Da un punto di vista concettuale è più facile pensare in termini di FM.

Modulazione di fase (PM) (2/2)

modulante: $m(t) = \sin 2\pi f_1 t$; portante: $p(t) = \cos 2\pi f_c t$
segnale PM: $y(t) = \cos(2\pi f_c t + k_\varphi \sin 2\pi f_1 t)$



Per $m(t) > 0$ gli attraversamenti dello zero di $y(t)$ sono in *anticipo* rispetto a $p(t)$; per $m(t) < 0$ gli attraversamenti dello zero di $y(t)$ sono in *ritardo* rispetto a $p(t)$.

La differenza non è solo formale poiché nell'implementazione reale dei sistemi di sintesi la FM è sempre una PM poiché permette una migliore gestione dei parametri ed in particolare consente l'automodulazione. Da un punto di vista concettuale è più facile pensare in termini di FM.

Frequenza istantanea del segnale PM

La frequenza istantanea del segnale PM è:

$$f(t) = \frac{1}{2\pi} \frac{d}{dt} (2\pi f_c t + k_\varphi m(t)) = f_c + \frac{k_\varphi}{2\pi} \frac{d}{dt} m(t)$$

$f(t)$ varia in modo **proporzionale alla derivata** della modulante:

- quando $m(t)$ aumenta, $f(t) > f_c$
- quando $m(t)$ diminuisce, $f(t) < f_c$

Modulazione di fase e di frequenza

$$f(t) = f_c + \frac{k_\varphi}{2\pi} \frac{d}{dt} m(t)$$

Una variazione istantanea della fase comporta anche una variazione della frequenza, che è la derivata della fase.

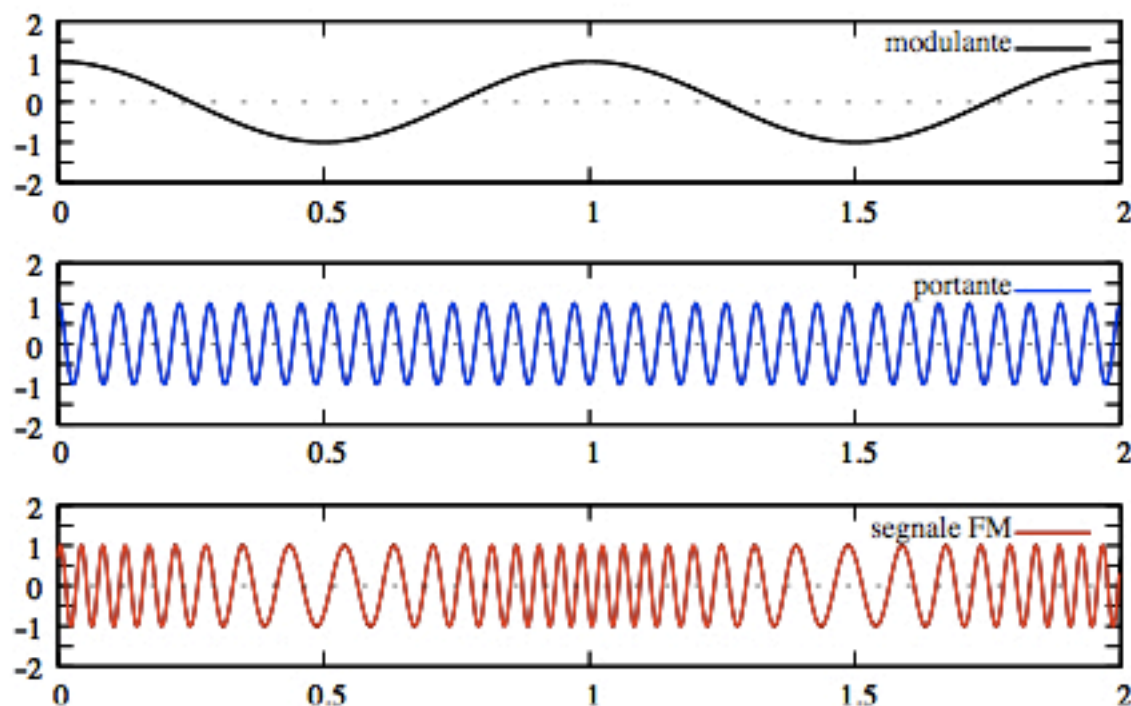
- È impossibile modulare la fase senza modulare contemporaneamente anche la frequenza, e viceversa.
- La modulazione di fase con $m(t)$ è equivalente alla modulazione di frequenza con $\frac{d}{dt}m(t)$.
- La modulazione di frequenza con $m(t)$ è equivalente alla modulazione di fase con $\int m(t) dt$.

La differenza non è solo formale poiché nell'implementazione reale dei sistemi di sintesi la FM è sempre una PM poiché permette una migliore gestione dei parametri ed in particolare consente l'automodulazione. Da un punto di vista concettuale è più facile pensare in termini di FM.

Modulazione di frequenza (FM)

modulante: $m(t) = \cos 2\pi f_1 t$; portante: $p(t) = \cos 2\pi f_c t$

segnale FM: $y(t) = \cos(2\pi f_c t + k_f \sin 2\pi f_1 t)$



Quando $m(t) > 0$ gli attraversamenti dello zero di $y(t)$ sono più frequenti; quando $m(t) < 0$ gli attraversamenti dello zero di $y(t)$ sono meno frequenti.

Indice di modulazione

Per una modulante sinusoidale $m(t) = \cos 2\pi f_1 t$, il segnale modulato $y(t) = \cos(2\pi f_c t + k_\varphi \sin 2\pi f_1 t)$ ha una frequenza istantanea

$$f(t) = f_c + \frac{1}{2\pi} \frac{d}{dt} (k_\varphi \sin 2\pi f_1 t) = f_c + k_\varphi f_1 \cos 2\pi f_1 t$$

La deviazione istantanea di frequenza Δf del segnale FM è:

$$\Delta f(t) = f(t) - f_c = k_\varphi f_1 \cos 2\pi f_1 t$$

La deviazione massima di frequenza Δf_{\max} è:

$$\Delta f_{\max} = k_\varphi f_1$$

e k_φ è detto **indice di modulazione**.

Spettro del segnale FM

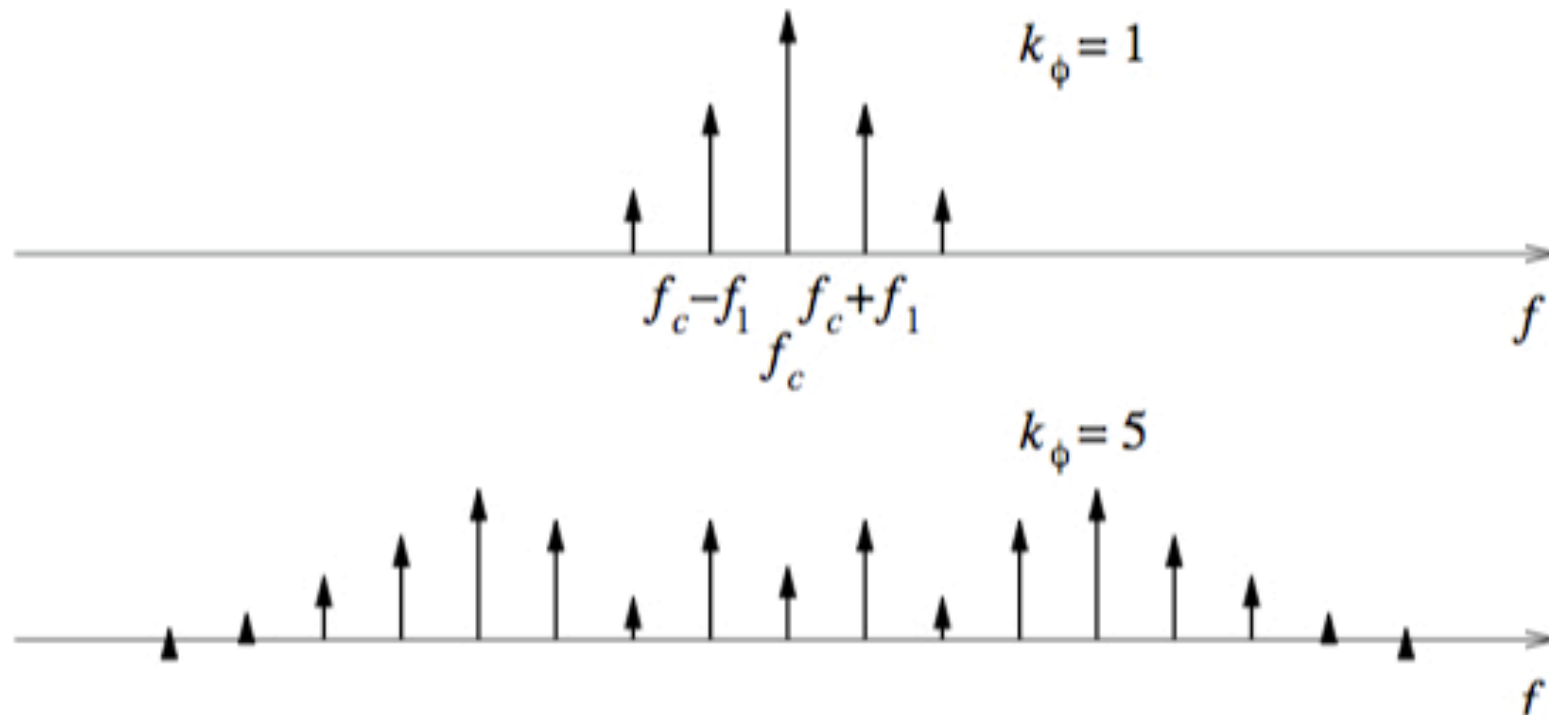
Lo spettro del segnale FM non può essere evidentemente calcolato in maniera elementare come nella AM/RM poiché in questo caso l'argomento di una funzione trigonometrica è a sua volta una funzione trigonometrica. Si deve ricorrere all'impiego di un artificio matematico che consiste nell'utilizzare uno sviluppo in serie utilizzando le funzioni di Bessel del primo tipo (J).

