

MSound

T H E M A T L A B S O U N D

Università degli Studi di Salerno
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica
Anno Accademico
2002/2003

Progetto sviluppato per il corso di

Matematica Applicata (Teoria dei Suoni)
tenuto dal Prof. Vittorio Cafagna

Introduzione

Tra i vari compilatori acustici, CSound, sviluppato al M.I.T. (Massachusetts Institute of Technology), è divenuto in breve tempo uno dei più diffusi, sia perché gratuito e ampiamente supportato, sia perché semplice da utilizzare e al tempo stesso molto potente.

CSound è un software di pubblico dominio: chiunque è libero non solo di utilizzarlo, ma anche di modificarlo e di ampliarlo. Sebbene, da un punto di vista applicativo sia relativamente semplice per l'utente finale, essendo scritto in linguaggio C, per comprenderne il funzionamento intrinseco, è necessario avere una conoscenza approfondita del linguaggio stesso, nonché una discreta familiarità con i concetti matematici relativi alla teoria dei suoni.

Matlab rappresenta un'ambiente matematico potente e flessibile caratterizzato inoltre da un linguaggio di scripting e da funzioni di alto livello che, oltre a consentire un rapido sviluppo di procedure matematiche, migliora notevolmente la comprensibilità dei sorgenti rispetto al linguaggio C.

In quest'ottica è maturata l'idea di riunire la semplicità di Matlab con le prerogative di CSound, dando vita al progetto MSound. Questo vuole essere una raccolta di procedure, aperte a sviluppi successivi, mirate a mettere in pratica i concetti teorici studiati nel modo più semplice e intuitivo possibile.

Nelle pagine seguenti, illustreremo i concetti di base per la creazione di un suono in Matlab, la realizzazione di oscillatori, l'analisi di segnali acustici mediante DFT (Discrete Fourier Transform), e approfondiremo quindi l'implementazione della Modulazione di Frequenza (FM), proponendo inoltre esempi che dimostrino le potenzialità di Matlab, e quindi di MSound.

Note per l'installazione:

I file di MSound sono contenuti nella cartella 'msound' nel CD allegato. Per il corretto funzionamento del software si consiglia di ricopiare la cartella 'msound' sul proprio disco locale, dopodiché aggiornare le pathlist di Matlab.

Capitolo 1

Il suono in Matlab

Rappresentazione dei segnali

La struttura dati centrale in Matlab è l'*array numerico*, una collezione ordinata di valori numerici interi, reali o complessi, la quale può avere due o più dimensioni. I segnali acustici, siano essi monofonici o stereofonici, vengono rappresentati in Matlab mediante array.

Rappresentazione dei vettori

Matlab rappresenta i segnali campionati come vettori. I vettori sono array di dimensione $(1,n)$ o $(n,1)$, dove n , in questo contesto, fa riferimento al numero di campioni nella sequenza. Un modo di introdurre una sequenza in Matlab è di dichiarare una lista di elementi al prompt dei comandi.

Esempio:

L'istruzione

```
x = [4 3 5 -2 7];
```

crea una semplice sequenza di 5 elementi sotto forma di vettore riga.

Operazione fondamentale è quella della **trasposizione**: l'orientamento per colonna è preferibile per i segnali monofonici in quanto si estende in modo intuitivo al caso stereofonico. Nei segnali stereofonici, infatti, la prima colonna rappresenta il canale sinistro, mentre la seconda quello destro.

In Matlab è possibile trasporre un vettore semplicemente apponendo il simbolo `'` all'espressione.

Esempio:

Per trasporre il vettore riga precedentemente creato, è sufficiente l'istruzione

```
x = x'
```

la quale darà come risultato

```
x =  
    4  
    3  
    5  
   -2  
    7
```

Generazione di forme d'onda

Matlab fornisce diverse funzioni per generare forme d'onda. La maggior parte di queste richiede la dichiarazione preliminare di un vettore tempo.

Considerando una *frequenza di campionamento (sample rate)*¹ pari a 22.050 hz, è possibile creare il vettore tempo mediante la scrittura:

```
t = linspace(0, 2, 2*22050) ;
```

L'operazione crea un vettore tempo di 22.050 elementi che rappresenta un tempo t che va da 0 a 2 (secondi), partizionato in 22.050 punti.

Forme d'onda periodiche: onde sinusoidali

Dato un vettore tempo t è possibile creare una forma d'onda sinusoidale di frequenza $freq$ mediante l'istruzione:

```
waveform = sin(2 * pi * freq * t);
```

Forme d'onda periodiche: onde a dente di sega

Dato un vettore tempo t è possibile creare una forma d'onda a dente di sega di frequenza $freq$ mediante l'istruzione:

```
waveform = sawtooth(2 * pi * freq * t);
```

Forme d'onda periodiche: onde quadre

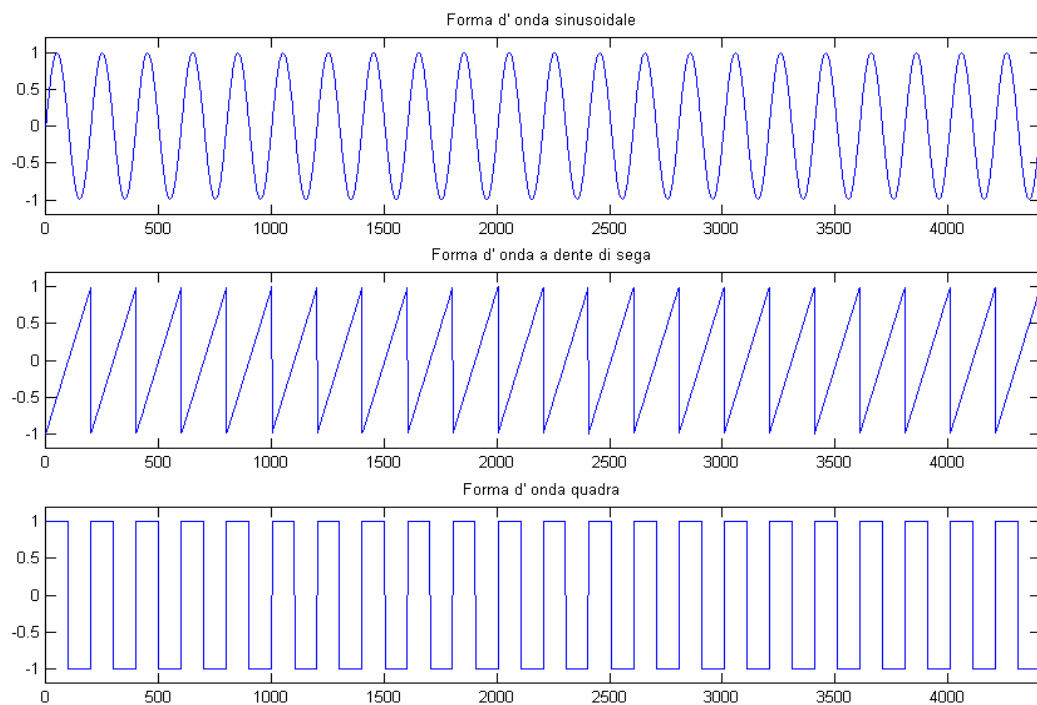
Dato un vettore tempo t è possibile creare una forma d'onda periodica quadra di frequenza $freq$ mediante l'istruzione:

```
waveform = square(2 * pi * freq * t);
```

¹ La frequenza di campionamento (sample rate) determina il numero di campioni al secondo con cui viene codificato il segnale. Frequenze di campionamento elevate producono segnali audio di elevata qualità, ma richiedono notevoli quantità di memoria. Per il teorema di Nyquist, la frequenza di campionamento deve essere pari almeno al doppio della massima frequenza contenuta nel segnale originale, quindi ad esempio, per riprodurre una frequenza di 10 KHz, sarà necessario una frequenza di campionamento pari ad almeno 20 KHz. E' possibile specificare in Matlab qualunque frequenza di campionamento, ma è bene ricordare che la maggior parte delle schede audio supporta solo alcune frequenze prestabilite, tipicamente:

8,000 Hz	Qualità telefonica
11,025 Hz	Qualità radio (AM)
16,000 Hz	Ragionevole compromesso tra 11 KHz e 22 KHz
22,050 Hz	Qualità radio (FM)
32,075 Hz	Qualità superiore a quella radio (FM)
44,100 Hz	Qualità CD
48,000 Hz	Qualità DAT

Nota: Nel seguito useremo solo forme d'onda sinusoidali.



Le forme d'onda generate dalle tre istruzioni illustrate

Funzioni MSound: vettore tempo

```
% Funzione: time  
% -----  
% Crea un vettore tempo.  
%  
% Prototipo: [vettore_tempo] = time(durata, sample_rate)  
% Input:  
% - durata: specifica la dimensione del vettore tempo in secondi  
% - sample_rate: specifica la frequenza di campionamento in Hertz  
% Output:  
% - vettore_tempo: il vettore tempo che verrà utilizzato nella creazione  
%                 dei suoni.  
  
function [timevec] = time(duration, srate);  
  
timevec = linspace(0, duration, duration * srate);
```

Funzioni MSound: generazione di forme d'onda

```
% Funzione: oscil
% -----
% Implementazione di un oscillatore sinusoidale.
%
% Prototipo: [snd] = oscil(amp, freq, time, srate)
% Input:
%   - amp: ampiezza del segnale (valori compresi nel range [-1,+1])
%   - freq: frequenza del segnale (Hz)
%   - t: vettore tempo
% Output:
%   - snd: segnale generato dall'oscillatore.
%       Usa la funzione 'sound' o 'soundsc' per ascoltare il suono
%       generato.
%
% N.B.:
% La funzione oscil può prendere come argomento sia un glissando (parametro
% freq) che un inviluppo di ampiezza (parametro amp), a patto che i vettori
% relativi al glissando e/o all'inviluppo di ampiezza abbiano la stessa
% dimensione del suono stesso.

function [snd] = oscil(amp, freq, t);

snd = amp .* sin(2 * pi * freq .* t);
```

```
% Funzione: foscil
% -----
% Implementazione di una coppia di oscillatori che realizzano una FM
% semplice.
%
% Prototipo: [snd] = foscil(amp, cps, car, mod, indx, t)
% Input:
%   - amp: ampiezza
%   - cps: frequenza nominale
%   - car: fattore per il quale si moltiplica la frequenza nominale
%         per ottenere la frequenza della portante
%   - mod: fattore per il quale si moltiplica la frequenza nominale
%         per ottenere la frequenza della modulante
%   - indx: indice di modulazione
%   - t: vettore tempo
% Output:
%   - snd: suono generato

function [snd] = foscil(amp, cps, car, mod, indx, t);

snd = amp .* sin(2 * pi * (cps * car) .* t + indx .* sin(2 * pi * (cps *
    mod) .* t));
```

Funzioni MSound: picchi e normalizzazione

```
% Funzione: findpeak  
% -----  
% Cerca il picco di ampiezza in un segnale  
%  
% Prototipo: [peak] = findpeak(snd)  
% Input:  
% - snd: suono (vettore) in cui cercare il picco di ampiezza.  
% Output:  
% - peak: picco di ampiezza.
```

```
function [peak] = findpeak(snd);  
  
peak = 0;  
for i = 1 : length(snd)  
    if (abs(snd(i)) > peak)  
        peak = abs(snd(i));  
    end  
end
```

```
% Funzione: normalize  
% -----  
% Normalizza un suono al livello di ampiezza desiderato.  
%  
% Prototipo: [normsnd] = normalize(snd, percentage)  
% Input:  
% - snd: il suono (vettore) da normalizzare  
% - percentage: livello percentuale di ampiezza da sottrarre/aggiungere al suono di input  
% Output:  
% - normsnd: suono normalizzato al livello desiderato.
```

```
function [normsnd] = normalize(snd,percentage);  
  
weight = percentage / 100;  
fprintf(' Finding peak...');  
peak = findpeak(snd);  
fprintf('%d\n',peak);  
fprintf(' Normalizing sound...\n');  
normsnd = snd * (weight / peak);
```

Funzioni MSound: disegno

```
% Funzione: drawsound
% -----
% Disegna a video il grafico del suono
%
% Prototipo: drawsound(snd)
% Input:
% - snd: suono (vettore) del quale si desidera il grafico

function drawsound(snd);

plot(snd);
axis([1 length(snd) min(snd) max(snd)]);
xlabel('Sample');
ylabel('Amplitude (absolute value)');
grid on;
zoom xon;
```

```
% Funzione: drawfft
% -----
% Disegna a video il grafico della DFT (calcolata mediante 'fft').
%
% Prototipo: drawfft(snd, srates, npoint)
% Input:
% - snd: suono (vettore) al quale applicare l'algoritmo 'fft'
% - srates: frequenza di campionamento (Hz)
% - npoint: numero di punti con cui applicare 'fft'

function drawfft(snd, srates, npoint);

if nargin == 2
    amp = abs(real(fft(snd)));
elseif nargin == 3
    amp = abs(real(fft(snd, npoint)));
end
[maxamp, index] = max(amp);
freqvec = (0:length(amp)-1) * srates / length(amp);
stem(freqvec, amp, '.');
axis([0 srates/2 min(amp) max(amp)]);
xlabel('Frequency (Hz)');
ylabel('Amplitude');
grid on;
zoom xon;
```


Funzioni Matlab per la gestione del suono

***sound* e *soundsc*: funzioni di ascolto**

Matlab fornisce due funzioni per l'ascolto dei suoni (vettori) creati. Entrambe sono compatibili con tutte le schede audio Windows e la maggior parte delle piattaforme Unix.

La funzione *sound*

Sintassi:

```
sound(y, Fs)  
sound(y)  
sound(y, Fs, bits)
```

Descrizione:

- *sound(y,Fs)* invia alla scheda audio il segnale nel vettore *y*, il quale ha frequenza di campionamento *Fs*. Si assume che i valori in *y* siano nel range $-1.0 \leq y \leq 1.0$. I valori al di fuori del range vengono tagliati.
- *sound(y)* esegue i suoni alla frequenza di campionamento di default di 8.192 Hz.
- *sound(y, Fs, bits)* esegue il suono usando un numero di bit per campione pari a *bits*. La maggior parte delle piattaforme supporta *bits* = 8 oppure *bits* = 16.

Nota: i suoni stereo vengono eseguiti solo su piattaforme che li supportano. *y* deve essere una matrice di dimensioni (n,2).

La funzione *soundsc*

Sintassi:

```
soundsc(y,Fs)  
soundsc(y)  
soundsc(y, Fs, bits)  
soundsc(y,...,slim)
```

Descrizione:

- *soundsc(y,Fs)* invia alla scheda audio il segnale nel vettore *y*, il quale ha frequenza di campionamento *Fs*. Il segnale *y* viene scalato nel range $-1.0 \leq y \leq 1.0$ prima di essere eseguito, in modo tale da evitare i tagli che la funzione *sound* apporta quando i valori in *y* sono al di fuori del range precedente.
- *soundsc(y)* esegue i suoni alla frequenza di campionamento di default di 8.192 Hz.
- *soundsc(y, Fs, bits)* esegue il suono usando *bits* bit per campione. La maggior parte delle piattaforme supporta *bits* = 8 oppure *bits* = 16.

- `soundsc(y,...,slim)`, dove `slim = [slow shigh]` mappa i valori in `y` compresi tra `slow` e `shigh` all'intero range sonoro. Il valore di default è `slim = [min(y) max(y)]`.

I soundfile: *auread*, *auwrite*, *wavread* e *wavwrite*

Matlab fornisce quattro funzioni per la lettura e la scrittura di soundfile. Le funzioni *auread* ed *auwrite* leggono e scrivono file in formato NeXT/SUN (.au), mentre le funzioni *wavread* e *wavwrite* leggono e scrivono file in formato Microsoft Wave (.wav).

Lettura: la funzione *auread*

Sintassi:

```
y = auread('aufile')
[y,Fs,bits] = auread('aufile')
[...] = auread('aufile',N)
[...] = auread('aufile',[N1,N2])
siz = auread('aufile','size')
```

Descrizione:

La funzione *auread* supporta segnali multi-canale nei seguenti formati:

- 8-bit mu-law
- 8, 16, e 32-bit linear
- floating-point
- `y = auread('aufile')` legge il soundfile specificato dalla stringa `aufile`, restituendo il segnale letto nel vettore (o nella matrice) `y`. L'estensione ".au" viene automaticamente considerata se non viene specificata. I valori dell'ampiezza sono nel range `[-1,+1]`.
- `[y,Fs,bits] = auread('aufile')` restituisce la frequenza di campionamento (`Fs`) in Hertz ed il numero di bit per campione (`bits`) usati per codificare le informazioni nel file.
- `[...] = auread('aufile',N)` restituisce solo i primi `N` campioni da ogni canale nel file.
- `[...] = auread('aufile',[N1 N2])` restituisce solo i campioni che vanno da `N1` a `N2` da ogni canale nel file.
- `siz = auread('aufile','size')` restituisce la dimensione delle informazioni audio contenute nel file, restituendo il vettore `siz = [campioni canali]`.

Lettura: la funzione *wavread*

Sintassi:

```
y = wavread('filename')
[y,Fs,bits] = wavread('filename')
[...] = wavread('filename',N)
[...] = wavread('filename',[N1 N2])
[...] = wavread('filename','size')
```

Descrizione:

wavread supporta dati multicanale, con oltre 16 bit per campione.

- $y = \text{wavread}('filename')$ legge un file wave, specificato tramite la stringa *filename*, restituendo il dato campionato in *y*. L'estensione ".wav" è aggiunta se non ne viene specificata alcuna. I valori di ampiezza sono nel range $[-1, +1]$.
- $[y, Fs, bits] = \text{wavread}('filename')$ restituisce il valore di campionamento (*Fs*) in Hertz e il numero di bit per campione (*bits*) usati per codificare i dati nel file.
- $[...] = \text{wavread}('filename', N)$ restituisce solo i primi *N* campioni da ogni canale nel file.
- $[...] = \text{wavread}('filename', [N1\ N2])$ restituisce solo i campioni che vanno da *N1* ad *N2* da ogni canale nel file.
- $siz = \text{wavread}('filename', 'size')$ restituisce la dimensione dei dati audio contenuti nel file restituendo il vettore $siz = [campioni\ canali]$.

Scrittura: la funzione auwrite**Sintassi:**

```
auwrite(y, 'aufile')  
auwrite(y, Fs, 'aufile')  
auwrite(y, Fs, N, 'aufile')  
auwrite(y, Fs, N, 'method', 'aufile')
```

Descrizione:

La funzione auwrite supporta dati multi-canale per i formati:

- 8-bit mu-law
- 8 e 16-bit linear
- $\text{auwrite}(y, 'aufile')$ crea il soundfile specificato dalla stringa *aufile*. Le informazioni devono essere strutturate con un canale per ogni colonna. I valori delle ampiezze al di fuori del range $[-1, +1]$ vengono tagliati prima della scrittura del file.
- $\text{auwrite}(y, Fs, 'aufile')$ specifica la frequenza di campionamento in Hertz.
- $\text{auwrite}(y, Fs, N, 'aufile')$ seleziona il numero di bit per la codifica. Valori ammissibili sono *N* = 8 ed *N* = 16.
- $\text{auwrite}(y, Fs, N, 'method', 'aufile')$ consente di selezionare il metodo di codifica, il quale può essere *mu* o *linear*. Si osservi che i file codificati con il metodo *mu* devono necessariamente usare una codifica a 8 bit per campione. Per default, *method* = 'mu'.

Scrittura: la funzione *wavwrite*

Sintassi:

```
wavwrite(y,'filename')  
wavwrite(y,Fs,'filename')  
wavwrite(y,Fs,N,'filename')
```

Descrizione:

La funzione *wavwrite* supporta dati audio multi-canali a 8 o 16 bit.

- *wavwrite(y,'filename')* crea il file specificato dalla stringa *filename*. Le informazioni devono essere strutturate con una colonna per ogni canale. I valori delle ampiezze fuori dal range $[-1,+1]$ vengono tagliati prima della scrittura del file.
- *wavwrite(y,Fs,'filename')* specifica la frequenza di campionamento *Fs* in Hertz.
- *wavwrite(y,Fs,N,'filename')* forza la scrittura del file con uno schema di codifica a *N* bit ($N \leq 16$).

Capitolo 2

Glissandi ed inviluppi

I glissandi

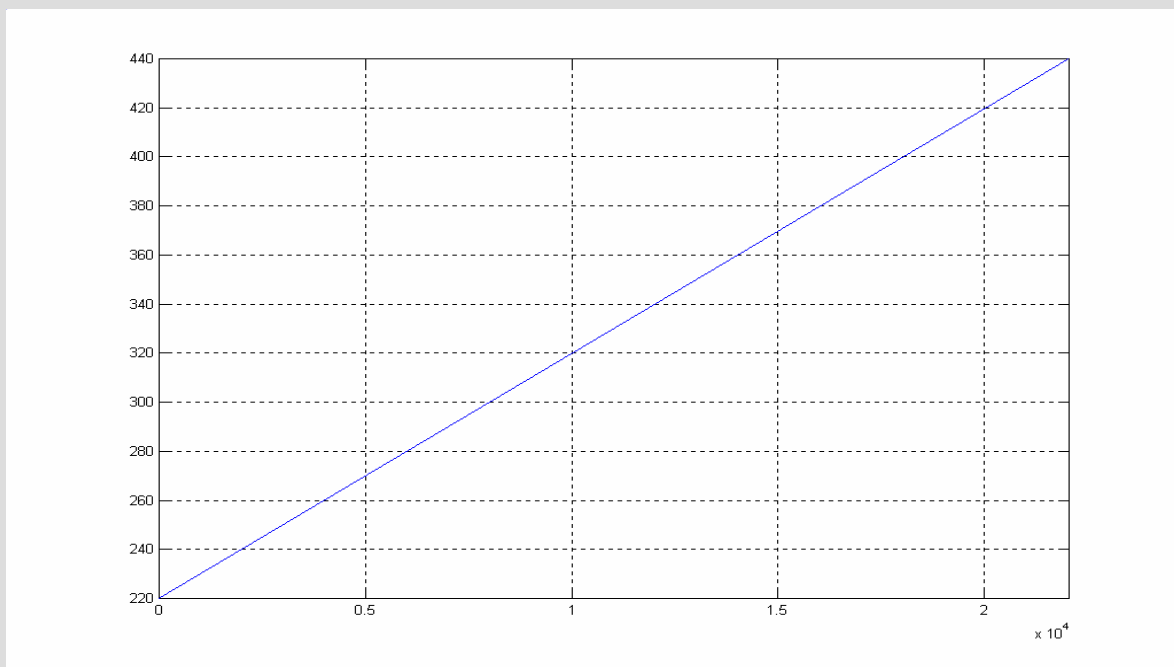
Le note create negli esempi del Capitolo 1 avevano frequenze fisse al loro interno. Per creare eventi sonori che glissano da una frequenza ad un'altra all'interno di una nota possiamo creare un segmento che conduca, in un determinato tempo, dal valore della frequenza iniziale a quello della frequenza finale desiderata e poi assegnare i valori di questo segmento al secondo argomento di *oscil*, quello della frequenza.

Creare un glissando in Matlab è un'operazione molto semplice. E' sufficiente infatti utilizzare la funzione *linspace* corredata da opportuni parametri.

Esempio:

```
gliss = linspace(220,440,1*22050);
```

Il primo parametro, 220, rappresenta la frequenza iniziale in Hertz. Il secondo parametro, 440, la frequenza finale. Il valore 1 è il tempo in secondi della durata del glissando, mentre 22.050 è la frequenza di campionamento espressa in Hertz.



La variazione lineare del glissando

E' necessaria però una precisazione: il glissando viene codificato in Matlab alla stessa frequenza di campionamento dichiarata per la funzione *oscil*, mentre quella dichiarata nella funzione *linspace* è una semplice convenzione temporale (crea il vettore la cui lunghezza rappresenta il numero di campioni). Quindi, per via della legge di Nyquist, è necessario ridefinire la frequenza finale dichiarata nella chiamata a *linspace* secondo la seguente relazione:

$$\text{freq_finale} = (\text{freq_finale_desiderata} + \text{freq_iniziale})/2;$$

Esempio:

per utilizzare correttamente il glissando definito nell'esempio precedente, è necessario correggere l'istruzione nel seguente modo:

```
gliss = linspace(220, (220+440)/2, 1*22050);
```

ovvero

```
gliss = linspace(220, 330, 1*22050);
```

Glissandi su più segmenti di retta

Volendo definire i glissandi più complessi (ad esempio ascendente e poi discendente), è sufficiente realizzare ciascun glissando singolarmente e quindi concatenarli.

Esempio:

```
% Crea un glissando da 220 a 440 Hz della durata di 0.5 secondi
```

```
gliss1 = linspace(220, 330, 0.5 * 22050);
```

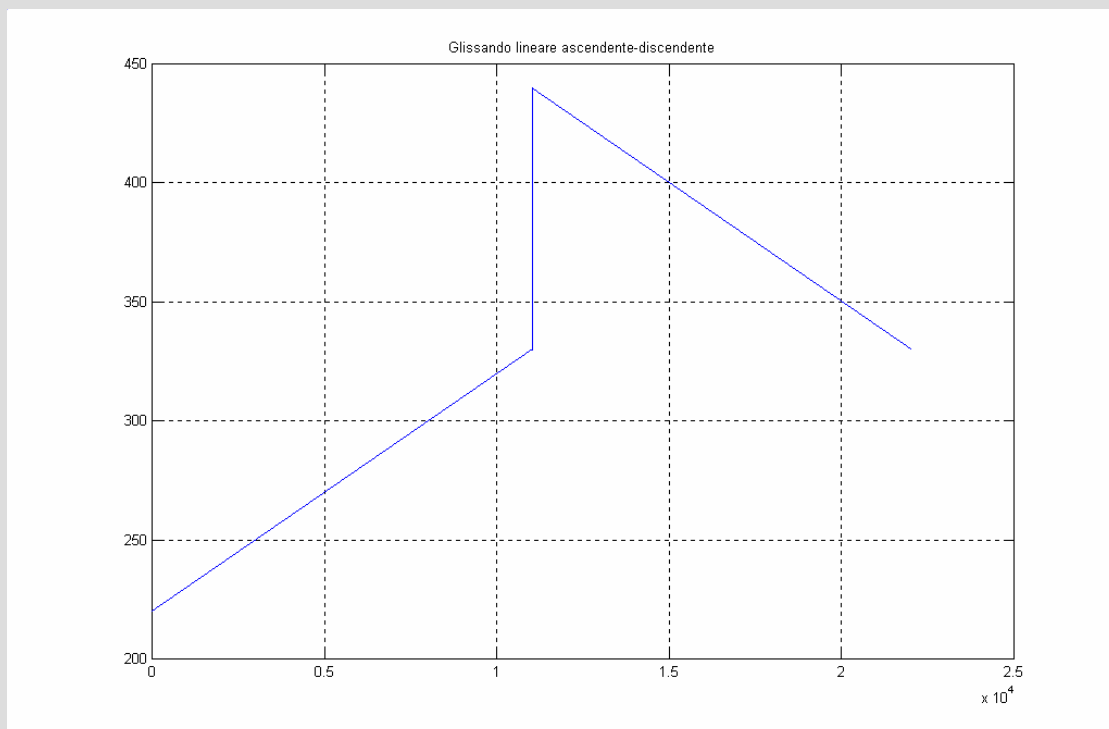
```
% Crea un glissando da 440 a 220 Hz della durata di 0.5 secondi
```

```
gliss2 = linspace(440, 330, 0.5 * 22050);
```

```
% Crea il glissando ascendente-discendente della durata di 1 secondo
```

```
% concatenando i due glissandi precedentemente definiti
```

```
gliss = [ gliss1 gliss2];
```



Il grafico mostra l'andamento del glissando appena creato. Le correzioni apportate alle frequenze finali possono trarre in inganno: sebbene il comportamento grafico sia anomalo, il risultato sonoro risulterà corretto.

