Chapter 5

Sound modeling: signal-based approaches

Giovanni De Poli *Carlo Drioli* *Federico Avanzini*

5.1 Introduzione

Negli strumenti musicali tradizionali il suono e' prodotto dalla vibrazione di parti meccaniche. Negli strumenti sintetici, la vibrazione e' descritta da funzioni nel tempo, dette segnali, che esprimono la variazione nel tempo della pressione acustica.

Per costruire uno strumento musicale tradizionale e' sufficiente realizzare un dispositivo che sfrutta uno dei tanti meccanismi fisici per la produzione di vibrazioni. Negli strumenti musicali sintetici, invece, l'obiettivo e' di generare una piu' astratta funzione nel tempo, detta segnale acustico. Allo scopo e' necessario implementare una rappresentazione semplificata ed astratta del modo di produzione del suono, detta *modello*. Il modello del segnale, quindi, gioca il ruolo del fenomeno fisico negli strumenti tradizionali: esso costituisce il nucleo centrale attorno al quale si sviluppa la realizzazione dello strumento musicale. Nel modello l'astrazione ha il significato di inscrivere il meccanismo di produzione in una classe piu' generale di cui esso rappresenta un caso particolare. La semplificazione tende a focalizzare la descrizione sugli aspetti ritenuti significativi nel caso in esame. Spesso nella descrizione del modello si ricorre a relazioni matematiche per legare le cause con gli effetti; in questo modo, usando il modello si puo' prevedere il comportamento del fenomeno in condizioni note. Le condizioni note includono i parametri del modello, eventualmente variabili nel tempo, e lo stato iniziale da cui procede l'evoluzione.

L'*algoritmo* e' il procedimento effettivo che consente di realizzare cio'. L'implementazione dell'algoritmo su un processore consente di calcolare l'evoluzione del fenomeno, eventualmente nella sua forma semplificata. In particolare algoritmi di sintesi dei segnali musicali, basati su modelli del segnale che si focalizzano su diverse e interessanti proprieta' degli stessi, consentono di ottenere l'evoluzione temporale del segnale. Al variare dei parametri si ottengono tutti i possibili segnali della classe identificata dal modello; e' cioe' possibile esplorare l'insieme dei timbri descritto dal modello.

In linea di principio qualsiasi variazione dei parametri di controllo di un algoritmo e' lecita. L'impiego per scopi musicali, tuttavia, impone alcune limitazioni alla liberta' di scegliere i parametri di controllo. I parametri di controllo a loro volta possono variare nel tempo, divenendo cosi a loro volta dei segnali (di controllo). La variazione dei segnali di controllo acquista un diverso significato secondo la scala dei tempi su cui si attua. Se il controllo si attua sulla scala di tempo della (frazione di) nota, parliamo di controllo della dinamica spettrale. Esso infatti viene spesso interpretato in relazione alla variazione a tempo breve dello spettro. Se il controllo si attua nella scala di tempo dell'organizzazione delle note in frasi o entita' superiori, parliamo di controllo espressivo. Ad esempio la variazione del pitch delle note rappresenta il controllo espressivo fondamentale nella musica occidentale.

La sintesi elettronica dei suoni sembra offrire una grande liberta' nella costruzione dei suoni, sia nella imitazione di quelli naturali, che nella produzione di sonorita' originali. Vi sono ormai varie tecniche per riprodurre i suoni desiderati con la fedelta' voluta. Tuttavia scopo della sintesi del suono non e' tanto la riproduzione di un segnale voluto, quanto la realizzazione di un generatore suonabile, caratterizzato cioe' da una articolazione timbrica paragonabile a quella degli strumenti classici. Il problema si sposta quindi alle possibilita' di controllo dell'algoritmo e dell'articolazione timbrica offerte dagli strumenti sintetici.

5.1.1 Obiettivi della modellazione audio

Tradizionalmente, nella musica occidentale, il suono e' caratterizzato da altezza, intensita', durata metrica, timbro e localizzazione spaziale. Sono questi i parametri che il musicista gestisce. La presenza del pitch presuppone un modello di segnale (quasi) periodico. Il pitch e' legato alla frequenza del suono e induce nello spettro del segnale una struttura a righe, dove cioe' l'energia e' concentrata in bande ristrette (righe) a intervalli regolari sullo spettro. Non tutti i suoni hanno altezza definita; in questi casi si parla di spettri continui, caratterizzati da assenza di regolarita' nello spettro. Il loudness e' legato all'energia del segnale, la durata metrica e' alla base della percezione ritmica. Lo spazio e soprattutto il timbro sono i parametri che offrono maggiori possibilita' di articolazione nei suoni sintetici o trasformati elettronicamente.

L'esigenza di manipolare questi parametri rimane un aspetto centrale anche nella musica elettronica. Lo scopo della sintesi del suono quindi dovrebbe tendere a realizzare strumenti suonabili piu' che generatori di segnale, in modo da preservare il rapporto di causa ed effetto che sussiste tra l'azione sul controllo ed il risultato sul suono. Si dovrebbe cioe' offrire al musicista uno *strumento* a tutti gli effetti, inteso come entita' caratterizzata da certi requisiti di coerenza interna, che si concretizzano in suonabilita', qualita' sonora, utilizzabilita' all'interno di una partitura.

Lo strumento musicale e' importante anche perche', oltre a rappresentare il processo di generazione, puo' essere visto come astrazione di una classe di suoni caratterizzati da un timbro, un comportamento dinamico, e da certe possibilita' espressive. Questo fatto puo' applicarsi oltre che agli strumenti tradizionali, anche agli strumenti sintetici. Ne risulta che si possono definire classi astratte di suoni sintetici in base al tipo di modello (e algoritmo) usato per la sintesi e per il tipo di controllo offerto al musicista. Una volta, la scelta dell'algoritmo di sintesi avveniva in base alla efficienza computazionale, anche a spese della sua controllabilita'. Oggi, con lo sviluppo della tecnologia, questo problema e' sempre meno importante.

Acquistano quindi sempre piu' importanza altri criteri di scelta, tra cui "migliore" metafora per il musicista e "migliore" risultato acustico. Al primo criterio corrisponde il grado di suggestione che l'algoritmo opera sul musicista-compositore; ad esempio la sintesi additiva suggerisce una visione armonica. Al secondo criterio corrisponde l'esigenza di un risultato acustico ben preciso, o di una particolare interfaccia verso l'esecutore; ad esempio la sintesi per modulazione di frequenza puo'

riprodurre facilmente suoni percussivi inarmonici (campane).

Gli strumenti sintetici, al pari degli strumenti classici, richiedono l'apprendimento della tecnica di esecuzione. Si deve infatti imparare con l'esperienza le relazioni tra i parametri di controllo e il risultato acustico. Queste relazioni spesso non sono intuitive nel controllo a basso livello degli algoritmi e quindi limitano di fatto la versatilita' dello strumento. Si puo' notare d'altra parte che la tendenza attuale e' quella di incorporare l'esecutore nello strumento; si cerca cioe' di realizzare uno strumento senza problemi di manualita' e controllabile con informazioni di alto livello, eventualmente per mezzo di esecutori automatici (sequencer). Nell'ottica di questo approccio devono quindi essere sviluppati sofisticati modelli del controllo timbrico che, a partire da poche e sintetiche informazioni, siano in grado di produrre un ventaglio espressivo paragonabile a quello di un esecutore umano.

5.1.2 Classificazione dei modelli audio

Di seguito sono presentati i principali algoritmi di sintesi con riferimento ai criteri di scelta sopra esposti. E' tuttavia possibile procedere ad una classificazione degli algoritmi di sintesi basata sull'analisi della loro struttura. Si puo' infatti notare che la complessita' della struttura ha forti riflessi sulla controllabilita' sia timbrica che espressiva di un algoritmo. Gli algoritmi definiti da una struttura semplice necessitano di un flusso di segnali di controllo molto articolato, in quanto caratterizzazione ed espressivita' timbrica devono essere garantiti proprio dai segnali di controllo. Invece gli algoritmi con complessita' strutturale elevata garantiscono di base una buona caratterizzazione timbrica e una buona coerenza interna, consentendo quindi un controllo molto piu' semplificato. Si possono quindi individuare le seguenti classi di algoritmi:

- generazione diretta: di questa classe fanno parte campionamento, sintesi additiva, granulare;
- feed-forward: sottrattiva, modulazioni, distorsione non lineare;
- feed-back: sintesi per modelli fisici

Ad esempio se prendiamo in considerazione uno strumento caratterizzato da un controllo gestuale assai semplice come il pianoforte, si identificano i seguenti requisiti per i segnali di controllo:

- sintesi additiva: supponendo di lavorare con 100 parziali la pressione del tasto attiva 100 inviluppi temporali e altrettanti inviluppi frequenziali con andamento in funzione della velocita' della pressione del tasto.
- sintesi FM: supponendo di lavorare con un algoritmo a 6 operatori la pressione del tasto attiva 6 inviluppi temporali e altrettanti inviluppi degli indici di modulazione con andamento funzione della velocita' della pressione del tasto.
- sintesi per modelli fisici: supponendo di lavorare con un algoritmo martelletto corda, la pressione del tasto fornisce l'unico parametro (la velocita' d'impatto del martelletto) all'algoritmo, che provvede autonomamente a generare la sonorita' attesa.

È possibile anche un'altra classificazione degli algoritmi di sintesi in base al tipo di modello con cui viene rappresentato il suono. In questo caso si possono distinguere

- *modelli del segnale* che rappresentano il suono che ci arriva all'orecchio, senza far riferimento al meccanismo fisico che sottosta' alla produzione del suono. La percezione del suono e' un fenomeno complesso, che analizza il segnale sia nel tempo che nella frequenza. Anche i modelli

del segnale possono essere divisi in due classi, secondo se possono essere interpretati dall'utente in termini di caratteristiche temporali o spettrali. Possiamo quindi includere nella prima classe il campionamento e la sintesi granulare, mentre la sintesi additiva e sottrattiva, le modulazioni e la distorsione non lineare sono della seconda classe (meglio interpretabili nel dominio della frequenza).

- *modelli della sorgente* che ottengono il segnale acustico come sottoprodotto di un modello di simulazione del meccanismo fisico di produzione del suono. Appartiene a questa categoria la sintesi per modelli fisici.

Va infine ricordato che quando si parla di segnali musicali generalmente si intendono i segnali sonori. Come detto pero' il risultato acustico che si ottiene da un modello dipende dal controllo che si effettua sui parametri del modello stesso. In molti casi questi parametri sono tempo varianti e si evolvono durante lo sviluppo del singolo suono. Sono cioe' essi stessi dei segnali, chiamati appunto di controllo, che pero' si differenziano dai segnali audio perche' si evolvono piu' lentamente. Inoltre essi vengono percepiti seguendo la loro evoluzione temporale e non analizzandoli in frequenza, come accade per i segnali audio. Nel seguito verranno esposti i principali algoritmi di sintesi dei segnali audio. Talvolta essi sono utili anche per i segnali di controllo.

5.2 Metodi di generazione diretta

In questa categoria troviamo i metodi che generano direttamente il segnale attraverso un'unico modello, o piu' modelli che pero' non si influenzano reciprocamente.

5.2.1 Generatori di forme d'onda

5.2.1.1 Oscillatori numerici

La caratteristica di molti suoni musicali e' di essere quasi periodici o armonici. E' questa proprieta' che determina la sensazione di altezza di un suono. Il piu' semplice metodo di sintesi consiste nel produrre un segnale periodico mediante la continua ripetizione di una certa forma d'onda. Un algoritmo che realizza questo metodo si chiama oscillatore. L'oscillatore piu' diffuso e' quello a forma d'onda tabulata (*table look-up oscillator*). In questo caso la forma d'onda e' memorizzata in una tabella in punti equispaziati. Per generare una forma d'onda periodica, basta leggere ripetutamente la tabella mandando i suoi campioni uno dopo l'altro in uscita. Se F_s e' la frequenza di campionamento e L e' la lunghezza della tabella, la frequenza f del suono periodico risulta $f = F_s/L$. Se si volesse un suono con la stessa forma d'onda ma di frequenza diversa, occorrerebbe una tabella contenete la stessa forma d'onda ma rappresentata con un numero diverso di valori. Si vorrebbe quindi una forma d'onda continua da cui prelevare di volta in volta il valore all'ascissa desiderata. A questo scopo si ricorre a tabelle contenenti la forma d'onda in (molti) punti equispaziati e poi prelevando di volta in volta il valore piu' opportuno o mediante interpolazione tra i due punti adiacenti o usando il valore di ascissa piu' prossima a quella desiderata (interpolazione di ordine zero). Naturalmente piu' fitti sono i punti, migliore e' l'approssimazione. Si usano tipicamente tabelle da 256 a 4096 punti. In questo modo l'oscillatore ricampiona la tabella per generare un suono di differente frequenza.

La distanza (in numero di campioni) fra due campioni della tabella prelevati in istanti successivi si chiama SI (*sampling increment*) ed e' proporzionale alla frequenza f del suono prodotto:

$$f = \frac{SI \cdot F_s}{L} \tag{5.1}$$

Se il passo di lettura SI e' maggiore di uno, puo' succedere che le frequenze delle componenti piu' alte siano maggiori della frequenza di Nyquist, dando luogo a foldover. Per evitare questo fenomeno, bisogna limitare la banda del segnale memorizzato. Se invece il passo e' minore di uno, come avviene spesso per i segnali di controllo, inviluppi di ampiezza etc., allora il problema non si pone in quanto la banda e' gia' sufficientemente limitata.

M-5.1

Implement a circular look-up from a table of length L and with sampling increment SI.

M-5.1 Solution

```
phi=mod(phi +SI,L);
s=tab[phi];
```

where `phi` is the reading point in the table, `A` is a scaling parameter, `s` is the output signal sample. The function `mod(x,y)` computes the remainder of the division x/y and is used here to implement circular reading of the table.

Segnali sinusoidali possono essere generati, oltre che tramite tabella, anche con metodi ricorsivi. Un primo metodo si basa sul risuonatore numerico, costituito da un filtro del secondo ordine con i poli (complessi coniugati) sul cerchio di raggio unitario. Esso e' dato dall'equazione ricorrente

$$y(n+1) = 2\cos(\omega)y(n) - y^2(n-1) \tag{5.2}$$

dove $\omega = 2\pi f/F_s$. Con condizioni iniziali $y(0) = 1$ e $y(-1) = \cos\omega$ il generatore produce $y(n) = \cos n\omega$; con $y(0) = 0$ e $y(-1) = -\sin\omega$ il generatore produce $y(n) = \sin n\omega$. In generale se $y(0) = \cos\phi$ e $y(-1) = \cos(\phi - \omega)$ il generatore produce $y(n) = \cos(n\omega + \phi)$. Questa proprieta' puo' anche essere verificata ricordando la relazione trigonometrica $cos\omega \cdot \cos\phi = 0.5[cos(\phi+\omega) + \cos(\phi-\omega)]$.

Un'altro metodo si basa sulla forma accoppiata decritta dalle equazioni

$$\begin{aligned} x(n+1) &= \cos\omega \cdot x(n) - \sin\omega \cdot y(n) \\ y(n+1) &= \sin\omega \cdot x(n) + \cos\omega \cdot y(n) \end{aligned}$$

Con $x(0) = 1$ e $y(0) = 0$ si ha $x(n) = \cos(n\omega)$ e $y(n) = \sin(n\omega)$; vengono generati contemporaneamente un seno e un coseno. Questa proprita' puo essere verificata considerando che se si definisce una variabile complessa $w(n) = x(n) + jy(n) = \exp(jn\omega)$, risulta $w(n+1) = \exp(j\omega) \cdot w(n)$. Prendendo la parte reale e immaginaria di questa relazione risulta la forma accoppiata.

In generale entrambi metodi hanno il problema che la quantizzazione dei coefficienti puo' causare instabilita' numerica e cioe' i poli non sono esattamente sul cerchio unitario. Le forme d'onda generate allora o tenderanno a smorzarsi o a crescere indefinitamente. A questo scopo e' opportuno periodicamente reinizializzare la ricorsione.

5.2.1.2 Amplitude/frequency controlled oscillators

The amplitude and frequency of a sound are usually required to be time-varying parameters. Amplitude control is needed in order to define suitable sound envelopes, or to create *tremolo* effects (quasi-periodic amplitude variations around an average value). Frequency control is needed to simulate *portamento* between two tones, or subtle pitch variations in the sound attack/release, or *vibrato* effects (quasi-periodic pitch variations around an average value), and so on.

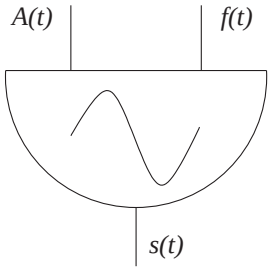


Figure 5.1: Symbol of the fixed waveform oscillator, with varying amplitude and frequency.

We then want to have at our disposal a digital oscillator of the form

$$s(n) = A(n) \cdot \text{tab}[\phi(n)], \tag{5.3}$$

where $A(n)$ scales the amplitude of the signal, and the phase $\phi(n)$ does not in general increase linearly in time and is computed as a function of the instantaneous frequency. Figure 5.1 shows the symbol usually adopted to depict an oscillator with fixed waveform and varying amplitude and frequency

Many sound synthesis languages (e.g., the well known *Csound*) define control signals at *frame rate*: a frame is a time window with pre-defined length (typically 5 to 50 ms), in which the control signals can be reasonably assumed to be approximately constant. This approximation clearly helps to reduce computational loads significantly.