Omettendo per brevità i passaggi intermedi, si perviene alla seguente formula finale tenendo conto che f_1 = frequenza modulante:

$$\begin{aligned} y(t) = & J_0(k_\varphi) \cos 2\pi f_c t \\ & - J_1(k_\varphi) (\cos 2\pi(f_c - f_1)t - \cos 2\pi(f_c + f_1)t) \\ & + J_2(k_\varphi) ((\cos 2\pi(f_c - 2f_1)t + \cos 2\pi(f_c + 2f_1)t) \\ & - J_3(k_\varphi) ((\cos 2\pi(f_c - 3f_1)t - \cos 2\pi(f_c + 3f_1)t) \\ & + \dots \end{aligned}$$

Spettro del segnale FM

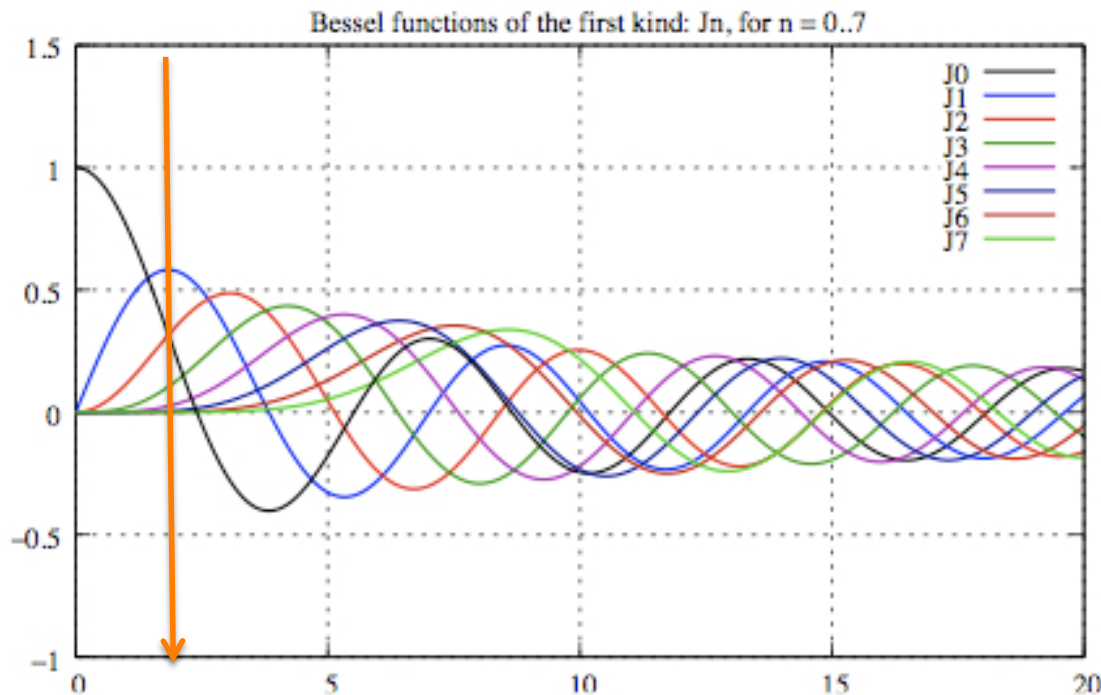
Lo spettro del segnale FM contiene tutte le frequenze $f_c \pm n f_1$; le ampiezze tendono a zero all'aumentare di n , ma al crescere dell'indice di modulazione k_ϕ tendono a zero più lentamente.



Funzioni di Bessel

Le $J_n(x)$ sono le funzioni di Bessel (del primo tipo), e costituiscono le soluzioni dell'equazione di Bessel

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2)y = 0$$



Le ampiezze delle componenti del segnale FM sono date dai valori delle funzioni di Bessel che intercettano un valore particolare dell'indice di modulazione. (Nell' esempio indicato con la freccia nel circa a 2.0)

Schema del calcolo dei segni delle bande laterali FM

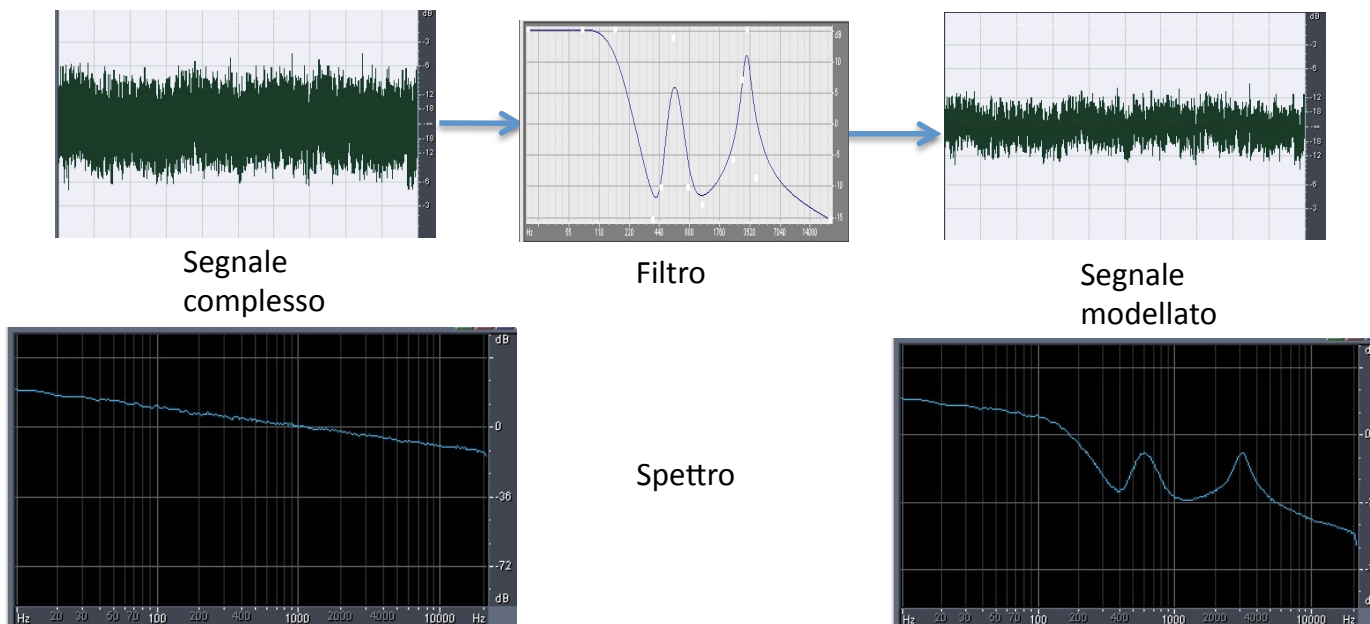
k	Inferiore		Superiore	
	Freq	Amp	Freq	Amp
1	$f_c - f_m$	$- J_1(l)$	$f_c + f_m$	$J_1(l)$
2	$f_c - 2f_m$	$J_2(l)$	$f_c + 2f_m$	$J_2(l)$
3	$f_c - 3f_m$	$- J_3(l)$	$f_c + 3f_m$	$J_3(l)$
4	$f_c - 4f_m$	$J_4(l)$	$f_c + 4f_m$	$J_4(l)$
5	$f_c - 5f_m$	$- J_5(l)$	$f_c + 5f_m$	$J_5(l)$
⋮	⋮	⋮	⋮	⋮

Sintesi Sottrattiva

La Sintesi Sottrattiva si basa sul principio che ogni suono viene generato modellando lo spettro di un segnale complesso. In altre parole, da un suono molto ricco spettralmente, per sottrazione, si arriva a un suono finale che abbia le caratteristiche timbriche desiderate. Il suono di origine può essere sia un suono reale pre-registrato o eventualmente acquisito in tempo reale, o un suono totalmente sintetico.

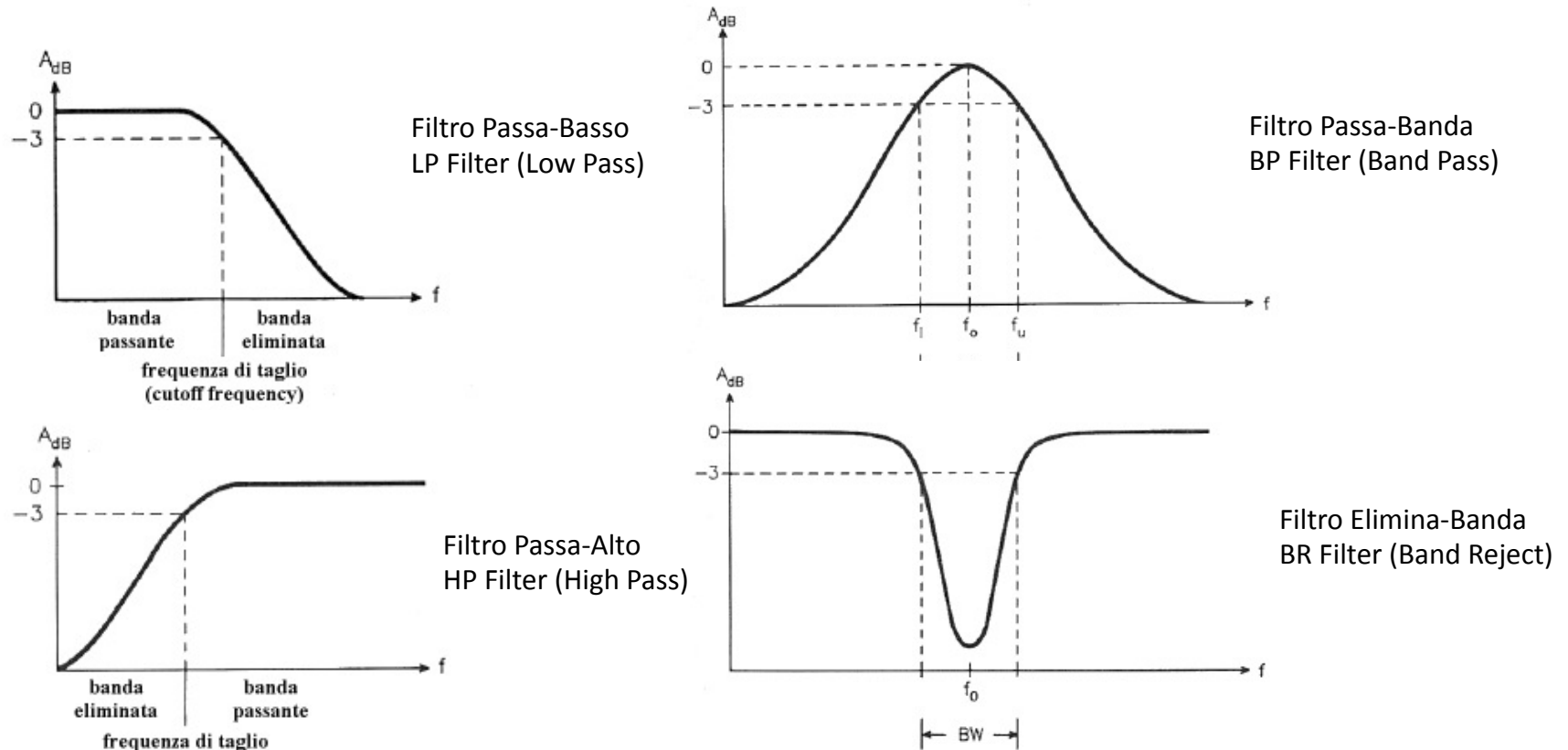
La sintesi sottrattiva si applica indistintamente ai suoni periodici e aperiodici, continui e impulsivi.