Inviluppi di ampiezza

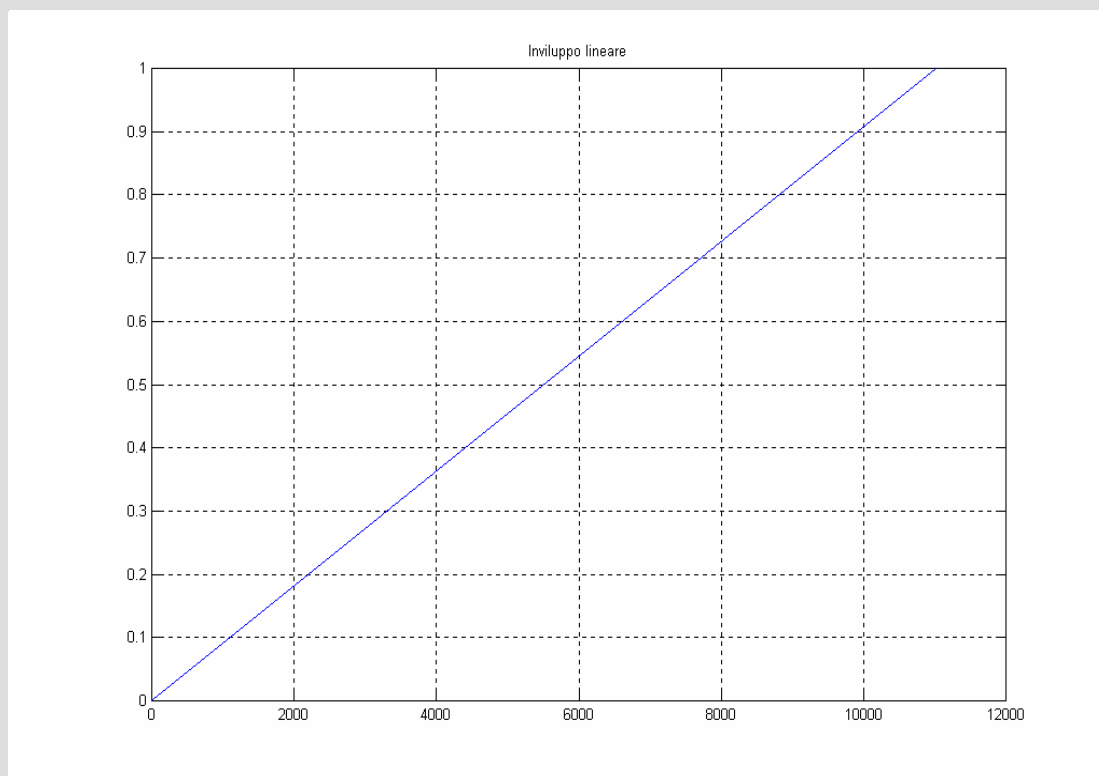
Lo stesso meccanismo utilizzato per i glissandi può essere applicato all'elaborazione dell'ampiezza di un suono: specificare un segmento per l'ampiezza da passare come primo argomento alla funzione *oscil*, creerà una dinamica interna al suono, cioè un inviluppo.

Esempio:

Nel caso più semplice potremmo voler creare un inviluppo lineare ascendente:

```
env = linspace(0, 1, 0.5*22050);
```

il parametro *0* corrisponde all'*ampiezza iniziale*, il parametro *1* all'*ampiezza finale*, *0.5* fa riferimento alla *durata* in secondi dell'inviluppo, mentre *22.050* è la *frequenza di campionamento* espressa in Hertz.



La variazione lineare dell'inviluppo

Nota: in questo caso non sono necessarie modifiche al valore finale dell'ampiezza, in quanto l'inviluppo funge da moltiplicatore della forma d'onda. Si ricordi però che i valori delle ampiezze devono essere nel range $[-1, +1]$.

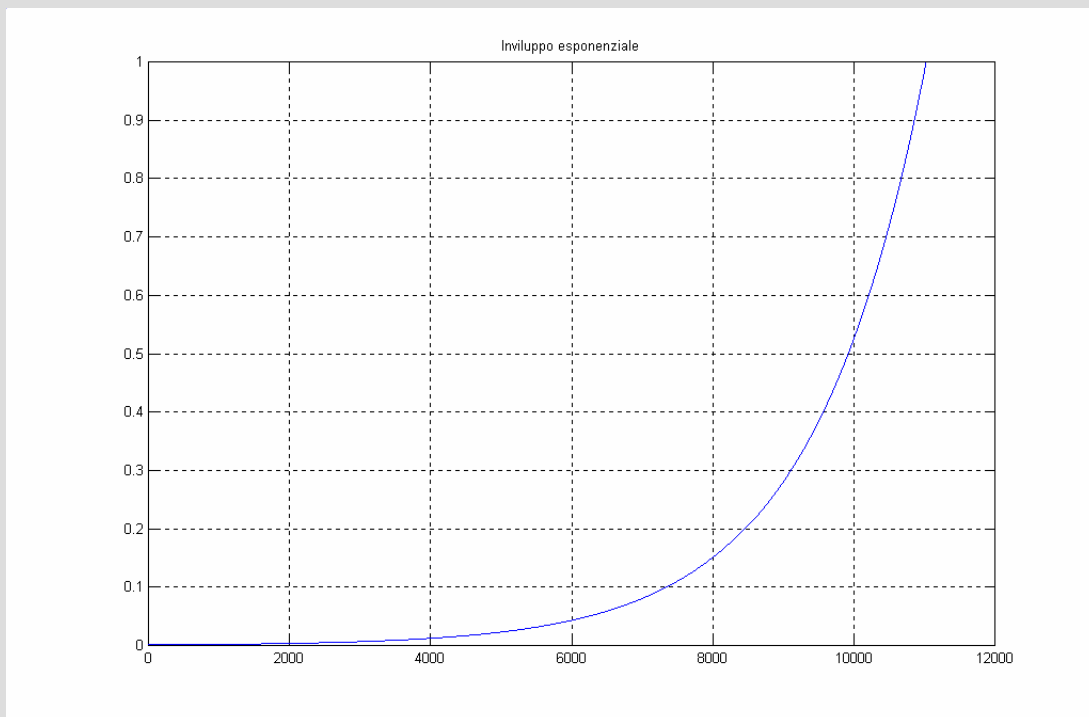
Inviluppi su segmenti di esponenziale

Con l'aggiunta di poche e semplici istruzioni alla chiamata di *linspace*, possiamo creare inviluppi che variano su segmenti di esponenziale.

Esempio:

```
env = exp(linspace(log(0.001), log(1), 0.5*22050));
```

L'istruzione `linspace(log(0.001), log(1), 0.5*22050)` crea un vettore di valori equispaziati che va da $\log(0.001)$ a $\log(1)$. Sarà poi la funzione `exp`, la quale agirà sul vettore appena creato, a generare l'inviluppo desiderato.



La variazione esponenziale dell'inviluppo

Nota: si osservi che la funzione `log` non può accettare come parametro il valore `0`. E' sufficiente in questi casi inserire un valore diverso da `0` adeguatamente piccolo (negli esempi, `0.001`).

Inviluppi ADSR

Generalmente gli inviluppi vengono descritti mediante quattro parametri:

- **attack** : la quantità di attack determina quanto velocemente l'inviluppo sale da 0 fino al livello massimo quando suoniamo una nota sulla tastiera.
- **decay** : la quantità di decay determina quanto velocemente l'inviluppo scende dal livello massimo al livello definito dal sustain.
- **sustain** : il livello di sustain determina di quanto l'inviluppo è mantenuto alto quando il gate (segnale generato dalla tastiera quando premiamo un tasto) è ON.
- **release** : la quantità di release determina quanto velocemente l'inviluppo scende a zero quando il gate diventa OFF (cioè quando il tasto premuto in precedenza è rilasciato).

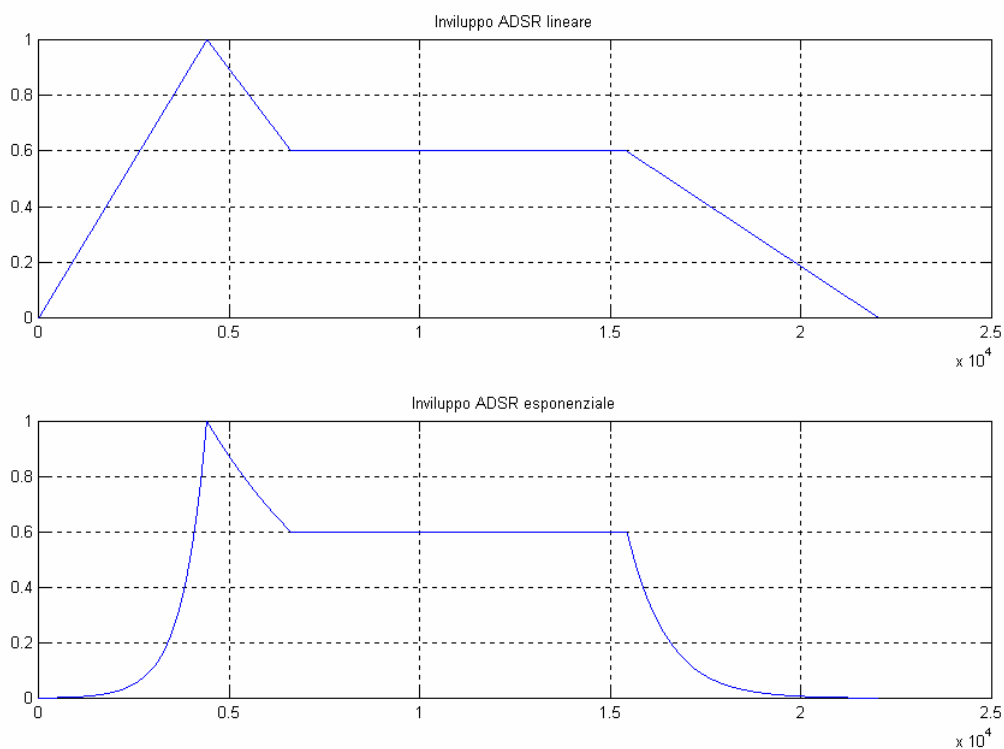
Esempio:

Creiamo un inviluppo ADSR lineare:

```
attack = linspace(0, 1, 0.2 * 22050);  
decay = linspace(1, 0.6, 0.1 * 22050);  
sustain = linspace(0.6, 0.6, 0.4 * 22050);  
release = linspace(0.6, 0, 0.3 * 22050);  
adsr_lin = [ attack decay sustain release];
```

A partire da questa forma lineare possiamo elaborare la forma esponenziale dell'inviluppo ADSR:

```
attack = exp(linspace(log(0.001), log(1), 0.2 * 22050));  
decay = exp(linspace(log(1), log(0.6), 0.1 * 22050));  
sustain = exp(linspace(log(0.6), log(0.6), 0.4 * 22050));  
release = exp(linspace(log(0.6), log(0.001), 0.3 * 22050));  
adsr_exp = [ attack decay sustain release];
```



I due inviluppi ADSR a confronto

Funzioni MSound: glissandi

```
% Funzione: lingliss
% -----
% Realizzazione del glissando su segmenti di retta.
%
% Prototipo: [seg] = lingliss(parameters,srate)
% Input:
%   - parameters: vettore dei parametri di lunghezza almeno pari a 3, o suoi multipli,
%                 laddove ciascuna tripla rappresenta:
%                 [frequenza_iniziale frequenza_finale durata_in_secondi]
%   - srate: frequenza di campionamento in Hertz
% Output:
%   - seg: vettore del glissando

function [seg] = lingliss(parameters,srate);

% Routine di controllo dei parametri di input
if (mod(length(parameters),3) ~= 0)
    fprintf(' lingliss error: incompatible parameters vector length\n');
    return
end

global_duration = 0;
for k = 1 : 3 : length(parameters)
    beg_val = parameters(k);
    end_val = parameters(k + 1);
    end_freq = (end_val + beg_val) / 2;
    duration = parameters(k + 2);
    global_duration = global_duration + duration;
    if (k == 1)
        seg = linspace(beg_val, end_freq, duration * srate);
    else
        seg = [ seg linspace(beg_val, end_freq, duration * srate) ];
    end
end
```

Funzioni MSound: inviluppi di ampiezza

```
% Funzione: linenv
% -----
% Realizzazione dell'inviluppo su segmenti di retta.
%
% Prototipo: [seg] = linenv(parameters,srate)
% Input:
%   - parameters: vettore dei parametri di lunghezza almeno pari a 3, o suoi multipli,
%                 laddove ciascuna tripla rappresenta:
%                 [ampiezza_iniziale ampiezza_finale durata_in_secondi]
%   - srate: frequenza di campionamento in Hertz
% Output:
%   - seg: vettore dell'inviluppo di ampiezza

function [seg] = linenv(parameters,srate);

% Routine di controllo dei parametri di input
if (mod(length(parameters),3) ~= 0)
    fprintf(' linenv error: incompatible parameters vector length\n');
    return
end

global_duration = 0;
for k = 1 : 3 : length(parameters)
    beg_val = parameters(k);
    end_val = parameters(k + 1);
    duration = parameters(k + 2);
    global_duration = global_duration + duration;
    if (k == 1)
        seg = linspace(beg_val, end_val, duration * srate);
    else
        seg = [ seg linspace(beg_val, end_val, duration * srate) ];
    end
end
end
```

```

% Funzione: expenv
% -----
% Realizzazione dell'involuppo su segmenti di esponenziale.
%
% Prototipo: [seg] = expenv(parameters,srate)
% Input:
% - parameters: vettore dei parametri di lunghezza almeno pari a 3, o suoi multipli,
%               laddove ciascuna tripla rappresenta:
%               [ampiezza_iniziale ampiezza_finale durata_in_secondi]
% - srate: frequenza di campionamento in Hertz
% Output:
% - seg: vettore dell'involuppo di ampiezza

function [seg] = expenv(parameters,srate);

% Routine di controllo dei parametri di input
if (mod(length(parameters),3) ~= 0)
    fprintf(' expenv error: incompatible parameters vector length\n');
    return
end

global_duration = 0;
for k = 1 : 3 : length(parameters)
    beg_val = parameters(k);
    end_val = parameters(k + 1);
    duration = parameters(k + 2);
    global_duration = global_duration + duration;
    if (k == 1)
        seg = exp(linspace(log(beg_val), log(end_val), duration * srate));
    else
        seg = [ seg exp(linspace(log(beg_val), log(end_val), duration * srate)) ];
    end
end
end

```

Capitolo 3

La trasformata discreta di Fourier in Matlab: la funzione fft e la sua inversa ifft

La *trasformata discreta di Fourier*, o *DFT*, è lo strumento principale per l'analisi dei segnali digitali. Si chiama trasformata di Fourier quell'operazione che consente di passare dalla rappresentazione di un segnale nel *dominio del tempo* (tempo/altezza) alla rappresentazione nel *dominio della frequenza* (frequenza/ampiezza), cioè al suo *spettro*.

Il calcolo della trasformata di Fourier è piuttosto lungo e complesso. Nel caso di segnali digitali si è trovato un mezzo per accelerare i calcoli, la cosiddetta Fast Fourier Transform (trasformata veloce di Fourier), con il solo vincolo che il numero di campioni da analizzare sia necessariamente una potenza intera di 2; quindi 32, 64, 128, 256, 1024, 2048, 4096, etc. Maggiore è il numero di punti, migliore è la risoluzione in frequenza, perchè, detto n il numero di punti su cui si effettua l'analisi e sr la frequenza di campionamento, la larghezza di banda bw di ogni canale di analisi è data dalla formula:

$$bw = sr/n$$

Esempio:

Per una frequenza di campionamento di 44100 Hz ed un numero di punti pari a 256, la larghezza di banda sarà

$$44100/256 = 172.27 \text{ Hz}$$

Questo equivale a dire che, se facciamo un'analisi di un suono campionato a 44100 Hz con un numero di punti pari a 256, lo spettro risultante conterrà le ampiezze delle seguenti bande di frequenza:

$$172.27, 344.5, 516.8, 689, 861.3, 1033.6, \dots$$

Matlab fornisce un metodo per calcolare la DFT con un tempo di esecuzione ridotto: la funzione **fft**, acronimo di *fast fourier transform*, e la sua inversa, **ifft**.

La funzione **fft**, con un singolo argomento x in input, calcola la DFT del vettore di input o della matrice. Se x è un vettore, **fft** calcola la DFT del vettore; se x è una matrice, **fft** calcola la DFT di ogni colonna.

Un secondo argomento a **fft** specifica il numero di punti n della trasformata, ovvero la lunghezza di DFT.

$$y = \text{fft}(x,n);$$

in questo caso, **fft** opera un padding (aggiunta di zeri) se la sequenza di input è minore di n , o tronca la sequenza se più lunga di n . Se n non è specificata il valore assunto è quello della sequenza di input.

Esempio: creiamo un vettore di tempo ed un segnale

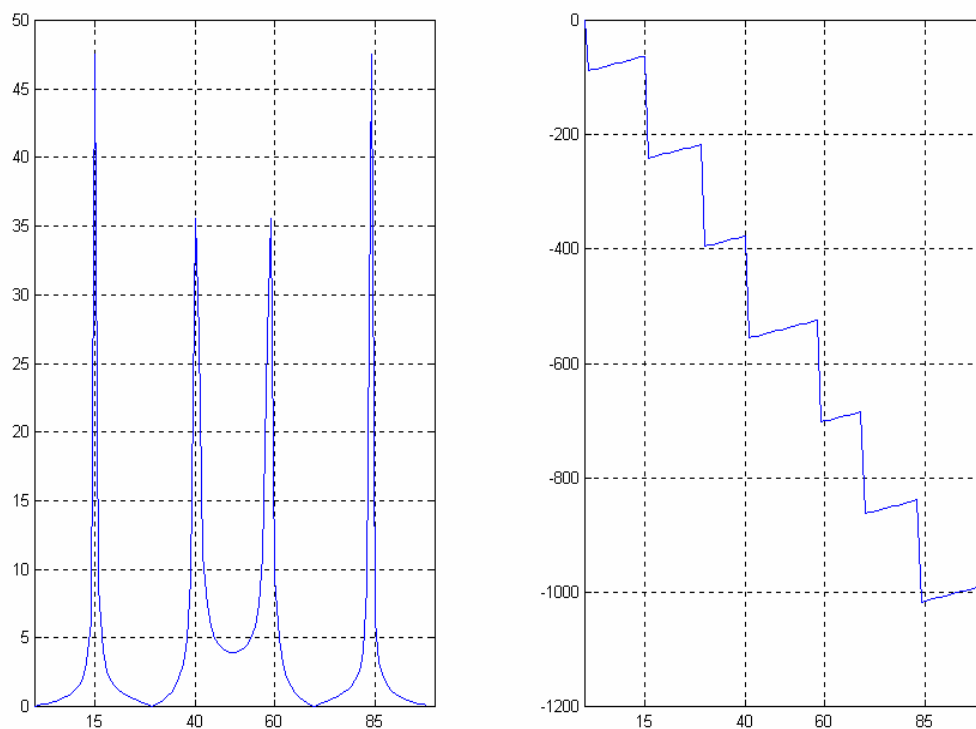
```
t = linspace(0, 1, 1*99); % vettore tempo
x = sin(2 * pi * 15 * t) + sin (2 * pi * 40 * t); % segnale
```

La DFT del segnale, l' ampiezza e la fase della trasformata, sono quindi:

```
y = fft (x) % calcola la DFT di x
m = abs (Y) % ampiezza
p = unwrap (angle(y)) % fase
```

Per disegnare l'ampiezza e la fase :

```
f = ( 0:length(y)-1)*99/length(y); % vettore delle frequenze
plot ( f, m)
set ( gca, 'XTick', [ 15 40 60 85]);
plot ( f, p*180/pi)
set ( gca,'XTick' , [ 15 40 60 85]);
```



Ampiezza e fase del segnale x (File sorgente: fourier.m)

Il tempo di esecuzione di fft dipende dalla lunghezza n della DFT da calcolare:

- se n è una potenza di 2, fft usa l' algoritmo radice di 2 ad alta velocità. Questo metodo è molto veloce, ed inoltre è altamente ottimizzato per x reali fornendo una velocità superiore del 40% rispetto al caso complesso;
- per ogni n che non è potenza di 2, fft usa un algoritmo a fattori primi. La velocità di questo algoritmo dipende sia dalla dimensione di n che dal numero di fattori primi che questa possiede. Ad esempio, sebbene 1013 e 1000 siano pressocchè vicini, fft

trasforma una sequenza di lunghezza 1000 molto più velocemente di una sequenza di lunghezza 1013;

- per un n numero primo, `fft` non può usare un algoritmo FFT. Il calcolo risulterà molto lento.

La funzione inversa della trasformata di Fourier `ifft` accetta una sequenza in input e, opzionalmente, il numero di punti desiderati per la trasformata. Nell'esempio seguente la sequenza originale `x` e quella ricostruita mediante `ifft` sono identiche (salvo errori di arrotondamento).

Matlab include anche la funzione FFT bidimensionale e la sua inversa, `fft2` e `ifft2` rispettivamente. Queste funzioni sono utili per segnali bidimensionali, o elaborazioni di immagini; non saranno trattate in questo testo.

Esempio:

```
t = linspace(0,1,1*22050);  
x = sin(2*pi*120*t);  
y = real(ifft(fft(x))) ;
```

E' talvolta conveniente riordinare l'output delle funzioni `fft` o `fft2` di modo che la componente in frequenza zero sia al centro della sequenza. La funzione Matlab `fftshift` sposta la componente in frequenza zero al centro di un vettore o di una matrice.

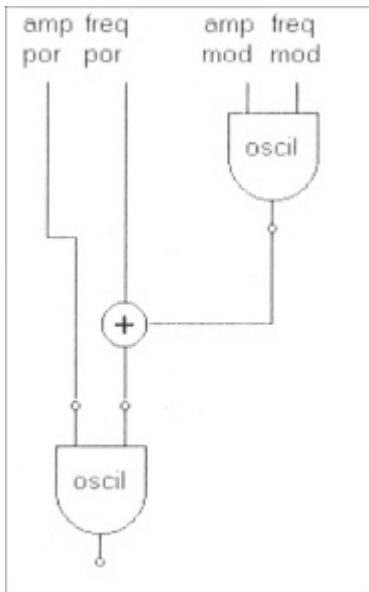
Capitolo 4

Modulazione di frequenza

La modulazione di frequenza o FM, è un semplice e potente metodo per la creazione ed il controllo di spettri complessi introdotto da John Chowning alla Stanford University nel 1973. Nella sua forma più semplice si basa su un oscillatore portante (**carrier**) la cui frequenza viene istantaneamente variata, o meglio modulata, da un oscillatore modulante (**modulator**). Questo modello viene spesso semplicemente chiamato FM o sine-waveFM. Altre forme di FM sono estensioni del modello di base.

Definizioni e terminologia:

- *Oscillatore* : dispositivo per generare forme d'onda;
- *FM* : Frequency (o Pitch) Modulation, si ha quando il pitch o frequenza di un oscillatore (la portante) è modulata da un altro oscillatore;
- *DX-FM* : FM nei sintetizzatori DX – Si ha quando entrambi gli oscillatori usano onde sinusoidali le cui frequenze sono "musicalmente intonate" sulla base della tastiera;
- *C* : Frequenza portante (*carrier*) – La frequenza dell'oscillatore che viene modulato;
- *M* : Frequenza modulante (*modulator*) – La frequenza dell'oscillatore che modula la portante.



Uno degli schemi più semplici per la modulazione di frequenza è quello illustrato in figura: abbiamo due oscillatori sinusoidali. Se l'ampiezza del segnale modulante è 0, non abbiamo modulazione e ciò che rimane è solo l'oscillazione sinusoidale dell'onda portante. Aumentando l'ampiezza del segnale modulante ha luogo la modulazione di frequenza e il segnale modulante devia verso l'acuto e verso il grave la frequenza dell'oscillatore portante. Più aumentiamo l'ampiezza del segnale modulante, maggiore è la deviazione di frequenza del segnale modulante.

Quando l'uscita dell'oscillatore modulante è positiva, la frequenza dell'oscillatore portante è più alta della sua frequenza di base. Al contrario se l'uscita dell'oscillatore modulante è negativa, la frequenza dell'oscillatore è più bassa della sua frequenza di base. Il massimo mutamento di frequenza che l'oscillatore portante subisce viene chiamato deviazione (***deviation***), oppure deviazione di picco (***peak frequency deviation***), ed è espresso in hertz.

Mentre con l'AM, nel caso che portante e modulante siano due sinusoidi, otteniamo una sola coppia di bande laterali, nella FM abbiamo una serie teoricamente infinita di coppie di bande laterali. Il numero delle bande laterali udibili dipende dalla deviazione: più è ampia la deviazione, maggiore è il numero di bande laterali che hanno ampiezza sufficiente per poter essere udite.

Teoricamente sono sempre presenti tutte le frequenze somma ($C+M$, $C+2M$, $C+3M$, ...) e le frequenze differenza ($C-M$, $C-2M$, $C-3M$, ...) fino all'infinito. In pratica, per indici di modulazione e frequenze modulanti basse le frequenze laterali di ordine elevato non sono udibili perché non hanno un'ampiezza sufficiente. Infatti:

$$\text{Deviazione} = \text{Indice_di_modulazione} * \text{Frequenza_modulante}$$

La gamma di frequenza delle bande laterali udibili sarà compresa fra la frequenza della portante meno la deviazione ($F_c - D$) e la frequenza della portante più la deviazione ($F_c + D$). Cosa succede quando abbiamo frequenze negative?

Le formule della FM:

L'idea di base della FM proposta da John Chowning è descritta nella seguente equazione:

$$x(t) = \sin[2 * \pi * f_c * t + I * \sin(2 * \pi * f_m * t)]$$

dove

- f_c è la frequenza del segnale portante;
- f_m è la frequenza del segnale modulante;
- I è l'indice di modulazione;
- t è il tempo (campione).