M-5.2

Assume that a function `sinosc(t0,a,f,ph0)` realizes a sinusoidal oscillator (`t0` is the initial time, `a,f` are the frame-rate amplitude and frequency vectors, and `ph0` is the initial phase). Then generate a sinusoid of length 2 s, with constant amplitude and frequency.

M-5.2 Solution

```
%%% headers %%%
global Fs;              %sample rate
global SpF;             %samples per Frame
Fs=22050;
ControlW=0.01;          %control window (in sec): 10 ms
SpF=round(Fs*ControlW);
Fc=Fs/SpF;              %control rate

%%% define controls %%%
slength=2;             %soundlength in seconds
nframes=slength*Fc;    %total number of frames
a=ones(1,nframes);     %constant amplitude
f=50*ones(1,nframes);  %constant frequency

%%% compute sound %%%
s=sinosc(0,a,f,0);     %sound signal
```

Note the structure of this simple example: in the “headers” section some global parameters are defined, that need to be known also to auxiliary functions; a second section defines the control parameters, and finally the audio signal is computed.

When the oscillator frequency is constant the phase is a linear function of time, $\phi(t) = 2\pi ft$, therefore in the digital domain ϕ can be computed as $\phi(n+1) = \phi(n) + \frac{2\pi f}{F_s}$. In the more general case in which the frequency varies at frame rate, we have to understand how to compute the phase of the oscillator. The starting point is the equation

$$f(t) = \frac{1}{2\pi}\frac{d\phi}{dt}(t), \tag{5.4}$$

which simply says that the radian frequency $\omega(t) = 2\pi f(t)$ is the instantaneous angular velocity of the time-varying phase $\phi(t)$. If $f(t)$ is varying slowly enough (i.e. it is varying at frame rate), we can say that in the Kth frame the first-order approximation

$$\frac{1}{2\pi}\frac{d\phi}{dt}(t) = f(t) \sim f(T_K) + F_c\left[f(T_{K+1}) - f(T_K)\right]\cdot(t - T_K) \tag{5.5}$$

holds, where T_K, T_{K+1} are the initial instants of frames K and $K+1$, respectively. The term $F_c\left[f(T_{K+1}) - f(T_K)\right]$ approximates the derivative df/dt inside the Kth frame. We can then find the phase by integrating equation (5.5):

$$\phi(t) = \phi(T_k) + 2\pi f(T_k)(t - T_k) + 2\pi F_c[f(T_{K+1}) - f(T_K)]\frac{(t - T_K)^2}{2},$$

$$\phi((K-1)\cdot \text{SpF} + n) = \phi(K) + 2\pi\frac{f(K)n}{F_s} + \pi\frac{f(K+1) - f(K)}{\text{SpF}\cdot F_s}n^2, \tag{5.6}$$

where $n = 0 \ldots (\text{SpF} - 1)$ spans the frame. In summary, equation (5.6) computes ϕ at sample rate, given the frame rate frequencies. The key ingredient of this derivation is the linear interpolation (5.5).

M-5.3

Realize the `sinosc(t0,a,f,ph0)` function that we have used in M-5.2. Use equation (5.6) to compute the phase given the frequency vector `f`.

M-5.3 Solution

```
function s = sinosc(t0,a,f,ph0);

global SpF;             %samples per frame
global Fs;              %sampling rate

nframes=length(a);      %total number of frames
if (length(f)==1) f=f*ones(1,nframes); end
if (length(f)~=nframes) %check
   error('f and a must have the same length!');
end

s=zeros(1,nframes*SpF);        %signal vector (initialized to 0)
lastampl=a(1);
lastfreq=f(1);
lastphase=ph0;
for i=1:nframes                  %cycle on the frames
   naux=1:SpF;                   %counts samples within frame
   ampl=lastampl +...            %compute amplitudes within frame
        (a(i)-lastampl)/SpF.*naux;
```

```
   phase=lastphase+pi/Fs.*naux.* ... %compute phases within frame
         (2*lastfreq +(1/SpF)*(f(i)-lastfreq).*naux);
   s(((i-1)*SpF+1):i*SpF)=ampl.*cos(phase); %read from table
   lastampl=a(i);                           %save last values
   lastfreq=f(i);                           %of amplitude,
   lastphase=phase(SpF);                    %frequency, phase
end


s=[zeros(1,round(t0*Fs+1)) s]; %add initial silence of t0 sec.
```

Both the amplitude `a` and frequency `f` envelopes are defined at frame rate and are interpolated at sample rate inside the function body. Note in particular the computation of the `phase` vector within each frame.

5.2.1.3 Generatori di inviluppi

In ogni linguaggio di sintesi subito dopo l'oscillatore sinusoidale si incontra, per importanza, la famiglia dei generatori di funzioni di inviluppi. Ad esempio un inviluppo d'ampiezza puo' essere descritto da una spezzata composta da vari punti connessi da linee rette. In particolare, un inviluppo tipico per suoni musicali e' l'inviluppo *ADSR*: l'andamento temporale dell' ampiezza di un suono e' suddiviso nelle quattro fasi di *Attack*, *Decay*, *Sustain e Release*. Se si vuole cambiare la durata dell'inviluppo, e' bene modificare poco le durate dei tratti corrispondenti all'attacco e decadimento del suono, mentre si puo' variare di piu' il tratto di regime. In questo modo si avranno differenti passi di lettura della tabella o distanza tra le ascisse dei punti generati a seconda di quale porzione dell'inviluppo si sta generando.

L'uso combinato dei due tipi di generatori appena visti (generatore di segnali e generatore di inviluppi) permette di creare agevolmente suoni con inviluppo di ampiezza e curve di altezza molto articolati.

M-5.4

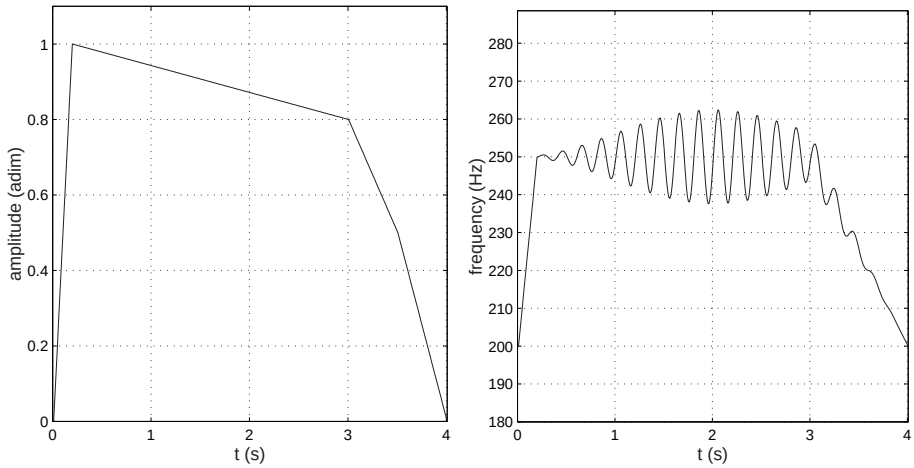
Write a function that realizes a line-segment envelope generator. The input to the function are a vector of time instants and a corresponding vector of envelope values.

M-5.4 Solution

```
function env = envgen(t,a,method);     %t vector of time instants
                                       %a vector of envelope vaues
global SpF;                            %samples per frame
global Fs;                             %sampling rate

if (nargin<3)
   method='linear';
end

frt=floor(t*Fs/SpF+1);                 %times instants as frame numbers
nframes=frt(length(frt));              %total number of frames
env=interp1(frt,a,[1:nframes],method); %linear interpolation
```



(a) (b)

Figure 5.2: Segnali di controllo di (a) ampiezza e (b) frequenza

A fronte della descrizione della forma di inviluppo con istanti temporali (in secondi) e la relativa ampiezza, la funzione genera la funzione al frame rate. Si noti che la funzione di interpolazione `interp1` permette di avere facilmente interpolazioni cubiche o $spline$.

M-5.5

Synthesize a modulated sinusoid using the functions `sinosc` and `envgen`.

M-5.5 Solution

```
%%% headers %%%
%[...]

%%% define controls %%%
a=envgen([0,.2,1,1.5,2],[0,1,.8,.5,0],'linear');  %ADSR amp. envelope
f=envgen([0,.2,1,2],[200,250,250,200],'linear');   %pitch envelope
f=f+max(f)*0.05*...                %pitch envelope with vibrato added
      sin(2*pi*5*(SpF/Fs)*[0:length(f)-1]).*hanning(length(f))';

%%% compute sound %%%
s=sinosc(0,a,f,0);
```

In fig. 5.2 sono illustrati i segnali di controllo `a` e `f`.

5.2.1.4 Generatori di rumori

Generazione di numeri aleatori .

Per generare un rumore si ricorre ai generatori di numeri pseudo-casuali. Ci sono molti metodi

e nessuno e' soddisfacente sotto tutti gli aspetti. Il metodo piu' diffuso si chiama congruenziale lineare e puo' generare sequenze piuttosto lunghe di numeri aleatori prima di ripetersi periodicamente. Dato un valore iniziale (seme) $I(0)$ nell'intervallo $0 \leq I(0) < m$, l'algoritmo di generazione si basa sulla ricorrenza

$$\begin{aligned} I(n) &= [aI(n-1)+c] mod(m) \\ s(n) &= I(n)/m \end{aligned}$$

dove a e c sono due costanti che devono essere scelte accuratamente in relazione al valore di m, per riuscire ad avere la sequenza di lunghezza massima. I numeri generati $s(n)$ sono uniformemente distribuiti nell'intervallo $0 \leq s(n) < 1$. Su questo intervallo la densita' di probabilita' e' piatta. Pertanto la media vale $E[u] = 1/2$ e la varianza $\sigma_u^2 = 1/12$. Per avere una sequnza a media nulla si fa $u(n) = s(n) - 0.5$. Questa sequenza corrisponde ad un rumore bianco in quanto i numeri generati possono essere considerati mutualmente indipendenti e la densita' spettrale di potenza e' data da $S(f) = \sigma_u^2$. Pertanto, essendo $S(f)$ piatto, la sequenza contiene tutte le frequenza in ugual misura e presenta parimenti variazioni lente e veloci.

Generazione di rumori a bassa frequenza .

Se si desidera che la sequenza vari piu' lentamente, si puo' generare un nuovo numero aleatorio ogni d campioni e mantenedo il precedente nell'intervallo (*holder*) o facendo interpolazione lineare tra i due valori generati. In questo caso la spettro di potenza e' dato da:

$$S(f) = |H(f)|^2 \frac{\sigma_u^2}{d}$$

con

$$|H(f)| = \left| \frac{\sin(\pi f d/F_s)}{\sin(\pi f/F_s)} \right|$$

nel caso dell'holder e

$$|H(f)| = \frac{1}{d} \left[\frac{\sin(\pi f d/F_s)}{\sin(\pi f/F_s)} \right]^2$$

nel caso di interpolazione lineare.

Generazione di rumori $1/f$.

Un rumore $1/f$, chiamato anche rumore rosa, e' caratterizzato da uno spettro di potenza $S(f)$ che decresce in frequenza secondo un andamento proporzionale a $1/f$

$$S(f) = \frac{A}{f} \tag{5.7}$$

In genere, per evitare un valore infinito a $f = 0$, si considera questa espressione valida per $f \geq f_{min}$, dove f_{min} e' la minima frequenza desiderata. Lo spettro (5.7) e' caratterizzato da un decadimento di 3 dB per ottava, cioe' quando la frequenza raddoppia, lo spettro di potenza si dimezza. L'ammontare di potenza contenuta in un intervallo di frequenza $[f_1, f_2]$ e'

$$\int_{f_1}^{f_2} S(f) df = A \ln(\frac{f_1}{f_2})$$

Questo implica che l'ammontare di potenza contenuta in ogni intervallo di ottava e' sempre la stessa. Il rumore $1/f$ e' presente in molti fenomeni naturali ed e' legato ai fenomeni frattali.

Nell'audio e' conosciuto come rumore rosa, per differenziarlo dal rumore bianco. Esso rappresenta l'equivalente psicoacustico del rumore bianco, in quanto contiene all'incirca la stessa potenza per ogni banda critica. In senso fisico esso dipende da processi che si evolvono su differenti scale temporali. Un modello per generare rumore $1/f$ consiste nella somma di vari rumori bianchi, ciascuno filtrato attraverso un filtro passa-basso del primo ordine e con costante di tempo via via crescente in progressione geometrica. Una variante consiste nel prendere la media di vari generatori y_i a tenuta di numeri aleatori con periodo di rinnovamento $d_i = 2^i$; cioe'

$$y(n) = \frac{1}{M} \sum_{i=1}^{M} y_i(n) \tag{5.8}$$

Lo spettro di potenza di (5.8) non ha esattamente un andamento del tipo $1/f$, ma lo approssima per frequenze $f \geq F_s/2^M$.

5.2.2 Campionamento

5.2.2.1 Definizioni e applicazioni

Trovare un modello matematico che imiti fedelmente un suono reale e' un compito estremamente difficile. Se pero' esiste un suono di riferimento, allora e' sempre possibile riprodurlo dopo averlo registrato digitalmente mediante campionamento (*sampling*). Tale metodo, anche se semplice nei suoi principi, e' molto usato negli strumenti musicali digitali e, appunto, nei campionatori. I campionatori infatti memorizzano una grande quantita' di esempi di suoni completi, usualmente prodotti da strumenti musicali reali. Quando si vuole sintetizzare un suono, basta scegliere uno dei suoni del repertorio memorizzati e riprodurlo direttamente. Ne risulta quindi una alta efficienza computazionale e una grande fedelta' al suono originale.

In molti casi tale tecnica viene presentata come un mezzo per riprodurre suoni naturali ed e' valutata facendo riferimento agli strumenti originali. Per questo essa e' molto usata nelle tastiere commerciali per produrre suoni imitativi degli strumenti meccanici, come ad esempio organo o piano elettronici. Naturalmente il metodo di campionamento non puo' realizzare tutte le possibilita' espressive degli strumenti originali. D'altra parte si puo' notare che i suoni memorizzati possono essere sintetici o derivare da modificazioni di altri suoni. Questo amplia le possibili applicazioni del metodo. Dal punto di vista della storia della musica, questo metodo rappresenta una versione attualizzata della Musica Concreta. Questo tipo di musica, nata a Parigi nel 1950 per opera soprattutto di Pierre Schaefer, inizio' ad usare come materiale sonoro delle composizioni musicali suoni di qualsiasi tipo registrati da microfono e poi eventualmente manipolati.

5.2.2.2 Elaborazioni: pitch shift, looping

Le possibilita' di elaboraazione sono piuttosto ridotte e sono spesso legate alla metafora del registratore a nastro o moviola. La modificazione piu' frequente consiste nel cambiare la frequenza del suono, variando la frequenza di lettura dei campioni. Non sono consigliabili grandi variazioni di frequenza, in quanto la compressione o espansione temporale di una forma d'onda produce un cambiamento inverso della scale delle frequenze e quindi un'espansione o compressione dello spettro. Tale fatto tende a produrre un risultato innaturale dal punto di vista timbrico, esattamente come accade se viene variata la velocita' di lettura di un nastro magnetico. E' pertanto necessario limitare le variazioni a pochi semitoni ed avere quindi molti suoni campionati distribuiti lungo la scala musicale. Speciale cura va posta in questo caso per non avere suoni adiacenti troppo diversi. Con un insieme di suoni (ad

esempio tre per ottava) e con la variazione di lettura dei campioni e' quindi possibile riprodurre tutta la gamma di altezze desiderate.

M-5.6

Import a .wav file of a single instrument tone. Scale it (compress and expand) to different extents and listen to the new sounds. Up to what scaling ratio are the results acceptable?

Spesso si vuole inoltre variare il suono anche in funzione di altri parametri, ad esempio l'intensita'. Per ottenere una variazione di intensita' non basta infatti cambiare l'ampiezza del suono, ma bisogna anche modificare timbricamente il suono. Tipicamente i suoni piu' intensi sono caratterizzati da un attacco piu' rapido e da una maggiore estensione dello spettro. In tal caso o si utilizza un unico prototipo (ad esempio registrato fortissimo) e poi lo si trasforma (ad esempio mediante filtraggio) per ottenere le altre intensita', o si ricorre ad insiemi diversi di note registrate per differenti valori del parametro (ad esempio con dinamica fortissimo, mezzo forte, pianissimo) e poi si procede a creare le varie sfumature con interpolazioni e/o ulteriori trasformazioni. In definitiva questa tecnica e' caratterizzata da alta efficienza computazionale e alta qualita' imitativa, ma bassa flessibilita' per i suoni non inizialmente previsti nel repertorio o non facilmente riconducibili a esso con semplici trasformazioni.

Per maggior efficienza nell'uso della memoria, spesso si ricorre a memorizzare solo parte del regime stazionario del suono e a ripeterlo (*looping*) nella sintesi. Naturalmente la ripetizione non deve essere di un segmento troppo breve per evitare un carattere troppo statico del suono. Ad esempio per allungare la durata di un suono, dopo che e' passato l'attacco si puo' ripetere ciclicamente la parte individuata finche' non si vuole terminare il suono. A quel punto si emette la parte finale del suono memorizzato. Per creare un ciclo senza artefatti, bisogna porre molta cura nello scegliere i punti di inizio e fine del ciclo. In genere si sceglie un numero intero di periodi inizianti con valore nullo in modo da non avere discontinuita' ne' di ampiezza ne' di fase. Queste discontinuita' infatti sono fastidiose all'ascolto.