Il processo di modellazione per sottrazione, o più genericamente per attenuazione o esaltazione, si ottiene attraverso l'uso di filtri.



Tipologie base dei filtri

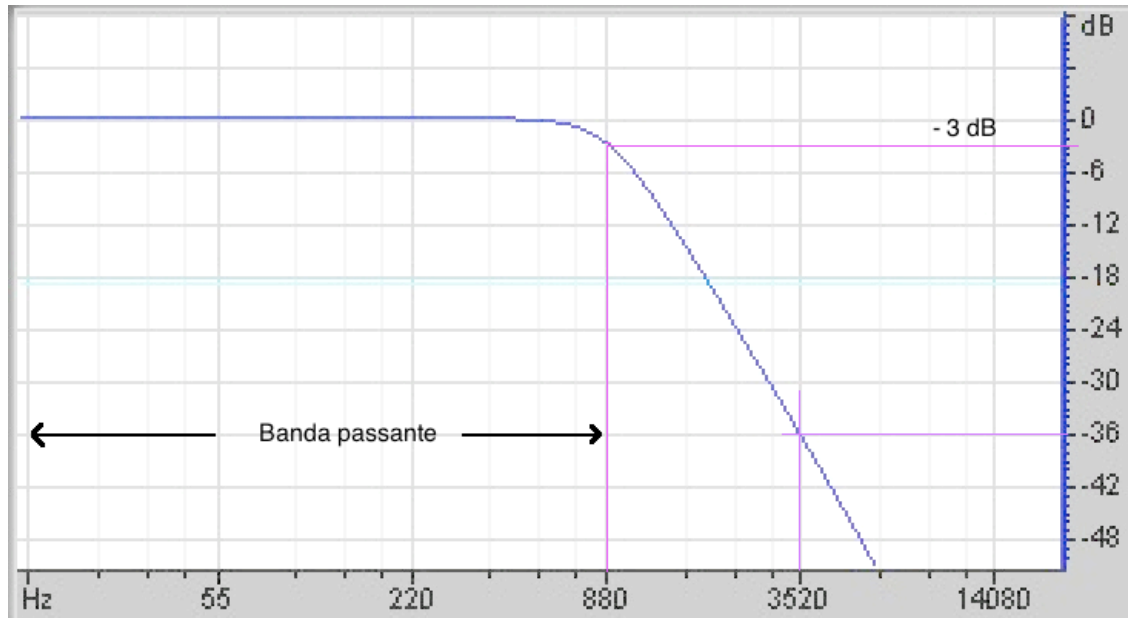
La sintesi sottrattiva digitale trae origine dall'omologa nel campo analogico dal quale ha ereditato alcune tipologie base di filtri con la possibilità di ampliare a dismisura le possibili forme di modellazione dello spettro. In ogni caso le tipologie basilari rappresentano un punto imprescindibile anche nei sistemi digitali. Tali tipologie base sono:



I diagrammi rappresentano la Risposta in Frequenza di tali filtri. In ordinata l'ampiezza in dB e in ascissa la frequenza in Hz.

Caratterizzazione dei filtri

I filtri sono caratterizzati, oltre che dalla tipologia della loro risposta, anche da altri fattori. Uno di questi è il loro grado di selettività cioè la capacità di attenuare il segnale al di fuori della banda passante

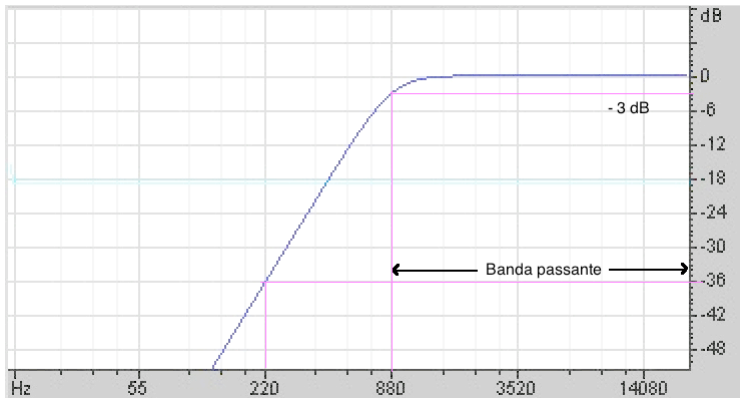


↑
Frequenza di taglio

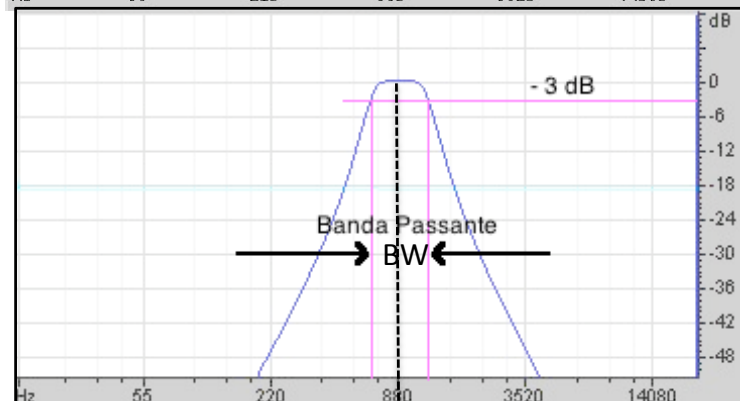
In questo esempio è mostrata la risposta in frequenza di un filtro passa-basso del terzo ordine con frequenza di taglio a 880 Hz. In corrispondenza a tale frequenza la risposta si è ridotta di -3 dB. La selettività misura la pendenza della risposta espressa come dB/ottava. Come si vede dalla figura, a 4 volte la frequenza di taglio ($880 \times 4 = 3520$) cioè 2 ottave sopra, la risposta si è ridotta a -36 dB, e quindi -18 dB per un'ottava. LA selettività quindi dipende dall'ordine del filtro:

- 1° ordine : 6 dB/oct
- 2° ordine : 12 dB/oct
- 3° ordine : 18 dB/oct
- ...etc....

Passa Alto- Passa Banda – Elimina Banda

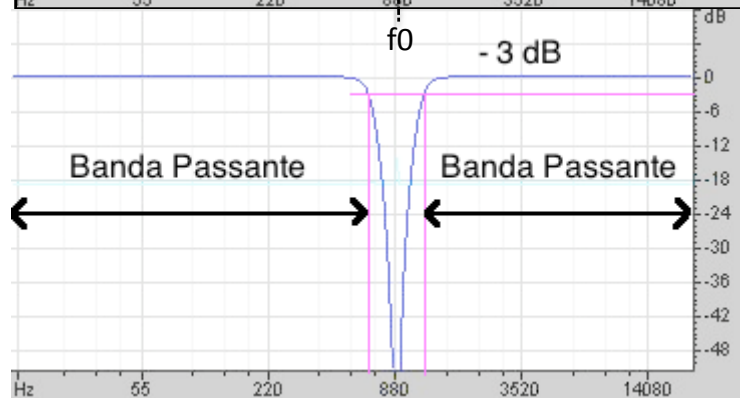


Filtro Passa Alto (HP) del 3° ordine con frequenza di taglio a 880 Hz (18 dB/ottava)



Filtro Passa Banda (BP) del 3° ordine con frequenza centrale a 880 Hz (18 dB/ottava). In questo caso la banda passante è delimitata dai due punti a – 3 dB superiormente e inferiormente.