In trigonometria la formula che ci dà l'ampiezza istantanea di una senoide di ampiezza di picco A_0 è:

$$A = A_0 * \sin(2 * \pi * \omega * t)$$

in cui ω è la velocità angolare. Poiché il termine $2\pi\omega t$ esprime la frequenza f , in modo trigonometricamente scorretto, ma più adatto ai nostri scopi, possiamo scrivere:

$$A' = A_1 * \sin(f_1)$$

Quando operiamo una modulazione di frequenza, non facciamo altro che modificare la frequenza f , cioè renderla variabile in dipendenza di una seconda senoide

$$A'' = A_2 * \sin(f_2)$$

e quindi possiamo scrivere

$$A = A_1 * \sin(f_1 + A_2 * \sin(f_2))$$

che è appunto l'equazione trigonometrica della FM.

Introduciamo adesso un importantissimo parametro, l'*indice di modulazione* I . L'indice di modulazione si definisce come

$$I = d/m$$

in cui d è la *deviazione di picco*, cioè la quantità massima di cui varia la frequenza della portante, e m è la frequenza della modulante.

L'equazione si può dunque scrivere come:

$$A = A_1 * \sin(f_1 + I * \sin(f_2))$$

Nel caso di modulante a sua volta modulata, l'equazione diventa:

$$A = A_1 * \sin(f_1 + I_1 * \sin(f_2 + I_2 * \sin(f_3)))$$

Tutte le frequenze sotto lo zero ricompaiono in controfase e a specchio nel campo positivo e quindi il suono diventa più complesso ogniquale volta la sua gamma di frequenza o anche solo le sue componenti laterali udibili si riflettono nel campo audio e dunque si sommano algebricamente con le altre.

Lo stesso effetto di riflessione nel campo udibile ha luogo nel caso che la gamma di frequenza o le componenti laterali superino la frequenza di Nyquist.

Per determinare quali bande laterali sono presenti è necessario studiare il rapporto tra la frequenza portante e la frequenza modulante piuttosto che i singoli valori in hertz.

Proprietà del rapporto C:M

Innanzitutto definiamo alcune convenzioni:

- tratteremo i soli rapporti detti *non riducibili*, ovvero quelli fra interi divisibili solo per 1 e da nessun altro intero. Ad esempio il rapporto 2:2 equivale ad 1:1 e può essere ridotto a quest'ultimo per gli scopi pratici. Allo stesso modo 10:4 equivale a 5:2, 9:6 equivale a 3:2 e così via;
- tutti i possibili rapporti verranno suddivisi in sottogruppi:
 - uno di questi gruppi sarà quello descritto dal rapporto **1:N** che vedremo avere particolari proprietà.
 - un altro gruppo è quello descritto dal rapporto **N:M** dove N ed M sono minori di 10. In questo caso la restrizione ad interi a singola cifra è applicata puramente per semplicità di calcolo.
 - l'ultimo gruppo comprende i rapporti fra numeri maggiori o uguali a 10.

Talvolta i rapporti C:M vengono espressi tramite numeri reali (ad esempio 1:1.4). Questi rapporti possono essere approssimati con degli interi (nel caso precedente, 5:7).

Nella FM un insieme di bande laterali viene "prodotto" intorno alla frequenza portante, equamente spaziate ad una distanza pari alla frequenza modulante. Infatti ci riferiremo spesso alle bande laterali intese come coppie: I , II , III, e così via.

Calcolo delle bande laterali

Le cosiddette *bande laterali superiori* (*upper sidebands*) sono quelle che giacciono al di sopra della portante. Le loro frequenze sono :

$$C + M \quad C + 2M \quad C + 3M \quad C + 4M \quad C + 5M \quad \dots$$

Ad esempio, se C:M è 1:2, la modulante è doppia rispetto alla portante quindi la prima banda superiore sarà

$$C + M = 1 + 2 = 3.$$

La seconda banda superiore sarà

$$C + 2M = 1 + (2*2) = 1 + 4 = 5.$$

In alternativa è possibile sommare il valore della modulante al valore della banda immediatamente precedente. Si osservi che in questo caso le frequenze delle bande laterali sono tutte dispari, ed essendo la portante 1, le bande laterali superiori sono tutte le armoniche dispari, con la portante che è la fondamentale (la più bassa frequenza nello spettro). E' questo il caso della simulazione del suono di un clarinetto, il quale produce esclusivamente armoniche dispari.

Qualora il rapporto C:M fosse stato 2:5 la prima banda laterale superiore sarebbe stata

$$C + M = 2 + 5 = 7.$$

Poiché 7 non è un multiplo di 2 la banda sarà chiamata inarmonica. Ma la seconda banda superiore sarebbe stata

$$C + 2M = 2 + (2 * 5) = 2 + 10 = 12$$

ovvero la sesta armonica. Abbiamo quindi visto che le bande laterali possono essere armoniche o inarmoniche.

Le *bande laterali inferiori* (*lower sidebands*) sono quelle che giacciono al di sotto della portante, ovvero le cui frequenze sono:

$$C-M \quad C-2M \quad C-3M \quad C-4M \quad C-5M \quad \dots$$

Tutte le frequenze sotto lo zero vengono riflesse perché eliminiamo semplicemente il segno meno e trattiamo il valore come un numero positivo. Acusticamente questo processo di riflessione comporta una inversione di fase. Matematicamente esprimiamo questa riflessione applicando al calcolo delle bande laterali il valore assoluto: $|C-M|$.

Ad esempio per 1:2, la prima banda laterale inferiore è

$$|C - M| = |1 - 2| = |-1| = 1.$$

La seconda banda inferiore è

$$|C - 2M| = |1 - (2 * 2)| = |1 - 4| = |-3| = 3.$$

Comunque per rendere le cose più semplici avremmo potuto aggiungere 2 alla prima banda inferiore (1), la quale era già riflessa, ottenendo 3.

Per il rapporto 1:1, la prima banda inferiore è 0 (non udibile) e la seconda, la terza e la quarta sono 1,2,3 rispettivamente.

Per 7:5 le bande inferiori sono 2, 3, 8 e 13 dove 3 è la prima riflessa. In questi casi ci si chiede, dato un rapporto C:M, se la frequenza portante è la frequenza più bassa nello spettro, ovvero se è la fondamentale. Se è così possiamo trattare la frequenza portante come la tonalità fondamentale nel timbro risultante.

Innanzitutto abbiamo il caso del rapporto 1:1 le cui bande laterali superiori sono 2, 3, 4,... e le cui bande laterali inferiori sono 0, 1, 2, 3,... Chiaramente la portante è la componente più piccola diversa da zero, e tutte le bande laterali sono armoniche (*multiple*). Poiché 1:1 è l'unico rapporto con una banda laterale inferiore pari a zero, ci troviamo di fronte ad un caso particolare. E' anche l'unico rapporto che produce l'intera serie di armoniche. Per altri rapporti potremmo calcolare le bande laterali e decidere se alcune di queste debbano essere inferiori alla portante.

Si noti che nel caso 1:2 la prima banda laterale inferiore è $|-1| = 1$, quindi ricadrà sulla portante. Inoltre, se M è maggiore del doppio rispetto a C, ad esempio 2:5, allora la prima banda laterale inferiore riflessa sarà sempre più grande della portante.

Regola: Affinché la portante sia la fondamentale, M deve essere maggiore o uguale al doppio di C, o in alternativa essere in rapporto 1:1 con la portante.

Una ulteriore caratteristica è quella delle *bande coincidenti*: le bande laterali riflesse sono coincidenti con quelle non riflesse. Acusticamente queste proprietà si riflettono sull'ampiezza delle due bande laterali le quali verranno addizionate tra loro, influenzate inoltre dall'inversione di fase della banda riflessa. Poiché l'ampiezza di ogni banda laterale varia in base allo spessore

della modulazione, espresso dall'indice di modulazione, la somma dei contributi di ciascuna banda laterale diventa complesso.

Si nota che le bande laterali generate dai rapporti 1:1 hanno la caratteristica di essere coincidenti. Un comportamento simile si avrà con tutti i rapporti di tipo N:1 (2:1, 3:1, 4:1, ...).

```
% Funzione: sidebands
% -----
% Calcolo delle frequenze delle bande laterali create da una modulazione di frequenza.
%
% Prototipo: sidebands(freq, c_ratio, m_ratio, mod_index)
% Input:
% - freq: frequenza del segnale portante
% - c_ratio: valore della portante nel rapporto C:M
% - m_ratio: valore della modulante nel rapporto C:M
% - mod_index: indice di modulazione
% Output:
% - La funzione produce a video le frequenze delle bande laterali
%   generate.

function sidebands(frequency, c_ratio, m_ratio, m_index);

for k=1: m_index +2
    uppersb = (c_ratio * frequency) + (k * frequency * m_ratio);
    lowersb = abs((c_ratio * frequency) - (k * frequency * m_ratio));
    fprintf ('With k at %d, upper sideband: %d Hz; lower sideband: %d Hz\n',k,uppersb,lowersb);
end
```

Il secondo tipo di rapporto che mostra la stessa proprietà è il rapporto 1:2. Le armoniche dispari sono situate sia nelle bande laterali inferiori che in quelle superiori. Possiamo quindi estrapolare il secondo caso, quello dei rapporti N:2 (1:2, 3:2, 5:2, 7:2, ...).

Nessun altro rapporto mostra questa proprietà. Tutti quelli diversi da N:1 ed N:2 hanno una disposizione asimmetrica delle bande laterali generate, ovvero la distanza in frequenza tra bande laterali adiacenti non è costante.

Vogliamo la distanza tra la prima banda laterale superiore (C+M) e la prima banda laterale inferiore riflessa (C-M), che per definizione è negativa e può essere resa positiva invertendo l'espressione (M-C). Sottraendo queste due frequenze avremo: $(C+M) - (M-C) = 2C$.

Poiché la distanza fra bande laterali consecutive, siano esse inferiori o superiori, è M, la spaziatura rimanente è $M - 2C$. Quindi possiamo enunciare la seguente regola:

Quando C è la frequenza più bassa nello spettro, la spaziatura delle bande laterali al di sotto di questa (eccezion fatta per i rapporti 1:1) sarà $M - 2C$ e $2C$.

Per l'esempio precedente (2:5), la spaziature è $2 * 2 = 4$ e $5 - 4 = 1$. Si osservi che dobbiamo considerare C come la fondamentale. Questa sarà la cosiddetta *forma normale* del rapporto C:M.

Forma normale del rapporto C:M

Def.: Un rapporto C:M è nella sua forma normale quando la portante è la fondamentale nello spettro prodotto.

Regola: affinché un rapporto sia in forma normale, M deve essere maggiore o uguale a due volte C, o in alternativa essere in rapporto 1:1.

Tramite la forma normale è possibile fornire una base per uno schema di classificazione di tutti i rapporti ed allo stesso tempo codificare una regola per identificare le condizioni in cui la portante è la fondamentale. Se consideriamo solo i rapporti che coinvolgono numeri interi fino a 9, possiamo elencarli tutti nella forma normale:

1:1 1:2 1:3 1:4 1:5 1:6 1:7 1:8 1:9

Questi rapporti sono stati elencati in un ordine detto anche serie di Farey, dove il valore di M/C è crescente (o C/M è decrescente). Ogni rapporto in forma normale sarà associato ad una famiglia di rapporti, come mostrato di seguito.

Le serie di Farey

Le serie di Farey di ordine n, F_n , sono sequenze di frazioni irriducibili comprese tra 0 e 1, i cui denominatori non superano n. Ad esempio, la serie di Farey di ordine 5 è:

$$\frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{4}{5}, \frac{1}{1}$$

Termini simmetrici di una serie di Farey sono complementari. Ad esempio, $(2/5) + (3/5) = 1$. Inoltre, si noti che ogni termine della serie di Farey gode della seguente proprietà: se a/b , c/d ed e/f sono tre termini consecutivi, allora vale la relazione:

$$\frac{c}{d} = \frac{a+e}{b+f}$$

Vale inoltre la seguente proprietà: se a/b e c/d sono termini consecutivi di una serie di Farey, allora vale la relazione:

$$bc - ad = 1.$$

Ridurre un forma non-normale alla forma normale

Quando il valore di M nel rapporto è minore rispetto al doppio di C, si dice che il rapporto C:M è in forma non-normale, ma può essere ridotto alla forma normale applicando la formula

$$C = |C-M|$$

ovvero si sottrae M a C (ignorando l'eventuale segno meno risultante) e si tratta il risultato come il nuovo valore di C. E' possibile applicare questa regola fin quando il rapporto soddisfa il criterio di forma normale.

Consideriamo ad esempio il rapporto 3:1. Calcoliamo $|C-M| = |3-1| = 2$, quindi consideriamo il rapporto 2:1, il quale non è ancora in forma normale. Una ulteriore applicazione della formula

darà il seguente risultato: $|C-M| = |2-1| = 1$, quindi il nuovo rapporto sarà 1:1, il quale è in forma normale per definizione.

Un altro esempio è 8:5. Nella prima riduzione otteniamo $|8-5| = 3$ e consideriamo il nuovo rapporto 3:5, che non è ancora in forma normale. Riducendo ulteriormente otteniamo $|3-5| = |-2| = 2$. Il nuovo rapporto, 2:5 è in forma normale, infatti $5 > 2*2$.

Si osservi che la riduzione a forma normale è l'operazione inversa della generazione di bande laterali: per generare bande laterali addizioniamo M e C, per ridurre il rapporto, operiamo una sottrazione.

Famiglie di rapporti C:M

Per ogni rapporto in forma normale esiste un insieme di rapporti il quale produce lo stesso insieme di bande laterali. Questo insieme di rapporti si dice "famiglia", ed identifica la famiglia in base al rapporto in forma normale.

Consideriamo i rapporti 2:5, 3:5 e 7:5. Le seguenti sono le bande laterali

2:5 → 2	3	7	8	12	13	17	18	22	23	27	28
3:5 → 3	2	8	7	13	12	18	17	23	22	28	27
7:5 → 7	2	12	3	17	8	22	13	27	18	32	23

Osserviamo che i tre rapporti generano le stesse bande laterali, ma con ordini differenti. 2:5 è il rapporto in forma normale, quindi i tre rapporti appartengono alla famiglia del rapporto 2:5.

Per generare l'intero insieme dei membri di una famiglia di rapporto, la regola è la seguente:

Per un rapporto C:M in forma normale, l'insieme dei membri della famiglia è:

$C+NM : M$ e $|C-NM| : M$ per $N = 1,2,3,4,\dots$

Semplicemente si prende ogni banda laterale e si usa come valore C del rapporto, mantenendo M costante. Ad esempio, la famiglia 2:5 comprende:

$C + M : M$	$2 + 5 : 5 = \mathbf{7 : 5}$	$ C - M : M$	$ 2 - 5 : 5 = \mathbf{3 : 5}$
$C + 2M : M$	$2 + 2*5 : 5 = 2 + 10 : 5 = \mathbf{12 : 5}$	$ C - 2M : M$	$ 2 - 2*5 : 5 = 2 - 10 : 5 = \mathbf{8 : 5}$
$C + 3M : M$	$2 + 3*5 : 5 = 2 + 15 : 5 = \mathbf{17 : 5}$	$ C - 3M : M$	$ 2 - 3*5 : 5 = 2 - 15 : 5 = \mathbf{13 : 5}$
$C + 4M : M$	$2 + 4*5 : 5 = 2 + 20 : 5 = \mathbf{22 : 5}$	$ C - 4M : M$	$ 2 - 4*5 : 5 = 2 - 20 : 5 = \mathbf{18 : 5}$
...

Rapporti C:M armonici e inarmonici

Quando abbiamo mostrato come generare bande laterali, abbiamo notato come alcune fossero armoniche ed altre inarmoniche, ovvero non multiple della fondamentale. Acusticamente c'è una differenza importante poiché ogni rapporto produce un timbro che può essere armonico o inarmonico.

La regola per determinare le armoniche/inarmoniche è molto semplice nel caso dei rapporti in forma normale:

I rapporti armonici in forma normale hanno sempre la forma 1:N; al contrario, i rapporti inarmonici in forma normale non hanno mai la forma 1:N.

Esempio:

Rapporti armonici:	1:1	1:2	1:3	1:4	1:5	1:6	...
Rapporti inarmonici:	2:9	2:7	3:8	2:5	4:9	3:5	...

Per stabilire quando un rapporto in forma non-normale è armonico o meno, è possibile ridurlo alla forma normale e riprendere la semplice regola precedente. Comunque, i rapporti con $M = 1, 2, 3, 4, 6$ sono sempre inarmonici. Per $M = 5, 7, 8, 9$ è necessario controllare se C è un multiplo di $M \pm 1$: se multiplo, il rapporto è armonico. Per esempio, 9:5, 11:5, 14:5 e 16:5 sono membri della famiglia dei rapporti 1:5 i cui valori di C sono $10 - 1$, $10 + 1$, $15 - 1$, $15 + 1$, e così via.

Calcolo della frequenza fondamentale

I membri della famiglia dei rapporti 2:5 sono 7:5, 8:5, 12:5, 13:5, ... e sono inarmonici. Ci chiediamo ora come sia possibile calcolare la giusta frequenza portante per un membro di una famiglia di rapporti che produce lo stesso spettro di un rapporto in forma normale di una data frequenza portante/fondamentale.

Consideriamo una frequenza portante/fondamentale di 100 Hz. L'equazione è

$$FC = (C \text{ ratio}) * (N.F. C) / (N.F. C \text{ ratio})$$

Ad esempio, se la frequenza portante in forma normale è 100 Hz per 1:1, allora la portante corrispondente per 3:1 è $(3 * 100) / 1 = 300$ Hz. Per ogni rapporto 1:N è possibile calcolare immediatamente il valore come $C * 100$. Per un rapporto C:M in forma normale, calcoliamo $C * 100 / NFC$: ad esempio, per 7:5 e 2:5, la portante per 7:5 è pari a $7 * 100 / 5 = 140$ Hz. Acusticamente ascoltiamo un membro di una famiglia di rapporti in forma normale su una portante alta essendo la sua energia distribuita fra le frequenze alte, piuttosto che centrata sulla fondamentale, per un indice di modulazione basso.

Quando vogliamo usare un rapporto in forma non-normale con una specifica frequenza portante, spesso vogliamo conoscere quale sarà la sua frequenza fondamentale, oppure desideriamo una fondamentale specifica e vogliamo sapere come calcolare la portante giusta. Per calcolare la fondamentale (FF) per un rapporto C:M in forma non-normale la cui portante è conosciuta, abbiamo bisogno di conoscere la forma normale del rapporto NFC:M, e quindi applicare l'equazione:

$$FF = (\text{Frequenza portante}) * (N.F. C \text{ ratio}) / (C \text{ ratio})$$

Ad esempio, con un rapporto 5:2 ed una portante di 550 Hz, il rapporto in forma normale è 1:2 e quindi la fondamentale è $(550 * 1/5) = 110$ Hz. Analogamente, per 7:5 ed una portante di 700 Hz, il rapporto in forma normale è 2:5, e quindi la fondamentale è $700 * 2 / 7 = 200$ Hz.

Se conosciamo la fondamentale e vogliamo conoscere la portante, possiamo usare l'equazione:

$$\text{Portante} = (\text{fondamentale}) / (C \text{ ratio}) / (N.F. C \text{ ratio})..$$

Serie generate da M:C

Seguono due tavole. Nella prima, dati i valori di M e C, è possibile trovare le serie che vengono generate. Nella seconda vengono mostrati gli spettri armonici delle serie.

M:C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1:1	2:1	3:1	4:1	5:1	6:1	7:1	8:1	9:1	10:1	11:1	12:1	13:1	14:1	15:1	16:1
2	1:1	1:1 x2	3:1	2:1 x2	5:2	3:1 x2	7:2	4:1 x2	9:2	5:1 x2	11:2	6:1 x2	13:2	7:1 x2	15:2	8:1 x2
3	1:1	2:1	1:1 x3	4:1	5:2	2:1 x3	7:3	8:3	3:1 x3	10:3	11:3	4:1 x3	13:3	14:3	5:1 x3	16:3
4	1:1	1:1 x2	3:1	1:1 x4	5:1	3:1 x2	7:3	2:1 x4	9:4	5:2 x2	11:4	3:1 x4	13:4	7:2 x2	15:4	4:1 x4
5	1:1	2:1	3:1	4:1	1:1 x5	6:1	7:2	8:3	9:4	2:1 x5	11:5	12:5	13:5	14:5	3:1 x5	16:5
6	1:1	1:1 x2	1:1 x3	2:1 x2	5:1	1:1 x6	7:1	4:1 x2	3:1 x3	5:2 x2	11:5	2:1 x6	13:6	7:3 x2	5:2 x3	8:3 x2
7	1:1	2:1	3:1	4:1	5:2	6:1	1:1 x7	8:1	9:2	10:3	11:4	12:5	13:6	2:1 x7	15:7	16:7
8	1:1	1:1 x2	3:1	1:1 x4	5:2	3:1 x2	7:1	1:1 x8	9:1	5:1 x2	11:3	3:1 x4	13:5	7:3 x2	15:7	2:1 x8
9	1:1	2:1	1:1 x3	4:1	5:1	2:1 x3	7:2	8:1	1:1 x9	10:1	11:2	4:1 x3	13:4	14:5	5:2 x3	16:7
10	1:1	1:1 x2	3:1	2:1 x2	1:1 x5	3:1 x2	7:3	4:1 x2	9:1	1:1 x10	11:1	6:1 x2	13:3	7:2 x2	3:1 x5	8:3 x2
11	1:1	2:1	3:1	4:1	5:1	6:1	7:3	8:3	9:2	10:1	1:1 x11	12:1	13:2	14:3	15:4	16:5
12	1:1	1:1 x2	1:1 x3	1:1 x4	5:2	1:1 x6	7:2	2:1 x4	3:1 x3	5:1 x2	11:1	1:1 x12	13:1	7:1 x2	5:1 x3	4:1 x4
13	1:1	2:1	3:1	4:1	5:2	6:1	7:1	8:3	9:4	10:3	11:2	12:1	1:1 x13	14:1	15:2	16:3
14	1:1	1:1 x2	3:1	2:1 x2	5:1	3:1 x2	1:1 x7	4:1 x2	9:4	5:2 x2	11:3	6:1 x2	13:1	1:1 x14	15:1	8:1 x2
15	1:1	2:1	1:1 x3	4:1	1:1 x5	2:1 x3	7:1	8:1	3:1 x3	2:1 x5	11:4	4:1 x3	13:2	14:1	1:1 x15	16:1
16	1:1	1:1 x2	3:1	1:1 x4	5:1	3:1 x2	7:2	1:1 x8	9:2	5:2 x2	11:5	3:1 x4	13:3	7:1 x2	15:1	1:1 x16

Serie armoniche generate da M:C																																			
Serie	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
1:1]	[]	[]	[]]	[]	[]	[]	[]]	[]	[]	[]	[]	[]	[]	[]	[]		
2:1]	[]	[]]	[]]		[]]]]]]]]]]			
3:1]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]		
4:1]		[]		[]		[]		[]		[]		[]		[]		[]		[
5:1]			[]		[]			[]			[]			[]		[]		[]			
6:1]			[]				[]				[]			[[]				[]]		
7:1]				[]						[]					[]						[]						
8:1]					[]							[]						[]							[]		
9:1]						[]								[]								[]]		
10:1]							[]									[]								[]				
11:1]							[]											[]										[
12:1]									[]											[]]	
13:1]										[]												[]]	
14:1]											[]													[]]	
15:1]												[]														[]]	
16:1]													[]																[]	
Serie	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
5:2]	[]	[]	[]	[]	[]	[]]	
7:2]		[]				[]			[]			[]]	
9:2]				[]						[]				[]]	
11:2]					[]								[]								[]	
13:2]								[]										[]]	
15:2]											[]								[[]	
7:3]	[]	[]	[]	[]		[
8:3]		[]		[]		[]		[]	
10:3]			[]					[]				[]	
11:3]				[]						[]						[]	
13:3]							[]									[]]	
14:3]							[]										[]		
16:3]										[]											[]	
Series	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
9:4]	[]	[]	[]		[
11:4]		[]			[]			[]	
13:4]				[]						[]]	
15:4]							[]						[]	
11:5]	[]	[]	[]	
12:5]		[]		[]		[]	
13:5]		[]			[]]	
14:5]			[]				[]	
16:5]						[]						[]
13:6]	[]	[]	
15:7]	[]	[]
16:7]	[[]

Tavola 2: Serie armoniche generate da M:C.

] [| indicano bande riflesse non coincidenti.

Se **C** appare in corrispondenza di una], allora [sarà una banda riflessa (e viceversa).

Il simbolo][indica bande riflesse coincidenti.

Indice di modulazione

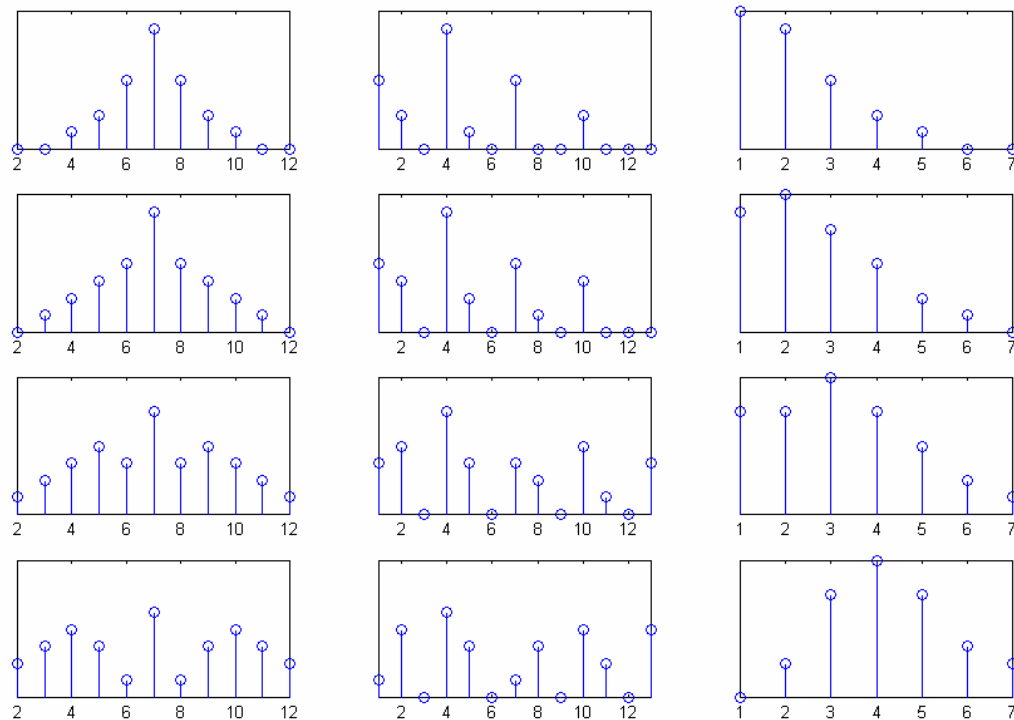
L'indice di modulazione determina la potenza (l'ampiezza) della portante e di ogni ordine di bande laterali. L'ordine si riferisce al livello di separazione delle bande laterali rispetto alla portante e per ogni ordine esistono due bande. L'uscita FM risultante sarà simmetrica per ogni coppia di bande laterali.