M-5.7

Import a .wav file of a single instrument tone. Find the stationary (sustain) part, isolate a section, and perform the looping operation. Listen to the results, and listen to the artifacts when the looped section does not start/end at zero-crossings.

Spesso si individuano nel regime alcuni brevi tratti significativi e nella sintesi si procede ad una interpolazione (*cross-fade*) tra i successivi tratti. In questo modo l'evoluzione temporale lungo la durata del suono puo' essere meglio controllata.

5.2.3 Sintesi granulare

La sintesi granulare si basa sulla successione di forme d'onda di breve durata (tipicamente da 1 a 100 msec) chiamate *grani*. Da questo punto di vista un grano e' un breve evento acustico la cui durata e' prossima alle soglie di discriminazione della durata, frequenza e intensita' nella percezione uditiva. E' un po' come nel cinema dove la successione veloce di immagini statiche, produce la sensazione di movimento. Questa idea base si articola poi in due casi principali a seconda della forma d'onda del grano.

5.2.3.1 Granulazione di suoni reali

Nel primo, forme d'onda complesse, prese da suoni reali o descritte come spettri, si susseguono in parte sovrapponendosi nel metodo chiamato Overlap and Add (OLA). Si possono cosi' sia riprodurre

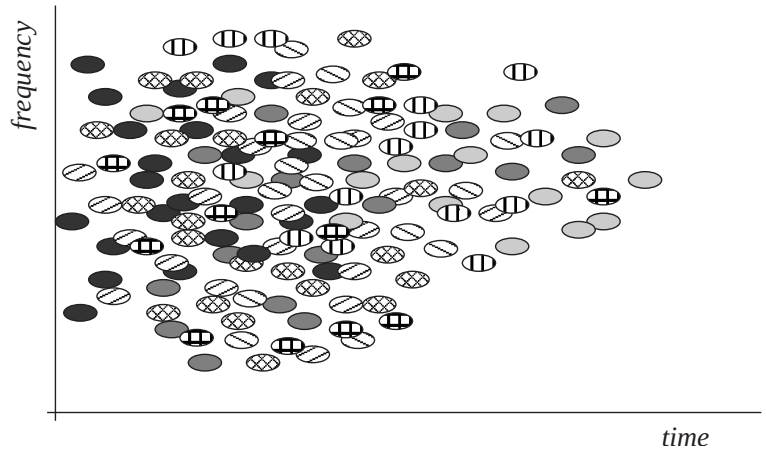


Figure 5.3: Classica rappresentazione grafica di sintesi granulare con grani ricavati da sorgenti diverse e mescolati in modo aleatorio nel tempo.

fedelmente suoni che modificarli nelle caratteristiche dinamiche. Come nella sintesi additiva era importante la coordinazione nella scelta delle frequenze, cosi qui e' importante curare l'allineamento temporale dei grani, allo scopo di evitare fenomeni di discontinuita' di fase, che producono effetti acustici poco gradevoli. Questo rende spesso il controllo difficile. Un esempio d'uso e' nella sintesi della componente stocastica del segnale; in questo caso viene infatti solo controllata l'evoluzione dell'inviluppo spettrale. A questo scopo per ogni frame si ricorre alla trasformata di Fourier inversa, mediante FFT, di uno spettro il cui modulo e' definito dall'inviluppo spettrale e la fase viene creata da un generatore di numeri casuali. Ogni frame viene poi moltiplicato per una finestra prima di fare l'OverLap-Add, cioe' la somma dei vari frames con parziale sovrapposizione temporale. Si puo' usare questo approccio anche come metodo di trasformazione di suoni campionati (granulazione di suoni). In questo caso i grani vengono costruiti prelevando piccole parti di un suono, registrato precedentemente o acquisito direttamente da un convertitore D/A, e poi applicando ad ognuna un inviluppo di ampiezza. Questi grani poi vengono emessi con ordine scelto dal compositore, ad esempio con velocita' variabile o mescolandoli con ordine casuale. I grani possono anche essere scelti da suoni diversi e poi emessi in modo interlacciato, creando quindi tessiture intermedie (fig. 5.3).

5.2.3.2 Grani sintetici

Nel secondo tipo invece si usano come grani funzioni gaussiane (o in generale di tipo passabasso) modulate in frequenza, in modo da localizzare l'energia nel piano tempo-frequenza. Abbiamo qui invece un'analogia con il mosaico, dove l'analogo del grano e' la singola tessera monocromatica e la giustapposizione di tessere di colori diversi fornisce un'immagine complessa. In questo caso la forma d'onda dell'i-esimo grano e' data da

$$g_i(n) = w_i(n) \cdot \cos\left(2\pi \frac{f_i}{F_s} n + \phi_i\right)$$

dove $w_i(n)$ e' una finestra di durata N_i campioni. La formula di sintesi e' data da

$$s(n) = \sum_i a_i \cdot g_i(n - n_i)$$

dove a_i e' un coefficiente di ampiezza dell'i-esimo grano e n_i e' il suo riferimento temporale. Ogni grano da' quindi un contributo di energia concentrato sul punto (n_i, f_i) nel piano tempo-frequenza .

Quando i grani sono collocati regolarmente su una griglia nel piano tempo frequenza, essa diventa una realizzazione della sintesi da analisi tempo-frequenza tipo STFT. In questo caso l'analogia e' l'immagine a colori sullo schermo di un computer, composta da una griglia di pixel di tre colori. Quando invece i grani sono sincroni con il periodo del segnale, si ha la cosiddetta sintesi granulare sincrona con il periodo, che fa riferimento alla sintesi sottrattiva come filtraggio di un segnale quasi periodico. Infatti ogni grano puo' essere interpretato come la risposta all'impulso di un filtro FIR e quindi il risultato puo' essere interpretato come un treno periodico di impulsi che eccita un banco di filtri FIR tempo varianti. Questa interpretazione fornisce anche i criteri per la scelta delle forme d'onda dei grani.

Il caso piu' importante e classico di sintesi granulare e' quando invece i grani semplici sono distribuiti in modo irregolare (*asynchronous granular synthesis*). Per esempio distribuendo casualmente i grani dentro una maschera che delimita una particolare regione nello spazio tempo-frequenza-ampiezza si ottiene come risultato una nuvola di microsuoni o tessitura (*texture*) musicale che varia nel tempo. Si puo' inoltre controllare la densita' dei grani dentro la maschera. Vengono cosi' modellati suoni articolati dove non interessa controllare esattamente la microstruttura. Si evitano cosi i problemi del controllo dettagliato delle caratteristiche temporali dei grani. La durata dei grani influenza la tessitura sonora: durate brevi danno un carattere scoppiettante, esplosivo, mentre durate piu' lunghe danno un'impressione molto piu' sfumata. Quando i grani vengono distribuiti in una larga regione frequenziale, la nuvola ha un carattere massiccio, mentre se la banda e' stretta, ne risulta un suono dotato di altezza propria. Densita' sparse di grani danno un effetto puntinistico.

5.3 Additive synthesis techniques

The idea of additive synthesis is to produce complex sounds through superposition of elementary (sinusoidal) components. If certain requirements are met (e.g. when the frequencies of each partial are integer multiples of a common value), the resulting signal is perceived as a unitary sound event.

The organ is an example of an acoustic instruments that makes use of additive synthesis techniques: single organ pipes produce spectrally simple sounds, and rich timbrical effects (typically harmonic spectra) are obtained by letting different pipes sound together.

5.3.1 Spectral modeling

Spectral analysis of the sounds produced by musical instruments, or by any physical system, shows that the spectral energy of the sound signals can be interpreted as the sum of two main components: a *deterministic* component that is concentrated on a discrete set of frequencies, and a *stochastic* component that has a broadband characteritics. The deterministic –or sinusoidal– component normally corresponds to the main modes of vibration of the system. The stochastic residual accounts for the energy produced by the excitation mechanism which is not turned into stationary vibrations by the system, and for any other energy component that is not sinusoidal.

As an example, consider the sound of a wind instrument: the deterministic signal results from self-sustained oscillations inside the bore, while the residual noisy signal is generated by the turbulent flow components due to air passing through narrow apertures inside the instrument. Similar considerations apply to other classes of instruments, as well as to voice sounds, and even to non-musical sounds.

In the remainder of this section we discuss the modeling of the deterministic sound signal and

introduce the main concepts of additive synthesis. Later on, in section 5.3.3 we will address the problem of including the stochastic component into the additive model.

5.3.1.1 Deterministic signal component

The term *deterministic* signal means in general any signal that is not noise. The class of deterministic signals that we consider here is restricted to sums of sinusoidal components with varying amplitude and frequency. Amplitude and frequency variations can be noticed e.g. in sound attacks: some partials that are relevant in the attack can disappear in the stationary part. In general, the frequencies can have arbitrary distributions: for quasi-periodic sounds the frequencies are approximately harmonic components (integer multiples of a common fundamental frequency), while for non-harmonic sounds (such as that of a bell) they have non-integer ratios.

The deterministic part of a discrete-time sound signal is therefore modeled by the equation

$$s(n) = \sum_k A_k(n) \cdot \sin\left(2\pi \frac{f_k(n)}{F_s} n + \phi_k\right). \tag{5.9}$$

Equation (5.9) has a great generality and can be used to faithfully reproduce many types of sound, especially in a "synthesis-by-analysis" framework (that we discuss in section 5.3.2 below). However, as already noted, it discards completely the noisy components that are always present in real signals. Another drawback of equation (5.9) is that it needs an extremely large number of control parameters: for each note that we want to reproduce, we need to provide the amplitude and frequency envelopes for all the partials. Moreover, the envelopes for a single note are not fixed, but depend in general on the intensity.

On the other hand, additive synthesis provides a very intuitive sound representation, and this is one of the reasons why it has been one of the earliest popular synthesis techniques in computer music.[1] Moreover, sound transformations performed on the parameters of the additive representation (e.g., time-scale modifications) are perceptually very robust.

5.3.1.2 Time- and frequency-domain implementations

Additive synthesis with equation (5.9) can be implemented either in the time domain or in the frequency domain. The more traditional time-domain implementation uses the digital sinusoidal oscillator in wavetable or recursive form, as discussed in section 5.2.1.1. The instantaneous amplitude and the instantaneous radian frequency of a particular partial are obtained by linear interpolation, as discussed previously. Figure 5.4 provides a block diagram of such a time-domain implementation.

M-5.8

Use the sinusoidal oscillator realized in M-5.3 to synthesize a sum of two sinusoids.

M-5.8 Solution

```
%%% headers %%%
%[...]

%%% define controls %%%
```

[1] Some composers have even used additive synthesis as a compositional metaphor, in which sound spectra are reinterpreted as harmonic structures.

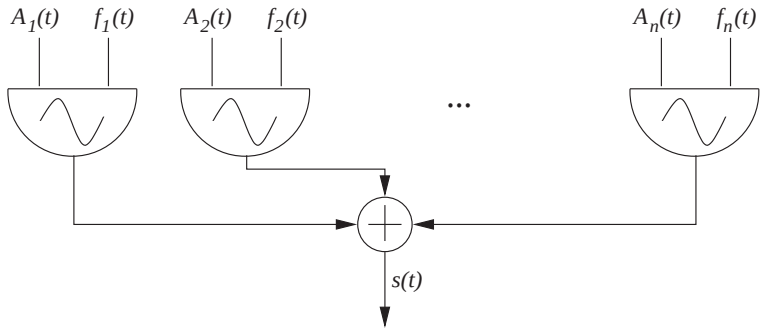


Figure 5.4: Sum of sinusoidal oscillators with time-varying amplitudes and frequencies.

```
a=envgen([0,.5,5,10,15,19.5,20],[0,1,1,1,1,1,0]); %fade in/out
f1=envgen([0,20],[200,200]);          %constant freq. envelope
f2=envgen([0,1,5,10,15,20],...        %increasing freq. envelope
          [200,200,205,220,270,300]);

%%% compute sound %%%
s=sinosc(0,a,f1,0)+sinosc(0,a,f2,0);
```

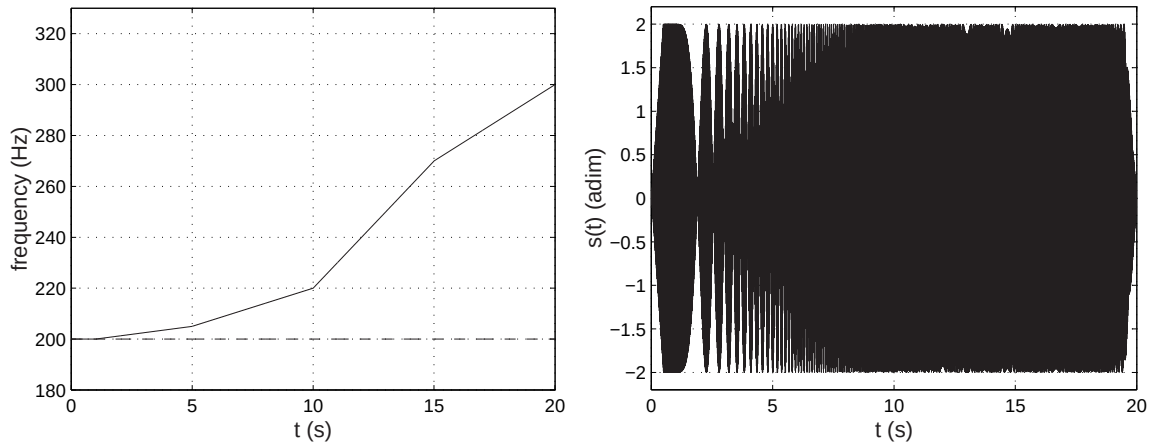
The sinusoidal oscillator controlled in frequency and amplitude is the fundamental building block for time-domain implementations of additive synthesis. Here we employ it to look at the beating phenomenon. We use two oscillators, of which one has constant frequency while the second is given a slowly increasing frequency envelope. Figure 5.5 shows the `f1, f2` control signals and the amplitude envelope of the resulting sound signal: note the beating effect.

In alternative to the time-domain approach, a very efficient implementation of additive synthesis can be developed in the frequency domain, using the inverse FFT. Consider a sinusoid in the time-domain: its STFT is obtained by first multiplying it for a time window $w(n)$ and then performing the Fourier transform. Therefore the transform of the windowed sinusoid is the transform of the window, centered on the frequency of the sinusoid, and multiplied by a complex number whose magnitude and phase are the magnitude and phase of the sine wave:

$$s(n) = A\sin(\omega_0 n/F_s + \phi) \quad \Rightarrow \quad \mathcal{F}[w \cdot s](\omega) = Ae^{i\phi}W(\omega - \omega_0). \tag{5.10}$$

If the window $W(\omega)$ has a sufficiently high sidelobe attenuation, the sinusoid can be generated in the spectral domain by calculating the samples in the main lobe of the window transform, with the appropriate magnitude, frequency and phase values. One can then synthesize as many sinusoids as desired, by adding a corresponding number of main lobes in the Fourier domain and performing an IFFT to obtain the resulting time-domain signal in a frame.

By an overlap-and-add process one then obtains the time-varying characteristics of the sound. Note however that, in order for the signal reconstruction to be free of artifacts, the overlap-and-add procedure must be carried out using a window with the property that its shifted copies overlap and add to give a constant. A particularly simple and effective window that satisfies this property is the triangular window.



(a) (b)

Figure 5.5: Beating effect: (a) frequency envelopes (`f1` dashed line, `f2` solid line) and (b) envelope of the resulting signal.

The FFT-based approach can be convenient with respect to time-domain techniques when a very high number of sinusoidal components must be reproduced: the reason is that the computational costs of this implementation are largely dominated by the cost of the FFT, which does not depend on the number of components. On the other hand, this approach is less flexible than the traditional oscillator bank implementation, especially for the instantaneous control of frequency and magnitude. Note also that the instantaneous phases are not preserved using this method. A final remark concerns the FFT size: in general one wants to have a high frame rate, so that frequencies and magnitudes need not to be interpolated inside a frame. At the same time, large FFT sizes are desirable in order to achieve good frequency resolution and separation of the sinusoidal components. As in every short-time based processes, one has to find a trade-off between time and frequency resolution.