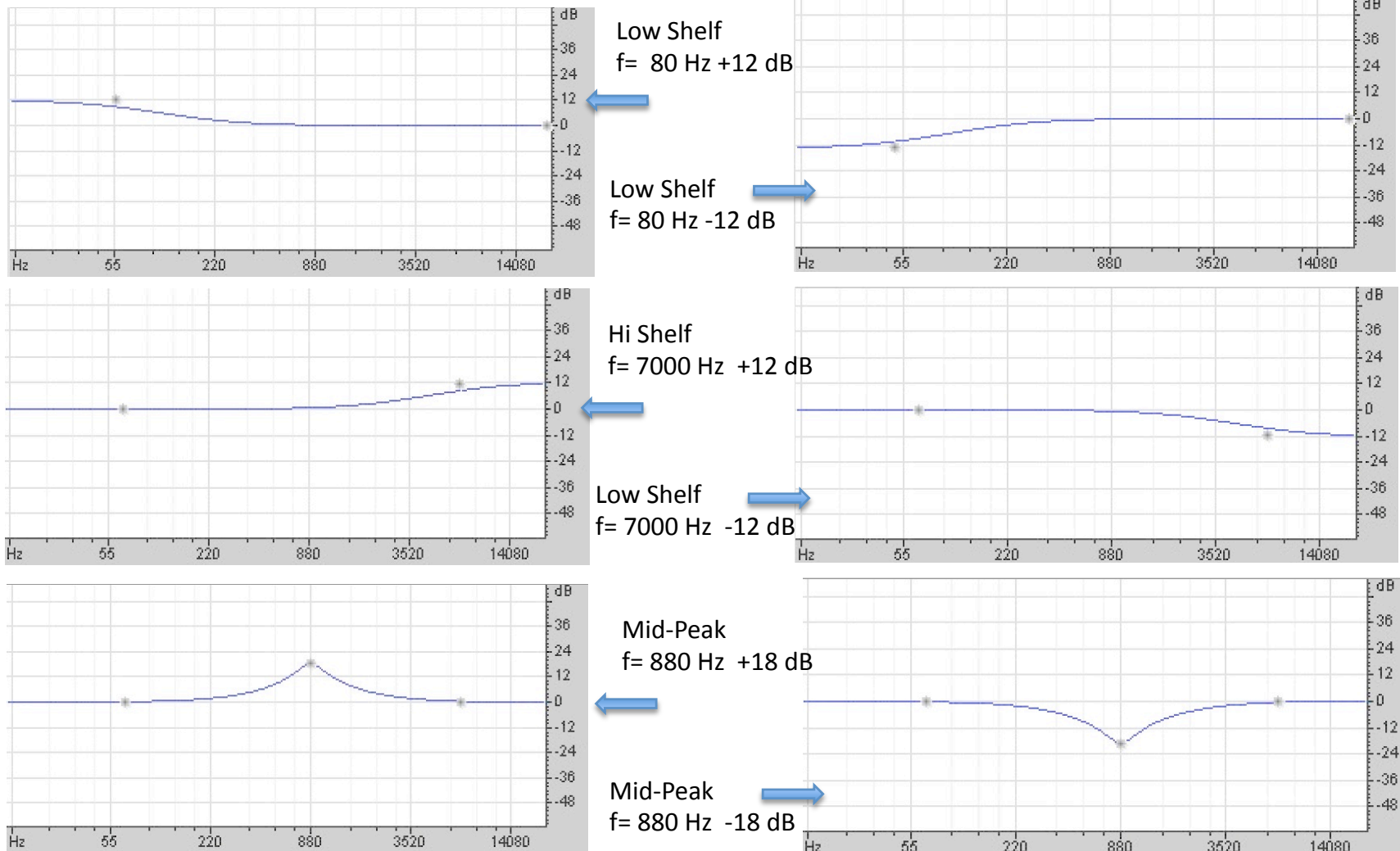
Un altro parametro caratterizzante è il fattore di merito Q dato da f_0/ BW



Filtro Elimina Banda (BR) del 3° ordine con frequenza centrale a 880 Hz (18 dB/ottava). In questo caso la banda passante è delimitata tra 0 e il primo punto a – 3dB e tra il secondo punto a – 3dB e la frequenza massima.

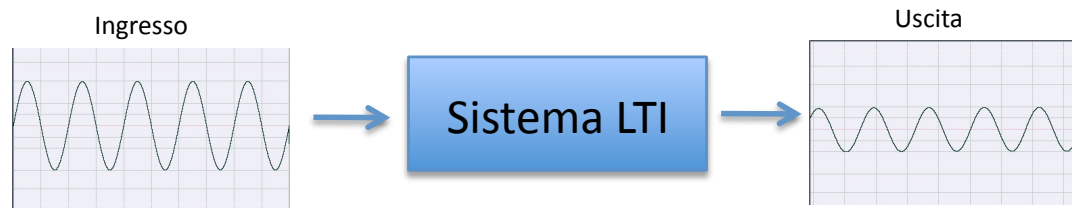
Filtri di Equalizzazione (Shelves)

Sono filtri che servono a modificare la risposta operando in zone specifiche della frequenza con azioni di *esaltazione* e di *attenuazione* ma lasciando invariato il resto della banda. Di seguito sono mostrate le tipologie base: *Low-Shelf / Hi-Shelf / Mid-Peak*. Trovano posto nei mixer analogici e digitali.



Definizione di Risposta

- Il comportamento di un filtro e in generale di un sistema che sia in grado di accettare un ingresso e produrre un'uscita è descritto interamente con la risposta nel tempo (**risposta impulsiva**) e nel dominio della frequenza (**risposta in frequenza**).
- Il termine “risposta” denota quindi la reazione d'uscita di un sistema ad un certo stimolo presentato al suo ingresso. Esiste una categoria di sistemi definiti LTI (Lineare a Tempo Invariante) per i quali tali risposte li descrivono completamente. I filtri precedentemente introdotti appartengono a tale categoria.
- Se un sistema allora è LTI, applicando al suo ingresso un segnale sinusoidale stazionario l'uscita sarà ancora un segnale sinusoidale di stessa frequenza ma di *ampiezza* e *fase* modificate.

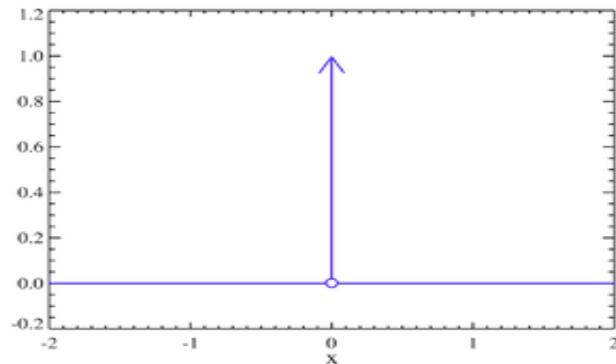


Se in tali condizioni, la frequenza del segnale d'ingresso viene fatta variare nel range di interesse (per esempio da 20 a 20 KHz nel caso dei segnali audio) per ogni valore di frequenza saremo in grado di conoscere i rispettivi valori di uscita di frequenza e fase e cioè appunto ciò che di fatto costituisce la risposta in frequenza del sistema LTI

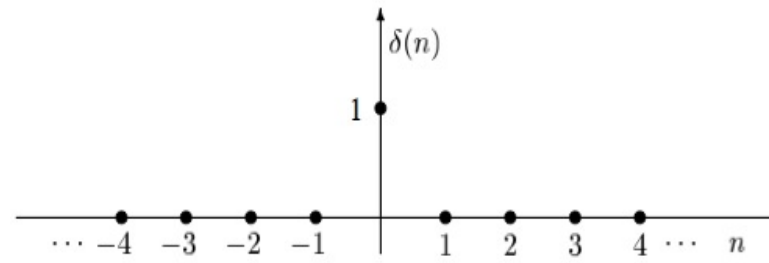
I diagrammi rappresentano la Risposta in Frequenza di tali filtri. In ordinata l'ampiezza in dB e in ascissa la frequenza in Hz.

Risposta impulsiva 1/2

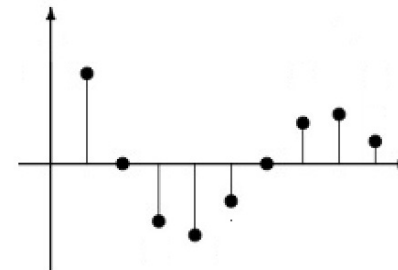
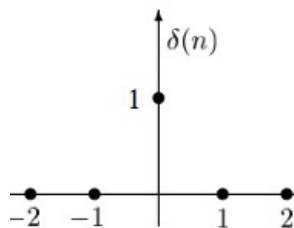
Un modo per ottenere invece la risposta temporale di un sistema è quella di stimolarlo con un segnale non periodico di tipo impulsivo. Nel caso dei segnali analogici il segnale di eccitazione è una particolare funzione detta impulso di Dirac o Delta di Dirac. La formulazione matematica di tale particolare funzione (non è realizzabile integralmente nella pratica e di fatto si accetta una certa approssimazione) prevederebbe che l'impulso sia nullo fuorchè per $t = 0$ e la sua area sottesa sia unitaria. Nel caso dei segnali discreti tale stimolo risulta di più semplice attuazione poichè rappresentato da una sequenza di campioni nulli tranne uno unitario nell'origine



Impulso di Dirac

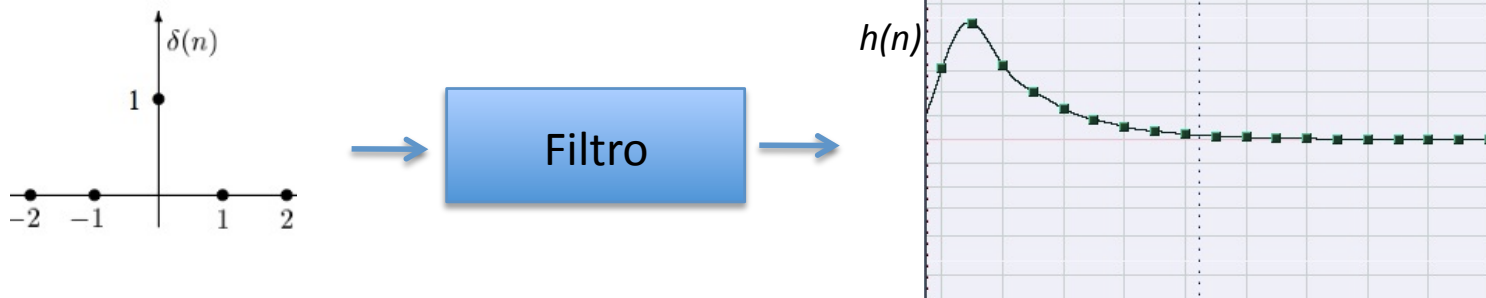


impulso discreto



Risposta Impulsiva 2/2

Sfruttando alcune proprietà matematiche dei sistemi LTI è possibile definire una fondamentale relazione che intercorre tra la risposta impulsiva e quella in frequenza. Tale relazione ha notevoli implicazioni pratiche nell'utilizzo nella modellazione del suono con le tecniche che fanno uso di filtri.