E' difficile calcolare le ampiezze esatte, ma è possibile definire dei comportamenti generici delle bande laterali in base all'indice di modulazione. Molto basilaramente, incrementando l'indice di modulazione aumenta il numero di bande laterali presenti:

- quando non c'è modulazione esiste solo la frequenza portante (non ci sono bande laterali);
- a livelli bassi di modulazione la distribuzione delle ampiezze (le altezze nello spettro armonico) assume una forma triangolare con l'apice in C;
- piccoli incrementi nella modulazione allargano la forma triangolare (si vengono a formare più bande);
- quando la modulazione arriva a livelli molto alti la distribuzione delle ampiezze cambia in una coppia di triangoli centrati sugli ordini medi, con un picco in C.

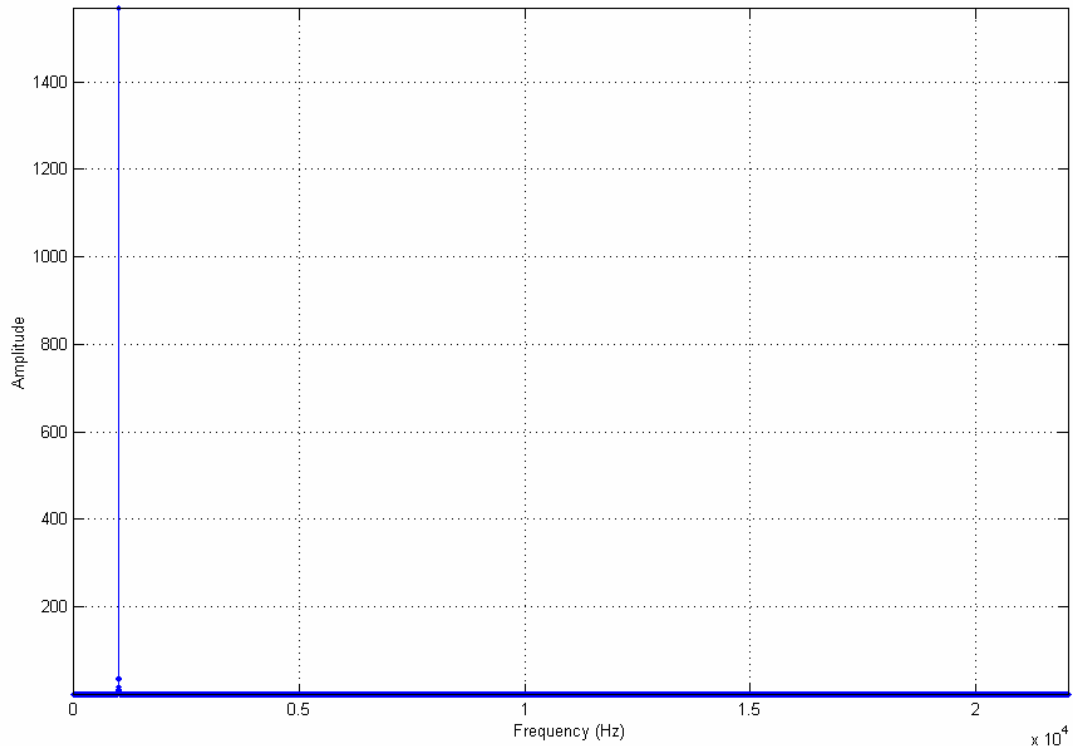
La tavola che segue mostra i cambiamenti di distribuzione in corrispondenza dell'aumento dell'indice di modulazione (il grafico in alto rappresenta minor modulazione e quello in basso maggior modulazione).

I grafici della colonna sinistra sono quelli relativi al rapporto $M:C = 1:7$ (senza bande laterali riflesse). Quelli della colonna centrale fanno riferimento al rapporto $M:C = 3:4$ (con bande laterali riflesse non coincidenti). Infine i grafici della colonna destra considerano il rapporto $M:C = 1:1$ (con bande laterali riflesse coincidenti).

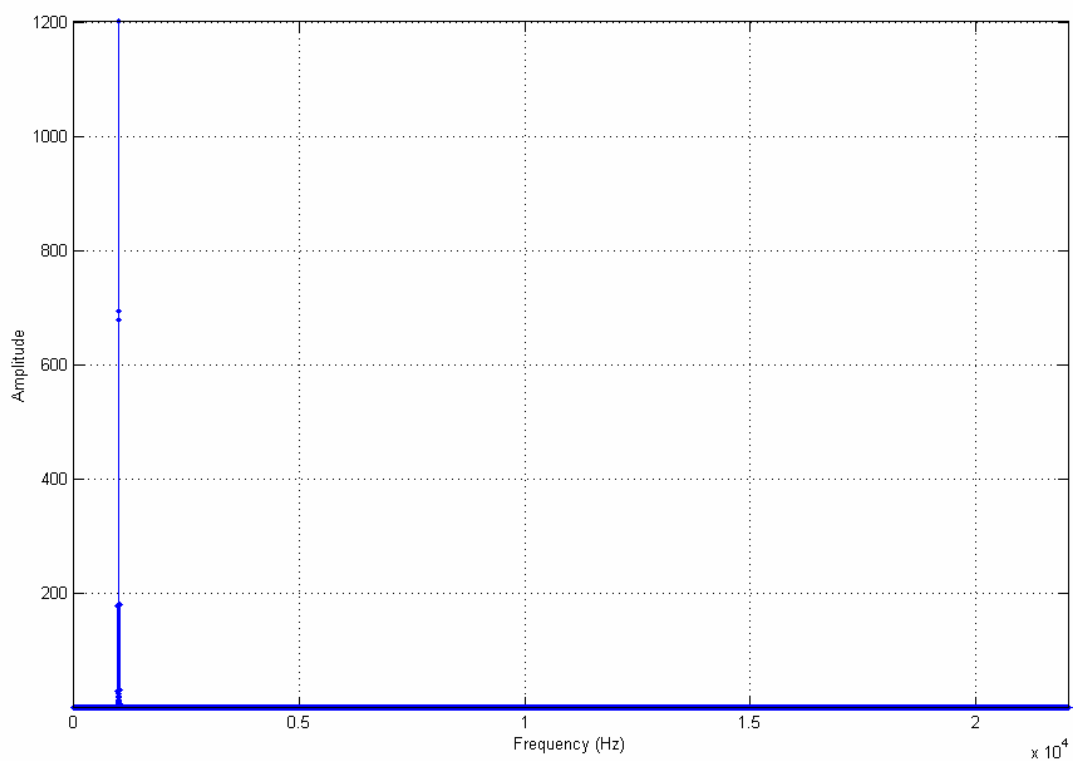


Nota: i grafici sono esempi guida, non sono accurati.

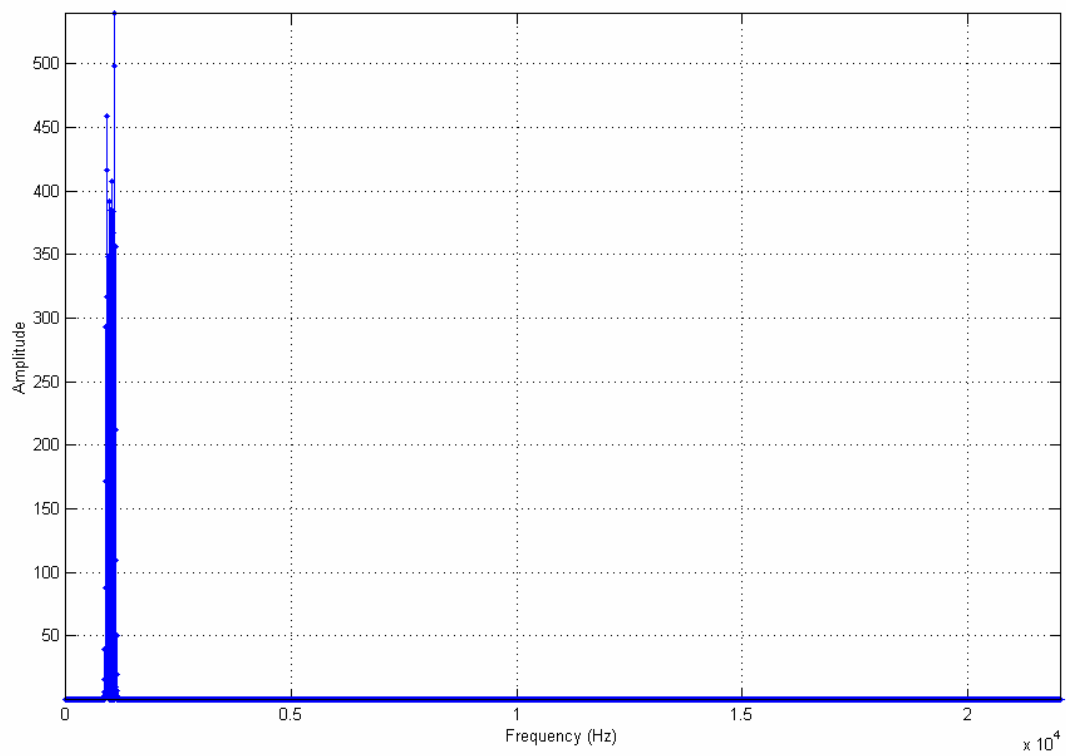
Osserviamo ora l'impatto dell'indice di modulazione sullo spettro, utilizzando l'analisi di Fourier e la funzione fft disponibile in Matlab. Consideriamo una frequenza portante di 1000 Hz ed una frequenza modulante di 10 Hz. Osserviamo come cambiano gli spettri al variare dell'indice di modulazione.



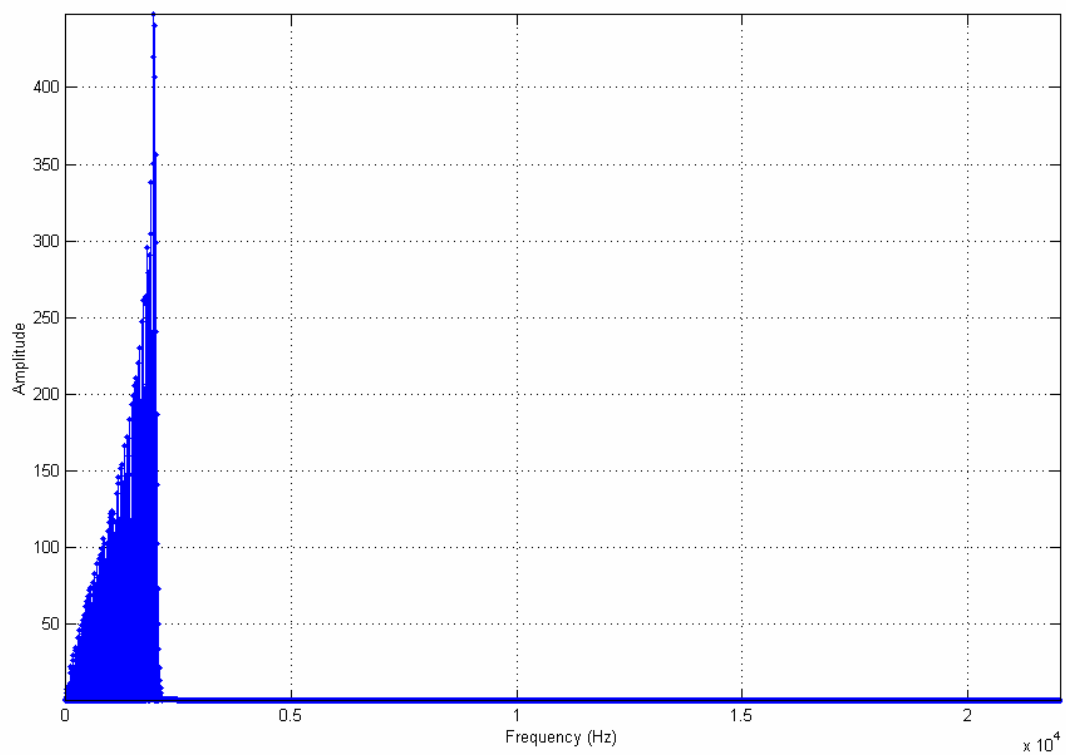
Indice di modulazione = 0



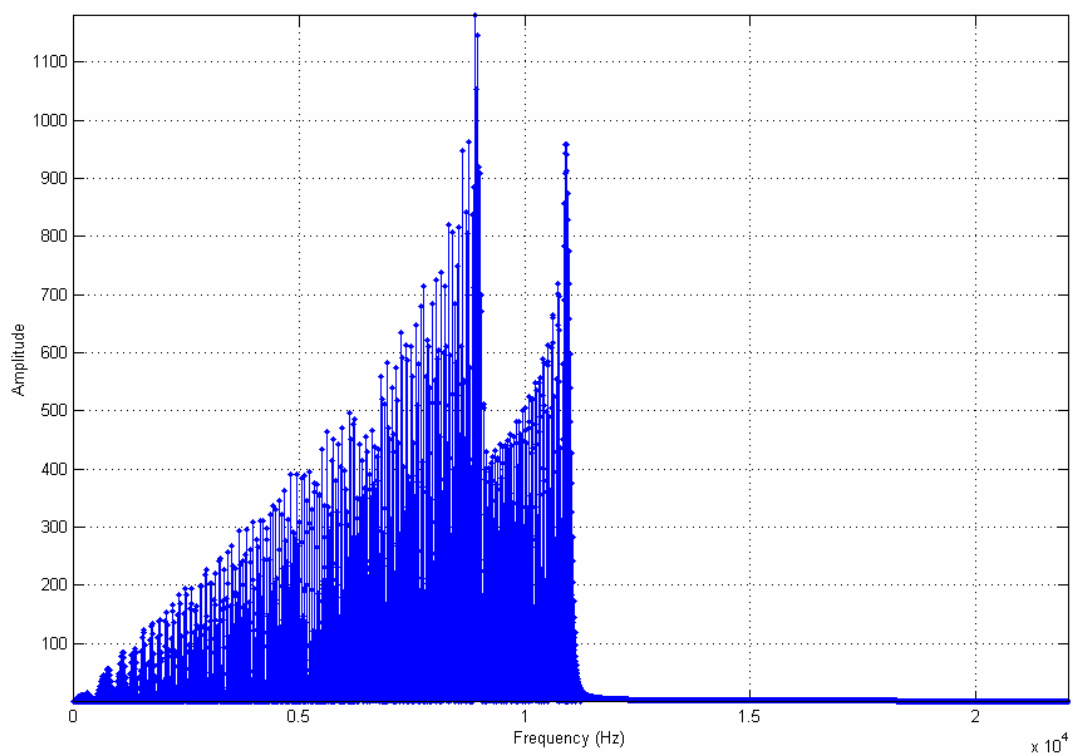
Indice di modulazione = 1



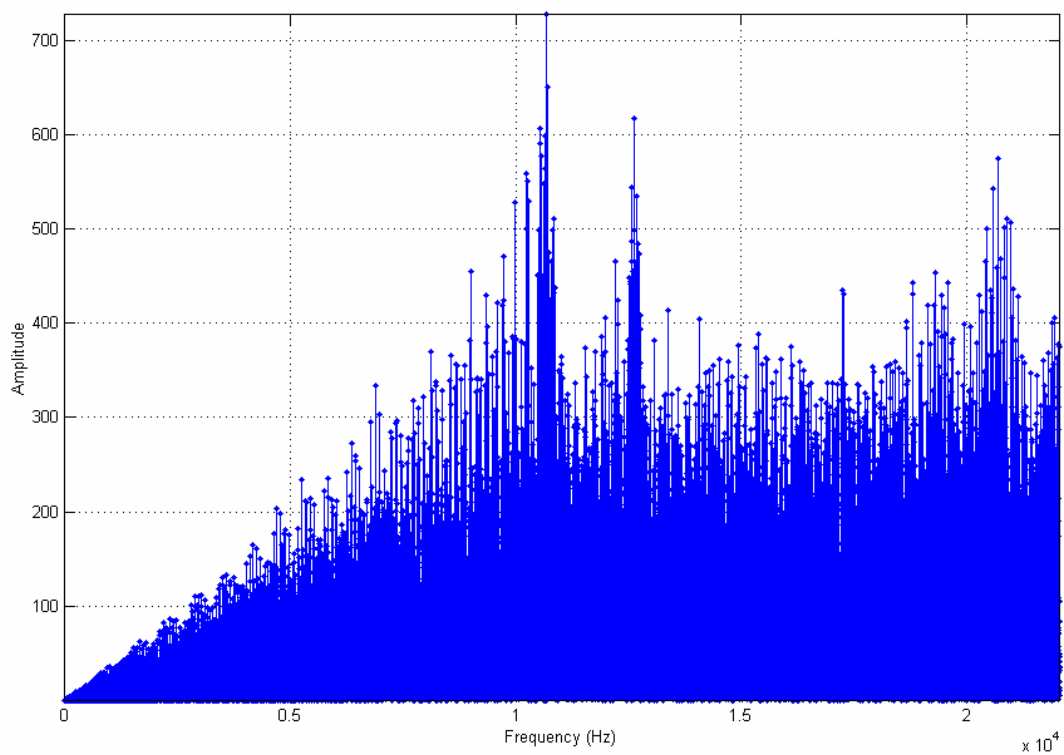
Indice di modulazione = 10



Indice di modulazione = 100

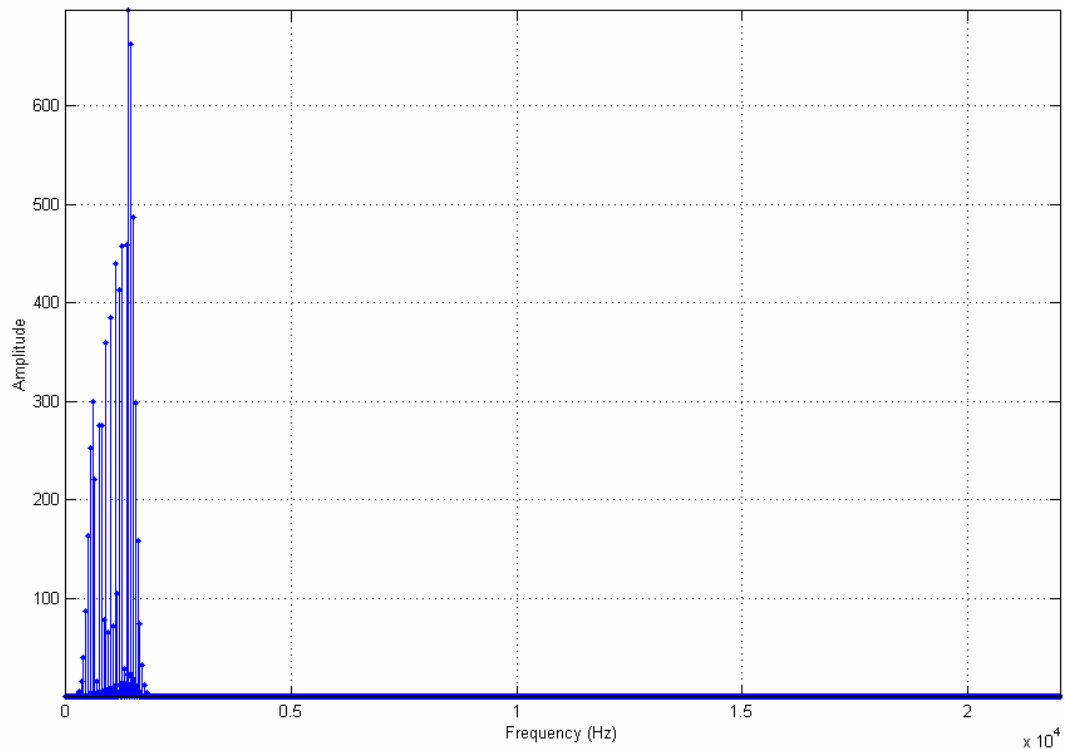


Indice di modulazione = 1000

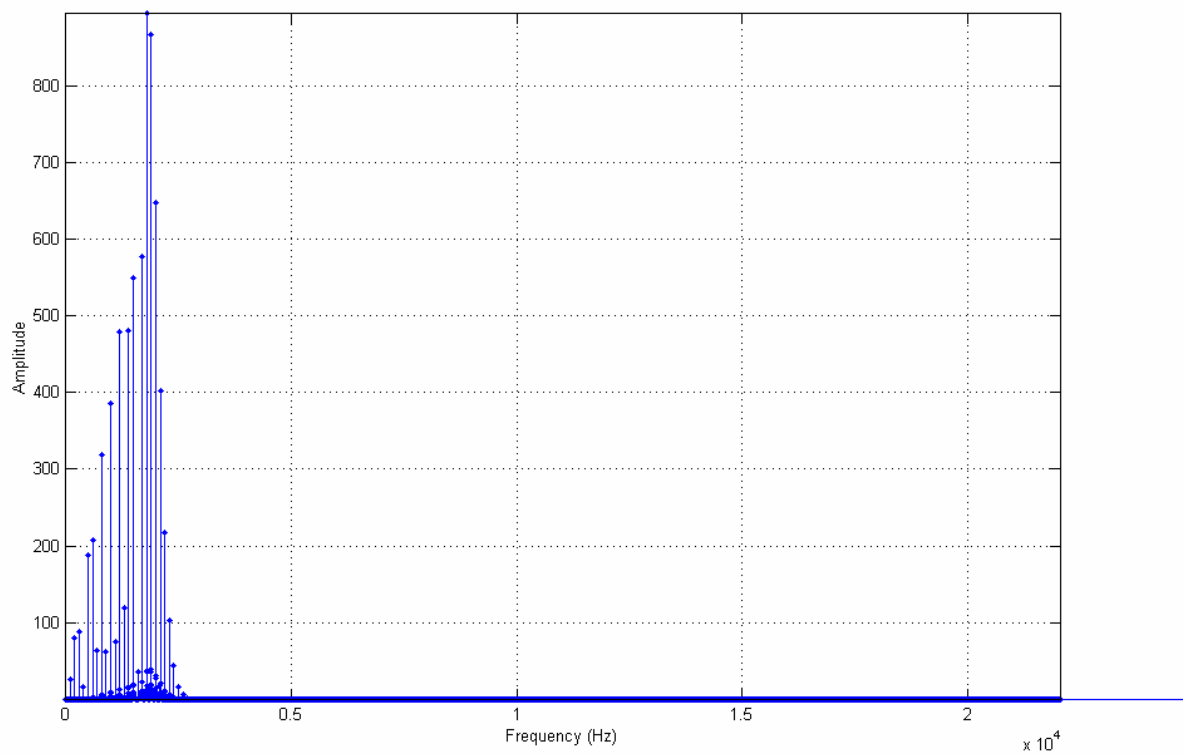


Indice di modulazione = 10000

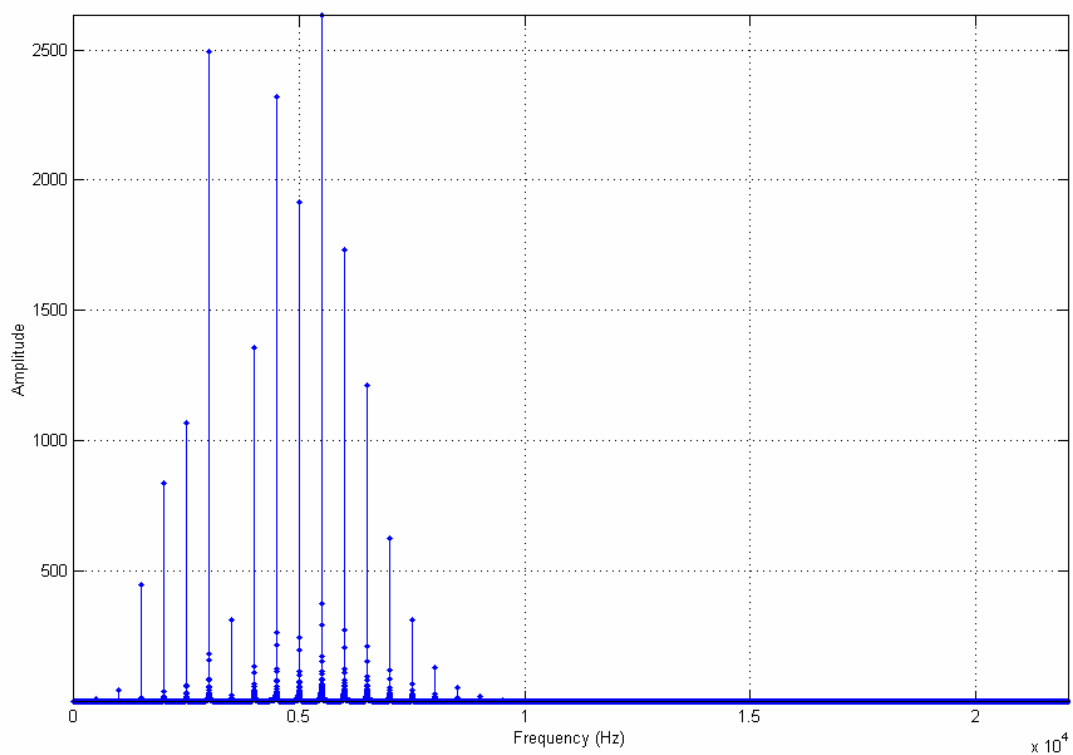
Osserviamo ora l'impatto della frequenza modulante sullo spettro, utilizzando il metodo precedente. Fissiamo l'indice di modulazione a 10, e consideriamo una frequenza portante di 1000 Hz. Osserviamo come cambiano gli spettri al variare della frequenza modulante.