5.3.2 Synthesis by analysis

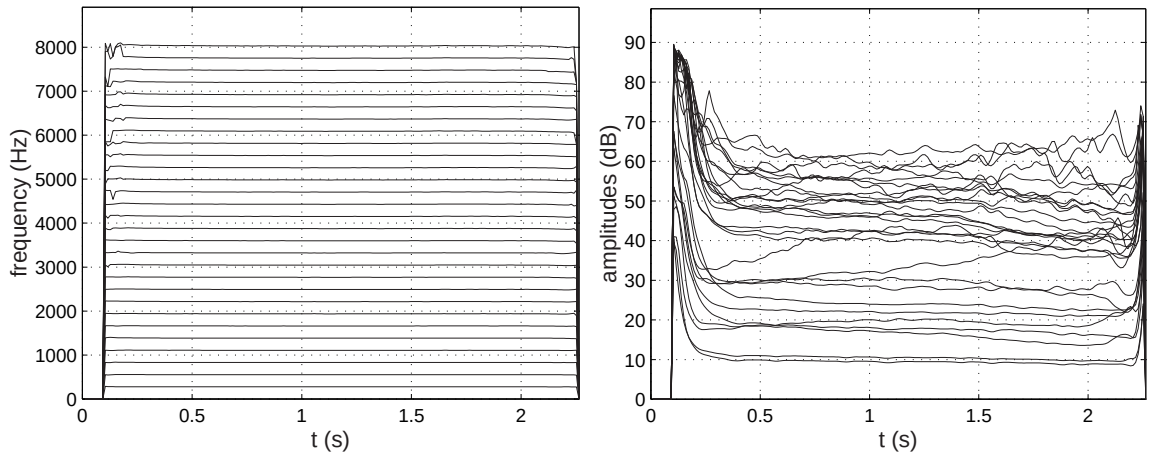
As already remarked, additive synthesis allows high quality sound reproduction if the amplitude and frequency control envelopes are extracted from Fourier analysis of real sounds. Figure 5.6 shows the result of this kind of analysis, in the case of a saxophone tone. Using these data, additive resynthesis is straightforward.

M-5.9

Assume that the script `sinan` imports two matrices `sinan_freqs` and `sinan_amps` with the partial frequency and amplitude envelopes of an analyzed sound. Resynthesize the sound.

M-5.9 Solution

```
%%% headers %%%
% [...]
```



(a) (b)

Figure 5.6: Fourier analysis of a saxophone tone: (a) frequency envelopes and (b) amplitude envelopes of the sinusoidal partials, as functions of time.

```
%%% define controls %%%
readsan;         %import analysis matrices sinan_freqs and sinan_amps
npart=size(sinan_amps,1);  %number of analyzed partials

%%% compute sound %%%
s=sinosc(...          %generate first partial
        0.5,sinan_amps(1,:),sinan_freqs(1,:),0);
for (i=2:npart) %generate higher partials and sum
  s=s+sinosc(0.5,sinan_amps(i,:),sinan_freqs(i,:),0);
end
```

5.3.2.1 Magnitude and Phase Spectra Computation

The first step of any analysis procedure that tracks frequencies and amplitudes of the sinusoidal components is the frame-by-frame computation of the sound magnitude and phase spectra. This is carried out through short-time Fourier transform. The subsequent tracking procedure will be performed in this spectral domain. The control parameters for the STFT are the window-type and size, the FFT-size, and the frame-rate. These must be set depending on the sound to be processed.

Note that the analysis step is completely independent from the synthesis, therefore the observations made in section 5.3.1.2 about FFT-based implementations (the window must overlap and add to a constant) do not apply here. Good resolution of the spectrum is needed in order to correctly resolve, identify, and track the peaks which correspond to the deterministic component.

If the analyzed sound is almost stationary, long windows (i.e. windows that cover several periods) that have good side-lobe rejection can be used, with a consequent good frequency resolution. Unfortunately most interesting sounds are not stationary and a compromise is required. For harmonic sounds one can scale the actual window size as a function of pitch, thus achieving a constant time-frequency

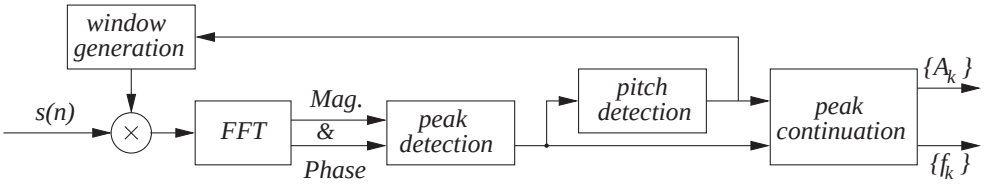


Figure 5.7: Block diagram of the sinusoid tracking process, where $s(n)$ is the analyzed sound signal and $A_k,\ f_k$ are the estimated amplitude and frequency of the kth partial in the current analysis frame.

trade-off. For inharmonic sounds the size should be set according to the minimum frequency difference that exists between partials.

The question is now how to perform automatic detection and tracking of the spectral peaks that correspond to sinusoidal components. In section 5.3.2.2 below we present the main guidelines of a general analysis framework, which is summarized in figure 5.7. First, the FFT of a sound frame is computed according to the above discussion. Next, the prominent spectral peaks are detected and incorporated into partial trajectories. If the sound is pseudo-harmonic, a pitch detection step can improve the analysis by providing information about the fundamental frequency information, and can also be used to choose the size of the analysis window.

Such a scheme is only one of the possible approaches that can be used to attack the problem. Hidden Markov Models (HMMs) are another one: a HMM can optimize groups of peaks trajectories according to given criteria, such as frequency continuity. This type of approach might be very valuable for tracking partials in polyphonic sounds and complex inharmonic tones.

5.3.2.2 A sinusoid tracking procedure

We now discuss in more detail the analysis steps depicted in figure 5.7. The first one is detection of the most prominent frequency peaks (i.e., local maxima in the magnitude spectrum) in the current analysis frame. Real sounds are not periodic, do not have clearly spaced and defined spectral peaks, exhibit interactions between components. Therefore, the best one can do at this point is to detect as many peaks as possible and postpone to later analysis steps the decision of which ones actually correspond to sinusoidal components. The peaks are then searched by only imposing two minimal constraints: they have to lie within a given frequency range, and above a given magnitude threshold. The detection of very soft peaks is hard: they have little resolution, and measurements are very sensitive to transformations because as soon as modifications are applied to the analysis data, parts of the sound that could not be heard in the original can become audible. Having a very clean sound with the maximum dynamic range, the magnitude threshold can be set to the amplitude of the background noise floor. In order to gain better resolution in the high frequency range, the sound may be pre-processed to introduce preemphasis, which has then to be compensated later on before the resynthesis.

After peak detection, many procedures can be used to decide whether a peak belongs to a sinusoidal partial or not. One possible strategy is to measure how close the peak shape is to the ideal sinusoidal peak (recall what we said about the transform of a windowed sinusoid and in particular equation (5.10). A second valuable source of additional information is pitch. If a fundamental frequency is actually present, it can be exploited in two ways. First, it helps the tracking of partials. Second, the size of the analysis window can be set according to the estimated pitch in order to keep the number of periods-per-frame constant, therefore achieving the best possible time-frequency trade-off (this is an example of a *pitch-synchronous* analysis). There are many possible pitch detection

strategies, which we do not want to discuss here.

A third and fundamental strategy for peak selection is to implement some sort of peak *continuation* algorithm. The basic idea is that a set of "guides" advance in time and follow appropriate frequency peaks (according to specified constraints that we discuss in the next paragraph) forming trajectories out of them. A guide is therefore an abstract entity which is used by the algorithm to create the trajectories, and the trajectories are the actual result of the peak continuation process. The guides are turned on, advanced, and finally turned off during the continuation algorithm, and their instantaneous state (frequency and magnitude) is continuously updated during the process. If the analyzed sound is harmonic and a fundamental has been estimated, then the guides are created at the beginning of the analysis, with frequencies set according to the estimated harmonic series. When no harmonic structure can be estimated, each guide is created when the first available peak is found. In the successive analysis frames, the guides modify their status depending on the last peak values. This past information is particularly relevant when the sound is not harmonic, or when the harmonics are not locked to each other and we cannot rely on the fundamental as a strong reference for all the harmonics.

The main constraints used to assign guides to spectral peaks are as follows. A peak is assigned to the guide that is closest to it and that is within an assigned frequency deviation. If a guide does not find a match, the corresponding trajectory can be turned off, and if a continuation peak is not found for a given amount of time the guide is killed. New guides and trajectories can be created starting from peaks of the current frame that have high magnitude and are not "claimed" by any of the existing trajectories. After a certain number of analysis frames, the algorithm can look at the trajectories created so far and adopt corrections: in particular, short trajectories can be deleted, and small gaps in longer trajectories can be filled by interpolating between the values of the gap edges.

One final refinement to this process can be added by noting that the sound attack is usually highly non-stationary and noisy, and the peak search is consequently difficult in this part. Therefore it is customary to perform the whole procedure backwards in time, starting from the end of the sound (which is usually a more stable part). When the attack is reached, a lot of relevant information has already been gained and non-relevant peaks can be evaluated and/or rejected.

5.3.3 "Sines-plus-noise" models

At the beginning of our discussion on additive modeling, we remarked that the spectral energy of the sound signals has a *deterministic* component that is concentrated on a discrete set of frequencies, and a *stochastic* component that has a broadband characteritics. So far we have discussed the problem of modeling the deterministic –or sinusoidal– component. Now we have to include the stochastic component into the model.

A sinusoidal representation may in principle be used also to simulate noise, since noise consists of sinusoids at every frequency within the band limits. It is clear however that such a representation would be computationally very demanding. Moreover it would not be a *flexible* sound representation. Therefore the most convenient sound model is of the form

$$s(n) = \sum_k A_k(n) \cdot \sin\left(2\pi \frac{f_k(n)}{F_s} n + \phi_k\right) + e(n), \tag{5.11}$$

where $e(t)$ represents the stochastic component and is modeled separately from the deterministic part.

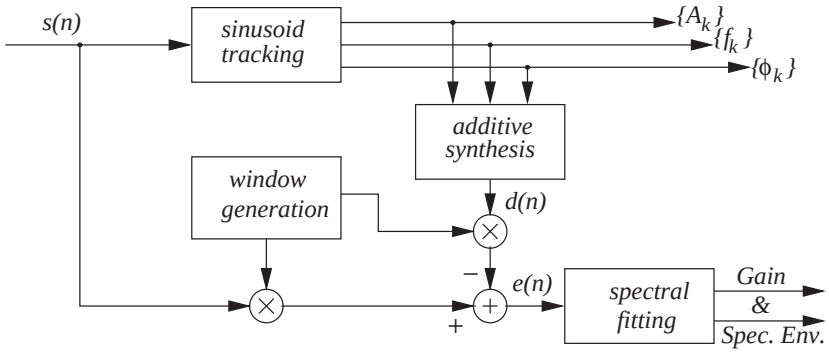


Figure 5.8: Block diagram of the stochastic analysis and modeling process, where $s(n)$ is the analyzed sound signal and $A_k,\ f_k,\ \phi_k$ are the estimated amplitude, frequency, and phase of the kth partial in the current analysis frame.

5.3.3.1 Stochastic analysis

The most straightforward approach to estimation of the stochastic component is through *subtraction* of the deterministic component from the original signal. Subtraction can be performed either in the time domain or in the frequency domain. Time domain subtraction must be done while preserving the phases of the original sound, and instantaneous phase preservation can be computationally very expensive. One the other hand, frequency-domain subtraction does not require phase preservation. However, time-domain subtraction provides much better results, and is usually favored despite the higher computational costs. For this reason we choose to examine time-domain subtraction in the remainder of this section. Figure 5.8 provides a block diagram.

Suppose that the deterministic component has been estimated in a given analysis frame, using for instance the general scheme described in section 5.3.2 (note however that in this case the analysis should be improved in order to provide estimates of the instantaneous phases as well). Then the first step in the subtraction process is the time-domain resynthesis of the deterministic component with the estimated parameters. This should be done by properly interpolating amplitude, frequency, and phase values in order to avoid artifacts in the resynthesized signal. The actual subtraction can be performed as

$$e(n) = w(n) \cdot [s(n) - d(n)], \qquad n = 0, 1, \ldots, N-1, \tag{5.12}$$

where $s(n)$ is the original sound signal and $d(n)$ is the re-synthesized deterministic part. The difference $(s - d)$ is multiplied by an analysis window w of size N, which deserves some discussion.

We have seen in 5.3.2 that high frequency resolution is needed for the deterministic part, and for this reason long analysis windows are used for its estimation. On the other hand, good time resolution is more important for the stochastic part of the signal, especially in sound attacks, while frequency resolution is not a major issue for noise analysis. A way to obtain good resolutions for both the components is to use two different analysis windows. Therefore w in equation (5.12) is not in general the same window used to estimate $d(n)$, and the size N is in general small.

Once the subtraction has been performed, there is one more step than can be used to improve the analysis, namely, test can be performed on the estimated residual in order to assess how good the analysis was. If the spectrum of the residual still contains some partials, then the analysis of the deterministic component has not been performed accurately and the sound should be re-analyzed until the residual is free of deterministic components. Ideally the residual should be as close as possible to

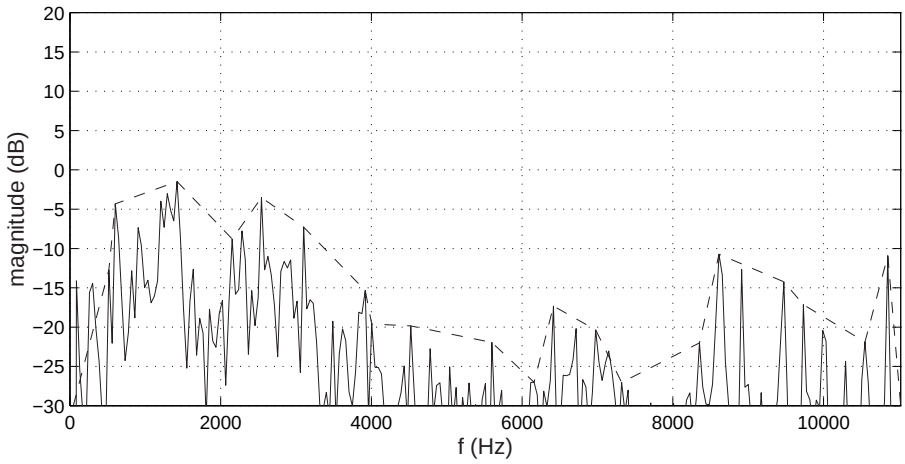


Figure 5.9: Example of residual magnitude spectrum (solid line) and its line-segment approximation (dashed line), in an analysis frame. The analyzed sound signal is the same saxophone tone used in figure 5.6.

a stochastic signal, therefore one possible test is a measure of correlation of the residual samples.[2]

5.3.3.2 Stochastic modeling

The assumption that the residual is a stochastic signal implies that it is fully described by its amplitude and its spectral envelope characteristics. Information on the instantaneous phase is not necessary. Based on these considerations, a frame of the stochastic residual can be completely characterized by a filter that models the amplitude and general frequency characteristics of the residual. The representation of the residual for the overall sound will then be a time-varying filter.

Within a given frame we therefore assume that $e(t)$ can be modeled as

$$E(\omega) = H(\omega)U(\omega), \tag{5.13}$$

where U is white noise and H is the frequency response of filter whose coefficients vary on a frame-by-frame basis. The stochastic modeling step is summarized in the last block of figure 5.8.

The filter design problem can be solved using different strategies. One approach that is often adopted uses some sort of curve fitting (line-segment approximation, spline interpolation, least squares approximation, and so on) of the magnitude spectrum of e in an analysis frame. As an example, line-segment approximation can be obtained by stepping through the magnitude spectrum, finding local maxima at each step, and connecting the maxima with straight lines. This procedure can approximate the spectral envelope with reasonable accuracy, depending on the number of points, which in turn can be set depending on the sound complexity. See figure 5.9 for an example.

Another possible approach to the filter design problem is Linear Predictive Coding (LPC), which is a popular technique in speech processing. However in this context curve fitting procedure on the noise spectrum (e.g., line-segment approximation) are usually considered to be more flexible approaches and are preferred to LPC. We will return on LPC in section 5.4.

The next question is how to implement the estimated time-varying filter in the resynthesis step.

[2] Note that if the analyzed sound has not been recorded in silent and anechoic settings the residual will contain not only the stochastic part of the sound, but also reverberation and/or background noise.

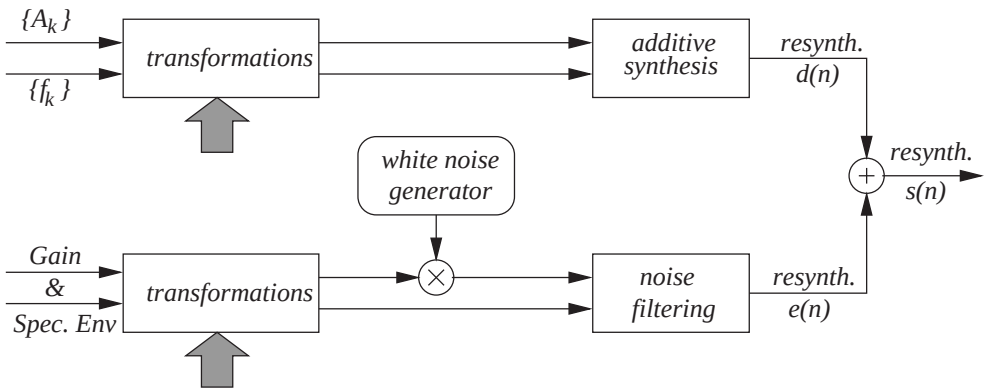


Figure 5.10: Block diagram of the sines-plus-noise synthesis process.

5.3.3.3 Resynthesis and modifications

Figure 5.10 shows the block diagram of the synthesis process. The deterministic signal, i.e., the sinusoidal component, results from the magnitude and frequency trajectories, or their transformation, by generating a sine wave for each trajectory (additive synthesis). As we have seen, this can either be implemented in the time domain with the traditional oscillator bank method or in the frequency domain using the inverse-FFT approach.