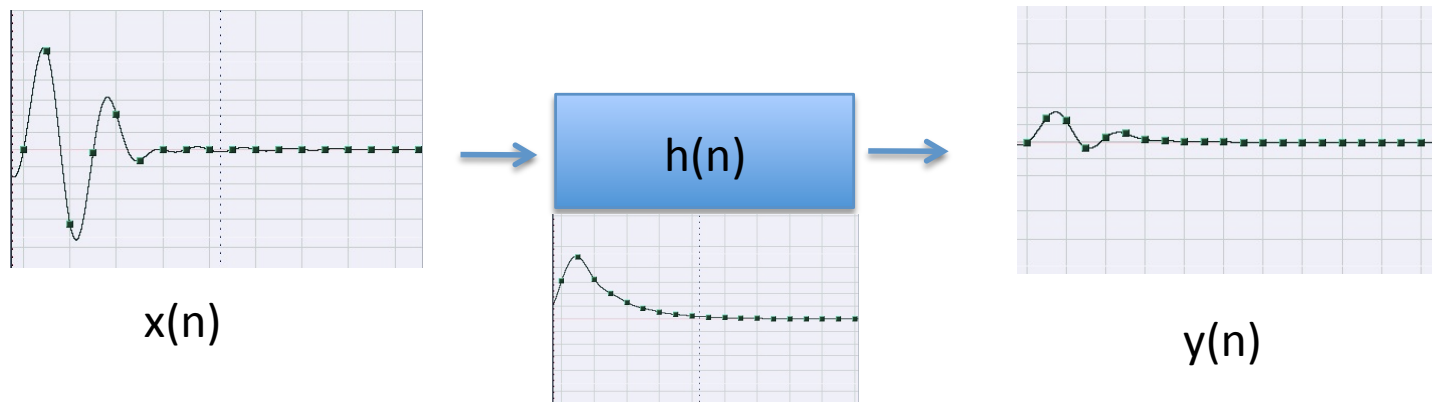
Quando un filtro viene stimolato con un impulso unitario, come già detto, la sua risposta temporale è detta risposta impulsiva. Tale risposta è una sequenza di impulsi successivi nel tempo di vario segno e ampiezza. Denominiamo tale sequenza come $h(n)$.

Se al posto dello stimolo unitario il filtro viene stimolato da un segnale, ogni impulso di tale segnale produrrà una risposta ritardata nel tempo e proporzionale alla singola ampiezza e segno. Ciascuna di queste singole risposte si sommano tra di loro formando così la risposta complessiva del filtro.

Convoluzione

Una volta che sia nota la risposta impulsiva $h(n)$ possiamo conoscere il comportamento del sistema per qualunque stimolo $x(n)$. Il processo di somma ritardata delle singole risposte descritto nella pagina precedente è formalizzabile con una operazione matematica detta *somma di convoluzione* o semplicemente **convoluzione**. Secondo questa formalizzazione quindi il segnale d'uscita si ottiene eseguendo la convoluzione tra il segnale d'ingresso e la sua risposta impulsiva:

$$y(n) = x(n) \star h(n)$$



Nota: in questo esempio la risposta $h(n)$ è relativa ad un filtro passa-basso del 1° ordine (6dB/oct) con frequenza di taglio $F_c = 3\text{Khz}$

Relazione tempo-frequenza

Il processo di convoluzione , nonostante sia concettualmente semplice da realizzare, nella pratica può diventare oneroso da un punto di vista computazionale poiché ciascun campione del segnale d'ingresso deve essere moltiplicato per tutti i campioni (ritardati) che costituiscono la risposta impulsiva e sommati tra loro. Fortunatamente, la conoscenza delle proprietà matematiche della trasformata di Fourier consentono di realizzare lo stesso processo nel dominio della frequenza. Vale allora la seguente e importante corrispondenza:

$$\begin{array}{ll} y(n) = x(n) \star h(n) & \text{(dominio del tempo)} \\ Y(f) = X(f) \cdot H(f) & \text{(dominio della frequenza)} \end{array}$$

dove $Y(f)$ e $X(f)$ sono le trasformate di Fourier rispettivamente dei segnali d'uscita, ingresso e $H(f)$ e la trasformata di Fourier della risposta impulsiva (corrispondente alla Risposta in frequenza). Ne consegue che l'operazione di convoluzione può essere sostituita dalla operazione più elementare di prodotto, a patto che si considerino le trasformate dei segnali in gioco.

Per riottenere il segnale nel dominio del tempo , una volta eseguito il prodotto delle trasformate occorre eseguire il processo di trasformata inversa di Fourier.

Origini della computer music



Max Mathews con una delle sue Invenzioni: il Radio Baton, un sistema per controllare gestualmente la performance di un brano generato da computer.

[Max Mathews](#) (1926, Columbus, Nebraska, USA - 2011, San Francisco, CA, USA)

E' considerato il padre delle computer music. Ha dato vita al primo linguaggio di programmazione per la generazione ed elaborazione digitale del suono. Ha prevalentemente svolto la sua attività presso i Bell Labs.

Inventore del sistema [GROOVE](#) (Real-time Generated Operations On Voltage-controlled Equipment) un sistema ibrido (parte di controllo digitale e parte sonora analogica) per l'impossibilità di produrre il suono digitale in real-time



Sistema Groove

Il primo linguaggio per la generazione del suono con il computer è stato Music I seguito da forme evolutive di tale linguaggio Music II ... fino a **Music V utilizzato da molti compositori di computer music negli anni 70**. Questi linguaggi sono catalogati come linguaggi Music-n e ancora oggi si utilizza un discendente di tale linea denominato **CSOUND**

Linguaggi Music-n

Lo sviluppo di tali linguaggi ha risentito, almeno nelle fasi iniziali dello sviluppo, del sistema informatico su cui venivano redatti. A titolo di esempio:

- **MUSIC I -II** IBM 704 a valvole– 32 Kbyte core mem. multiply in 250 usec Assembler - 1957 -Bell Labs
- **MUSIC III** IBM 7094 (transistori)
- .etc..
- **MUSIC V** IBM 360 - FORTRAN - 1968