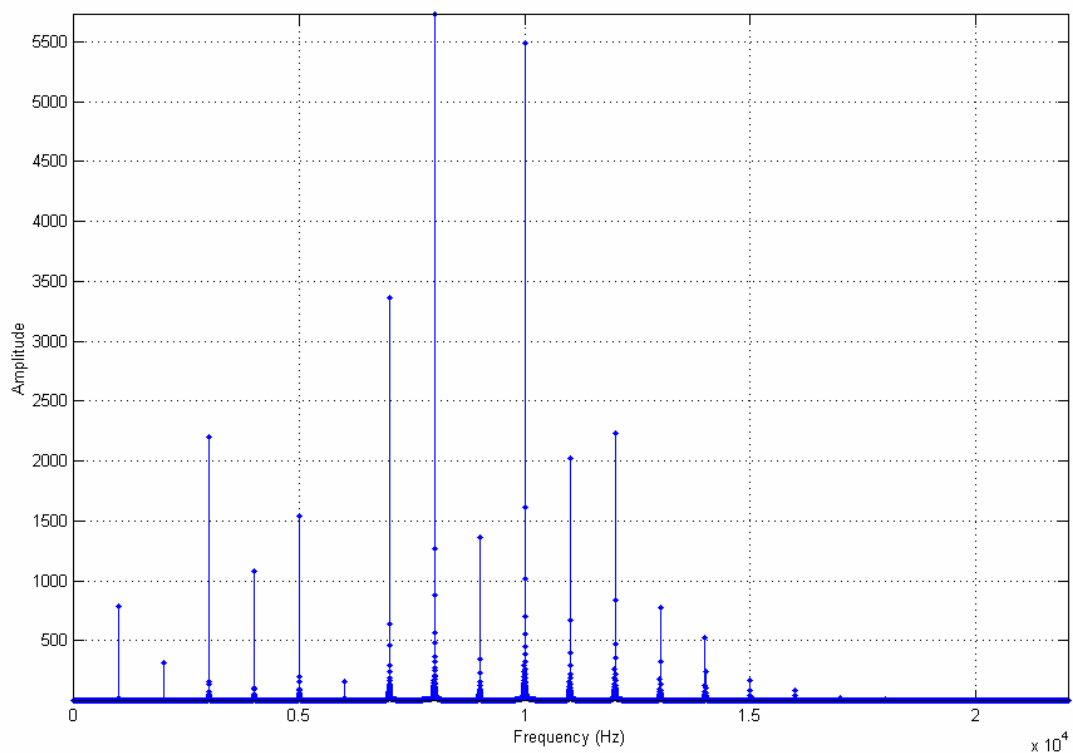
Frequenza modulante = 50



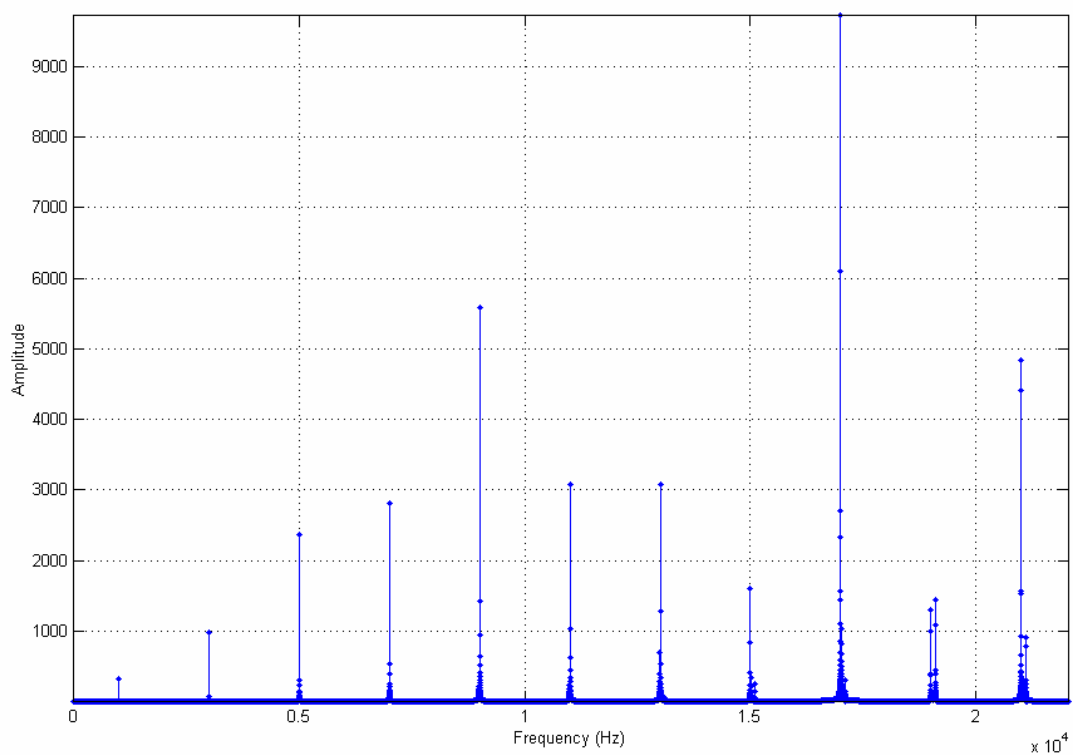
Frequenza modulante = 100



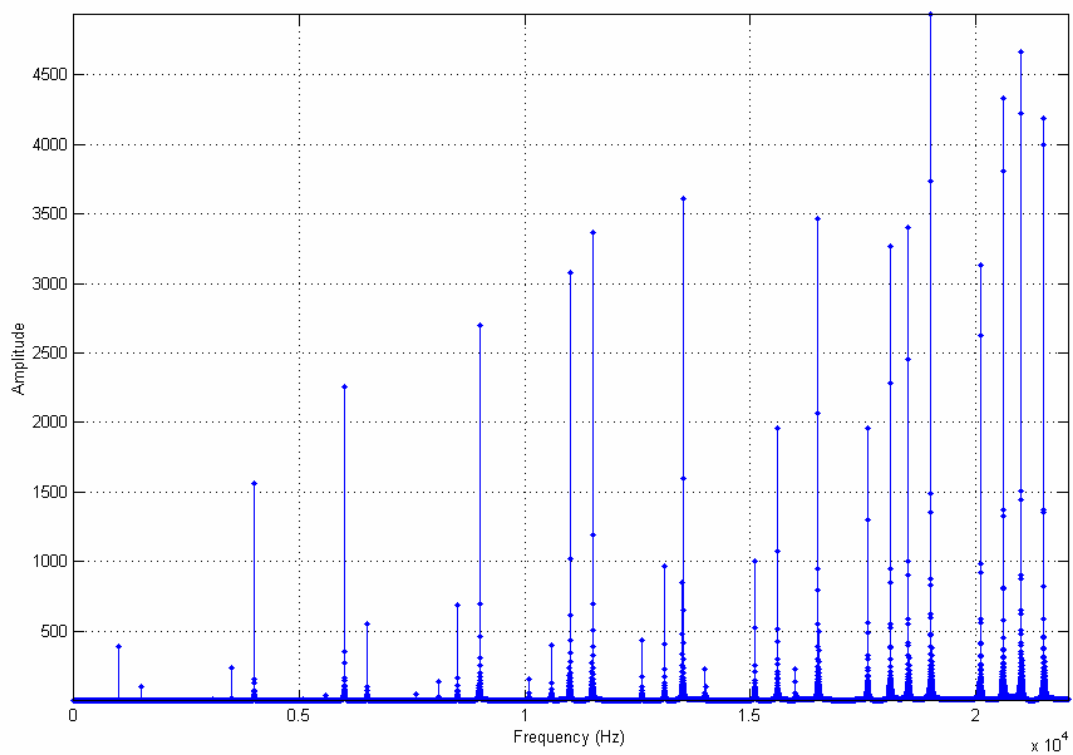
Frequenza modulante = 500



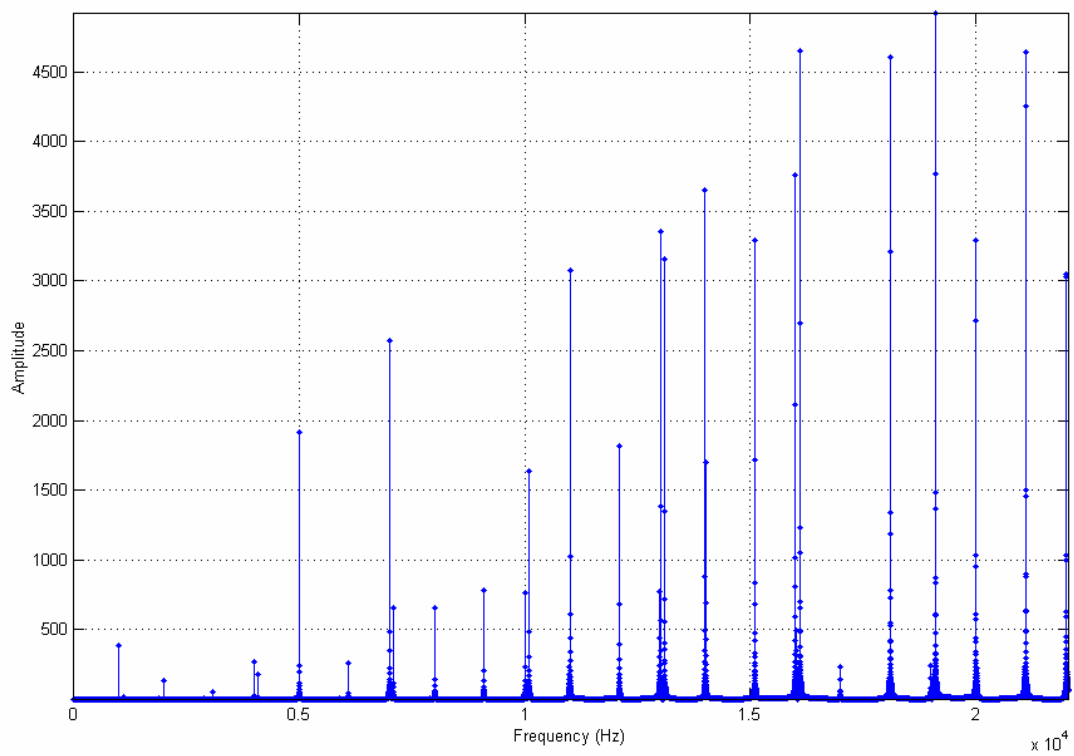
Frequenza modulante = 1000



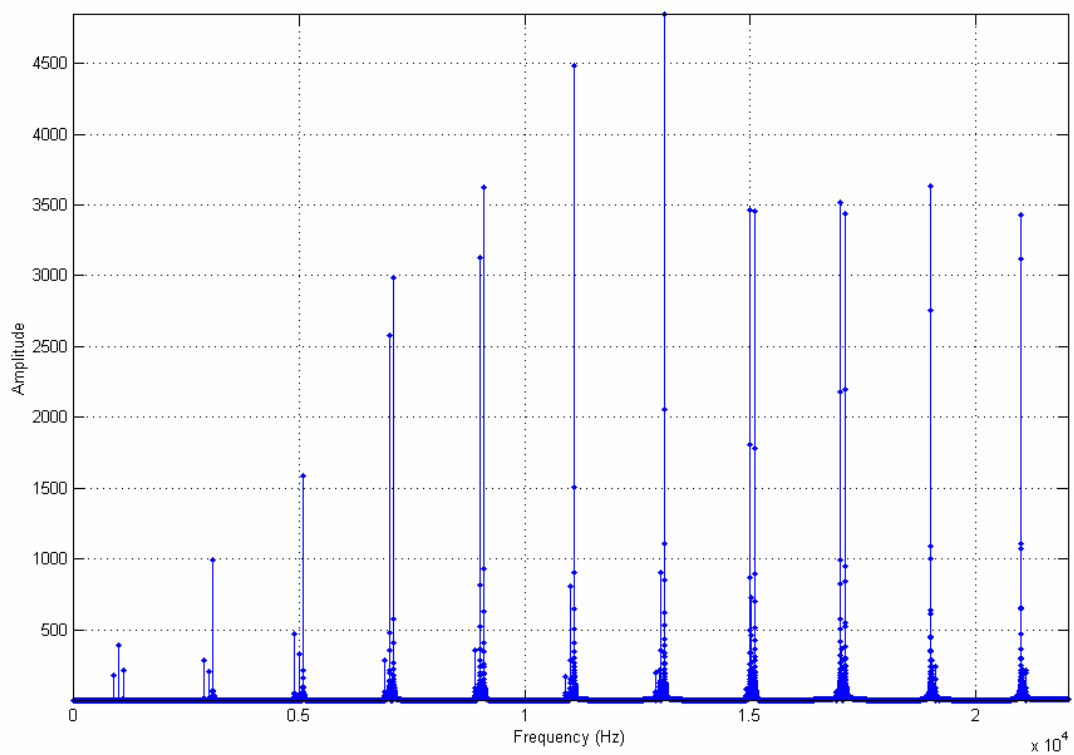
Frequenza modulante = 2000



Frequenza modulante = 2500



Frequenza modulante = 3000



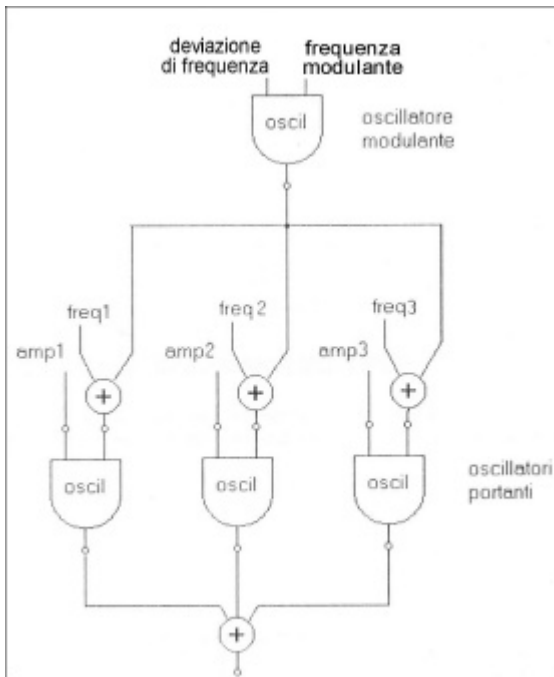
Frequenza modulante = 4000

Come si evince dai grafici, cambiando la frequenza modulante piuttosto che l'indice di modulazione, si hanno effetti differenti. Nel primo caso vengono create nuove componenti con frequenze diverse, mentre nel secondo caso si ha una maggiorazione di spettro nell'intorno della frequenza portante.

Inoltre il numero di componenti addizionali apparse non necessariamente cresce proporzionalmente alla frequenza modulante: la frequenza modulante di 2500 ha più componenti rispetto alla frequenza modulante pari a 1000, ma passando da una modulazione di frequenza di 3000 ad una di 4000 il numero di componenti decresce. Questo comportamento potrebbe sembrare casuale ma non lo è, infatti le modulazioni di frequenza di 1000, 2000, 3000 e 4000 sono multiple intere della frequenza portante mentre la frequenza modulante di 2500 non lo è. Quando la frequenza di modulazione utilizzata è un multiplo intero della frequenza portante, la frequenza modulante viene detta *formante*: in questo caso le componenti prodotte tendono ad essere armoniche rispetto alla frequenza portante. Le frequenze modulanti formanti tendono a produrre suoni che più si avvicinano a strumenti reali. Usando invece frequenze di modulazione piccole si produce un particolare effetto, il vibrato.

Generalmente si utilizzano segnali portanti multipli con diverse frequenze modulanti, piuttosto che un unico segnale portante (come riportato negli esempi per semplicità). Inoltre è possibile usare modulatori di frequenza multipli in modo da generare una grande varietà di armoniche arricchendo ulteriormente il segnale utilizzando tipologie diverse di segnale (ad esempio combinando segnali sinusoidali e segnali cosinusoidali).

La generazione di un suono mediante modulazione di frequenza è però solo il primo passo del processo di sintesi sonora: sebbene il suono prodotto sia appropriato per diverse applicazioni è necessario un ulteriore lavoro (applicazione di filtri e/o involuppi) per creare suoni che approssimano strumenti reali.



FM con portanti multiple: esempi pratici

Costruiamo ora uno strumento FM basato su un oscillatore che modula allo stesso tempo tre oscillatori portanti, ognuno dei quali ha un'ampiezza e una frequenza indipendente. Questo strumento genererà un suono di spettro molto più complesso rispetto ad uno strumento con una sola portante. L'oscillatore in alto genera l'unica modulante che viene sommata alle tre frequenze dei tre oscillatori portanti *freq1*, *freq2*, *freq3*.

Esempio 1:

```
srate = 44100;
t = linspace(0,5,5*srate);
```

% Sample rate
% Vettore tempo

```
deviation = 10;
mod_freq = 220;
```

% Deviazione di modulazione
% Frequenza di modulazione

% Genera il segnale modulante

```
modulator = oscil(deviation, mod_freq, t);
```

% Genera la prima portante

```
amp1 = 0.5;
freq1 = 110;
carrier1 = oscil(amp1, freq1 + modulator, t);
```

% Genera la seconda portante

```
amp2 = 0.5;
freq2 = 330;
carrier2 = oscil(amp2, freq2 + modulator, t);
```

% Genera la terza portante

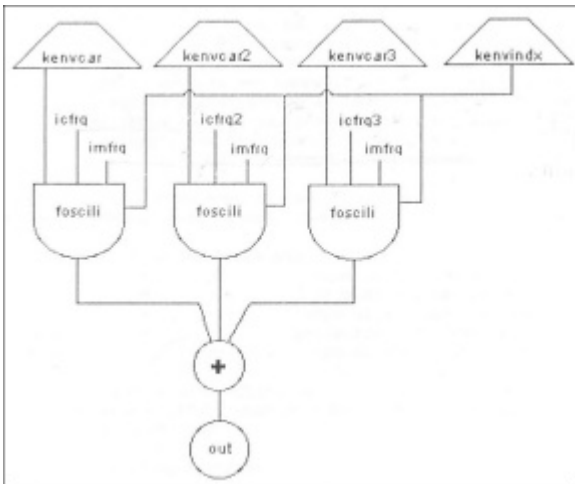
```
amp3 = 0.5;
freq3 = 440;
carrier3 = oscil(amp3, freq3 + modulator, t);
```

% Genera il suono, lo normalizza e lo suona.

```
snd = carrier1 + carrier2 + carrier3;
snd = normalize(snd,50);
sound(snd,srate);
```

% Livello di normalizzazione: 50%

Nome file sorgente: mul_car1.m
Nome file audio: mul_car1.wav



Implementiamo ora la seguente configurazione facendo uso dell'oscillatore foscili. In questo schema si utilizzano tre differenti involucri per le ampiezze di foscili (*kenvcar*, *kenvcar2*, *kenvcar3*) e di un involuipo per l'indice di modulazione, *kenvindx*.

Esempio 2:

```
srate = 44100;
t = linspace(0, 6, 6 * srate);
silence = zeros(1, 1*srate);                                     % Crea un secondo di silenzio

% Definizione degli involuipi d'ampiezza
kenvcar = [ linspace(0, 1, 3 * srate) linspace(1, 0, 3 * srate) ];
kenvcar2 = kenvcar;
kenvcar3 = kenvcar;

% Definizione dell'involuppo dell'indice di modulazione
kenvindx = [ linspace(0, 5, 3 * srate) linspace(5, 0, 3 * srate) ];

% Costruzione del primo suono
snd1_1 = foscili(kenvcar, 1, 100, 100, 5, t);
snd1_2 = foscili(kenvcar2, 1, 200, 100, 5, t);
snd1_3 = foscili(kenvcar3, 1, 300, 100, 5, t);
snd1 = snd1_1 + snd1_2 + snd1_3;

% Costruzione del secondo suono
snd2_1 = foscili(kenvcar, 1, 100, 113, 5, t);
snd2_2 = foscili(kenvcar2, 1, 258, 113, 5, t);
snd2_3 = foscili(kenvcar3, 1, 356, 113, 5, t);
snd2 = snd2_1 + snd2_2 + snd2_3;

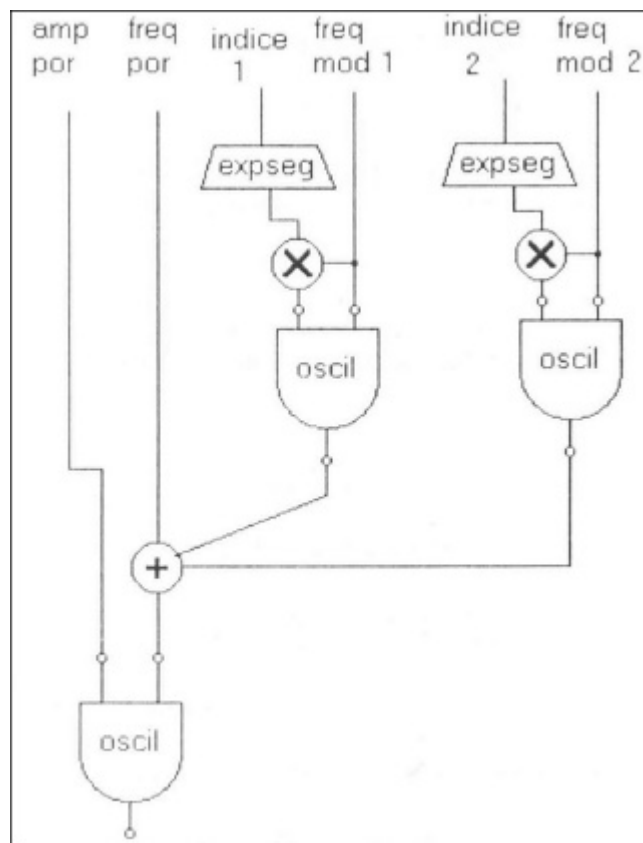
% Costruzione del terzo suono
snd3_1 = foscili(kenvcar, 1, 100, 107, 5, t);
snd3_2 = foscili(kenvcar2, 1, 111, 107, 5, t);
snd3_3 = foscili(kenvcar3, 1, 117, 107, 5, t);
snd3 = snd3_1 + snd3_2 + snd3_3;

snd = [ snd1 silence snd2 silence snd3 ];
snd = normalize(snd,50);
sound(snd,srate);
```

Nome file sorgente: mul_car2.m
Nome file audio: mul_car2.wav

FM con modulanti multiple

Costruiamo ora uno strumento FM complesso, basato su due oscillatori con inviluppi di indice di modulazione indipendente, che modulano un oscillatore portante, con un inviluppo d'ampiezza indipendente.



Esempio 3:

```
srate = 44100;
t = linspace( 0, 10, 10*srate);

% Primo oscillatore modulante
mod_freq = 100;
indmax = 0.8;
indatk = 2;
indperc = 0.8;
indrel = 1;

% Primo inviluppo di indice di modulazione
indenv_1 = exp(linspace(log(0.001),log(indmax),indatk*srate));
indenv_2 = exp(linspace(log(indmax),log(indmax*indperc),(10 - indatk - indrel)*srate));
indenv_3 = exp(linspace(log(indmax*indperc),log(0.001),indrel*srate));
indenv = [ indenv_1 indenv_2 indenv_3 ];

% Secondo oscillatore modulante
mod_freq2 = 125;
indmax2 = 1;
indatk2 = 2;
indperc2 = 0.6;
indrel2 = 2;

% Secondo inviluppo di indice di modulazione
indenv2_1 = exp(linspace(log(0.001),log(indmax2),indatk2*srate));
indenv2_2 = exp(linspace(log(indmax2),log(indmax2*indperc2),(10 - indatk2 -
indrel2)*srate));
indenv2_3 = exp(linspace(log(indmax2*indperc2),log(0.001),indrel2*srate));
indenv2 = [ indenv2_1 indenv2_2 indenv2_3 ];

% Oscillatore portante
amp = 1;
carr_freq = 200;
atk = 0.1;
perc = 0.9;
rel = 3;

% Inviluppo di ampiezza oscillatore portante
envc_1 = exp(linspace(log(0.001),log(amp),atk*srate));
envc_2 = exp(linspace(log(amp),log(amp*perc),(10 - atk - rel)*srate));
envc_3 = exp(linspace(log(amp*perc),log(0.001),rel*srate));
envc = [ envc_1 envc_2 envc_3 ];

mod1 = oscil((indenv*mod_freq/100), mod_freq, t);
mod2 = oscil((indenv2*mod_freq2/100), mod_freq2, t);
car = oscil(envc, carr_freq + mod1 + mod2,t);
car = normalize(car,50);
sound(car,srate);
```

Nome file sorgente: mul_mod.m
Nome file audio: mul_mod.wav

Capitolo 5

Esempio pratico: stesura di una partitura ed implementazione di uno strumento

Per ottenere qualsiasi tipo di suono Csound richiede la scrittura di due testi che vengono denominati rispettivamente *orchestra* e *partitura*. Il primo definisce le caratteristiche di uno o più strumenti, il secondo definisce le note e gli effetti che lo strumento suonerà.

E' possibile ricreare questo comportamento in Matlab:

- nel caso dell'orchestra sarà necessario scrivere una funzione Matlab, la quale implementerà le caratteristiche dello strumento.
- nel caso della partitura è possibile utilizzare una particolare struttura dati fornita da Matlab: l'array di celle.

In entrambi i casi l' estensione dei file è ".m", a differenza del Csound che prevede (obbligatoriamente in ambiente Windows) file ".orc" per le orchestre, e file ".sco" per le partiture.

La suddivisione in orchestre e partiture presenta il notevole vantaggio di fornire un supporto alla "programmazione acustica modulare": una partitura potrà essere suonata da una qualunque orchestra.

MATLAB: Array di celle

Possiamo considerare l'array di celle fornito da Matlab come una matrice di celle, contenitori di altre strutture dati Matlab. Ad esempio una cellula potrà contenere una matrice reale, un'altra un vettore di stringhe, ed un'altra un vettore di valori complessi.

Nota : l'array di celle non può avere posizioni vuote. Nel caso di partiture complesse questo può rappresentare uno svantaggio: ad esempio, applicando un effetto ad una singola nota, dichiarandone i parametri nella posizione riga-colonna corretta, il programmatore è costretto a riempire tutti gli spazi di quella colonna con determinati valori. Tuttavia l'estrema flessibilità della struttura la rende adatta alla scrittura di partitura.

Dichiarare un array di celle è molto semplice: è sufficiente assegnare ad un nome di variabile una serie di strutture separate da virgole e racchiuse da parentesi graffe. Per estendere l'array su più righe è sufficiente separarle con ";".