Concerning the stochastic component, a frequency-domain implementation is usually preferred to a direct implementation of the time-domain convolution (5.13), due to its computational efficiency[3] and flexibility. In each frame, the stochastic signal is generated by an inverse-FFT of the spectral envelopes. Similarly to what we have seen for the deterministic synthesis in section 5.3.1.2, the time-varying characteristics of the stochastic signal is then obtained using an overlap-and-add process.

In order to perform the IFFT, a magnitude and a phase responses have to be generated starting from the estimated frequency envelope. Generation of the magnitude spectrum is straightforwadly obtained by first linearly interpolating the spectral envelope to a curve with half the length of the FFT-size, and then multiplying it by a gain that corresponds to the average magnitude extracted in the analysis. The estimated spectral envelope gives no information on the phase response. However, since the phase response of noise is noise, a phase response can be created from scratch using a random signal generator. In order to avoid periodicities at the frame rate, new random values should be generated at every frame.

The sines-plus-noise representation is well suited for modification purposes.

- By only working on the deterministic representation and modifying the amplitude-frequency pairs or the original sound partials, many kinds of frequency and magnitude transformations can be obtained. As an example, partials can be transposed in frequency. It is also possible to decouple the sinusoidal frequencies from their amplitude, obtaining pitch-shift effects that preserve the formant structure.

- Time-stretching transformations can obtained by resampling the analysis points in time, thus slowing down or speeding up the sound while maintaining pitch and formant structure. Given

[3] In fact, by using a frequency-domain implementation for both the deterministic and the stochastic synthesis one can add the two spectra and resynthesize both the components at the cost of a sigle IFFT per frame.

the stochastic model that we are using, the noise remains noise and faithful signal resynthesis is possible even with extreme stretching parameters.

- By acting on the relative amplitude of the two components, interesting effects can be obtained in which either the deterministic or the stochastic parts are emphasized. As an example, the amount of "breathiness" of a voiced sound or a wind instrument tone can be adjusted in this way. One must keep in mind however that, when different transformations are applied to the two representations, the deterministic and stochastic components in the resulting signal may not be perceived as a single sound event anymore.
- Sound morphing (or *cross-synthesis* transformations can be obtained by interpolating data from two or more analysis files. This transformations are particularly effective in the case of quasi-harmonic sounds with smooth parameter curves.

5.3.4 Sinusoidal description of transients

So far we have seen how to extend the sinusoidal model by using a "sines-plus-noise" approach that explicitly describes the residual as slowly varying filtered white noise. Although this technique is very powerful, transients do not fit well into a filtered noise description, because they lose sharpness and are smeared. This consideration motivates us to handle transients separately.

One straightforward approach, that is sometimes used, is removing transient regions from the residual, performing the sines-plus-noise analysis, and adding the transients back into the signal. This approach obviously requires memory where the sampled transients must be stored, but since the transient residuals remain largely invariant throughout most of the range of an instrument, only a few residuals are needed in order to cover all the sounds of a single instrument. Although this approach works well, it is not flexible because there is no model for the transients. In addition, identifying transients as everything that is neither sinusoidal nor transient is not entirely correct. Therefore we look for a suitable transient model, that can be embedded in the additive description to obtain a "sines-plus-transients-plus-noise" representation.

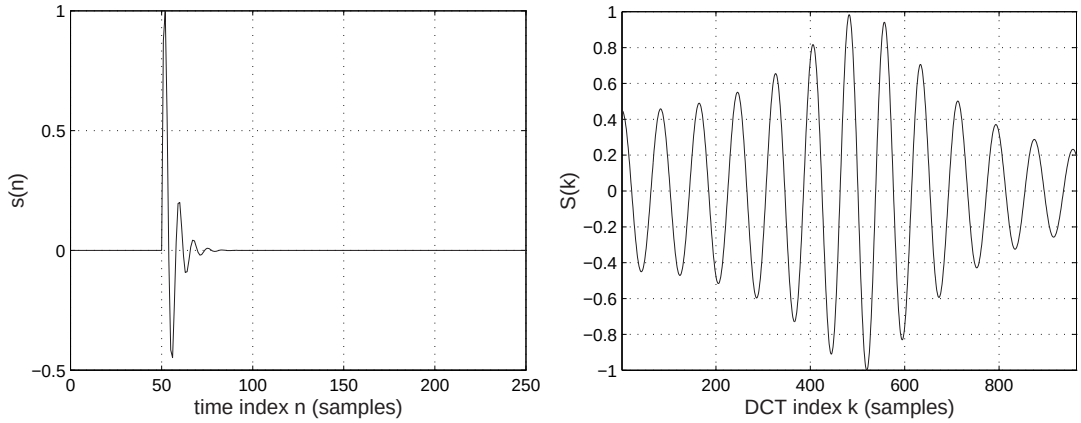
5.3.4.1 The DCT domain

In the following we adopt a further modified version of the additive sound representation (5.9), in which the sound transients are explicitly modeled by an additional signal:

$$s(n) = \sum_k A_k(n) \cdot \sin\left(2\pi \frac{f_k(n)}{F_s} n + \phi_k\right) + e_t(n) + e_r(n), \tag{5.14}$$

where e_t is the signal associated to transients and e_r is the noisy residual. The transient model is based on a main undelying idea: we have seen that a slowly varying sinusoidal signal is impulsive in the frequency domain, and sinusoidal models perform short-time Fourier analysis in order to track slowly varying spectral peaks (the tips of the impulsive signals) over time. Transients are very much dual to sinusoidal components: they are impulsive in the time domain, and consequently they must be oscillatory in the frequency domain. Therefore, although transient cannot be tracked by a short-time analysis (because their STFT will not contain meaningful peaks), we can track them by performing sinusoidal modeling in a properly chosen frequency domain. The mapping that we choose to use is the one provided by the discrete cosine transform (DCT):

$$S(k) = \beta(k) \sum_{n=0}^{N-1} s(n) \cos\left[\frac{(2n-1)k\pi}{2N}\right], \quad \text{for } n, k = 0, 1, \ldots, N-1, \tag{5.15}$$



(a) (b)

Figure 5.11: Example of DCT mapping: (a) an impulsive transient (an exponentially decaying sinusoid) and (b) its DCT as a slowly varying sinusoid.

where $\beta(1) = \sqrt{1/N}$ and $\beta(k) = \sqrt{2/N}$ otherwise. From equation (5.15) one can see that an ideal impulse $\delta(n - n_0)$ (i.e., a Kronecker delta function centered in n_0) is transformed into a cosine whose frequency increases with n_0. Figure 5.11(a) shows a more realistic transient signal, a one-sided exponentially decaying sine wave. Figure 5.11(b) shows the DCT of the transient signal: a slowly varying sinusoid. This considerations suggest that the time-frequency duality can be exploited to develop a transient model: the same kind of parameters that characterize the sinusoidal components of a signal can also characterize the transient components of a signal, although in a different domain.

5.3.4.2 Transient analysis and modeling

Having transformed the transient into the DCT domain, the most natural way to proceed is performing sinusoidal modeling in this domain: STFT analysis of the DCT-domain signal can be used to find meaningful peaks, and then the signal can be resynthesized in the DCT domain and back-transformed to the time domain with an inverse DCT transform (IDCT). This process is shown in figure 5.12. We now discuss the main steps involved in this block diagram.

First the input signal $s(n)$ is divided into non-overlapping blocks in which DCT analysis will be performed. The block length should be chosen so that a transient appears as "short", therefore large block sizes (e.g., 1 s) are usually chosen. The block DCT is followed by a sinusoidal analysis/modeling process which is identical to what we have seen is section 5.3.2. The analysis can optionally embed some information about transient location withing the block: there are many possible transient detection strategies, which we do not want to discuss here. Also, the analysis can perform better if the sinusoid tracking procedure starts from the end of the DCT-domain signal and moves backwards toward the beginning, because the beginning of a DCT frame is usually spectrally rich and this can deteriorate the performance of the analysis (similar considerations were done in section 5.3.2 when discussing sinusoid tracking in the time domain).

The analysis yields parameters that correspond to slowly varying sinusoids in the DCT domain: each transient is associated to a triplet $\{A_k, f_k, \phi_k\}$, amplitude, frequency, and phase of the kth "par-

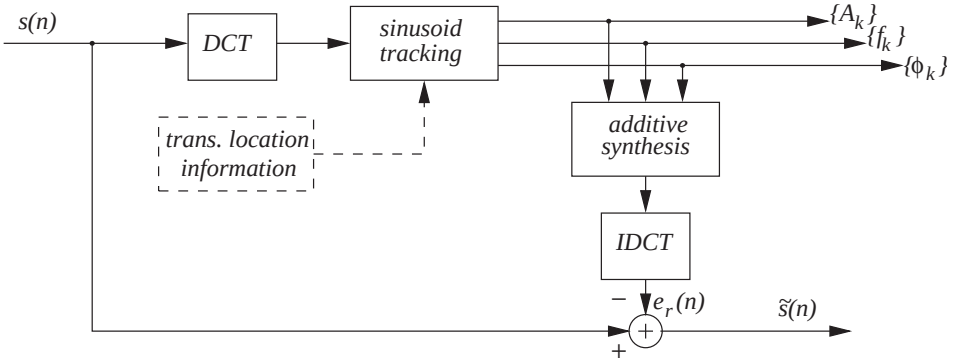


Figure 5.12: Block diagram of the transient analysis and modeling process, where $s(n)$ is the analyzed sound signal and $A_k,\ f_k,\ \phi_k$ are the estimated amplitude, frequency, and phase of the kth DCT-transformed transient in the current analysis frame.

tial" in each STFT analysis frame within a DCT block. By recalling the properties of the DCT one can see that f_k correspond to onset locations, A_k is the amplitude of the time-domain signal also, and ϕ_k is related to the time direction (positive or negative) in which the transient evolves. Resynthesis of the transients is then performed using these parameters to reconstruct the sinusoids in the DCT domain. Finally an inverse discrete cosine transform (IDCT) on each of the reconstructed signals is used to obtain the transients in each time-domain block, and the blocks are concatenated to obtain the transients for the entire signal.

It is relatively straightforward to implement a "fast transient reconstruction" algorithm. Without entering the details, we just note that the whole procedure can be reformulated using FFT transformations only: in fact one could verify that the DCT can be implemented using an FFT block plus some post-processing (multiplication of the real and imaginary parts of the FFT by appropriate cosinusoidal and sinusoidal signals followed by a sum of the two parts). Furthermore, this kind of approach naturally leads to a FFT-based implementation of the additive synthesis step (see section 5.3.1.2).

One nice property of this transient modeling approach is that it fits well within the sines-plus-noise analysis that we have examined in the previous sections. The processing block depicted in figure 5.12 returns an output signal $\tilde{s}(n)$ in which the transient components e_t have been removed by subtraction: this signal can be used as the input to the sines-plus-noise analysis, in which the remaining components (deterministic and stochastic) will be analyzed and modeled. From the implementation viewpoint, one advantage is that the core components of the transient-modeling algorithm (sinusoid tracking and additive resynthesis) are identical to those used for the deterministic model. Therefore the same processing blocks can be used in the two stages, although working on different domains.

5.4 Sintesi sottrattiva

Mentre la sintesi additiva costruisce suoni complessi sommando insieme semplici suoni sinusoidali tempo varianti, la sintesi sottrattiva e' basata sull'idea complementare di passare un segnale a larga banda attraverso un filtro tempo variante per produrre la forma d'onda desiderata. La sintesi sottrattiva trae la sua origine nel campo analogico, dove si usava produrre segnali a partire da forme d'onda semplici, come onde quadre o a dente di sega e poi sagomare lo spettro mediante filtraggio eventualmente variabile. Nel campo numerico si ha il vantaggio di poter controllare in modo molto

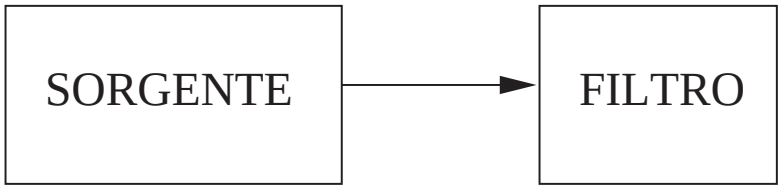


Figure 5.13: Sintesi sottrattiva

piu' preciso i parametri dei filtri. D'altra parte spesso conviene generare direttamente la forma d'onda voluta con altri metodi. Essa pertanto si e' piu' specializzata nell'uso con filtri piu' sofisticati o i cui parametri possano essere stimati a partire da suoni reali. Invece i filtri sono molto usati per produrre trasformazioni di suoni nel post-processing. Mediante filtri cioe' si arricchisce un segnale sintetizzato o registrato di vari effetti, si variano le sue caratteristiche spettrali, pur mantenendo la dinamica intrinseca del suono, si possono produrre effetti di riverberazione e spazializzazione e cosi' via.

5.4.1 Modelli sorgente-filtro

5.4.1.1 Blocchi di generazione

L'interpretazione fisica della sintesi sottrattiva consiste in una sorgente di segnale di eccitazione usata come ingresso di un sistema risonante, in una struttura di tipo feed-forward (fig. 5.13). Questa descrizione si adatta in prima approssimazione a vari strumenti musicali tradizionali. Ad esempio le corde vibranti di un violino sono accoppiate attraverso il ponticello alla cassa risonante, che in questo caso si comporta come filtro tempo invariante. Anche il suono della voce puo essere modellato come una sorgente di eccitazione, che puo' essere di tipo impulsivo data dalle vibrazioni delle corde vocali e rumorosa data dal flusso turbolento dell'aria in qualche costrizione del tratto vocale. Questa sorgente viene trasmessa attraverso il tratto vocale, la cavita' orale, la cavita' nasale, l'apertura delle labbra che filtrano e modificano spettralmente la sorgente, in modo approssimativamente lineare. Va osservato pero' che, nel caso della voce si puo' considerare la sorgente indipendente dal tratto vocale, mentre in molti casi di strumenti musicali per ottenere un modello efficace non si puo' trascurare l'influenza della risonanza sulla sorgente. La sintesi sottrattiva si applica bene nel primo caso, mentre nel secondo e' bene ricorrere alle tecniche della sintesi per modelli fisici.

A seconda della risposta in frequenza del filtro, si puo' variare l'andamento globale dello spettro del segnale in ingresso, ad esempio estraendo una piccola porzione del suo spettro. Se il filtro e' statico, cioe' se i parametri del filtro non variano, resta costante anche il suo effetto. Se invece i parametri sono tempo varianti, cambia anche la risposta in frequenza del filtro. In uscita si avra' una combinazione delle variazioni spettrali nel tempo del segnale in ingresso e di quelle del filtro. I parametri del filtro sono quindi scelti in base alla risposta in frequenza voluta e alla dinamica timbrica desiderata. Se viene usata per la sintesi e' bene che il segnale di ingresso non sia di frequenza fissa, ma abbia ad esempio un po' di tremolo. Solo in questo modo infatti viene percepita la forma dell'inviluppo spettrale a causa delle variazioni d'ampiezza delle varie parziali che seguono l'inviluppo spettrale.

La scomposizione effettuata offre la possibilita' di controllare separatamente le caratteristiche della sorgente da quelle del filtro dando quindi una maggiore flessibilita' nel controllo parametrico e una migliore interpretazione dei parametri di controllo.

5.4.1.2 Applicazioni nel campo audio

Il filtro passa-basso (LP) con risonanza e' usato spesso per simulare l'effetto di strutture risonanti; il filtro passa-alto (HP) invece per rimuovere componenti a bassa frequenza indesiderate; il filtro passa-banda (BP) puo' produrre effetti come imitazione di una linea telefonica, o la sordina in uno strumento musicale; il filtro elimina-banda (BR) puo' dividere lo spettro udibile in due bande separate che sembrino incorrelate. Il filtro risonante puo' essere usato per introdurre risonanze artificiali ad un suono; mentre il filtro notch (che elimina tutte le frequenze in una stretta banda attorno alla frequenza di risonanza) serve per eliminare disturbi quasi sinusoidali come ad esempio i 50 Hz dovuti all'alimentazione dei dispositivi elettronici. Un insieme di filtri notch usati in combinazione sul segnale di ingresso, puo' produrre l'effetto di phasing.

Va segnalato che il filtraggio puo' cambiare molto l'intensita' del suono filtrato. Infatti il filtro puo' produrre l'effetto desiderato, ma il risultato non puo' poi essere usato perche' diventato troppo debole o forte. Il metodo per compensare queste variazioni si chiama normalizzazione. In genere i metodi di normalizzazione impiegano norme del tipo L_1, L_2 e L_∞ sul modulo della risposta in frequenza del filtro. La norma L_1 e' usata quando il filtro non deve essere sovraccaricato in nessuna circostanza. Spesso pero' questo significa attenuare troppo il segnale. La norma L_2 (normalizzazione del valore efficace) e' usata per normalizzare l'intensita' del segnale. Questo metodo e' accurato per segnali a larga banda e adatto in molte applicazioni musicali. La norma L_∞ normalizza la risposta in frequenza rispetto al suo massimo ed e' efficace quando il segnale da filtrare e' sinusoidale o periodico.

Un banco di filtri consiste in un gruppo di filtri che sono alimentati con le stesso segnale. Ciascun filtro e' tipicamente un filtro passa-banda stretto impostato ad una propria frequenza centrale. Spesso i segnali filtrati vengono poi sommati per produrre il suono in uscita. Quando si puo' controllare il livello di ciascun filtro il banco di filtri viene chiamato anche equalizzatore in quanto si puo' usare per compensare una risposta in frequenza non piatta del sistema di trasmissione o riproduzione.