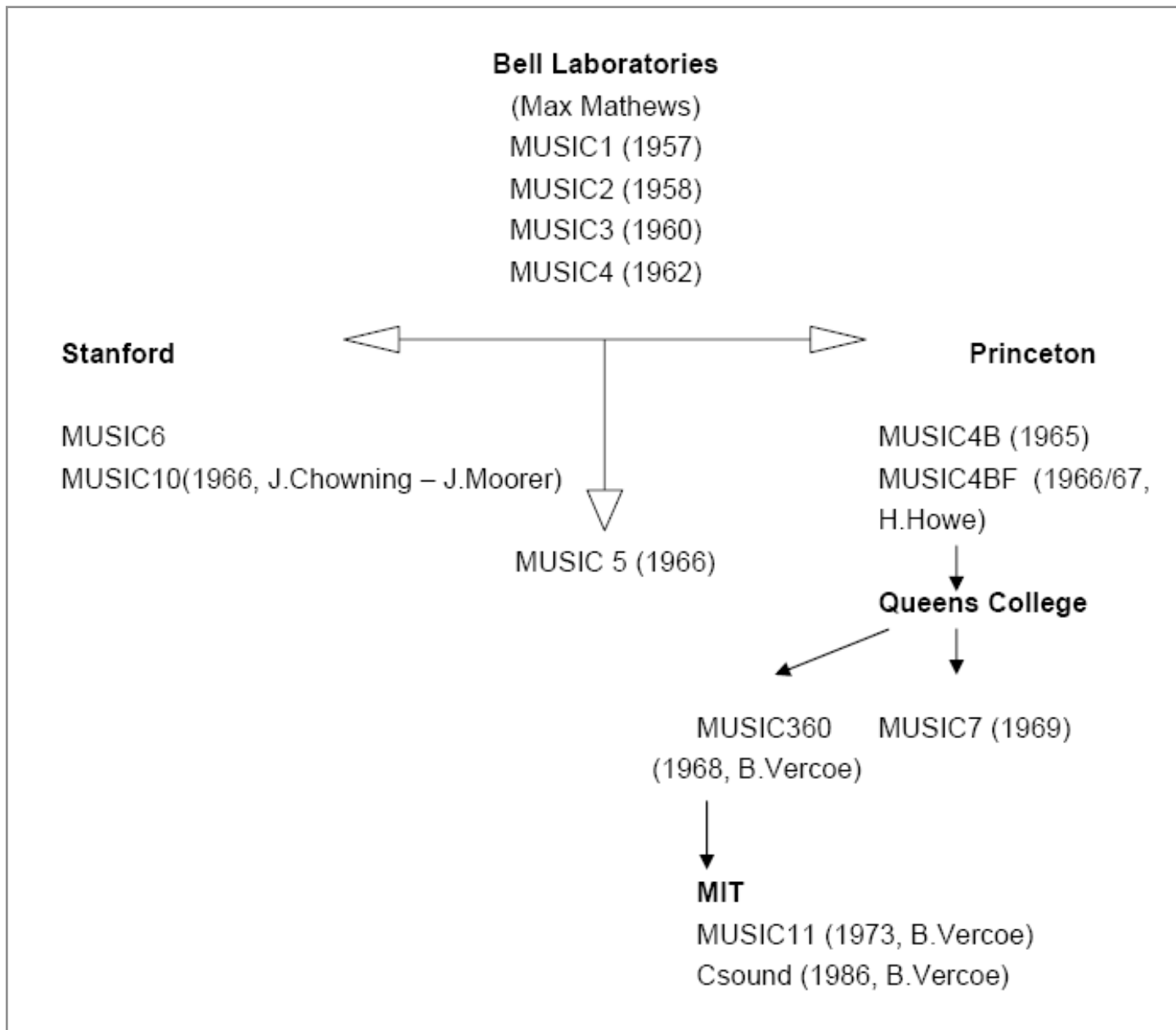


IBM 360



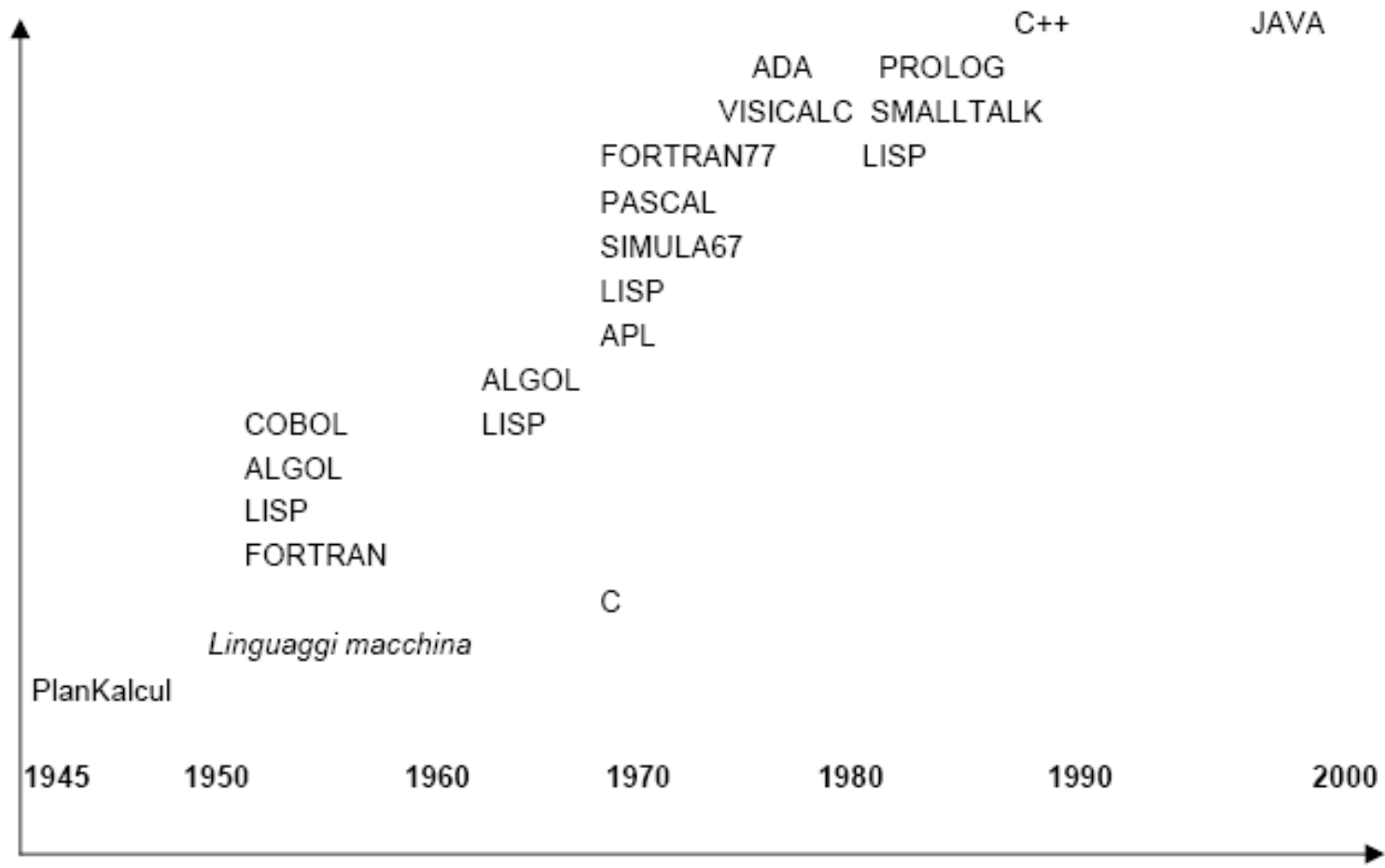
IBM 704

Albero di sviluppo dei linguaggi di sintesi audio



I linguaggi di tipo Music-n sono veri e propri linguaggi di programmazione ma a differenza di questi ultimi che sono per uso generale (*general purpose*) e che necessitano di alti tempi di apprendimento, sono molto più semplici da utilizzare e molto potenti per le applicazioni specifiche dell'audio.

Timeline dei linguaggi di programmazione general purpose



Esigenze computazionali per la generazione sonora

Secondo uno standard medio il suono digitale viene prodotto con frequenza di campionamento pari a 44100. Ciò comporta per il sistema digitale:

1 secondo di suono → 44100 numeri binari a 16 bit per canale audio (standard CD audio)

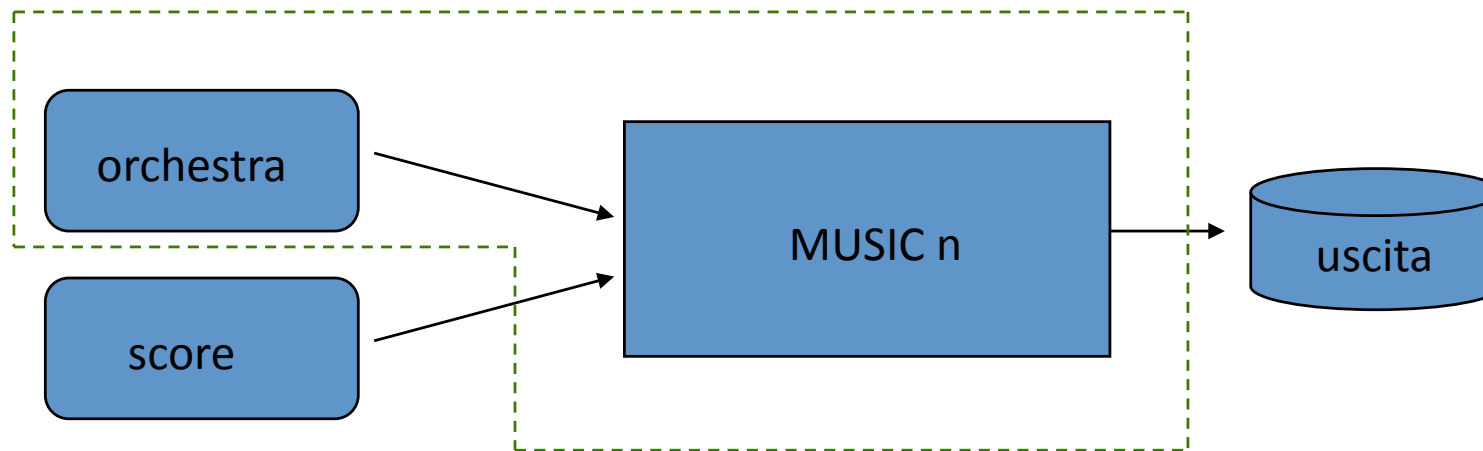
44100 è la quantità che esprime il SAMPLING RATE (frequenza di campionamento)

Generazione differita: viene eseguito il calcolo per la generazione/elaborazione (1 secondo di suono generato in un tempo $T_{gen} > 1 \text{ sec}$) e successivamente convertito da numerico/analogico e riprodotto ove T_{gen} è il tempo di calcolo effettivo.

Generazione in tempo reale: viene eseguito il calcolo e contemporaneamente convertito (1 secondo generato in un tempo $T_{gen} \leq 1 \text{ sec}$)

Struttura dei linguaggi Music-n

I linguaggi Music-n sono tutti basati su un paradigma di fondo che è dato dal dualismo delle struttura dati elaborata dal programma. In pratica si basa sulla descrizione di uno o più algoritmi di generazione/elaborazione del suono organizzati all'interno di un file descrittivo a caratteri ASCII denominato **ORCHESTRA** e su un secondo file (sempre ASCII) che definisce la temporizzazione degli strumenti e il controllo dei parametri d'esecuzione detto **SCORE** secondo lo schema indicato.



L'utente utilizza quindi un programma che è un **interprete di linguaggio** e che è in grado di decodificare le varie direttive e istruzioni specificate nei files relativi. La programmazione avviene quindi per **script** (scrivendo le varie Istruzioni all'interno di un file editabile di tipo ASCII).

Unità generative e connettività

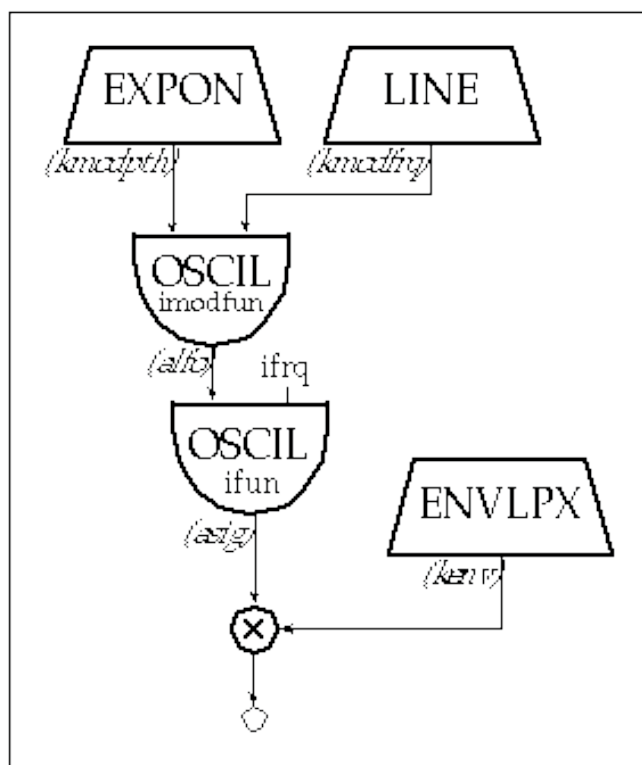
L'ORCHESTRA contiene quindi, come detto, la descrizione dell'algoritmo (degli algoritmi) di sintesi ed elaborazione del suono che consiste principalmente nella combinazione di varie unità primitive di elaborazione o unità generative (oscillatori, filtri, involuppi, operatori matematici, ecc..) attraverso un processo di connessione tra tali unità. Da un punto di vista sinottico, questa operazione può essere rappresentata in forma di schemi di flusso in cui le *ugens* (unità generative) sono raffigurate come box di varia forma da cui fuoriescono e/o entrano terminali di collegamento.