Ad esempio, l'assegnamento:

$C = \{ [1 \ 2], [3 \ 4]; [5 \ 6], [7 \ 8] \}$

crea la struttura

Cella 1,1 [1 2]	Cella 1,2 [3 4]
Cella 2,1 [5 6]	Cella 2,2 [7 8]

Seguono ora le partiture relative allo **Studio Op. 740 Num. 14** di **Carl Czerny**.


```
% File: czerny_left.m
% -----
% Realizzazione della
% partitura per lo
% Studio Op.740 Num.14
% di Carl Czerny (Mano
% Sinistra)
%
% • Nella prima
% colonna è
% rappresentato il
% tempo di inizio
% della nota;
% • Nella seconda
% colonna è
% rappresentato il
% tempo di durata
% della nota;
% • Nella terza
% colonna è
% rappresentata la
% frequenza (pitch)
% della nota.
% In questo caso sono
% state usate le macro
% predefinite (Scala
% Cromatica
% Temperata);
% • Nella quarta
% colonna è
% rappresentata
% l'ampiezza della
% nota.
% Si osservi che ogni
% riga rappresenta una
% nota. Note
% consecutive con lo
% stesso tempo di
% inizio e la stessa
% durata si
% sovrappongono. Per
% facilitare la
% lettura del listato
% sono stati aggiunti
% i numeri di
% sequenza: è
% possibile seguire le
% note sugli spartiti
% presenti in
% Appendice.

left_hand = {

% 1
0,    1.6,  sol2  1
0,    1.6,  sol1, 1

% 2
1.6,  1.6,  fad2, 1
1.6,  1.6,  fad1, 1
```

```
% 3
3.2,  1.6,  do3,  1
3.2,  1.6,  do2,  1

% 4
4.8,  1.6,  lad2,  1
4.8,  1.6,  lad1,  1

% 5
6.4,  1.6,  si2,  1
6.4,  1.6,  si1,  1

% 6
8,    1.6,  do3,  1
8,    1.6,  do2,  1

% 7
9.6,  1.6,  dod3,  1
9.6,  1.6,  dod2,  1

% 8
11.2, 1.6,  re3,  1
11.2, 1.6,  re2,  1

% 9
12.8, 1.6,  sol3,  1
12.8, 1.6,  re3,  1
12.8, 1.6,  sol2,  1

% 10
14.4, 1.6,  fad3,  1
14.4, 1.6,  re3,  1
14.4, 1.6,  fad1,  1

% 11
16,   1.6,  fa3,  1
16,   1.6,  re3,  1
16,   1.6,  sold2,1
16,   1.6,  fa2,  1

% 12
17.6, 1.6,  mi3,  1
17.6, 1.6,  do3,  1
17.6, 1.6,  sol2,  1
17.6, 1.6,  mi2,  1

% 13
19.2, 1.6,  red3,  1

19.2, 1.6,  do3,  1
19.2, 1.6,  sol2,  1
19.2, 1.6,  red2,  1

% 14
20.8, 0.8,  re3,  1
20.8, 0.8,  la2,  1
20.8, 0.8,  fad2,  1
20.8, 0.8,  re2,  1
```

```
21.6, 0.8,  dod3,  1
21.6, 0.8,  lad2,  1
21.6, 0.8,  sol2,  1
21.6, 0.8,  dod2,  1

% 15
22.4, 0.8,  re3,  1
22.4, 0.8,  lad2,  1
22.4, 0.8,  sol2,  1
22.4, 0.8,  re2,  1

23.2, 0.8,  re3,  1
23.2, 0.8,  lad2,  1
23.2, 0.8,  fad2,  1
23.2, 0.8,  re2,  1

% 16
24,   0.8,  sol3,  0.84
24,   0.8,  re3,  0.84
24,   0.8,  lad2,  0.84
24,   0.8,  sol2,  0.84

24.8, 0.4,  sol2,  0.84
24.8, 0.4,  sol1,  0.84

25.2, 0.4,  la2,   0.80
25.2, 0.4,  la1,   0.80

% 17
25.6, 0.4,  lad2,  0.64
25.6, 0.4,  lad1,  0.64

26,   0.4,  fa3,   0.64

26.4, 0.4,  re4,   0.64
26.4, 0.4,  lad3,  0.64

26.8, 0.4,  fa3,   0.64

% 18
27.2, 0.4,  fa2,   0.64
27.2, 0.4,  fa1,   0.64

27.6, 0.4,  fa3,   0.64

28,   0.4,  fa4,   0.64
28,   0.4,  do4,   0.64
28,   0.4,  la3,   0.64

28.4, 0.4,  fa3,   0.64

% 19
28.8, 0.4,  do3,   0.64
28.8, 0.4,  do2,   0.64

29.2, 0.4,  sol3,  0.64

29.6, 0.4,  red4,  0.64
29.6, 0.4,  do4,   0.64
```

30, 0.4, sol3, 0.64

% 20

30.4, 0.4, sol2, 0.64
30.4, 0.4, sol1, 0.64
30.8, 0.4, sol3, 0.64

31.2, 0.4, sol3, 0.64
31.2, 0.4, sol4, 0.64
31.2, 0.4, lad3, 0.64

31.6, 0.4, sol3, 0.64

% 21

32, 0.4, red3, 0.64
32, 0.4, red2, 0.64

32.4, 0.4, lad3, 0.64

32.8, 0.4, sol4, 0.64
32.8, 0.4, red4, 0.64

33.2, 0.4, lad3, 0.64

% 22

33.6, 0.4, re3, 0.64
33.6, 0.4, re2, 0.64

34, 0.4, fa4, 0.64
34, 0.4, re4, 0.64
34, 0.4, lad3, 0.64

34.4, 0.4, red3, 0.64
34.4, 0.4, red2, 0.64

34.8, 0.4, red4, 0.64
34.8, 0.4, do4, 0.64
34.8, 0.4, sol3, 0.64

% 23

35.2, 0.4, fa3, 0.64
35.2, 0.4, fa2, 0.64

35.6, 0.4, re4, 0.64
35.6, 0.4, lad3, 0.64
35.6, 0.4, fa3, 0.64

36, 0.4, fa3, 0.64
36, 0.4, fa2, 0.64

36.4, 0.4, do4, 0.64

36.4, 0.4, la3, 0.64
36.4, 0.4, fa3, 0.64

% 24

36.8, 0.1, lad2, 0.64

36.9, 0.3, 0, 0

37.2, 0.1, re4, 0.64
37.2, 0.1, lad3, 0.64
37.2, 0.1, fa3, 0.64

37.3, 0.3, 0, 0

37.6, 0.1, lad2, 0.64

37.7, 0.3, 0, 0

38, 0.1, re4, 0.64
38, 0.1, lad3, 0.64
38, 0.1, fa3, 0.64

38.1, 0.3, 0, 0

% 25

38.4, 0.1, la2, 0.64

38.5, 0.3, 0, 0

38.8, 0.1, red4, 0.64
38.8, 0.1, do4, 0.64
38.8, 0.1, fa3, 0.64

38.9, 0.3, 0, 0

39.2, 0.1, fa3, 0.64
39.2, 0.1, fa2, 0.64

39.3, 0.3, 0, 0

39.6, 0.1, red4, 0.64
39.6, 0.1, do4, 0.64
39.6, 0.1, fa3, 0.64

39.7, 0.3, 0, 0

% 26

40, 0.1, lad2, 0.64

40.1, 0.3, 0, 0

40.4, 0.1, re4, 0.64
40.4, 0.1, lad3, 0.64
40.4, 0.1, fa3, 0.64

40.5, 0.3, 0, 0

40.8, 0.1, lad2, 0.64

40.9, 0.3, 0, 0

41.2, 0.1, re4, 0.64
41.2, 0.1, lad3, 0.64
41.2, 0.1, fa3, 0.64

41.3, 0.3, 0, 0

% 27

41.6, 0.1, la2, 0.64

41.7, 0.3, 0, 0

42, 0.1, red4, 0.64
42, 0.1, do4, 0.64
42, 0.1, fa3, 0.64

42.1, 0.3, 0, 0

42.4, 0.1, fa3, 0.64
42.4, 0.1, fa2, 0.64

42.5, 0.3, 0, 0

42.8, 0.1, red4, 0.64
42.8, 0.1, do4, 0.64
42.8, 0.1, fa3, 0.64

42.9, 0.3, 0, 0

% 28

43.2, 0.8, lad2, 0.64
43.2, 0.8, lad1, 0.64

44, 0.8, sold2, 0.64
44, 0.8, sold2, 0.64

% 29

44.8, 1.6, sol2, 0.64
44.8, 1.6, sol1, 0.64

% 30

46.4, 0.8, do3, 0.64
46.4, 0.8, do2, 0.64

47.2, 0.8, lad2, 0.64
47.2, 0.8, lad1, 0.64

% 31

48, 1.6, la2, 0.64
48, 1.6, la1, 0.64

% 32

49.6, 0.8, re3, 0.64
49.6, 0.8, re2, 0.64

50.4, 0.8, do3, 0.64
50.4, 0.8, do2, 0.64

% 33

51.2, 0.8, lad2, 0.64
51.2, 0.8, lad1, 0.64

52, 0.8, sol2, 0.64
52, 0.8, sol1, 0.64

% 34 - 35

52.8, 2.0, dod3, 1.01
52.8, 2.0, lad2, 1.01
52.8, 2.0, sol2, 1.01
52.8, 2.0, dod2, 1.01

54.8, 0.4, dod3, 0.64
54.8, 0.4, dod2, 0.64

55.2, 0.4, dod3, 0.64
55.2, 0.4, dod2, 0.64

55.6, 0.4, dod3, 0.64

55.6, 0.4, dod2, 0.64
56, 0.1, re2, 0.84
56.1, 0.1, 0, 0
56.2, 0.1, fad2, 0.84
56.3, 0.1, 0, 0
% 36
56.4, 0.1, la2, 0.84
56.5, 0.1, 0, 0
56.6, 0.1, do3, 0.84
56.7, 0.1, 0, 0
56.8, 0.1, sol2, 0.84
56.8, 0.1, sol1, 0.84
56.9, 0.1, 0, 0
57, 0.1, lad2, 0.84
57.1, 0.1, 0, 0
57.2, 0.1, re3, 0.84
57.3, 0.1, 0, 0
57.4, 0.1, sol3, 0.84
57.5, 0.1, 0, 0
% 37
57.6, 0.1, re3, 0.84
57.6, 0.1, re2, 0.84
57.7, 0.1, 0, 0
57.8, 0.1, fad3, 0.84
57.9, 0.1, 0, 0
58, 0.1, la3, 0.84
58.1, 0.1, 0, 0
58.2, 0.1, do4, 0.84
58.3, 0.1, 0, 0
58.4, 0.1, sol3, 0.84
58.4, 0.1, sol2, 0.84
58.5, 0.1, 0, 0
58.6, 0.1, lad3, 0.84
58.7, 0.1, 0, 0

58.8, 0.1, re4, 0.84
58.9, 0.1, 0, 0
59, 0.1, sol4, 0.84
59.1, 0.1, 0, 0
% 38
59.2, 0.1, re4, 0.84
59.2, 0.1, re3, 0.84
59.3, 0.1, 0, 0
59.4, 0.1, fad4, 0.84
59.5, 0.1, 0, 0
59.6, 0.1, la4, 0.84
59.7, 0.1, 0, 0
59.8, 0.1, do5, 0.84
59.9, 0.1, 0, 0
60, 0.1, sol4, 0.84
60, 0.1, sol3, 0.84
60.1, 0.1, 0, 0
60.2, 0.1, lad4, 0.84
60.3, 0.1, 0, 0
60.4, 0.1, re5, 0.84
60.5, 0.1, 0, 0
60.6, 0.1, sol5, 0.84
60.7, 0.1, 0, 0
% 39
60.8, 0.4, re5, 1
60.8, 0.4, la4, 1
60.8, 0.4, fad4, 1
60.8, 0.4, re4, 1
61.2, 1.3, 0, 0
% 40
62.5, 1.6, sol2, 0.84
62.5, 1.6, sol1, 0.84
% 41
64.1, 1.6, fad2, 0.84
64.1, 1.6, fad1, 0.84
% 42
65.7, 1.6, do3, 0.84
65.7, 1.6, do2, 0.84

% 43
67.3, 1.6, lad2, 0.84
67.3, 1.6, lad1, 0.84
% 44
68.9, 1.6, si2, 0.84
68.9, 1.6, si1, 0.84
% 45
70.5, 1.6, do3, 0.84
70.5, 1.6, do2, 0.84
% 46
72.1, 1.6, dod3, 0.84
72.1, 1.6, dod2, 0.84
% 47
73.7, 1.6, re3, 0.84
73.7, 1.6, re2, 0.84
% 49
75.3, 1.6, sol3, 0.84
75.3, 1.6, re3, 0.84
75.3, 1.6, sol2, 0.84
% 49
76.9, 1.6, fad3, 0.84
76.9, 1.6, re3, 0.84
76.9, 1.6, fad2, 0.84
% 50
78.5, 1.6, fad3, 0.84
78.5, 1.6, re3, 0.84
78.5, 1.6, sold2, 0.84
78.5, 1.6, fa2, 0.84
% 51
80.1, 1.6, mi3, 0.84
80.1, 1.6, do3, 0.84
80.1, 1.6, sol2, 0.84
80.1, 1.6, mi2, 0.84
% 52
81.7, 1.6, red3, 0.84
81.7, 1.6, do3, 0.84
81.7, 1.6, sol2, 0.84
81.7, 1.6, red2, 0.84
% 53
83.3, 0.8, re3, 0.84
83.3, 0.8, la2, 0.84
83.3, 0.8, fad2, 0.84
83.3, 0.8, re2, 0.84
84.1, 0.8, dod3, 0.84
84.1, 0.8, lad2, 0.84
84.1, 0.8, sol2, 0.84
84.1, 0.8, dod2, 0.84
% 54
84.9, 0.8, re3, 0.84
84.9, 0.8, lad2, 0.84
84.9, 0.8, sol2, 0.84

84.9, 0.8, re2, 0.84

85.7, 0.8, re3, 0.84
85.7, 0.8, la2, 0.84
85.7, 0.8, fad2, 0.84
85.7, 0.8, re2, 0.84

% 55
86.5, 0.1, sol3, 1
86.5, 0.1, re3, 1
86.5, 0.1, sol2, 1

86.6, 0.3, 0, 0

86.9, 0.4, sol2, 0.84
86.9, 0.4, sol1, 0.84

87.3, 0.4, la2, 0.84
87.3, 0.4, la1, 0.84

87.7, 0.4, si2, 0.84
87.7, 0.4, si1, 0.84

% 56
88.1, 0.4, do3, 0.84
88.1, 0.4, do2, 0.84

88.5, 0.4, re3, 0.84
88.5, 0.4, re2, 0.84

88.9, 0.4, mi3, 0.84
88.9, 0.4, mi2, 0.84

89.3, 0.4, fad3, 0.84
89.3, 0.4, fad2, 0.84

% 57
89.7, 0.4, sol3, 0.84
89.7, 0.4, sol2, 0.84

90.1, 0.4, red3, 0.84
90.1, 0.4, red2, 0.84

90.5, 0.4, do3, 0.84
90.5, 0.4, do2, 0.84

90.9, 0.4, dod3, 0.84
90.9, 0.4, dod3, 0.84

% 58
91.3, 0.1, re3, 1
91.3, 0.1, re2, 1

91.4, 0.3, 0, 0

91.7, 0.1, lad3, 1

91.7, 0.1, sol3, 1
91.7, 0.1, re3, 1

91.8, 0.3, 0, 0

92.1, 0.1, re3, 1
92.1, 0.1, re2, 1

92.2, 0.3, 0, 0

92.5, 0.1, do4, 0.84
92.5, 0.1, la3, 0.84
92.5, 0.1, fad3, 0.84
92.5, 0.1, re3, 0.84

92.6, 0.3, 0, 0

% 59
92.9, 0.1, sol3, 0.84
92.9, 0.1, re3, 0.84
92.9, 0.1, sol2, 0.84

93, 0.3, 0, 0

93.3, 0.4, sol2, 0.84
93.3, 0.4, sol1, 0.84

93.7, 0.4, la2, 0.84
93.7, 0.4, la1, 0.84

94.1, 0.4, si2, 0.84
94.1, 0.4, si1, 0.84

% 60
94.5, 0.4, do3, 0.84
94.5, 0.4, do2, 0.84

94.9, 0.4, re3, 0.84
94.9, 0.4, re2, 0.84

95.3, 0.4, mi3, 0.84
95.3, 0.4, mi2, 0.84

95.7, 0.4, fad3, 0.84
95.7, 0.4, fad2, 0.84

% 61
96.1, 0.4, sol3, 0.84
96.1, 0.4, sol2, 0.84

96.5, 0.4, red3, 0.84
96.5, 0.4, red2, 0.84

96.9, 0.4, do3, 0.84
96.9, 0.4, do2, 0.84

97.3, 0.4, dod3, 0.84
97.3, 0.4, dod2, 0.84

% 62
97.7, 0.1, re3, 1
97.7, 0.1, re2, 1

97.8, 0.3, 0, 0

98.1, 0.1, lad3, 1
98.1, 0.1, sol3, 1

98.1, 0.1, re3, 1

98.2, 0.3, 0, 0

98.5, 0.1, re3, 1
98.5, 0.1, re2, 1
98.6, 0.3, 0, 0

98.9, 0.1, do4, 0.84
98.9, 0.1, la3, 0.84
98.9, 0.1, fad3, 0.84
98.9, 0.1, re3, 0.84

99, 0.3, 0, 0

% 63
99.3, 0.4, sol3, 0.84
99.3, 0.4, sol2, 0.84

99.7, 0.4, lad3, 0.84
99.7, 0.4, sol3, 0.84
99.7, 0.4, re3, 0.84

100.1, 0.4, re3, 0.84
100.1, 0.4, re2, 0.84

100.5, 0.4, do4, 0.84
100.5, 0.4, la3, 0.84
100.5, 0.4, fad3, 0.84
100.5, 0.4, re3, 0.84

% 64

100.9, 0.4, sol3, 0.84
100.9, 0.4, sol2, 0.84

101.3, 0.4, lad3, 0.84
101.3, 0.4, sol3, 0.84
101.3, 0.4, re3, 0.84

101.7, 0.4, re3, 0.84
101.7, 0.4, re2, 0.84

102.1, 0.4, do4, 0.84
102.1, 0.4, la3, 0.84
102.1, 0.4, fad3, 0.84
102.1, 0.4, re3, 0.84

% 65
102.5, 1.5, sol3, 1
102.5, 1.5, re3, 1
102.5, 1.5, lad2, 1
102.5, 1.5, sol2, 1

104, 0.1, 0, 0

% 66
104.1, 0.2, 0, 0

104.3, 0.1, sol2, 0.84

104.4, 0.1, 0, 0

```

104.5,0.1, re2, 0.84
104.6,0.1, 0, 0
104.7,0.1, lad1, 0.84
104.8,0.1, 0, 0
104.9,0.1, sol1, 0.84
105, 0.1, 0, 0
105.1,0.2, 0, 0
105.3,0.2, sol3, 1
105.3,0.2, re3, 1
105.3,0.2, lad2, 1
105.3,0.2, sol2, 1
105.5,0.2, 0, 0

% 67
105.7,1.6, sol2, 1
105.7,1.6, sol1, 1
% 68
107.3,1.6, sol2, 1
107.3,1.6, sol1, 1

% 69
108.9,0.8, sol2, 1
108.9,0.8, sol2, 1
};

```

```

% File: czerny_right.m
% -----
% Realizzazione della
% partitura per lo
% Studio Op.740 Num.14
% di Carl Czerny (Mano
% Destra)
%
% • Nella prima
% colonna è
% rappresentato il
% tempo di inizio
% della nota;
% • Nella seconda
% colonna è
% rappresentato il
% tempo di durata
% della nota;
% • Nella terza
% colonna è
% rappresentata la
% frequenza (pitch)
% della nota.
% In questo caso sono
% state usate le macro
% predefinite (Scala
% Cromatica
% Temperata);
% • Nella quarta
% colonna è
% rappresentata
% l'ampiezza della
% nota.
%
% Si osservi che ogni
% riga rappresenta una
% nota. Note
% consecutive con lo
% stesso tempo di
% inizio e la stessa
% durata si
% sovrappongono. Per
% facilitare la
% lettura del listato
% sono stati aggiunti
% i numeri di
% sequenza: è
% possibile seguire le
% note sugli spartiti
% presenti in
% Appendice.

right_hand = {

% 1
0, 0.1, sol3, 0.84
0.1, 0.1, re3, 0.84
0.2, 0.1, lad3, 0.84
0.3, 0.1, sol3, 0.84
0.4, 0.1, re4, 0.84
0.5, 0.1, lad3, 0.84
0.6, 0.1, sol4, 0.84
0.7, 0.1, re4, 0.84

0.8, 0.1, lad4, 0.84
0.9, 0.1, sol4, 0.84
1, 0.1, re5, 0.84
1.1, 0.1, lad4, 0.84
1.2, 0.1, sol5, 0.84
1.3, 0.1, re5, 0.84
1.4, 0.1, lad5, 0.84
1.5, 0.1, sol5, 0.84

% 2
1.6, 0.1, re6, 0.84
1.7, 0.1, la5, 0.84
1.8, 0.1, fad5, 0.84
1.9, 0.1, la5, 0.84
2, 0.1, re5, 0.84
2.1, 0.1, fad5, 0.84
2.2, 0.1, la4, 0.84
2.3, 0.1, re5, 0.84
2.4, 0.1, fad4, 0.84
2.5, 0.1, la4, 0.84
2.6, 0.1, re4, 0.84
2.7, 0.1, fad4, 0.84
2.8, 0.1, la3, 0.84
2.9, 0.1, re4, 0.84
3, 0.1, fad3, 0.84
3.1, 0.1, la3, 0.84

% 3
3.2, 0.1, re4, 0.84
3.3, 0.1, la3, 0.84
3.4, 0.1, fad4, 0.84
3.5, 0.1, re4, 0.84
3.6, 0.1, la4, 0.84
3.7, 0.1, fad4, 0.84
3.8, 0.1, re5, 0.84
3.9, 0.1, la4, 0.84
4, 0.1, fad5, 0.84
4.1, 0.1, re5, 0.84
4.2, 0.1, la5, 0.84
4.3, 0.1, fad5, 0.84
4.4, 0.1, re6, 0.84
4.5, 0.1, la5, 0.84
4.6, 0.1, fad6, 0.84
4.7, 0.1, re6, 0.84

% 4
4.8, 0.1, sol6, 0.84
4.9, 0.1, re6, 0.84
5, 0.1, lad5, 0.84
5.1, 0.1, re6, 0.84
5.2, 0.1, sol5, 0.84
5.3, 0.1, lad5, 0.84
5.4, 0.1, re5, 0.84
5.5, 0.1, sol5, 0.84
5.6, 0.1, lad4, 0.84
5.7, 0.1, re5, 0.84
5.8, 0.1, sol4, 0.84
5.9, 0.1, lad4, 0.84
6, 0.1, re4, 0.84
6.1, 0.1, sol4, 0.84
6.2, 0.1, sol3, 0.84
6.3, 0.1, re4, 0.84

% 5
6.4, 0.1, fa4, 0.84
6.5, 0.1, re4, 0.84
6.6, 0.1, sold4, 0.84
6.7, 0.1, fa4, 0.84
6.8, 0.1, re5, 0.84
6.9, 0.1, sold4, 0.84
7, 0.1, fa5, 0.84
7.1, 0.1, re5, 0.84
7.2, 0.1, sold5, 0.84
7.3, 0.1, fa5, 0.84
7.4, 0.1, re6, 0.84
7.5, 0.1, sold5, 0.84
7.6, 0.1, fa6, 0.84
7.7, 0.1, re6, 0.84
7.8, 0.1, sold6, 0.84
7.9, 0.1, fa6, 0.84

% 6
8, 0.1, red6, 0.84
8.1, 0.1, sol6, 0.84
8.2, 0.1, do6, 0.84
8.3, 0.1, red6, 0.84
8.4, 0.1, sol5, 0.84
8.5, 0.1, do6, 0.84
8.6, 0.1, red5, 0.84
8.7, 0.1, sol5, 0.84
8.8, 0.1, do5, 0.84
8.9, 0.1, red5, 0.84
9, 0.1, sol4, 0.84
9.1, 0.1, do5, 0.84
9.2, 0.1, red4, 0.84
9.3, 0.1, sol4, 0.84
9.4, 0.1, do4, 0.84
9.5, 0.1, red4, 0.84

% 7
9.6, 0.1, mi4, 0.84
9.7, 0.1, sol4, 0.84
9.8, 0.1, lad4, 0.84
9.9, 0.1, sol4, 0.84
10, 0.1, mi5, 0.84
10.1, 0.1, lad4, 0.84
10.2, 0.1, sol5, 0.84
10.3, 0.1, mi5, 0.84
10.4, 0.1, lad5, 0.84
10.5, 0.1, sol5, 0.84
10.6, 0.1, mi6, 0.84
10.7, 0.1, lad5, 0.84
10.8, 0.1, sol6, 0.84
10.9, 0.1, mi6, 0.84
11, 0.1, lad6, 0.84
11.1, 0.1, sol6, 0.84

% 8
11.2, 0.1, fad6, 0.84
11.3, 0.1, la6, 0.84
11.4, 0.1, re7, 0.84
11.5, 0.1, re6, 0.84
11.6, 0.1, la5, 0.84
11.7, 0.1, fad5, 0.84
11.8, 0.1, re6, 0.84

```

11.9, 0.1, re5, 0.84
 12, 0.1, la4, 0.84
 12.1, 0.1, fad4, 0.84
 12.2, 0.1, re5, 0.84
 12.3, 0.1, re4, 0.84
 12.4, 0.1, do4, 0.84
 12.5, 0.1, re4, 0.84
 12.6, 0.1, la3, 0.84
 12.7, 0.1, re4, 0.84

% 9

12.8, 0.1, lad3, 0.84
 12.9, 0.1, re4, 0.84
 13, 0.1, sol4, 0.84
 13.1, 0.1, lad4, 0.84
 13.2, 0.1, re5, 0.84
 13.3, 0.1, lad4, 0.84
 13.4, 0.1, sol5, 0.84
 13.5, 0.1, re5, 0.84
 13.6, 0.1, lad5, 0.84
 13.7, 0.1, sol5, 0.84
 13.8, 0.1, re6, 0.84
 13.9, 0.1, lad5, 0.84
 14, 0.1, sol6, 0.84
 14.1, 0.1, re6, 0.84
 14.2, 0.1, lad6, 0.84
 14.3, 0.1, sol6, 0.84

% 10

14.4, 0.1, re7, 0.84
 14.5, 0.1, la6, 0.84
 14.6, 0.1, fad6, 0.84
 14.7, 0.1, re6, 0.84
 14.8, 0.1, la5, 0.84
 14.9, 0.1, re6, 0.84
 15, 0.1, fad5, 0.84
 15.1, 0.1, la5, 0.84
 15.2, 0.1, re5, 0.84
 15.3, 0.1, fad5, 0.84
 15.4, 0.1, la4, 0.84
 15.5, 0.1, re5, 0.84
 15.6, 0.1, fad4, 0.84
 15.7, 0.1, la4, 0.84
 15.8, 0.1, re4, 0.84
 15.9, 0.1, la4, 0.84

% 11

16, 0.1, si4, 0.84
 16.1, 0.1, sold4, 0.84
 16.2, 0.1, re5, 0.84
 16.3, 0.1, si4, 0.84
 16.4, 0.1, sold5, 0.84
 16.5, 0.1, re5, 0.84
 16.6, 0.1, si5, 0.84
 16.7, 0.1, sold5, 0.84
 16.8, 0.1, re6, 0.84
 16.9, 0.1, si5, 0.84
 17, 0.1, sold6, 0.84
 17.1, 0.1, re6, 0.84
 17.2, 0.1, si6, 0.84
 17.3, 0.1, sold6, 0.84
 17.4, 0.1, re7, 0.84
 17.5, 0.1, si6, 0.84