Se si puo' controllare frequenza, banda e livello di ciascun filtro, si ha un sintetizzatore a formanti parallelo. Se le risposte dei singoli filtri non sono troppo sovrapposte, si riesce a controllare separatamente l'andamento dei singoli formanti. Questo puo' essere usato nella sintesi della voce, dove le transizioni tra i formanti devono essere accurate.

La tecnica vista si presta bene a sintetizzare sia gli inviluppi spettrali poco variabili nel tempo, come l'effetto delle casse armoniche o le risposte acustiche ambientali (in questo caso il filtro e' caratterizzato da ritardi consistenti, che vengono meglio interpretati nel tempo, come echi, riverberi o come ripetizioni periodiche del segnale in ingresso), sia gli inviluppi spettrali rapidamente variabili, come gli effetti di sordina, la voce parlata e cantata e i suoni caratterizzati da grande dinamica timbrica. Si osservi che il modello non e' limitato da assunti sulla periodicita' del segnale sorgente, ma anzi puo' utilmente essere impiegato per la simulazione di segnali non intonati, come le percussioni. Per questi ultimi sono normalmente impiegate sorgenti di segnali rumorosi, caratterizzati da spettri continui. In quest'ultimo caso il modello sorgente di rumore bianco - filtro diventa un valido mezzo per descrivere i processi stocastici; esso infatti permette la caratterizzazione dell'inviluppo spettrale, eventualmente considerato tempo-variante, che e' il parametro percettivamente piu' significativo.

5.4.1.3 Implementazione e controllo dei modelli

Dal punto di vista implementativo va detto che l'operazione di filtraggio lineare puo' essere realizzata con diverse strutture che realizzano l'equazione alle differenze sopra vista, oppure come convoluzione con una risposta all'impulso, che di fatto descrive un filtro, o la risposta all'impulso di un ambiente. Una maniera alternativa consiste nel fare il filtraggio in frequenza, facendo il prodotto dello spettro

del segnale, suddiviso in blocchi, con la risposta in frequenza del filtro, e antitrasformando il risultato.

La sintesi sottrattiva fa riferimento ad una interpretazione in frequenza. Le varie tecniche di implementazione dei filtri offrono diverse possibilita' di controllo parametrico, che vanno scelte in base alle applicazioni. Ad esempio per la sintesi di suoni vocalici e' utile poter controllare la frequenza e la banda dei formanti, dove si concentra l'energia e che caratterizza specialmente l'identita' del suono. Inoltre spesso e' utile combinare sorgenti periodiche a sorgenti stocastiche.

Se si possono fare ipotesi semplificative sull'ingresso, e' possibile stimare sia i parametri della sorgente che del filtro a partire da un suono dato. La procedura piu' nota e' il metodo di predizione lineare (LPC) che usa una sorgente composta da treno di impulsi o da rumore bianco ed e' usata per la sintesi della voce. Questo metodo verra' presentato piu' estesamente nel paragrafo 5.4.2. Analizzando una sequenza di segmenti di suono si ottengono parametri tempo varianti che possono essere usati nella sintesi. Il vantaggio di avere un modello parametrico e' che si puo' dare un'interpretazione fisica o spettrale a questi parametri e quindi avere un criterio di riferimento per la loro modificazione, sintetizzando quindi varianti del suono. Per esempio la stima dei parametri LPC della voce fornisce un filtro tempo variante che contiene l'andamento nel tempo dell'inviluppo spettrale e quindi delle formanti. Questi sono parametri particolarmente importanti per la percezione della voce. Per cui essi possono essere modificati in senso spettrale cambiando il carattere della voce, o in senso temporale, facendo quindi una compressione o espansione della scala temporale, oppure si puo' cambiare il pitch della voce senza cambiare il suo inviluppo e quindi mantenendo le caratteristiche della voce originaria. Una possibilita' usata spesso dai musicisti consiste nell'usare il filtro, con parametri stimati sul una voce parlata, applicando all'ingresso suoni d'altro tipo ricchi spettralmente. Vengono cosi' combinate le caratteristiche tempo-frequenza dei due suoni ottenendo, ad esempio, un'orchestra che canta. Questa tecnica viene chiamata sintesi incrociata.

5.4.1.4 Sorgenti e filtri notevoli

In linea di principio, qualsiasi segnale puo' essere usato come sorgente di un algoritmo di sintesi sottrattiva. Esistono tuttavia due generatori di segnale notevoli che, per la ricchezza spettrale dei segnali prodotti, sono considerati particolarmente adatti a questo scopo: il *generatore di rumore* e il *generatore di impulsi*. Il primo produce un segnale non periodico e a spettro continuo su tutta la banda di frequenze. Il secondo produce un segnale periodico con energia distribuita uniformemente su tutta la banda di frequenze disponibile.

M-5.10

Write a function `noisegen(t0,amp)` that realizes the noise generator, and a function `buzz(t0,amp,f)` that realizes the impulse generator. The parameters `(t0,amp,f)` are initial time, amplitude envelope, and frequency envelope, respectively.

M-5.10 Solution

Hint for the noise generator: simply use the function `rand()`.

Hint for the impulse generator: use additive synthesis, and sum up all the harmonic components $\cos(2k\pi f_0 t)$ $(k \in \mathbb{N})$ of the fundamental frequency f_0, up to the Nyquist frequency $F_s/2$. Figure 5.14 shows the resulting signal, in the time and frequency domains

---

La teoria del progetto dei filtri numerici e' argomento vasto e non e' affrontato in questa sede. A titolo di esempio e' invece tratto il caso notevole del progetto di celle IIR del secondo ordine:

$$H(z) = \frac{b_1}{1 + a_2 z^{-1} + a_3 z^{-2}} \tag{5.16}$$

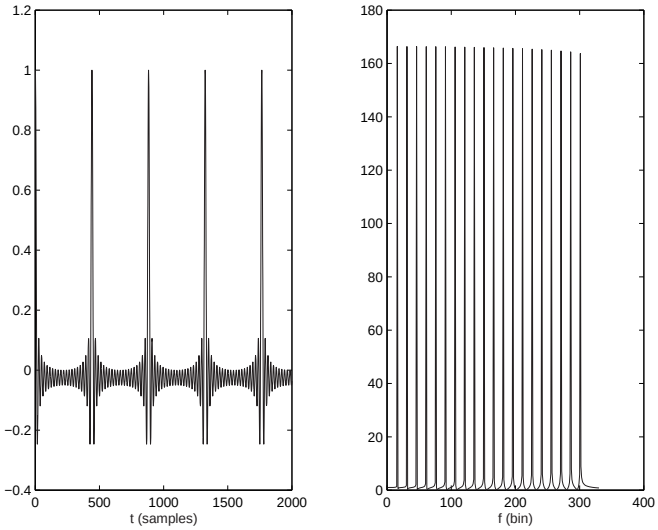


Figure 5.14: Generatore di impulsi

dati B larghezza di banda e f_c frequenza centrale della risonanza, i coefficienti del filtro si legano con buona approssimazione a questi parametri mediante le relazioni

$$a_2 = -2r\cos(\omega_c),\ \ a_3 = r^2,\ \ b_1 = (1-r)\sqrt{1-2r\cos(2\omega_c)+r^2}, \tag{5.17}$$

dove i parametri ausiliari r, ω_c sono definiti come $r = e^{\frac{-\pi B}{F_s}}$ e $\omega_c = 2\pi f_c/F_s$. Il parametro b_1, fattore di normalizzazione del guadagno, e' calcolato normalizzando la risposta in ampiezza rispetto alla risonanza, ovvero ponendo $|H(\omega_c)| = 1$.

M-5.11

Write a function `baIIR2(fc,B)` that computes the coefficients of the 2nd order resonant filter given the center frequency f_c and the bandwidth B.

M-5.11 Solution

```
function [b,a]=baIIR2(fc,B); %computes coeff. a,b of II order cells
                             %(the no. of cells to be computed depends
                             %on the length of the vectors fc,B)
global Fs;
global Fc;
nfilters=length(fc);         %no. of cells to be computed

r=exp(-(pi.*B)/Fs)
a2=-(2*r.*cos(2*pi*fc/Fs))'
a3=r'.^2
a1=ones(nfilters,1)

b1=(1-r).*sqrt(1-2.*r.*cos(2*2*pi.*fc/Fs)+r.*r);
b1=b1';

a=[a1 a2 a3];
b=[b1 zeros(nfilters,1) zeros(nfilters,1)];
```

Note that we have followed the Octave/Matlab convention in defining the coefficients `b,a`. See the help for the function `filter(b,a,in)`

Usando le sorgenti e gli elementi filtranti appena descritti, e' possibile sperimentare l'effetto del filtraggio a parametri costanti.

M-5.12

Using the functions `buzz` and `baIIR2`, realize a parallel formant synthesizer. Use 3 2nd order IIR cells, corresponding to the first 3 vowel formant frequencies.

M-5.12 Solution

```
%%% headers %%%
%[...]

%%% define controls %%%
amp=envgen([0,.2,1,1.8,2],[0,1,.8,1,0],'linear');  % amp. envelope
f=envgen([0,.2,1.8,2],[200,250,250,200],'linear'); % pitch envelope
f=f+max(f)*0.05*...                                % add vibrato
  sin(2*pi*5*(SpF/Fs)*[0:length(f)-1]).*hanning(length(f))';

[b_i,a_i]=baIIR2([300 2400 3000],[200 200 500]); %spec. envelope /i/
[b_a,a_a]=baIIR2([700 1200 2500],[200 300 500]); %spec. envelope /a/
[b_e,a_e]=baIIR2([570 1950 3000],[100 100 800]); %spec. envelope /e/

%%% compute sound %%%
s=buzz(0,amp,f);   %impulse source
si=filter(b_i(1,:),a_i(1,:),s)+...    %synthesize /i/
   filter(b_i(2,:),a_i(2,:),s)+filter(b_i(3,:),a_i(3,:),s);
sa=filter(b_a(1,:),a_a(1,:),s)+...    %synthesize /a/
   filter(b_a(2,:),a_a(2,:),s)+filter(b_a(3,:),a_a(3,:),s);
se=filter_e(b(1,:),a_e(1,:),s)+...    %synthesize /e/
     filter(b_e(2,:),a_e(2,:),s)+filter(b_e(3,:),a_e(3,:),s);
```

Note the use of the `filter` function. Figure 5.15 shows the spectrum of the original source signal, the spectral envelope defined by the filtering elements, and the spectrum of the final signal for two different pitches

---

5.4.1.5 Effetti audio

Il banco di filtri puo' naturalmente essere applicato anche al suono campionato di uno strumento acustico. Questo tipo di operazione e' adatta, per esempio, a produrre effetti di variazione timbrica a partire dal timbro originario di una nota dello strumento da riprodurre. Un caso tipico e' la riproduzione dell'effetto di una sordina su uno strumento a fiato. Di seguito e' riportato un semplice esempio di manipolazione timbrica di un file audio.

M-5.13

Process a recorded sample signal in order to obtain a "mute" effect.

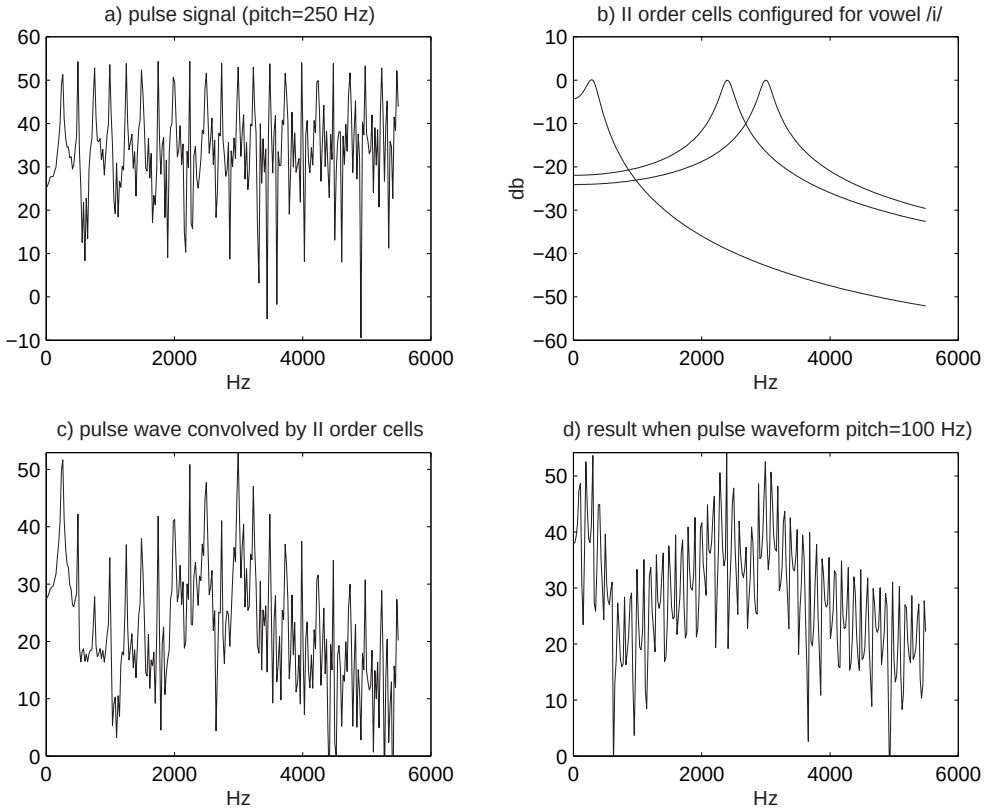


Figure 5.15: Inviluppo formantico con banco di filtri in parallelo

M-5.13 Solution

```
%%% headers %%%
s=wavread('sample.wav'); %import audio file
%[...]                   %define Fs accordingly

%%% define controls %%%
[b,a]=baIIR2([300 2400 3000],[200 200 200]);

%%% compute audio %%%
si=filter(b(1,:),a(1,:),s)+filter(b(2,:),a(2,:),s)
    +filter(b(3,:),a(3,:),s);
```

Nei casi in cui si vogliono ottenere cambiamenti di timbro continui nel tempo, e' necessario cambiare i parametri del filtro nel tempo. Questa e' una operazione non banale che da' occasione di accennare ad alcuni problemi fondamentali dei filtri tempo-varianti: il verificarsi di discontinuita' e transitori spuri nel suono di uscita a fronte di variazione a scalino dei parametri; l'interpolazione lineare dei parametri non sempre basta a risolvere i problemi ed e' necessario scegliere accuratamente le strutture realizzative dei filtri che presentano maggior robustezza alle variazioni parametriche.

M-5.14

Write a function `IIRcell(in,fc,B)` that takes the vectors of center frequencies and bandwidths (`fc,B`) of a 2nd order cell, and an input vector `in` of samples, and returns a filtered sample vector.

M-5.14 Solution

```
function out = IIRcell(in,fc,B);

global SpF;
nframes=length(fc);

[b,a]=baIIR(fc,B);           %compute filter params. for each frame
initstate=zeros(1,2);        %initialize filter state
out=zeros(1,nframes*SpF);    %initialize output vector

for (i=1:nframes)                     %compute chunks of the output
   framein=in(((i-1)*SpF+1):i*SpF);   %vector in each frame
   [out(((i-1)*SpF+1):i*SpF),endstate]=...
          filter(b(i,:),a(i,:),framein,initstate);
   initsate=endstate;                 %update filter state
end
```

5.4.2 Sintesi della voce per predizione lineare

5.4.2.1 L'apparato di fonazione

La voce umana e' prodotta dal flusso di aria attraverso l'apparato di fonazione. Esso e' composto da tre cavita' principali: la cavita' nasale, la cavita' orale e la cavita' faringale, schematizzate in fig. 5.16. La cavita' nasale e' principalmente ossea e quindi la sua forma e' fissa. Essa puo' essere isolata dal resto dell'apparato vocale se si solleva il velo palatino, o palato molle. Cosi' facendo si chiude il diaframma rinovelare che mette in comunicazione la cavita' nasale con quella orale e faringale. Quando l'apparato vocale e' in posizione di riposo, il velo pende giu' e il diaframma e' quindi aperto. Durante la produzione della maggior parte dei suoni linguistici il velo e' sollevato e il diaframma e' chiuso, ma nel caso di suoni nasali o nasalizzati esso rimane aperto, in modo che l'aria sfugge attraverso la cavita' nasale, conferendo al suono una caratteristica colorazione nasale.

La sommita' della cavita' orale e' formata dalla struttura ossea del palato e dal palato molle. La conformazione della cavita' puo' essere modificata in modo considerevole dal movimento della mandibola, che puo' aprire o chiudere la bocca; dalle labbra, la cui disposizione puo' variare dall'estremo appiattimento all'estremo arrotondamento; dalla lingua che puo' assumere una quantita' di posizioni diverse. La cavita' faringale si estende fino al fondo della gola. Essa puo' essere compressa ritraendo indietro la radice della lingua verso la parete della faringe. Nella sua parte inferiore essa termina con le corde vocali, una coppia di membrane carnose che l'aria attraversa provenendo dai polmoni. Lo spazio tra esse e' detto glottide. Durante la produzione di un suono, essa puo' essere completamente aperta, con le corde vocali in posizione di quiete, parzialmente chiusa con le corde vocali in vibrazione o completamente chiusa, isolando cosi' la cavita' faringale dai polmoni.