Nella figura sono visibili 3 generatori di involuppo (*expon*, *line* e *envlpx*), 2 oscillatori (*oscil*) e un operatore prodotto (*x*)

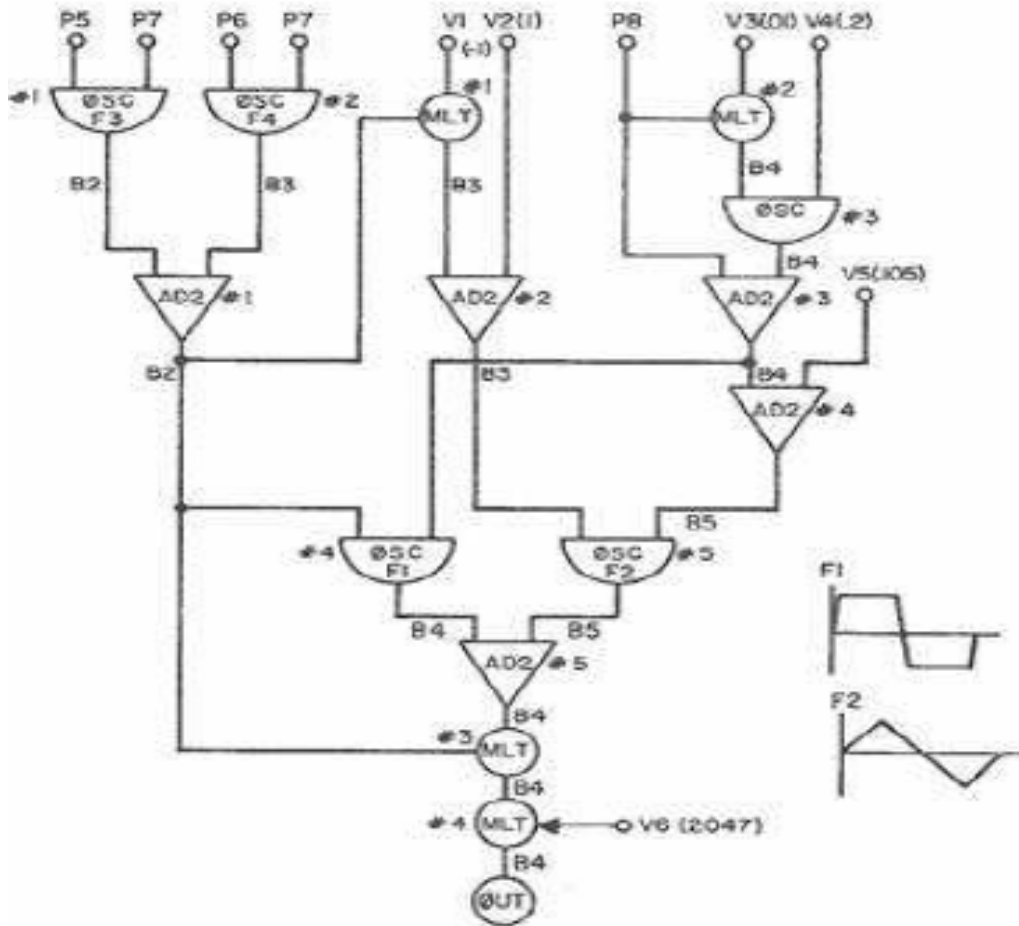
L'uscita di *expon* controlla l'ampiezza del primo oscillatore mentre *line* controlla la sua frequenza. A sua volta il primo oscillatore controlla l'ampiezza del secondo mentre la sua frequenza rimane costante (*ifrq*). Infine l'uscita del secondo oscillatore viene moltiplicata per il terzo involuppo *envlpx*.

Si noti come i collegamenti tra i moduli vengano indicati con nomi di variabili (*kmodpth*, *kmodfrq*, *alfa*, *ifrq*, *asig*, *kenv*). Tali nomi sono scelti arbitrariamente dal programmatore ma devono essere preceduti da lettere chiave (a, k o i) che ne identificano la tipologia di appartenenza:

- a** : variabile che identifica flusso audio
- k** : variabile che identifica flusso di controllo
- i** : variabile che identifica inizializzazione



Schema Music V e codifica



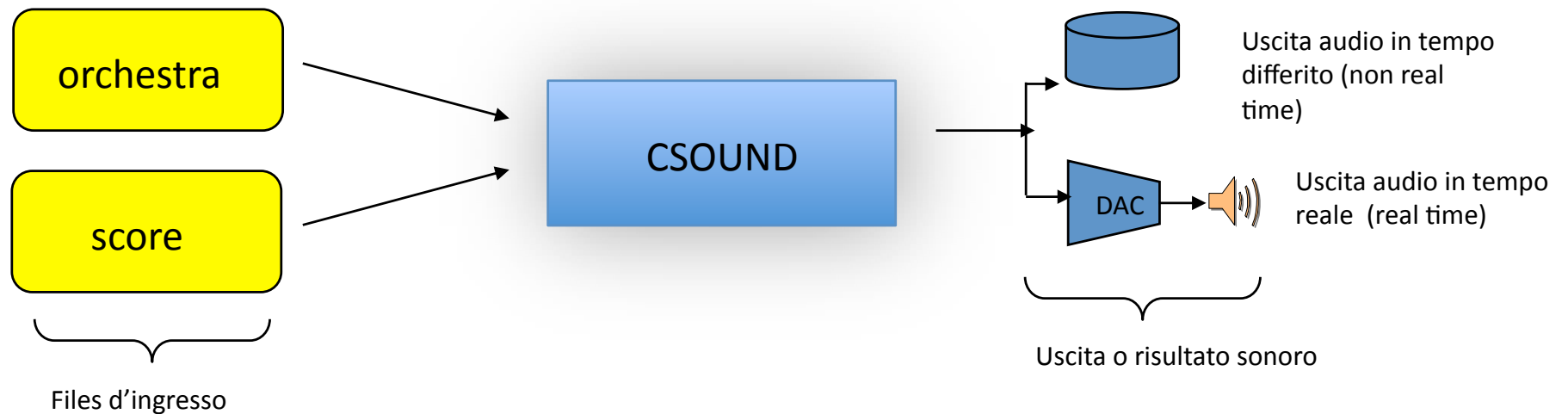
```

INS 0 4 ;
ØSC P5 P7 B2 F3 P30 ;
ØSC P6 P7 B3 F4 P29 ;
AD2 B2 B3 B2 ;
MLT B2 V1 B3 ;
AD2 B3 V2 B3 ;
MLT P8 V3 B4 ;
ØSC B4 V4 B4 F5 P28 ;
AD2 P8 B4 B4 ;
AD2 B4 V5 B5 ;
ØSC B3 B5 B5 F2 V7 ;
ØSC B2 B4 B4 F1 V8 ;
MLT B2 B4 B4 ;
MLT B4 V6 B4 ;
ØUT B4 B1 ;
END ;
    
```

Questo esempio mostra come uno schema a blocchi costituito da varie ugens viene trasformato in uno script all'interno del linguaggio Music V. Come si può vedere, la codifica era di basso livello cioè con un tipo di linguaggio **machine oriented** e quindi non semplice da utilizzare. Nelle versioni attuali di tali programmi il linguaggio si è molto semplificato (**man oriented**)

Csound

Csound proviene da uno sviluppo di Music V ed è stato concepito all'interno dei laboratori EMS (Electronic Music Studio) del MEDIA Lab MIT) da [Barry Vercoe](#) (1985)



Aspetti innovativi di Csound

- Ridefinisce la modalità di interpretazione dei dati d'ingresso
- Scritto in C, permette di usare script di programmazione C-like semplici all'uso
- Portabilità (da Unix a Dos , Windows, OS, OSX, Linux)
- Maggiore efficienza, migliorano e aumentano le UGs specializzate