% 12

17.6, 0.1, do7, 0.84
 17.7, 0.1, sol6, 0.84
 17.8, 0.1, mi6, 0.84
 17.9, 0.1, sol6, 0.84
 18, 0.1, do6, 0.84
 18.1, 0.1, mi6, 0.84
 18.2, 0.1, sol5, 0.84
 18.3, 0.1, do6, 0.84
 18.4, 0.1, mi5, 0.84
 18.5, 0.1, sol5, 0.84
 18.6, 0.1, do5, 0.84
 18.7, 0.1, mi5, 0.84
 18.8, 0.1, sol4, 0.84
 18.9, 0.1, do5, 0.84
 19, 0.1, do4, 0.84
 19.1, 0.1, sol4, 0.84

% 13

19.2, 0.1, do5, 0.84
 19.3, 0.1, sol4, 0.84
 19.4, 0.1, do5, 0.84
 19.5, 0.1, sol4, 0.84
 19.6, 0.1, red5, 0.84
 19.7, 0.1, do5, 0.84
 19.8, 0.1, sol5, 0.84
 19.9, 0.1, red5, 0.84
 20.0, 0.1, do6, 0.84
 20.1, 0.1, sol5, 0.84
 20.2, 0.1, red6, 0.84
 20.3, 0.1, do6, 0.84
 20.4, 0.1, sol6, 0.84
 20.5, 0.1, red6, 0.84
 20.6, 0.1, do7, 0.84
 20.7, 0.1, sol6, 0.84

% 14

20.8, 0.1, do7, 0.84
 20.9, 0.1, la6, 0.84
 21.0, 0.1, fad6, 0.84
 21.1, 0.1, la6, 0.84
 21.2, 0.1, do6, 0.84
 21.3, 0.1, fad6, 0.84
 21.4, 0.1, la5, 0.84
 21.5, 0.1, do6, 0.84
 21.6, 0.1, lad6, 0.84
 21.7, 0.1, sol6, 0.84
 21.8, 0.1, mi6, 0.84
 21.9, 0.1, sol6, 0.84
 22.0, 0.1, lad5, 0.84
 22.1, 0.1, mi6, 0.84
 22.2, 0.1, sol5, 0.84
 22.3, 0.1, lad5, 0.84

% 15

22.4, 0.1, re6, 0.84
 22.5, 0.1, lad5, 0.84
 22.6, 0.1, sol5, 0.84
 22.7, 0.1, lad5, 0.84
 22.8, 0.1, re5, 0.84
 22.9, 0.1, sol5, 0.84
 23.0, 0.1, lad4, 0.84
 23.1, 0.1, re5, 0.84

23.2, 0.1, fad5, 0.84
 23.3, 0.1, re5, 0.84
 23.4, 0.1, do5, 0.84
 23.5, 0.1, re5, 0.84
 23.6, 0.1, la4, 0.84
 23.7, 0.1, do5, 0.84
 23.8, 0.1, fad4, 0.84
 23.9, 0.1, la4, 0.84

% 16

24.0, 0.1, lad4, 0.84
 24.1, 0.1, re5, 0.84
 24.2, 0.1, sol5, 0.84
 24.3, 0.1, re5, 0.84
 24.4, 0.1, lad4, 0.84
 24.5, 0.1, sol4, 0.84
 24.6, 0.1, re4, 0.84
 24.7, 0.1, lad3, 0.84
 24.8, 0.1, sol3, 0.84
 24.9, 0.1, lad3, 0.84
 25.0, 0.1, re4, 0.84
 25.1, 0.1, sol4, 0.84
 25.2, 0.1, fa3, 0.84
 25.3, 0.1, do4, 0.84
 25.4, 0.1, red4, 0.84
 25.5, 0.1, fa4, 0.84

% 17

25.6, 0.1, re4, 0.84
 25.7, 0.1, lad3, 0.84
 25.8, 0.1, fa4, 0.84
 25.9, 0.1, re4, 0.84
 26.0, 0.1, lad4, 0.84
 26.1, 0.1, fa4, 0.84
 26.2, 0.1, re5, 0.84
 26.3, 0.1, lad4, 0.84
 26.4, 0.1, fa5, 0.84
 26.5, 0.1, re5, 0.84
 26.6, 0.1, lad5, 0.84
 26.7, 0.1, fa5, 0.84
 26.8, 0.1, red6, 0.84
 26.9, 0.1, lad5, 0.84
 27.0, 0.1, fa6, 0.84
 27.1, 0.1, re6, 0.84

% 18

27.2, 0.1, do6, 0.84
 27.3, 0.1, la5, 0.84
 27.4, 0.1, fa6, 0.84
 27.5, 0.1, do6, 0.84
 27.6, 0.1, la6, 0.84
 27.7, 0.1, fa6, 0.84
 27.8, 0.1, do7, 0.84
 27.9, 0.1, la6, 0.84
 28.0, 0.1, fa6, 0.84
 28.1, 0.1, do6, 0.84
 28.2, 0.1, la5, 0.84
 28.3, 0.1, fa5, 0.84
 28.4, 0.1, do5, 0.84
 28.5, 0.1, la4, 0.84
 28.6, 0.1, fa4, 0.84
 28.7, 0.1, do4, 0.84

% 19

28.8, 0.1, red4, 0.84
 28.9, 0.1, do4, 0.84
 29.0, 0.1, sol4, 0.84
 29.1, 0.1, red4, 0.84
 29.2, 0.1, do5, 0.84
 29.3, 0.1, sol4, 0.84
 29.4, 0.1, red5, 0.84
 29.5, 0.1, do5, 0.84
 29.6, 0.1, sol5, 0.84
 29.7, 0.1, red5, 0.84
 29.8, 0.1, do6, 0.84
 29.9, 0.1, sol5, 0.84
 30.0, 0.1, red6, 0.84
 30.1, 0.1, do6, 0.84
 30.2, 0.1, sol6, 0.84
 30.3, 0.1, red6, 0.84

% 20

30.4, 0.1, re6, 0.84
 30.5, 0.1, lad5, 0.84
 30.6, 0.1, sol6, 0.84
 30.7, 0.1, re6, 0.84
 30.8, 0.1, lad6, 0.84
 30.9, 0.1, sol6, 0.84
 31, 0.1, re7, 0.84
 31.1, 0.1, lad6, 0.84
 31.2, 0.1, sol6, 0.84
 31.3, 0.1, re6, 0.84
 31.4, 0.1, lad5, 0.84
 31.5, 0.1, sol5, 0.84
 31.6, 0.1, re5, 0.84
 31.7, 0.1, lad4, 0.84
 31.8, 0.1, sol4, 0.84
 31.9, 0.1, re4, 0.84

% 21

32, 0.1, sol4, 0.84
 32.1, 0.1, red4, 0.84
 32.2, 0.1, lad4, 0.84
 32.3, 0.1, sol4, 0.84
 32.4, 0.1, red5, 0.84
 32.5, 0.1, lad4, 0.84
 32.6, 0.1, sol5, 0.84
 32.7, 0.1, red5, 0.84
 32.8, 0.1, lad5, 0.84
 32.9, 0.1, sol5, 0.84
 33, 0.1, red6, 0.84
 33.1, 0.1, lad5, 0.84
 33.2, 0.1, sol6, 0.84
 33.3, 0.1, red6, 0.84
 33.4, 0.1, lad6, 0.84
 33.5, 0.1, sol6, 0.84

% 22

33.6, 0.1, fa6, 0.84
 33.7, 0.1, fa7, 0.84
 33.8, 0.1, lad6, 0.84
 33.9, 0.1, re7, 0.84
 34, 0.1, fa6, 0.84
 34.1, 0.1, lad6, 0.84
 34.2, 0.1, re6, 0.84
 34.3, 0.1, fa6, 0.84

34.4, 0.1, red6, 0.84
 34.5, 0.1, red7, 0.84
 34.6, 0.1, sol6, 0.84
 34.7, 0.1, do7, 0.84
 34.8, 0.1, red6, 0.84
 34.9, 0.1, sol6, 0.84
 35, 0.1, do6, 0.84
 35.1, 0.1, red6, 0.84

% 23

35.2, 0.1, re6, 0.84
 35.3, 0.1, re7, 0.84
 35.4, 0.1, fa6, 0.84
 35.5, 0.1, lad6, 0.84
 35.6, 0.1, re6, 0.84
 35.7, 0.1, fa6, 0.84
 35.8, 0.1, lad5, 0.84
 35.9, 0.1, re6, 0.84

36, 0.1, do6, 0.84
 36.1, 0.1, do7, 0.84
 36.2, 0.1, red6, 0.84
 36.3, 0.1, la6, 0.84
 36.4, 0.1, do6, 0.84
 36.5, 0.1, red6, 0.84
 36.6, 0.1, la5, 0.84
 36.7, 0.1, do6, 0.84

% 24

36.8, 0.1, lad5, 0.84
 36.9, 0.1, re6, 0.84
 37, 0.1, fa6, 0.84
 37.1, 0.1, fa5, 0.84
 37.2, 0.1, lad4, 0.84
 37.3, 0.1, fa4, 0.84
 37.4, 0.1, re5, 0.84
 37.5, 0.1, fa5, 0.84
 37.6, 0.1, lad5, 0.84
 37.7, 0.1, re6, 0.84
 37.8, 0.1, fa6, 0.84
 37.9, 0.1, fa5, 0.84
 38, 0.1, lad4, 0.84
 38.1, 0.1, fa4, 0.84
 38.2, 0.1, re5, 0.84
 38.3, 0.1, fa5, 0.84

% 25

38.4, 0.1, do6, 0.84
 38.5, 0.1, red6, 0.84
 38.6, 0.1, fa6, 0.84
 38.7, 0.1, fa5, 0.84
 38.8, 0.1, do5, 0.84
 38.9, 0.1, fa4, 0.84
 39, 0.1, red5, 0.84
 39.1, 0.1, fa5, 0.84
 39.2, 0.1, la5, 0.84
 39.3, 0.1, do6, 0.84
 39.4, 0.1, red6, 0.84
 39.5, 0.05, fa6, 0.84
 39.55, 0.05, la6, 0.84
 39.6, 0.05, do7, 0.84
 39.65, 0.05, red7, 0.84
 39.7, 0.05, 0, 0

39.75, 0.05, 0, 0
 39.8, 0.05, 0, 0
 39.85, 0.1, do6, 0.84
 39.95, 0.1, la5, 0.84

% 26

40.05, 0.1, lad5, 0.84
 40.15, 0.1, re6, 0.84
 40.25, 0.1, fa6, 0.84
 40.35, 0.1, fa5, 0.84
 40.45, 0.1, lad4, 0.84
 40.55, 0.1, fa4, 0.84
 40.65, 0.1, re5, 0.84
 40.75, 0.1, fa5, 0.84
 40.85, 0.1, lad5, 0.84
 40.95, 0.1, re6, 0.84
 41.15, 0.1, fa6, 0.84
 41.25, 0.1, fa5, 0.84
 41.35, 0.1, lad4, 0.84
 41.45, 0.1, fa4, 0.84
 41.55, 0.1, re5, 0.84

% 27

41.65, 0.1, fa5, 0.84
 41.75, 0.1, do6, 0.84
 41.85, 0.1, red6, 0.84
 41.95, 0.1, fa6, 0.84
 42.05, 0.1, fa5, 0.84
 42.15, 0.1, do5, 0.84
 42.25, 0.1, fa4, 0.84
 42.35, 0.1, red5, 0.84
 42.45, 0.1, fa5, 0.84
 42.55, 0.1, la5, 0.84
 42.65, 0.1, do6, 0.84
 42.75, 0.1, red6, 0.84
 42.85, 0.05, fa6, 0.84
 42.9, 0.05, la6, 0.84
 42.95, 0.05, do7, 0.84
 43, 0.05, red7, 0.84
 43.05, 0.05, 0, 0
 43.1, 0.05, 0, 0
 43.15, 0.05, 0, 0

% 28

43.2, 0.1, do6, 0.84
 43.3, 0.1, la5, 0.84
 43.4, 0.1, lad5, 0.84
 43.5, 0.1, re6, 0.84
 43.6, 0.1, fa6, 0.84
 43.7, 0.1, re6, 0.84
 43.8, 0.1, lad5, 0.84
 43.9, 0.1, re6, 0.84
 44, 0.1, fa5, 0.84
 44.1, 0.1, lad5, 0.84
 44.2, 0.1, re5, 0.84
 44.3, 0.1, fa5, 0.84
 44.4, 0.1, lad4, 0.84
 44.5, 0.1, re5, 0.84
 44.6, 0.1, fa4, 0.84
 44.7, 0.1, lad4, 0.84

% 29

44.8, 0.1, re4, 0.84

44.9, 0.1, fa4, 0.84
 45, 0.1, 0, 0
 45.1, 0.1, si5, 0.84
 45.2, 0.1, re6, 0.84
 45.3, 0.1, si5, 0.84
 45.4, 0.1, fa5, 0.84
 45.5, 0.1, si5, 0.84
 45.6, 0.1, re5, 0.84
 45.7, 0.1, fa5, 0.84
 45.8, 0.1, si4, 0.84
 45.9, 0.1, re5, 0.84
 46, 0.1, fa4, 0.84
 46.1, 0.1, si4, 0.84
 46.2, 0.1, re4, 0.84
 46.3, 0.1, fa4, 0.84

% 30

46.4, 0.1, si3, 0.84
 46.5, 0.1, re4, 0.84
 46.6, 0.1, 0, 0
 46.7, 0.1, red6, 0.84
 46.8, 0.1, sol6, 0.84
 46.9, 0.1, red6, 0.84
 47, 0.1, do6, 0.84
 47.1, 0.1, red6, 0.84
 47.2, 0.1, sol5, 0.84
 47.3, 0.1, do6, 0.84
 47.4, 0.1, mi5, 0.84
 47.5, 0.1, sol5, 0.84
 47.6, 0.1, do5, 0.84
 47.7, 0.1, mi5, 0.84
 47.8, 0.1, sol4, 0.84
 47.9, 0.1, do5, 0.84

% 31

48, 0.1, mi4, 0.84
 48.1, 0.1, sol4, 0.84
 48.2, 0.1, 0, 0
 48.3, 0.1, dod6, 0.84
 48.4, 0.1, mi6, 0.84
 48.5, 0.1, dod6, 0.84
 48.6, 0.1, sol5, 0.84
 48.7, 0.1, dod6, 0.84
 48.8, 0.1, mi5, 0.84
 48.9, 0.1, sol5, 0.84
 49, 0.1, dod5, 0.84

49.1, 0.1, mi5, 0.84
 49.2, 0.1, sol4, 0.84
 49.3, 0.1, dod5, 0.84
 49.4, 0.1, mi4, 0.84
 49.5, 0.1, sol4, 0.84

% 32

49.6, 0.1, dod4, 0.84
 49.7, 0.1, mi4, 0.84
 49.8, 0.1, 0, 0
 49.9, 0.1, fad6, 0.84
 50, 0.1, la6, 0.84
 50.1, 0.1, fad6, 0.84
 50.2, 0.1, re6, 0.84
 50.3, 0.1, fad6, 0.84
 50.4, 0.1, la5, 0.84

50.5, 0.1, re6, 0.84
 50.6, 0.1, fad5, 0.84
 50.7, 0.1, la5, 0.84
 50.8, 0.1, re5, 0.84
 50.9, 0.1, fad5, 0.84
 51, 0.1, la4, 0.84
 51.1, 0.1, re5, 0.84

% 33

51.2, 0.1, fad4, 0.84
 51.3, 0.1, la4, 0.84
 51.4, 0.1, 0, 0
 51.5, 0.1, sol6, 0.84
 51.6, 0.1, lad6, 0.84
 51.7, 0.1, sol6, 0.84
 51.8, 0.1, re6, 0.84
 51.9, 0.1, sol6, 0.84
 52, 0.1, lad5, 0.84
 52.1, 0.1, re6, 0.84
 52.2, 0.1, sol5, 0.84
 52.3, 0.1, lad5, 0.84
 52.4, 0.1, re5, 0.84
 52.5, 0.1, sol5, 0.84
 52.6, 0.1, lad4, 0.84
 52.7, 0.1, re5, 0.84

% 34

52.8, 0.1, sol4, 0.84
 52.9, 0.1, lad4, 0.84
 53, 0.1, 0, 0
 53.1, 0.1, lad6, 0.84
 53.2, 0.1, mi7, 0.84
 53.3, 0.1, lad6, 0.84
 53.4, 0.1, sol6, 0.84
 53.5, 0.1, lad6, 0.84
 53.6, 0.1, mi6, 0.84
 53.7, 0.1, sol6, 0.84
 53.8, 0.1, lad5, 0.84
 53.9, 0.1, mi6, 0.84
 54, 0.1, sol5, 0.84
 54.1, 0.1, lad5, 0.84
 54.2, 0.1, mi5, 0.84
 54.3, 0.1, sol5, 0.84

% 35

54.4, 0.1, lad4, 0.84
 54.5, 0.1, mi5, 0.84
 54.6, 0.1, sol4, 0.84
 54.7, 0.1, lad4, 0.84
 54.8, 0.1, mi4, 0.84
 54.9, 0.1, sol4, 0.84
 55, 0.1, lad3, 0.84
 55.1, 0.1, mi4, 0.84
 55.2, 0.1, sol3, 0.84
 55.3, 0.1, lad3, 0.84
 55.4, 0.1, mi3, 0.84
 55.5, 0.1, sol3, 0
 55.6, 0.1, la3, 0.84
 55.7, 0.1, lad3, 0.84
 55.8, 0.1, la3, 0.84
 55.9, 0.1, sol3, 0.84

% 36

56, 0.1, fad3, 0.84
 56.1, 0.1, sol3, 0.84
 56.2, 0.1, fad3, 0.84
 56.3, 0.1, re3, 0.84
 56.4, 0.1, la3, 0.84
 56.5, 0.1, fad3, 0.84
 56.6, 0.1, do4, 0.84
 56.7, 0.1, la3, 0.84
 56.8, 0.1, red4, 0.84
 56.9, 0.1, do4, 0.84
 57, 0.1, lad3, 0.84
 57.1, 0.1, sol3, 0.84
 57.2, 0.1, re4, 0.84
 57.3, 0.1, lad3, 0.84
 57.4, 0.1, sol4, 0.84
 57.5, 0.1, re4, 0.84

% 37

57.6, 0.1, lad4, 0.84
 57.7, 0.1, sol4, 0.84
 57.8, 0.1, fad4, 0.84
 57.9, 0.1, re4, 0.84
 58, 0.1, la4, 0.84
 58.1, 0.1, fad4, 0.84
 58.2, 0.1, do5, 0.84
 58.3, 0.1, la4, 0.84
 58.4, 0.1, red5, 0.84
 58.5, 0.1, do5, 0.84
 58.6, 0.1, lad4, 0.84
 58.7, 0.1, sol4, 0.84
 58.8, 0.1, re5, 0.84
 58.9, 0.1, lad4, 0.84
 59, 0.1, sol5, 0.84
 59.1, 0.1, re5, 0.84

% 38

59.2, 0.1, lad5, 0.84
 59.3, 0.1, sol5, 0.84
 59.4, 0.1, fad5, 0.84
 59.5, 0.1, re5, 0.84
 59.6, 0.1, la5, 0.84
 59.7, 0.1, fad5, 0.84
 59.8, 0.1, do6, 0.84
 59.9, 0.1, la5, 0.84
 60, 0.1, red6, 0.84
 60.1, 0.1, do6, 0.84
 60.2, 0.1, lad5, 0.84
 60.3, 0.1, sol5, 0.84
 60.4, 0.1, re6, 0.84
 60.5, 0.1, lad5, 0.84
 60.6, 0.1, sol6, 0.84
 60.7, 0.1, re6, 0.84

% 39

60.8, 0.1, lad6, 0.84
 60.9, 0.1, sol6, 0.84
 61, 0.1, fad6, 0.84
 61.1, 0.1, la6, 0.84
 61.2, 0.1, re7, 0.84
 61.3, 0.1, la6, 0.84
 61.4, 0.1, fad6, 0.84
 61.5, 0.1, re6, 0.84

61.6, 0.1, la5, 0.84
 61.7, 0.1, fad5, 0.84
 61.8, 0.1, re5, 0.84
 61.9, 0.1, la4, 0.84
 62, 0.1, fad4, 0.84
 62.1, 0.1, re4, 0.84
 62.2, 0.1, do4, 0.84
 62.3, 0.1, re4, 0.84

% 40

62.4, 0.1, la3, 0.84
 62.5, 0.1, re4, 0.84
 62.6, 0.1, sol3, 0.84
 62.7, 0.1, re3, 0.84
 62.8, 0.1, lad3, 0.84
 62.9, 0.1, sol3, 0.84
 63, 0.1, re4, 0.84
 63.1, 0.1, lad3, 0.84
 63.2, 0.1, sol4, 0.84
 63.3, 0.1, re4, 0.84
 63.4, 0.1, lad4, 0.84
 63.5, 0.1, sol4, 0.84
 63.6, 0.1, re5, 0.84
 63.7, 0.1, lad4, 0.84
 63.8, 0.1, sol5, 0.84
 63.9, 0.1, re5, 0.84

% 41

64, 0.1, lad5, 0.84
 64.1, 0.1, sol5, 0.84
 64.2, 0.1, re6, 0.84
 64.3, 0.1, la5, 0.84
 64.4, 0.1, fad5, 0.84
 64.5, 0.1, la5, 0.84
 64.6, 0.1, re5, 0.84
 64.7, 0.1, fad5, 0.84
 64.8, 0.1, la4, 0.84
 64.9, 0.1, re5, 0.84
 65, 0.1, fad4, 0.84
 65.1, 0.1, la3, 0.84
 65.2, 0.1, re4, 0.84
 65.3, 0.1, fad4, 0.84
 65.4, 0.1, la3, 0.84
 65.5, 0.1, re4, 0.84

% 42

65.6, 0.1, fad3, 0.84
 65.7, 0.1, la3, 0.84
 65.8, 0.1, re4, 0.84
 65.9, 0.1, la3, 0.84
 66, 0.1, fad4, 0.84
 66.1, 0.1, re4, 0.84
 66.2, 0.1, la4, 0.84
 66.3, 0.1, fad4, 0.84
 66.4, 0.1, re5, 0.84
 66.5, 0.1, la4, 0.84
 66.6, 0.1, fad5, 0.84
 66.7, 0.1, re5, 0.84
 66.8, 0.1, la5, 0.84
 66.9, 0.1, fad5, 0.84
 67, 0.1, re6, 0.84
 67.1, 0.1, la5, 0.84

% 43

67.2, 0.1, fad6, 0.84
 67.3, 0.1, re6, 0.84
 67.4, 0.1, sol6, 0.84
 67.5, 0.1, re6, 0.84
 67.6, 0.1, lad5, 0.84
 67.7, 0.1, re6, 0.84
 67.8, 0.1, sol5, 0.84
 67.9, 0.1, lad5, 0.84
 68, 0.1, re5, 0.84
 68.1, 0.1, sol5, 0.84
 68.2, 0.1, lad4, 0.84
 68.3, 0.1, re5, 0.84
 68.4, 0.1, sol4, 0.84
 68.5, 0.1, lad4, 0.84
 68.6, 0.1, re4, 0.84
 68.7, 0.1, sol4, 0.84

% 44

68.8, 0.1, sol3, 0.84
 68.9, 0.1, re4, 0.84
 69, 0.1, fa4, 0.84
 69.1, 0.1, re4, 0.84
 69.2, 0.1, sold4, 0.84
 69.3, 0.1, fa4, 0.84
 69.4, 0.1, re5, 0.84
 69.5, 0.1, sold4, 0.84
 69.6, 0.1, fa5, 0.84
 69.7, 0.1, re5, 0.84
 69.8, 0.1, sold5, 0.84
 69.9, 0.1, fa5, 0.84
 70, 0.1, re6, 0.84
 70.1, 0.1, sold5, 0.84
 70.2, 0.1, fa6, 0.84
 70.3, 0.1, re6, 0.84

% 45

70.4, 0.1, sold6, 0.84
 70.5, 0.1, fa6, 0.84
 70.6, 0.1, red6, 0.84
 70.7, 0.1, sol6, 0.84
 70.8, 0.1, do6, 0.84
 70.9, 0.1, red6, 0.84
 71, 0.1, sol5, 0.84
 71.1, 0.1, do6, 0.84
 71.2, 0.1, red5, 0.84
 71.3, 0.1, sol5, 0.84
 71.4, 0.1, do5, 0.84
 71.5, 0.1, red5, 0.84
 71.6, 0.1, sol4, 0.84
 71.7, 0.1, do5, 0.84
 71.8, 0.1, red4, 0.84
 71.9, 0.1, sol4, 0.84

% 46

72, 0.1, do4, 0.84
 72.1, 0.1, red4, 0.84
 72.2, 0.1, mi4, 0.84
 72.3, 0.1, sol4, 0.84
 72.4, 0.1, lad4, 0.84
 72.5, 0.1, sol4, 0.84
 72.6, 0.1, mi5, 0.84
 72.7, 0.1, lad4, 0.84

72.8, 0.1, sol5, 0.84
 72.9, 0.1, mi5, 0.84
 73, 0.1, lad5, 0.84
 73.1, 0.1, sol5, 0.84
 73.2, 0.1, mi6, 0.84
 73.3, 0.1, lad5, 0.84
 73.4, 0.1, sol6, 0.84
 73.5, 0.1, mi6, 0.84

% 47

73.6, 0.1, lad6, 0.84
 73.7, 0.1, sol6, 0.84
 73.8, 0.1, fad6, 0.84
 73.9, 0.1, la6, 0.84
 74, 0.1, re7, 0.84
 74.1, 0.1, re6, 0.84
 74.2, 0.1, la5, 0.84
 74.3, 0.1, fad5, 0.84
 74.4, 0.1, re6, 0.84
 74.5, 0.1, re5, 0.84
 74.6, 0.1, la4, 0.84
 74.7, 0.1, fad4, 0.84
 74.8, 0.1, re5, 0.84
 74.9, 0.1, re4, 0.84
 75, 0.1, do4, 0.84
 75.1, 0.1, re4, 0.84

% 48

75.2, 0.1, la3, 0.84
 75.3, 0.1, re4, 0.84
 75.4, 0.1, lad3, 0.84
 75.5, 0.1, re4, 0.84
 75.6, 0.1, sol4, 0.84
 75.7, 0.1, lad4, 0.84
 75.8, 0.1, re5, 0.84
 75.9, 0.1, lad4, 0.84
 76, 0.1, sol5, 0.84
 76.1, 0.1, re5, 0.84
 76.2, 0.1, lad5, 0.84
 76.3, 0.1, sol5, 0.84
 76.4, 0.1, re6, 0.84
 76.5, 0.1, lad5, 0.84
 76.6, 0.1, sol6, 0.84
 76.7, 0.1, re6, 0.84

% 49

76.8, 0.1, lad6, 0.84
 76.9, 0.1, sol6, 0.84
 77, 0.1, re7, 0.84
 77.1, 0.1, la6, 0.84
 77.2, 0.1, fad6, 0.84
 77.3, 0.1, re6, 0.84
 77.4, 0.1, la5, 0.84
 77.5, 0.1, re6, 0.84
 77.6, 0.1, fad5, 0.84
 77.7, 0.1, la5, 0.84
 77.8, 0.1, re5, 0.84
 77.9, 0.1, fad5, 0.84
 78, 0.1, la4, 0.84
 78.1, 0.1, re5, 0.84
 78.2, 0.1, fad4, 0.84
 78.3, 0.1, la4, 0.84