La forma d'onda del segnale vocale e' quella di un'onda di pressione acustica originata da movimenti fisiologici dell'apparato di fonazione. L'aria e' spinta dai polmoni nella trachea e quindi forzata attraverso le corde vocali. Durante la generazione di suoni *vocalizzati* (quasi periodici), come la vocale /a/, l'aria spinta dai polmoni causa la vibrazione delle corse vocali e quindi la modulazione del flusso

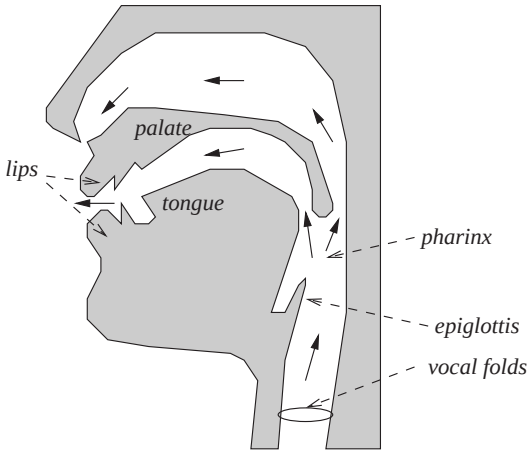


Figure 5.16: A schematic view of the phonatory system. Solid arrows indicates the direction of the airflow generated by lung pressure.

d'aria ad una frequenza dipendente dalla pressione nella trachea e dalla disposizione (lunghezza, spessore, tensione) delle corde vocali. Piu' grande e' la tensione delle corde vocali, piu' alta e' la frequenza della voce. La velocita' volumetrica (portata) del flusso di aria che attraversa la glottide definisce l'ingresso o eccitazione del tratto vocale.

I suoni non vocalizzati come /f/ sono generati tenendo volontariamente aperte le corde vocali, forzando l'aria attraverso la glottide e quindi usando l'articolazione per creare una costrizione lungo il tratto vocale (ad esempio posando i denti superiori sul labbro inferiore per il fonema /f/). Con contemporanea costrizione e vibrazione delle corde vocali si generano le fricative vocalizzate come la /z/ di rosa o la /dz/ di zanzara. I suoni esplosivi come /p/ sono generati aumentando la pressione dell'aria nella bocca e facendola quindi uscire improvvisamente.

Alla produzione di ogni fonema corrisponde una certa configurazione anatomica del tratto vocale, il quale agisce come risonatore meccanico allo scopo di modificare lo spettro dell'eccitazione glottale. Le frequenze di risonanza del tratto vocale vengono dette formanti. Ogni suono e' caratterizzato dal valore assunto dalle formanti che tipicamente sono in numero di quattro nell'intervallo 0-4 kHz. Nel parlato continuo, la configurazione del tratto vocale varia nel tempo, per cui l'evoluzione temporale delle formanti costituisce un efficace metodo di rappresentazione del segnale vocale. Inoltre, considerando che la velocita' di variazione degli organi articolatori e' abbastanza lenta, e' possibile schematizzare il processo come stazionario a tempo breve (dell'ordine di 10 - 50 msec).

5.4.2.2 Un modello di analisi/sintesi vocale: predizione lineare

Come esempio di sintesi sottrattiva presentiamo ora un algoritmo semplificato per la sintesi della voce. In esso l'ingresso $u(t)$ del filtro e' un treno di impulsi con frequenza f_0 per i suoni vocalizzati e rumore bianco per i suoni non vocalizzati. La velocita' volumetrica che esce dalla glottide e' modellata come uscita di un filtro passa-basso $G(z)$ a due poli con frequenza di taglio stimata a circa 100Hz. Il tratto vocale e' modellato con un filtro $V(z)$ a soli poli, consistente in una cascata di un numero ridotto di risuonatori (filtri) del secondo ordine. Ogni risonanza e' definita come un *formante* con una frequenza

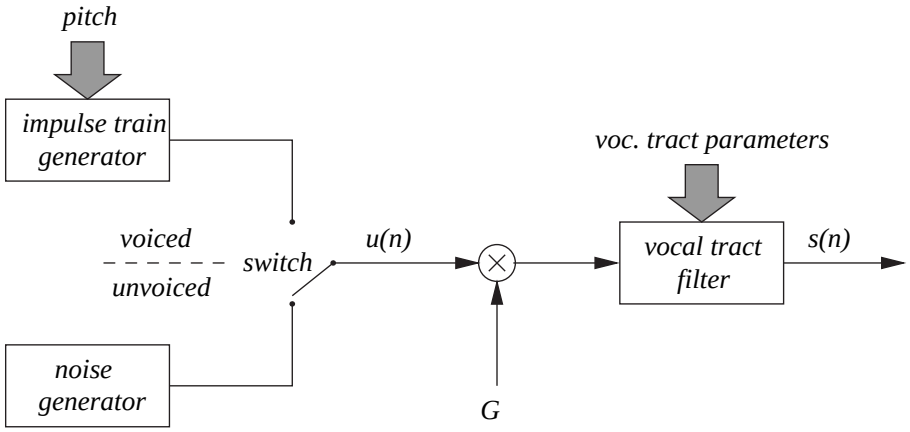


Figure 5.17: Diagramma a blocchi del modello semplificato di produzione del parlato

centrale f_i e una larghezza di banda B_i.[4] Infine, la forma d'onda della velocita' volumetrica alle labbra e' trasformata nella forma della pressione acustica fuori dalle labbra per mezzo di un modello $L(z)$ di radiazione delle labbra. In definitiva si ha

$$S(z) = [G(z) \cdot V(z) \cdot L(z)] \cdot U(z), \tag{5.18}$$

dove $S(z)$ e' il segnale prodotto.

Il modello della glottide e' della forma

$$G(z) = \frac{1}{[1 - \exp(-c/F_s)z^{-1}]^2}$$

e il modello di radiazione delle labbra e' della forma

$$L(z) = 1 - z^{-1}$$

Il filtro $V_i(z)$ dell'i-esimo formante di frequenza f_i e banda B_i e' nella forma (5.16). Il modello cascata del tratto vocale e' dato quindi da;

$$V(z) = \prod_{i=1}^{K} V_i(z)$$

5.4.2.3 Sintesi LPC

Il modello descritto prevede come ingresso un treno di impulsi periodici o un rumore e come parametri le frequenze e le larghezze di banda dei formanti (fig. 5.17). Nella sintesi della voce questi parametri vengono aggiornati ogni 5-10 msec o all'inizio di ogni periodo di pitch (sintesi sincrona con il periodo). Si puo' osservare che il termine al numeratore $L(z) = 1 - z^{-1}$ quasi coincide con un polo

[4] Un modello piu' accurato dovrebbe comprendere un infinito numero di risonanze, il cui effetto principale alle frequenze piu' basse e' di aumentare il livello spettrale. Quindi quando si deve rappresentare accuratamente solo il funzionamento del sistema solo alle frequenza medio-basse, (la parte piu' importante per la percezione del parlato), e' necessario introdurre una correzione che rappresenti l'effetto dei poli piu' alti trascurati.

di $G(z)$ che vale $1 - \exp(-c/F_s)z^{-1}$, in quanto $c/F_s \ll 1$. Si puo' quindi approssimare la composizione degli effetti spettrali della radiazione, tratto vocale ed eccitazione glottale con un filtro senza zeri e $p = 2K + 1$ poli. Esso e' pertanto rappresentabile con

$$S(z) = H(z) \cdot U(z), \qquad H(z) = \frac{g}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \tag{5.19}$$

Il filtro $H(z)$ rappresenta le caratteristiche complessive ingresso uscita del modello. In definitiva i parametri che definiscono il modello di produzione del parlato qui visto sono i coefficienti a_k, il guadagno g, e i parametri dell'ingresso vocalizzato o non (pitch e ampiezza). Per stimarli a partire da un segnale vocale si dovra' pertanto prima decidere se usare un generatore casuale (per suoni non vocalizzati) o periodico, poi si stimera' la frequenza della fondamentale e il guadagno. Infine si stimano i coefficienti a_k tramite algoritmi predittivi, che possono essere ricondotti essenzialmente a due classi: metodo della covarianza e metodo dell'autocorrelazione. Il metodo dell'autocorrelazione e' quello attualmente piu' usato per l'esistenza di algoritmi piu' robusti e piu' efficienti.

Il modello semplificato a soli poli e' una rappresentazione naturale per i suoni non nasali, mentre invece per i suoni nasali e fricativi si dovrebbe tener conto anche degli zeri. D'altra parte se l'ordine p e' sufficientemente alto, anche il modello a soli poli produce una buona rappresentazione per quasi tutti i suoni del parlato. Il grande vantaggio e' che i parametri possono essere stimati in modo semplice.

Dall'equazione 5.19 risulta che i campioni $s(n)$ sintetizzati sono legati all'ingresso dall'equazione

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + g u(n) \tag{5.20}$$

L'equazione 5.19 e' detta modello di sintesi, in quanto applicando a questo filtro l'ingresso si ottiene il suono vocale in uscita.

5.4.2.4 Analisi LPC

Per la stima dei parametri si considera sconosciuto l'ingresso $u(n)$. Si consideri di stimare l'uscita del sistema approssimativamente da una somma pesata dei campioni precedenti. Si ha quindi un predittore lineare di ordine p, con coefficienti $\tilde{a}_k$, definito dalla relazione

$$\tilde{s}(n) = \sum_{k=1}^{p} \tilde{a}_k s(n-k) \tag{5.21}$$

Viene quindi definito l'errore di predizione $e(n)$ (chiamato anche residuo) la differenza tra il valore attuale $s(n)$ e il valore predetto $\tilde{s}(n)$

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} \tilde{a}_k s(n-k) \tag{5.22}$$

$$\Rightarrow E(z) = A(z)S(z), \qquad A(z) = 1 - \sum_{k=1}^{p} \tilde{a}_k z^{-k}$$

Confrontando le equazioni 5.20 e 5.22 si vede che se il segnale seguisse esattamente il modello di eq. 5.20 e se $\tilde{a}_k = a_k$, allora risulterebbe

$$e(n) = g u(n), \qquad H(z) = \frac{g}{A(z)}. \tag{5.23}$$

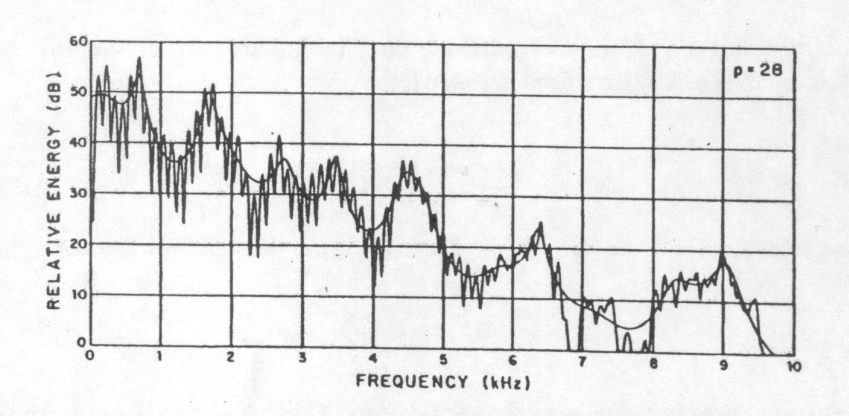


Figure 5.18: Spettro LPC con 28 poli confrontato con quello ottenuto da analisi mediante FFT

Si stimano i parametri del modello direttamente dai campioni del segnale vocale cercando di ottenere anche una buona stima delle proprieta' spettrali del segnale ottenuto utilizzando il modello per la sintesi (eq. 5.20). A causa della natura tempo variante del segnale vocale si fara' la stima su segmenti corti del segnale o a blocchi di campioni. L'approccio che viene seguito si basa sul metodo dei minimi quadrati, minimizzando quindi il quadrato dell'errore $e(n)$ di predizione su un segmento di suono ($E = \sum_m e^2(m)$, dove la somma e' estesa ai campioni del segmento analizzato). I parametri risultanti sono assunti essere i parametri della funzione del sistema $H(z)$ nel modello di produzione del parlato. Ricordando la relazione 5.23, si stima quindi g confrontando l'energia del segnale errore con quello scelto come eccitazione mediante la relazione

$$g^2 = \frac{\sum_m e^2(m)}{\sum_m u^2(m)}$$

Per minimizzare E, si ottengono le cosiddette equazioni di Yule-Walker, che consentono di determinare i coefficienti del filtro. Si noti che la minimizzazione ai minimi quadrati di E tende a produrre un segnale di errore con modulo dello spettro piatto (rumore bianco); per cui il filtro $A(z)$ e' chiamato anche *whitening filter*. Se il modello approssima bene il segnale vocalizzato, allora il residuo e' composto da un treno di impulsi che si ripetono alla frequenza di vibrazione delle corde vocali. Pertanto gli errori massimi di predizione si verificheranno con frequenza uguale al pitch del segnale. Nel dominio del tempo quindi la maggior parte dell'energia si localizza in vicinanza di questi picchi.

E' interessante notare che l'interpretazione in frequenza di E comporta che il metodo LPC stimato con l'autocorrelazione approssima meglio lo spettro nelle regioni di segnale ad alta energia, cioe' vicino ai picchi dello spettro rispetto alle regioni a bassa energia (valli). Un esempio e' riportato in fig. 5.18.

I parametri cosi' stimati sono assunti essere i parametri del modello. Da essi si possono ricavare altri parametri percettualmente piu' significativi come la frequenza e banda dei formanti. Questo approccio ha il vantaggio di avere metodi di stima efficienti e che si sono rivelati anche fornire una rappresentazione accurata del segnale vocale.

M-5.15

Write an example of an lpc analyis/synthesis procedure that analyzes a portion of speech and resynthesizes it using the model described above.

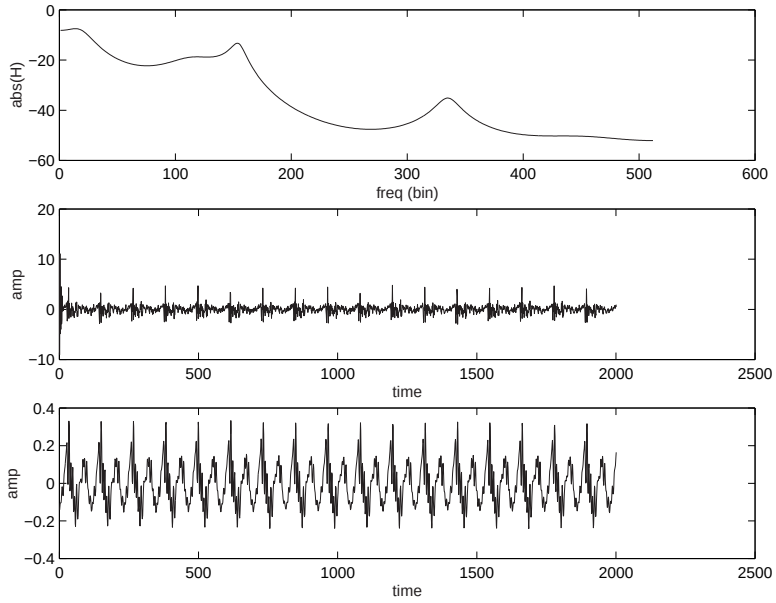


Figure 5.19: Analisi e risintesi della voce mediante LPC

M-5.15 Solution

```
%%% headers %%%
svoce=wavread('voce.wav');   %read audio sample
% [...]                      %define Fs accordingly

%%% analysis %%%
s=svoce(8000:10000);         %select signal portion
Nc=10;                       %no. of lpc coeff. to be computed
[a,g]=lpc(s,Nc);             %compute lpc coeff.
freqz([g 0 0],[a]);          %plot lpc filter response

%%% synthesis %%%
u=filter([a],[g 0 0],s);     %generate glottal excitation
                             %through inverse filtering
snew=filter([g,0,0],[a],u);  %resynthesize signal portion
```

Note that we have used the native function `lpc(s,N)` , where `s` is the input signal and `N` is the order of the prediction filter. Figure 5.19 shows the frequency response of the formant filter. the glottal excitation, and the resynthesized waveform.

5.5 Sintesi non lineari

Le trasformazioni viste sopra non possono cambiare le frequenze delle componenti in ingresso, in quanto sono trasformazioni lineari. Se si usano invece trasformazioni non lineari, le frequenze possono cambiare anche di molto. Ne consegue la possibilita' di cambiare sostanzialmente la natura del suono in ingresso. Queste possibilita' vengono anche usate nella sintesi del suono.

L'interpretazione della sintesi non lineare non e' basata sull'acustica fisica, ma piuttosto deriva dalla teoria della modulazione nelle comunicazioni elettriche, applicata ai segnali musicali. Questi metodi sono stati molto usati nella musica elettronica analogica e sono poi stati anche sviluppati nel digitale. Pertanto la sintesi non lineare ne eredita parzialmente l'interpretazione analogica come usata nella musica elettronica e inoltre e' diventata, specie con la modulazione di frequenza, una nuova metafora per i musicisti informatici.

Ci sono due effetti principali legati alla trasformazione non lineari: arricchimento dello spettro e traslazione dello spettro. Il primo effetto deriva dalla distorsione non lineare di un segnale e consente di controllare la brillantezza di un suono, mentre il secondo e' dovuto alla sua moltiplicazione per una sinusoide (portante) e sposta lo spettro attorno alla frequenza del segnale portante, alterando il rapporto armonico tra le righe del segnale modulante. La possibilita di traslare lo spettro e' molto efficace nelle applicazioni musicali. A partire da semplici componenti, si possono creare suoni armonici e inarmonici e stabilire differenti relazioni armoniche tra le parziali.