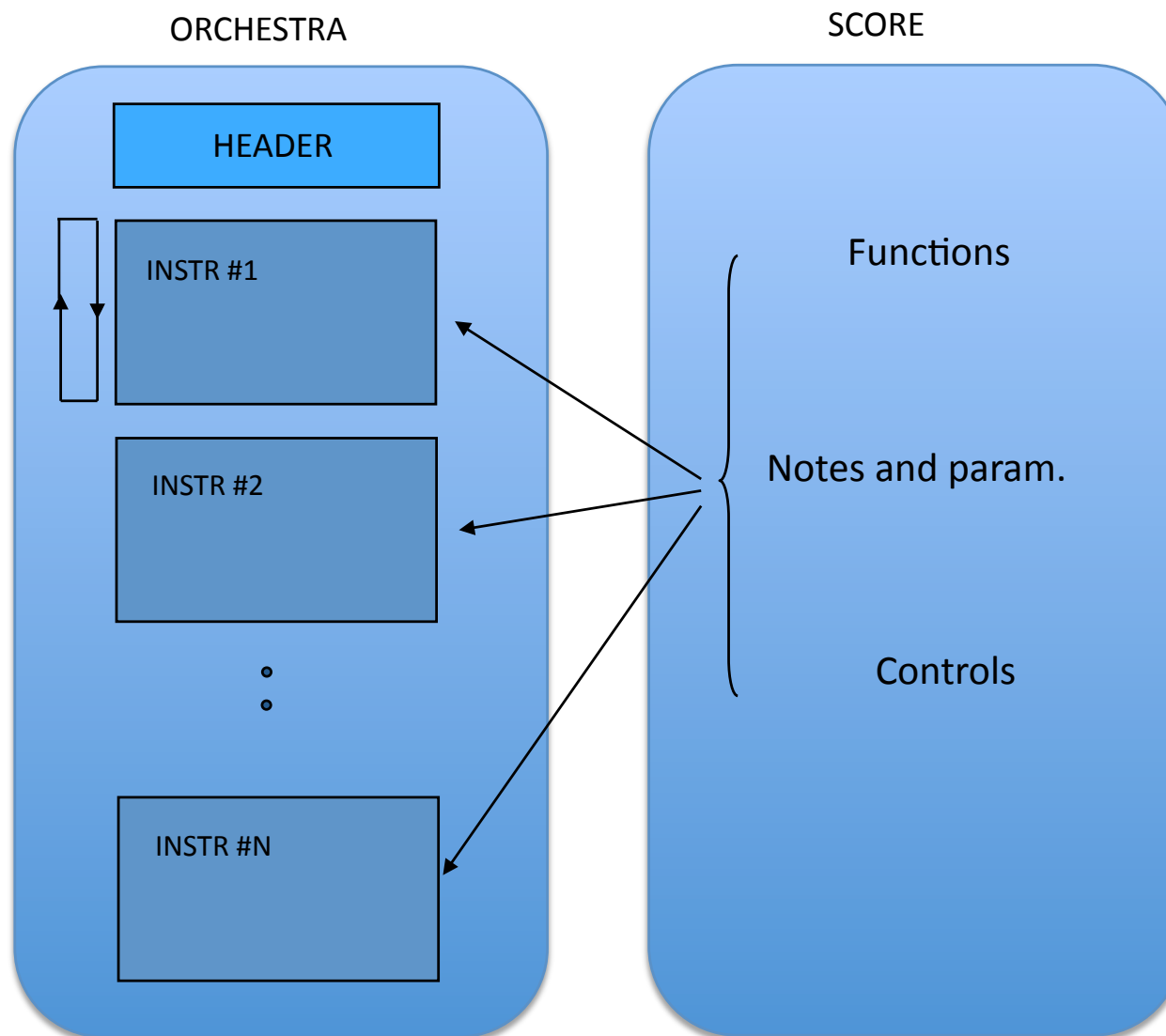
Struttura dei file orchestra e score in Csound

L'ORCHESTRA è costituita dall'insieme degli strumenti che costituiscono la generazione/elaborazione del suono.

Prima degli strumenti viene specificato un HEADER che definisce alcuni parametri globali (freq. di campionamento, numero di canali, frequenza di controllo).

Ogni strumento è numerato e al suo interno si avvia un ciclo continuo di calcolo del suono: ogni giro equivale ad un frame di campionamento.

Nello SCORE trovano luogo invece la temporizzazione dei vari strumenti (Notes and param.) più controlli e funzioni.



Aspetto del linguaggio Csound

```
; Initialize the global variables.
sr = 44100
kr = 441
ksmps = 100|
nchnls = 1

; Instrument #1 - a basic oscillator.
instr 1
  kamp = 10000
  kcps = 440
  ifn = 1

  a1 oscil kamp, kcps, ifn
  out a1
endin
; Instrument #2|.
instr 2
  ; Generate a random number between 220 and 440.
  kmin init 220
  kmax init 440
  k1 random kmin, kmax

  printks "k1 = %f\\n", 0.1, k1
endin
```

*NOTA : le parole e i simboli chiave sono mostrate
In colore azzurro. I commenti in verde e preceduti
da punto e virgola*

```
; Table #1, a sine wave.
f 1 0 16384 10 1

; Play Instrument #1 for 2 seconds.
i 1 0 2
; Play Instrument #2 for 5 seconds.
i 2 1.5 5
e
```

In questa pagina è mostrato il contenuto di un file ORCHESTRA (a sx) e un file SCORE (dx) indispensabili per fare funzionare l'interprete Csound.

Si possono individuare i due strumenti di cui è formata l'ORCHESTRA: *instr 1* e *instr 2* (si noti che la fine ogni strumento è delimitata dalla parola chiave *endin*).

L'ORCHESTRA contiene anche una intestazione (header dove sono specificati parametri globali, come frequenza di campionamento (sr) o numero di canali (nchnls)

Il file di dx (SCORE) contiene le informazioni di attivazione dell'orchestra:

i1 0 2 significa attiva strum. 1 al tempo zero e durata 2 sec
i2 1.5 5 " " strum. 2 al tempo 1.5 e durata 5 sec.

Sviluppi nelle versioni moderne di Csound

- 1984 Barry Vercoe rilascia la prima versione di Csound (primo linguaggio di sintesi audio scritto interamente in C) (MIT)
- 1991 John fficht porta Csound sotto DOS (University of Bath)

Attualmente esistono versioni per tutte le principali piattaforme (Unix, Windows, OSX, Linux)

- Rispetto a molti software commerciali ha un numero molto superiore di unità generative
- Documentazione estesa
- Facile espandibilità
- Uso dell'aritmetica in doppia precisione per il calcolo

Principali versioni di Csound

Sono disponibili molte versioni di Csound. Le più diffuse:

- CsoundAV (Gabriel Maldonado) (Windows)
- Csound Linux (Nicola Bernardini) (Linux)
- MacCsound (Ma++Ingalls) (OSX)
- Csound5 (Michael Gogins) (Tutte)
- QuteCsound (Andres Cabrera) (Tutte)

Interfaccia grafica (GUI) basata su FLTK



[SimpleKnob.csd](#)



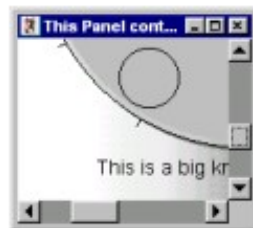
[simpleButton.csd](#)



[ShowValues.csd](#)



[simpleSlider.csd](#)



[simpleScroll.csd](#)



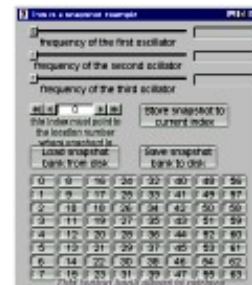
[ModifyAppearance.csd](#)



[simpleRoller.csd](#)



[simpleTabs.csd](#)



[Snapshots.csd](#)



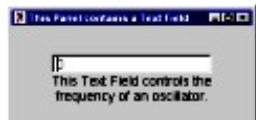
[simpleJoystick.csd](#)



[simpleButtonBank.csd](#)



[FLprintk.csd](#)



[simpleTextField.csd](#)



Viene impiegato un subset della libreria grafica Fast Light Tool Kit (open source) FLTK che mette a disposizione un elevato numero di “*widget*” (componente grafico di una interfaccia utente)

PLUS

- veloce
- supporta lo standard OpenGL

MINUS

- occorre scrivere codice per richiamare i widgets grafici

Interfaccia grafica (GUI) basata su Qt

Attraverso l'uso di una libreria grafica open-source Qt (sviluppata prevalentemente da Nokia) è possibile progettare un'interfaccia utente piazzando rapidamente attraverso il mouse una serie di widgets predefiniti. Tali oggetti possono essere facilmente collocabili, ridimensionabili e personalizzabili (varianti di forma, colore, funzione).

Una delle ultime versioni di Csound è basata sull' engine di generazione (**Csound versione 5**) e su tale interfaccia grafica. In rete è disponibile gratuitamente sotto il pacchetto unico denominato **QuteCsound**

The screenshot displays the 'FilatReverb' GUI interface, which is divided into several sections:

- ALL PASS SINGLE PARAM:** A table of parameters for six all-pass filters (AP1 to AP6) for both LEFT and RIGHT channels. The parameters are organized into three rows: 'primes', 'samples', and 'msec'. Values are color-coded: green for primes, blue for samples, and white for msec.
- DELAY and GAIN:** Vertical sliders for delay and gain for each AP filter.
- Controls and Display:** A panel on the right containing a 'Scope ch:all' display, a 'Sample' dropdown menu, a 'Source Select' button, a 'Bypass' checkbox, a 'LP Cut Freq' knob (set to 3812.000 Hz), a 'Direct On/Off' checkbox, and a 'RevAmount' knob (set to 0.890).

Blue dashed arrows point from the text below to the 'Sample' dropdown, the 'LP Cut Freq' knob, and the 'RevAmount' knob.

Parte di una interfaccia utente costruita in ambiente QuteCsound. Sono visibili "slider" verticali, "displays", "knobs", "menù" e molti altri.

Caratteri innovativi e sviluppi di Csound5 (ver. 5.xx)

- Licenziato sotto GNU Lesser General Public License (Licenza Open Source)
- Utilizzazione di librerie open source:
 1. libsndfile for soundfile input and output.
 2. PortAudio with ASIO drivers for low-latency, real-time audio input and output.
 3. FLTK for graphical widgets that can be programmed in orchestra code.
 4. PortMidi for real-time MIDI input and output.
- Sistema di bufferizzazione audio semplificato
- MIDI interop opcodes (intercambiabilità flusso dati MIDI)
- Uso di opcode (UG) di tipo “plug in”
(Python richiamabile all’interno)
(STK di Perry Cook per la modellazione fisica) ...etc...
- Csound API includibili in C/C++/Python/Java/Lisp
- Rientrate: multiple istanze girano sullo stesso processo
- **Riscrittura di un nuovo parser attraverso un *parser generator* (j. fficht)**