% 50

78.4, 0.1, re4, 0.84
 78.5, 0.1, la4, 0.84
 78.6, 0.1, si4, 0.84
 78.7, 0.1, sold4, 0.84
 78.8, 0.1, re5, 0.84
 78.9, 0.1, si4, 0.84
 79, 0.1, sold5, 0.84
 79.1, 0.1, re5, 0.84
 79.2, 0.1, si5, 0.84
 79.3, 0.1, sold5, 0.84
 79.4, 0.1, re6, 0.84
 79.5, 0.1, si5, 0.84
 79.6, 0.1, sold6, 0.84
 79.7, 0.1, re6, 0.84
 79.8, 0.1, si6, 0.84
 79.9, 0.1, sold6, 0.84

% 51

80, 0.1, re7, 0.84
 80.1, 0.1, si6, 0.84
 80.2, 0.1, do7, 0.84
 80.3, 0.1, sol6, 0.84
 80.4, 0.1, mi6, 0.84
 80.5, 0.1, sol6, 0.84
 80.6, 0.1, do6, 0.84
 80.7, 0.1, mi6, 0.84
 80.8, 0.1, sol5, 0.84
 80.9, 0.1, do6, 0.84
 81, 0.1, mi5, 0.84
 81.1, 0.1, sol5, 0.84
 81.2, 0.1, do5, 0.84
 81.3, 0.1, mi5, 0.84
 81.4, 0.1, sol4, 0.84
 81.5, 0.1, do5, 0.84

% 52

81.6, 0.1, do4, 0.84
 81.7, 0.1, sol4, 0.84
 81.8, 0.1, do5, 0.84
 81.9, 0.1, sol4, 0.84
 82, 0.1, do5, 0.84
 82.1, 0.1, sol4, 0.84
 82.2, 0.1, red5, 0.84
 82.3, 0.1, do5, 0.84
 82.4, 0.1, sol5, 0.84
 82.5, 0.1, red5, 0.84
 82.6, 0.1, do6, 0.84
 82.7, 0.1, sol5, 0.84
 82.8, 0.1, red6, 0.84
 82.9, 0.1, do6, 0.84
 83, 0.1, sol6, 0.84
 83.1, 0.1, red6, 0.84

% 53

83.2, 0.1, do7, 0.84
 83.3, 0.1, sol6, 0.84
 83.4, 0.1, do7, 0.84
 83.5, 0.1, la6, 0.84
 83.6, 0.1, fad6, 0.84
 83.7, 0.1, la6, 0.84
 83.8, 0.1, do6, 0.84
 83.9, 0.1, fad6, 0.84

84, 0.1, la5, 0.84
 84.1, 0.1, do6, 0.84
 84.2, 0.1, lad6, 0.84
 84.3, 0.1, sol6, 0.84
 84.4, 0.1, mi6, 0.84
 84.5, 0.1, sol6, 0.84
 84.6, 0.1, lad5, 0.84
 84.7, 0.1, mi6, 0.84

% 54

84.8, 0.1, sol5, 0.84
 84.9, 0.1, lad5, 0.84
 85, 0.1, re6, 0.84
 85.1, 0.1, lad5, 0.84
 85.2, 0.1, sol5, 0.84
 85.3, 0.1, lad5, 0.84
 85.4, 0.1, re5, 0.84
 85.5, 0.1, sol5, 0.84
 85.6, 0.1, lad4, 0.84
 85.7, 0.1, re5, 0.84
 85.8, 0.1, fad5, 0.84
 85.9, 0.1, re5, 0.84
 86, 0.1, do5, 0.84
 86.1, 0.1, re5, 0.84
 86.2, 0.1, la4, 0.84
 86.3, 0.1, do5, 0.84

% 55

86.4, 0.1, fad4, 0.84
 86.5, 0.1, la4, 0.84
 86.6, 0.1, lad4, 0.84
 86.7, 0.1, sol4, 0.84
 86.8, 0.1, re5, 0.84
 86.9, 0.1, lad4, 0.84
 87, 0.1, sol5, 0.84
 87.1, 0.1, lad4, 0.84
 87.2, 0.1, re5, 0.84
 87.3, 0.1, sol4, 0.84
 87.4, 0.1, fad5, 0.84
 87.5, 0.1, do5, 0.84
 87.6, 0.1, red5, 0.84
 87.7, 0.1, fad4, 0.84
 87.8, 0.1, fa5, 0.84
 87.9, 0.1, sold4, 0.84

% 56

88, 0.1, re5, 0.84
 88.1, 0.1, fa4, 0.84
 88.2, 0.1, red5, 0.84
 88.3, 0.1, sol4, 0.84
 88.4, 0.1, do5, 0.84
 88.5, 0.1, red4, 0.84
 88.6, 0.1, re5, 0.84
 88.7, 0.1, fad4, 0.84
 88.8, 0.1, la4, 0.84
 88.9, 0.1, re4, 0.84
 89, 0.1, dod5, 0.84
 89.1, 0.1, sol4, 0.84
 89.2, 0.1, lad4, 0.84
 89.3, 0.1, dod4, 0.84
 89.4, 0.1, do5, 0.84
 89.5, 0.1, red4, 0.84

% 57

89.6, 0.1, la4, 0.84
 89.7, 0.1, do4, 0.84
 89.8, 0.1, si4, 0.84
 89.9, 0.1, re4, 0.84
 90, 0.1, sol4, 0.84
 90.1, 0.1, si3, 0.84
 90.2, 0.1, lad4, 0.84
 90.3, 0.1, red4, 0.84
 90.4, 0.1, sol4, 0.84
 90.5, 0.1, lad3, 0.84
 90.6, 0.1, sold4, 0.84
 90.7, 0.1, do4, 0.84
 90.8, 0.1, red4, 0.84
 90.9, 0.1, sold3, 0.84
 91, 0.1, sol4, 0.84
 91.1, 0.1, lad3, 0.84

% 58

91.2, 0.1, mi4, 0.84
 91.3, 0.1, sol3, 0.84
 91.4, 0.1, re4, 0.84
 91.5, 0.1, lad3, 0.84
 91.6, 0.1, sol4, 0.84
 91.7, 0.1, re4, 0.84
 91.8, 0.1, lad4, 0.84
 91.9, 0.1, sol4, 0.84
 92, 0.1, re5, 0.84
 92.1, 0.1, lad4, 0.84
 92.2, 0.1, do5, 0.84
 92.3, 0.1, red5, 0.84
 92.4, 0.1, la4, 0.84
 92.5, 0.1, do5, 0.84
 92.6, 0.1, fad4, 0.84
 92.7, 0.1, la4, 0.84

% 59

92.8, 0.1, re4, 0.84
 92.9, 0.1, re5, 0.84
 93, 0.1, lad4, 0.84
 93.1, 0.1, sol4, 0.84
 93.2, 0.1, re5, 0.84
 93.3, 0.1, lad4, 0.84
 93.4, 0.1, sol5, 0.84
 93.5, 0.1, lad4, 0.84
 93.6, 0.1, re5, 0.84
 93.7, 0.1, sol4, 0.84
 93.8, 0.1, fad5, 0.84
 93.9, 0.1, do5, 0.84
 94, 0.1, red5, 0.84
 94.1, 0.1, fad4, 0.84
 94.2, 0.1, fa5, 0.84
 94.3, 0.1, sold4, 0.84

% 60

94.4, 0.1, re5, 0.84
 94.5, 0.1, fa4, 0.84
 94.6, 0.1, red5, 0.84
 94.7, 0.1, sol4, 0.84
 94.8, 0.1, do5, 0.84
 94.9, 0.1, red4, 0.84
 95, 0.1, re5, 0.84
 95.1, 0.1, fad4, 0.84

95.2, 0.1, la4, 0.84
 95.3, 0.1, re4, 0.84
 95.4, 0.1, dod5, 0.84
 95.5, 0.1, sol4, 0.84
 95.6, 0.1, lad4, 0.84
 95.7, 0.1, dod4, 0.84
 95.8, 0.1, do5, 0.84
 95.9, 0.1, red4, 0.84

% 61

96, 0.1, la4, 0.84
 96.1, 0.1, do4, 0.84
 96.2, 0.1, si4, 0.84
 96.3, 0.1, re4, 0.84
 96.4, 0.1, sol4, 0.84
 96.5, 0.1, si3, 0.84
 96.6, 0.1, lad4, 0.84
 96.7, 0.1, red4, 0.84
 96.8, 0.1, sol4, 0.84
 96.9, 0.1, lad3, 0.84
 97, 0.1, sol4, 0.84
 97.1, 0.1, do4, 0.84
 97.2, 0.1, red4, 0.84
 97.3, 0.1, sold3, 0.84
 97.4, 0.1, sol4, 0.84
 97.5, 0.1, lad3, 0.84

% 62

97.6, 0.1, mi4, 0.84
 97.7, 0.1, sol3, 0.84
 97.8, 0.1, re4, 0.84
 97.9, 0.1, lad3, 0.84
 98, 0.1, sol4, 0.84
 98.1, 0.1, re4, 0.84
 98.2, 0.1, lad4, 0.84
 98.3, 0.1, sol4, 0.84
 98.4, 0.1, re5, 0.84
 98.5, 0.1, lad4, 0.84
 98.6, 0.1, do5, 0.84
 98.7, 0.1, red5, 0.84
 98.8, 0.1, la4, 0.84
 98.9, 0.1, do5, 0.84
 99, 0.1, fad4, 0.84
 99.1, 0.1, la4, 0.84

% 63

99.2, 0.1, re4, 0.84
 99.3, 0.1, re5, 0.84
 99.4, 0.1, sol4, 0.84
 99.5, 0.1, re4, 0.84
 99.6, 0.1, lad4, 0.84
 99.7, 0.1, sol4, 0.84
 99.8, 0.1, re5, 0.84
 99.9, 0.1, lad4, 0.84
 100, 0.1, sol5, 0.84
 100.1, 0.1, re5, 0.84
 100.2, 0.1, lad5, 0.84
 100.3, 0.1, re5, 0.84
 100.4, 0.1, re6, 0.84
 100.5, 0.1, re5, 0.84
 100.6, 0.1, fad5, 0.84
 100.7, 0.1, re5, 0.84

% 64

100.8, 0.1, la5, 0.84
 100.9, 0.1, re5, 0.84
 101, 0.1, sol5, 0.84
 101.1, 0.1, re5, 0.84
 101.2, 0.1, lad5, 0.84
 101.3, 0.1, sol5, 0.84
 101.4, 0.1, re6, 0.84
 101.5, 0.1, lad5, 0.84
 101.6, 0.1, sol6, 0.84
 101.7, 0.1, re6, 0.84
 101.8, 0.1, lad6, 0.84
 101.9, 0.1, re6, 0.84
 102, 0.1, re7, 0.84
 102.1, 0.1, re6, 0.84
 102.2, 0.1, fad6, 0.84
 102.3, 0.1, re6, 0.84

% 65

102.4, 0.1, la6, 0.84
 102.5, 0.1, re6, 0.84
 102.6, 0.1, sol6, 0.84
 102.7, 0.1, lad6, 0.84
 102.8, 0.1, re6, 0.84
 102.9, 0.1, sol6, 0.84
 103, 0.1, lad5, 0.84
 103.1, 0.1, re6, 0.84
 103.2, 0.1, sol5, 0.84
 103.3, 0.1, lad5, 0.84
 103.4, 0.1, re5, 0.84
 103.5, 0.1, sol5, 0.84
 103.6, 0.1, lad4, 0.84
 103.7, 0.1, re5, 0.84
 103.8, 0.1, sol4, 0.84
 103.9, 0.1, lad4, 0.84

% 66

104, 0.1, re4, 0.84
 104.1, 0.1, sol4, 0.84
 104.2, 0.1, lad3, 0.84
 104.3, 0.1, re4, 0.84
 104.4, 0.1, sol3, 0.84
 104.5, 0.1, lad3, 0.84
 104.6, 0.1, re3, 0.84
 104.7, 0.1, sol3, 0.84
 104.8, 0.1, lad2, 0.84
 104.9, 0.1, re3, 0.84
 105, 0.1, sol2, 0.84
 105.1, 0.1, 0, 0
 105.2, 0.1, 0, 0
 105.3, 0.1, 0, 0
 105.4, 0.2, lad5, 0.84
 105.4, 0.2, sol5, 0.84
 105.4, 0.2, re5, 0.84
 105.4, 0.2, lad4, 0.84

% 67 - 68 - 69

105.6, 0.2, 0, 0

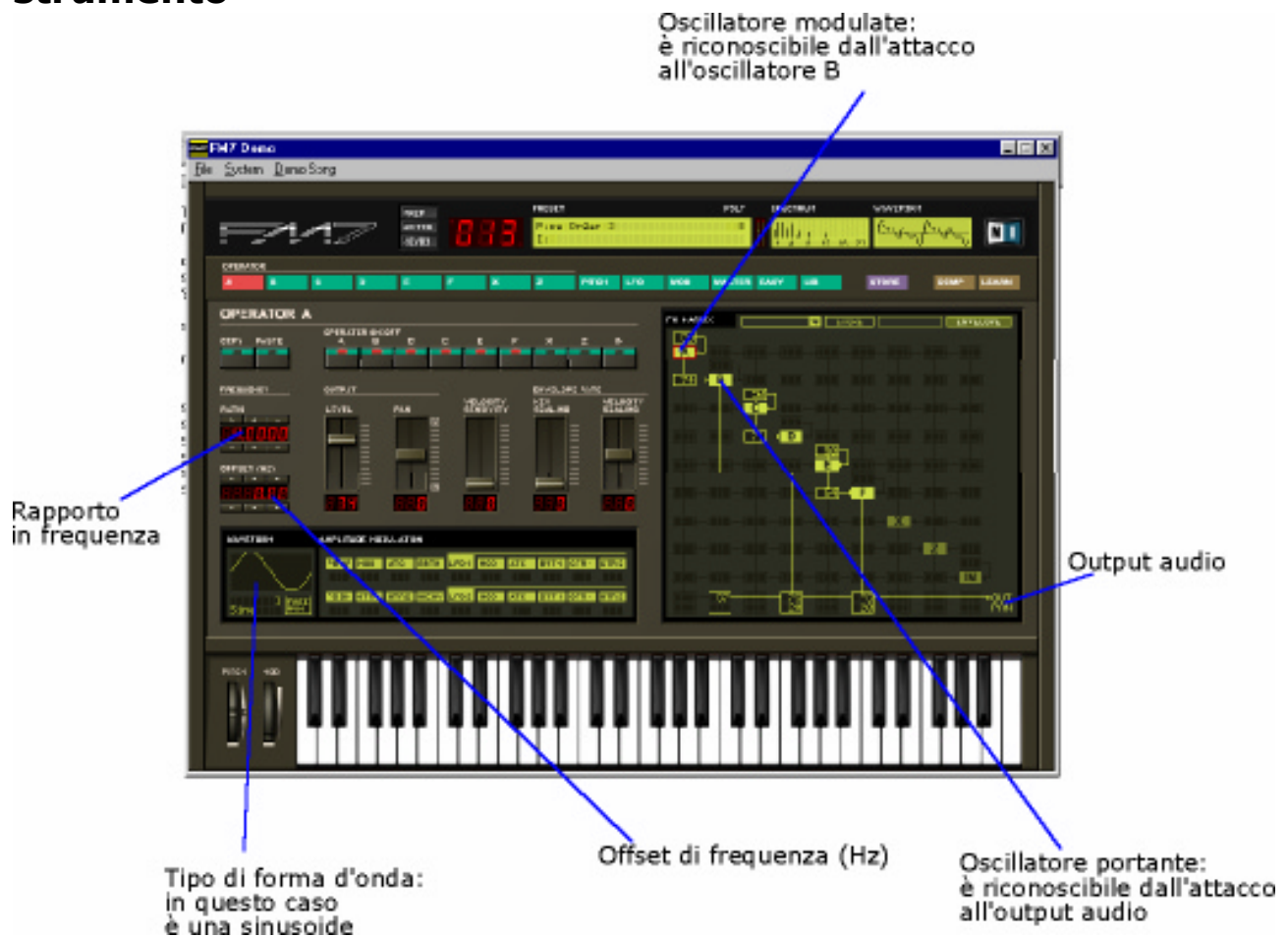
 105.8, 1.6, sol5, 0.84
 105.8, 1.6, re5, 0.84
 105.8, 1.6, lad4, 0.84

105.8, 1.6, sol4, 0.84

107.4, 1.6, sol5, 0.84
 107.4, 1.6, re5, 0.84
 107.4, 1.6, lad4, 0.84
 107.4, 1.6, sol4, 0.84

109, 1.5, sol5, 0.84
 109, 1.5, re5, 0.84
 109, 1.5, lad4, 0.84
 109, 1.5, sol4, 0.84
 };

Implementazione di uno strumento



Dallo studio di FM7, noto sintetizzatore software sviluppato dalla Native Instruments, è possibile osservare i parametri da prendere in considerazione per la costruzione di uno strumento mediante modulazione di frequenza.

Prendiamo in esame il caso dello strumento numero 13: Pipe Organ 2.

Cliccando su "Pitch" (in alto, tra i riquadri verdi) ed in seguito su "Matrix", nel riquadro che compare sulla destra è possibile osservare in che modo gli oscillatori sono collegati fra loro. Si ricorda che FM7 possiede 6 oscillatori (chiamati A, B, C, D, E ed F), modulanti o portanti in base alla configurazione, più altri due oscillatori speciali (X ed Y) che qui non useremo.

Un oscillatore modulante è riconoscibile dall'attacco ad un altro oscillatore, mentre un oscillatore portante è riconoscibile dall'attacco diretto al canale output. E' possibile instaurare anche legami di feedback, cioè fare in modo che un oscillatore moduli se stesso con il proprio output: è visibile in figura sugli oscillatori A, C ed E.

Uno sguardo ai collegamenti dà una prima idea sulla struttura di base dello strumento. E' necessario ora prendere in considerazione i parametri che implementano la modulazione di frequenza quali l'indice di modulazione ed il rapporto frequenza portante/frequenza modulante. Sulla sinistra, come evidenziato in figura, il riquadro "Frequency ratio" rappresenta il rapporto in frequenza portante/modulante relativo all'oscillatore selezionato: in pratica, questo rappresenta il divisore della nota che si desidera suonare.

L'indice di modulazione è rappresentato dai livelli di ampiezza, ben visibili nella matrice: ad esempio l'oscillatore A si automodula con un indice pari a 50 e modula l'oscillatore B con un indice pari a 74. I valori di ampiezza in FM7 sono compresi nell'intervallo [0,100], mentre in

Matlab sono compresi tra [0,1]. E' quindi ragionevole convertire i valori proposti ad FM7, dividendoli per 100.

Infine, ad ogni oscillatore è associato un inviluppo di ampiezza, visibile cliccando su "Envelope". E' possibile ricreare qualunque tipo di inviluppo in Matlab, come spiegato nel Capitolo 3. Tutti gli altri parametri sono stati ignorati per semplicità.

Utilizzando quindi lo schema classico di definizione di funzioni in Matlab abbiamo ricreato con discreto successo il suono definito in FM7. Di seguito la definizione dello strumento:

```
% File : organ.m

function [snd] = organ(note,sec,srate)

% Qualora non venga specificato alcun sample rate nella chiamata a funzione,
% di default viene assegnato un sample rate di 44100 allo strumento.
if nargin == 2
    srate = 44100;
end

% Creazione del vettore tempo
t = linspace(0, sec, sec * srate);

% Creazione degli inviluppi di ampiezza relativi a ciascun oscillatore
env_A = [ linspace(1.0,0.84,srate*0.4*sec) linspace(0.84,0.0,srate*0.6*sec) ];
env_B = [ linspace(0.0,1,srate*0.2*sec) linspace(1,0.84,srate*0.4*sec)
          linspace(0.84, 0.0, srate * 0.4 *sec) ];
env_C = env_A;
env_D = env_B;
env_E = env_A;
env_F = env_B;

% Creazione della modulazione di feedback, del segnale modulante e del portante
modulato
% Oscillatori A-B
feed_a = env_a .* sin(2 * pi * (note/4) * t);
mod_a = env_a .* sin(2 * pi * (note/4) * t + 0.50 * feed_a);
car_b = 0.57 * env_b .* sin(2 * pi * (note + 0.19) * t + 0.74 * mod_a);

% Creazione della modulazione di feedback, del segnale modulante e del portante
modulato
% Oscillatori C-D
feed_c = env_c .* sin(2 * pi * ((note/2) - 0.20) * t);
mod_c = env_c .* sin(2 * pi * ((note/2) - 0.20) * t + 0.50 * feed_c);
car_d = 0.75 * env_d .* sin(2 * pi * (note + 0.17) * t + 0.71 * mod_c);

% Creazione della modulazione di feedback, del segnale modulante e del portante
modulato
% Oscillatori E-F
feed_e = env_e .* sin(2 * pi * (note + 0.20) * t);
mod_e = env_e .* sin(2 * pi * (note + 0.20) * t + 0.50 * feed_e);
car_f = 0.75 * env_f .* sin(2 * pi * (note + 0.15) * t + 0.84 * mod_e);

% Creazione del segnale acustico
snd = car_b + car_d + car_f;
```

Elaborazione della partitura e scripting

Una volta creato uno strumento e definito una partitura, è necessario scrivere una procedura automatizzata per la generazione del file audio. Osserviamo dunque la seguente procedura:

```
% Funzione: compose(partiture,srate)
% -----
% Elabora la partitura specificate con il sample rate specificato (srate).
% Per modificare lo strumento è sufficiente cambiare la chiamata a funzione
% della riga 26.
%
% Si osservi l'uso di un file temporaneo di supporto: questo previene
% l'esaurimento della memoria di sistema, ed inoltre velocizza l'intera
% procedura, la quale elabora singolarmente ciascuna nota della partitura,
% elaborando una notevole mole di dati.

function [snd] = compose(partiture,srate);

% Scrittura a video di informazioni sulla partitura e sui parametri passati
fprintf('\n Partiture statistics:\n');
fprintf(' Sample rate: %d\n', srate);
[row,col] = size(partiture);
fprintf(' Number of notes: %d\n', row);

% Calcolo del tempo totale della partitura
global_time = partiture{row,1} + partiture{row,2};
fprintf(' Global time : %d sample (%d sec.)\n', global_time * srate,
        global_time);

% Apertura del file temporaneo in scrittura
fprintf(' Opening temporary file...\n');
fid = fopen('temp.snd','w');

fprintf('\n Composing partiture:\n');

creation_time = partiture{1,1};      % Lettura del tempo di creazione della nota
action_time = partiture{1,2};        % Lettura del tempo di durata della nota
atime = linspace(0, action_time, action_time * srate);
                                     % Creazione del vettore tempo per la nota
amplitude = partiture{1,4};          % Lettura dell'ampiezza della nota
frequency = partiture{1,3};          % Lettura della frequenza della nota
waveform = amplitude * organ(frequency,action_time,srate);
                                     % Creazione della prima nota

% Stampa a video informazioni sulla nota elaborata (debug)
fprintf(' Note: #1 - Action time: %d sec. (%d samples)\n', action_time,
        action_time * srate);

temp = waveform;
snd = [];
% Genera le note successive alla prima
for i = 2 : row
    creation_time = partiture{i,1};
    action_time = partiture{i,2};
    fprintf(' Note: #%d - Action time: %d sec. (%d samples) - Op.: ', i,
            action_time, action_time * srate);
    atime = linspace(0, action_time, action_time * srate);
    amplitude = partiture{i,4};
    frequency = partiture{i,3};
    waveform = amplitude * organ(frequency,action_time,srate);

    % Controlla se i tempi d'inizio e di durata della nota attuale sono uguali a
```

```

% quelli della nota precedente: in caso affermativo le due note si
% sovrappongono.
prev_ctime = partiture{i - 1,1};
prev_atime = partiture{i - 1,2};

if (creation_time == prev_ctime) & (action_time == prev_atime)
    fprintf(' Updating\n');
    temp = temp + waveform;
else
    fprintf(' Adding\n');
    snd = [snd temp];
    fwrite(fid,snd,'single');
    snd = [];
    temp = waveform;
end
end

fclose(fid);          % Chiude il file temporaneo in scrittura
fid = fopen('temp.snd','r'); % Riapre il file temporaneo in lettura
snd = fread(fid,'single'); % Legge dal file temporaneo il suono elaborato
fclose(fid);          % Chiude il file temporaneo il lettura

```

In grassetto è evidenziata l'istruzione da modificare per utilizzare un altro strumento. A questo punto, seppur non necessario, è conveniente scrivere uno script che esegua in modo del tutto automatico i passi visti finora. A tal proposito osserviamo il seguente script, il quale realizza la "compilazione" delle partiture precedentemente definite, mediante lo strumento "organ" definito ed infine suona il segnale generato.

```

% File: czerny.m
% -----
% Script per la "compilazione" delle partiture e dello strumento predefiniti.

srate = 22050;          % Il sample rate di default è pari a 22050.
                        % Per cambiarlo, modificare il valore

czerny_left; % Chiamata alla partitura relativa alla mano sinistra
czerny_right; % Chiamata alla partitura relativa alla mano destra

snd1 = compose(left_hand,srate); % "Compilazione" della prima partitura
snd2 = compose(right_hand,srate); % "Compilazione" della seconda partitura

% Routine di padding
if (length(snd1) < length(snd2))
    snd1 = [ snd1 ; zeros(1,length(snd2)-length(snd1))' ];
elseif (length(snd1) > length(snd2))
    snd2 = [ snd2 ; zeros(1,length(snd1)-length(snd2))' ];
end

snd = snd1 + snd2;      % Combinazione dei due segnali generati

snd = normalize(snd,50); % Normalizzazione del segnale generato
sound(snd,srate);       % Suono del segnale generato

```

Si osservi l'uso del padding: Matlab è molto rigido nelle operazioni che coinvolgono vettori o matrici. Potrebbe accadere che i segnali generati non abbiano le stesse dimensioni e basta un "campione" di differenza per bloccare l'intera procedura. La routine proposta serve proprio ad uniformare i segnali generati, in modo tale da poterli poi combinare e quindi suonare.

Appendice

Carl Czerny Studio Op. 740 Num. 14

*Gli spartiti sono stati importati
da MIDIFile ed elaborati mediante
Steinberg Cubase VST/24*

Studio op. 740 num. 14

Carl Czerny - Mano Sinistra

MIDI

1

5

9

13

17

21

25

29

33

37

41

45

49

53

MIDI
57



61



65



69



Studio op. 740 num. 14

Carl Czerny - Mano Destra

MIDI

5

9

13

17

21

25

29

33

37

44

45

46

53

MIDI
57



61



65



69