5.5.1 Synthesis by frequency modulation

This techniques does not derive from models of sound signals or sound production, instead it is based on an abstract mathematical description. The definition of "FM synthesis" denotes an entire family of techniques in which the instantaneous frequency of a periodic signal (*carrier*) is itself a signal that varies at sample rate (*modulating*). We have already seen in section 5.2.1.2 how to compute the signal phase when the frequency is varying at frame rate. We now face the problem of computing $\phi(n)$ when the frequency varies at audio rate.

A way of approximating $\phi(n)$ is through a first-order expansion:

$$\phi(n) = \phi(n-1) + \frac{d\phi}{dt}(n-1) \cdot \frac{1}{F_s}. \tag{5.24}$$

Recalling equation (5.4), that relates phase and instantaneous frequency, we approximate it as

$$\frac{d\phi}{dt}(n-1) = 2\pi \left[\frac{f(n) + f(n-1)}{2} \right], \tag{5.25}$$

where the frequency $f(t)$ has been approximated as the average of $f(n)$ at two consecutive instants. Using the two equations above, $\phi(n)$ is finally written as

$$\phi(n) = \phi(n-1) + \frac{\pi}{F_s}[f(n) + f(n-1)]. \tag{5.26}$$

M-5.16

Write a function `FMosc(t0,a,f,ph0)` that realizes a FM sinusoidal oscillator (the parameters `(t0,a,ph0)` are defined as in M-5.3, while `f` is now the sample-rate frequency vector).

M-5.16 Solution

```
function s=FMosc(t0,a,f,ph0)

global SpF;             %samples per frame
global Fs;              %sampling rate

nframes=length(a);      %total number of frames
```

```
s=zeros(1,nframes*SpF); %signal vector (initialized to 0)

lastfreq=f(1);
lastphase=ph0;
for (i=1:nframes)            %cycle on the frames
   phase=zeros(1,SpF);       %phase vector in a frame
   for(k=1:SpF)              %cycle through samples in a frame
      phase(k)=lastphase+... %compute phase at sample rate
               pi/Fs*(f((i-1)*SpF+k)+lastfreq);
      lastphase=phase(k);
      lastfreq=f((i-1)*SpF+k);
   end
   s(((i-1)*SpF+1):i*SpF)=a(i).*cos(phase);
end

s=[zeros(1,round(t0*Fs+1)) s]; %add initial silence of t0 sec.
```

Compare this function with the `sinosc` function in M-5.3. The only difference is that in this case the frequency is given at audio rate. Consequently the the phase computation differs.

Although early realizations of FM synthesis were implemented in this fashion, in the next sections we will follow an equivalent "phase-modulation" formulation. According to such formulation, the FM oscillator is written as:

$$s(t) = \sin(2\pi f_c t + \phi(t)), \tag{5.27}$$

where $\phi(t)$ is the input modulating signal and f_c is the carrier frequency.

5.5.1.1 Modulante semplice e spettri $f_c \pm k f_m$

Se la modulante e' una sinusoide di ampiezza I (indice di modulazione) e frequenza f_m data quindi da

$$\phi(t) = I \sin(2\pi f_m t)$$

la modulazione di frequenza semplice da':

$$\begin{aligned} s(t) &= \sin[2\pi f_c t + I \sin(2\pi f_m t)] \\ &= \textstyle\sum_{k=-\infty}^{\infty} J_k(I) \sin[2\pi(f_c + k f_m)t] \end{aligned} \tag{5.28}$$

dove $J_k(I)$ e' la funzione di Bessel del primo tipo di ordine k. Dall'equazione 5.28 si vede che il segnale prodotto ha uno spettro a righe di frequenza $f_c \pm k f_m$ e di ampiezza data da $J_k(I)$. Pur essendo la sommatoria estesa ad un numero infinito di termini, solo pochi di essi, attorno a $k = 0$ sono significativi. Infatti solo le funzioni di Bessel di ordine basso sono significative per valori piccoli dell'indice di modulazione. Quando l'indice I cresce, aumenta in corrispondenza anche il numero di funzioni significative. Il numero M di frequenze laterali di ampiezza maggiore di un centesimo e' dato da $M = I + 2.4 \cdot I^{0.27}$. In pratica si puo' considerare $M = 1.5 * I$. In questo modo si controlla la larghezza di banda attorno a f_c. Ne risulta un effetto tipo filtro dinamico, analogo a quello che i musicisti sperimentano nell'impiego della sintesi sottrattiva. Inoltre l'ampiezza di ogni funzione varia in modo oscillante al variare dell'indice. Questo fatto produce una caratteristica ondulazione delle

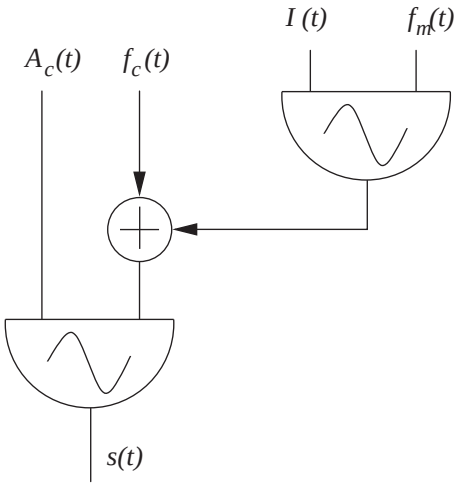


Figure 5.20: Modulazione di frequenza con modulante semplice

ampiezze delle parziali componenti quando l'indice varia in modo continuo e consente di ottenere facilmente spettri dinamici. Vale inoltre la proprieta' che l'ampiezza massima e la l'energia non cambiano al variare dell'indice I. Pertanto vengono evitati i problemi di normalizzazione di ampiezza che ci sono nella sintesi per distorsione non lineare.

E' interessante ora vedere l'equivalente formulazione di 5.28 come modulazione di frequenza. La frequenza istantanea di un segnale $s(t) = \sin[\psi(t)]$ e data da $f(t) = [1/2\pi][d\psi(t)/dt]$. Pertanto la frequenza istantanea $f_i(t)$ del segnale di eq. 5.28 vale

$$f_i(t) = f_c + I f_m \cos(2\pi f_m t) \tag{5.29}$$

Essa varia quindi attorno a f_c con una deviazione massima $d = I \cdot f_m$. In figura 5.20 e' riportato il caso di modulazione con portante semplice realizzato mediante oscillatore controllato in frequenza. Si osservi infine che un cambio della differenza di fase tra portante e modulante produce solo un cambiamento delle fasi reciproche delle parziali generate. Questo normalmente non e' percettualmente significativo. Solo nel caso in cui alcune parziali coincidano in frequenza, bisogna tenere conto della loro relazione di fase per calcolare l'ampiezza risultante.

L'equazione 5.28 mostra che nel caso di modulante semplice risulta uno spettro a righe di frequenza $|f_c \pm k f_m|$, con $k = 0, 1, \ldots$. Spettri di questo tipo sono caratterizzabili mediante il rapporto f_c/f_m. Quando il rapporto puo' essere rappresentato con una frazione irriducibile $f_c/f_m = N_1/N_2$ con N_1 e N_2 interi primi tra loro, il suono risultante e' armonico, nel senso che tutte le componenti sono multiple intere di una fondamentale. La frequenza fondamentale risulta

$$f_0 = \frac{f_c}{N_1} = \frac{f_m}{N_2},$$

e $f_c,\ f_m$ coincidono con la N_1-esima e N_2-esima armonica:

$$f_c = N_1 f_0, \qquad f_m = N_2 f_0.$$

Se $N_2 = 1$, tutte le armoniche sono presenti e le componenti laterali con k negativo si sovrappongono a quello con k positivo. Se $N_2 = 2$, sono presenti solo le armoniche dispari e le componenti

si sovrappongono ancora. Se $N_2 = 3$, mancano le armoniche multiple di 3. In generale il rapporto N_1/N_2 e' anche un indice dell'armonicita' dello spettro. Intuitivamente il suono e' piu' armonioso, quando il rapporto N_1/N_2 e semplice ossia quanto piu' il prodotto $N_1 \cdot N_2$ e' piccolo.

I rapporti possono essere raggruppati in famiglie. Tutti i rapporti del tipo $|f_c \pm kf_m|/f_m$ possono produrre le stesse componenti del rapporto f_c/f_m. Cambia solo quale parziale coincide con f_c. Ad esempio i rapporti 2/3, 5/3, 1/3, 4/3, 7/3 e cosi' via appartengono alla stessa famiglia. Sono presenti tutte le armoniche ad esclusione di quelle multiple di 3 (essendo $N_2 = 3$) e f_c coincidera' rispettivamente con la seconda, quinta, prima, quarta e settima armonica. Il rapporto che distingue la famiglia si dice in forma normalizzata se e' minore o uguale a $1/2$. Nell' esempio precedente esso e' uguale a $1/3$. Ciascuna famiglia e' quindi caratterizzata da un rapporto in forma normalizzata. Spettri simili possono essere ottenuti da suoni della stessa famiglia. Si vede quindi che il denominatore N_2 e' caratterizzante lo spettro. In particolare per $N_2 < 5$ ogni denominatore definisce una sola famiglia.

Se il rapporto e' irrazionale, il suono risultante non e' piu' periodico. Questa possibilita' viene usata per creare facilmente suoni inarmonici. Ad esempio se $f_c/f_m = 1/\sqrt{2}$ la spettro consiste in componenti a frequenza $f_c \pm k\sqrt{2}$. Non c'e' quindi nessuna fondamentale implicita. Un comportamento simile si ottiene per rapporti non semplici come $f_c/f_m = 5/7$.Di particolare interesse e' il caso in cui il rapporto f_c/f_m approssimi un valore razionale, cioe'

$$\frac{f_c}{f_m} = \frac{N_1}{N_2} + \epsilon.$$

In questo caso il suono non e' piu' rigorosamente periodico. La fondamentale e' ancora $f_0 = f_m/N_2$ e le parziali sono spostate dal loro preciso valore di $\pm\epsilon f_m$. Pertanto un piccolo spostamento della portante non cambia l'altezza del suono e lo rende molto piu' vivo grazie ai battimenti tra le componenti vicine. Si noti invece che lo stesso spostamento della modulante f_m cambia la fondamentale.

M-5.17

Synthesize a frequency modulated sinusoid, in the case of sinusoidal modulation. Plot the signal spectrum for increasing values of the modulation index.

M-5.17 Solution

```
%%% headers %%%
Fs=22050;         % sampling frequency

%%% define controls %%%
fc=700;           %carrier freq.
fm=100;           %modulating freq.
I=2;              %modulation index
t=0:(1/Fs):3;     %time vector (in s)

%%% compute sound %%%
s=sin(2*pi*fc*t+I*sin(2*pi*fm*t));
```

Figure 5.21 shows the signal spectrum for 3 values of the modulation index. Note that as the index increases the energy of the carrier frequency is progressively transferred to the lateral bands, according to the predicted behavior.

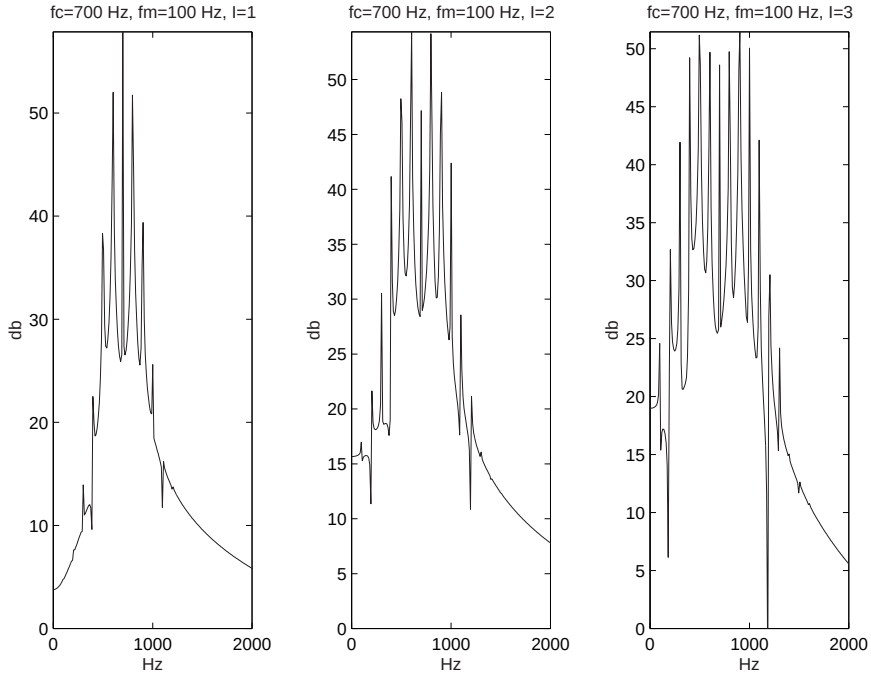


Figure 5.21: Spettro relativo a modulazione di frequenza con portante a 700 Hz, modulante sinusoidale a 100 Hz e indice di modulazione I crescente da 1 a 3

5.5.1.2 Portante composta

Consideriamo ora una portante periodica ma non sinusoidale.

$$s(t) = \sum_{l=0}^{L} A_l \sin[2\pi l f_c t + \phi_l(t)]$$

Se essa viene modulata, e' come se ciascuna sua armonica fosse modulata dalla stessa modulante. Se la modulante e' sinusoidale, nello spettro attorno ad ogni armonica della portante saranno presenti righe di ampiezza proporzionale all'armonica. Ne risulta uno spettro di righe a frequenza $|l f_c \pm k f_m|$ e di ampiezza $A_l J_k(I)$ con $|l| \leq L$ e $|k| \leq M$, essendo L il numero di armoniche significative.

$$s(t) = \sum_{l=1}^{L} \sum_{k=-M}^{M} A_l J_k(I) \sin[2\pi(l f_c + k f_m)t]$$

In generale ci possono essere varie portanti indipendenti modulate dalla stessa modulante o da differenti modulanti (fig. 5.22). Ne risulta una specie di sintesi additiva in cui invece che addendi sinusoidali, si hanno addendi piu' complessi

$$s(t) = \sum_{l=0}^{L} A_l \sin[2\pi f_c l n + \phi_l(t)]$$

Per esempio con portanti di frequenza multipla della frequenza della modulante f_m si possono creare suoni armonici complessi di frequenza fondamentale $f_0 = f_m$ controllando le varie regioni dello spettro in modo indipendente. La frequenza di ciascuna portante determina la regione che viene influenzata e in un certo senso la posizione di un formante.

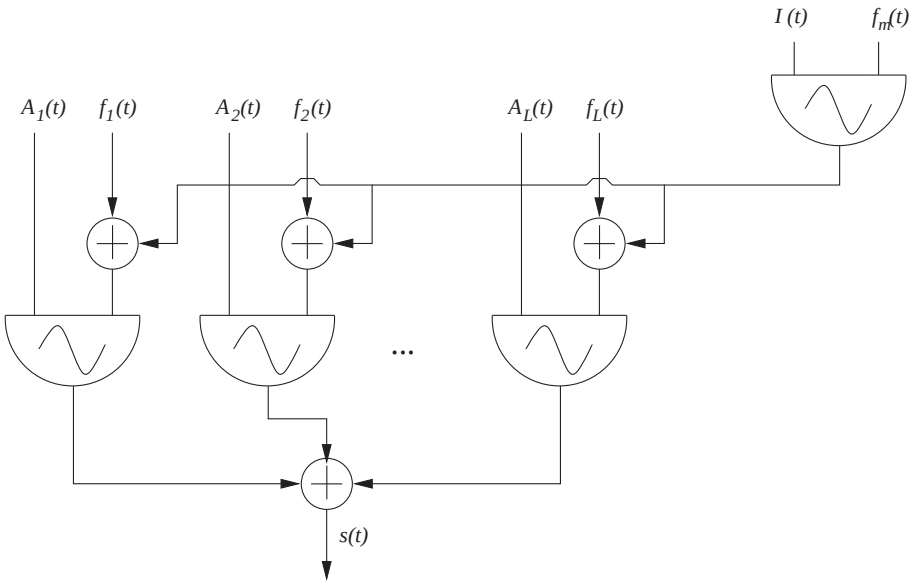


Figure 5.22: Modulazione di frequenza con L portanti modulate dalla stessa modulante.

M-5.18

Synthesize a frequency modulated signal in the case of composite carrier. Plot the signal spectrum.

M-5.18 Solution

```
%%% headers %%%
%[...]

%%% define controls %%%
fc1=300;     %carrier freq. 1
fc2=1000;    %carrier freq. 2
fc3=3000;    %carrier freq. 3
fm=100;      %modulating freq.
I1=1;        %modulation index 1
I2=2;        %modulation index 2
I3=3;        %modulation index 3
t=0:(1/Fs):3;%time vector (in s)

%%% compute sound %%%
theta=sin(2*pi*fm*t);           %modulation signal
s=sin(2*pi*fc1*t+I1*theta)+...  %sound signal
  sin(2*pi*fc2*t+I2*theta)+...
  sin(2*pi*fc3*t+I3*theta);
```

Figure 5.23 shows the resulting spectrum.

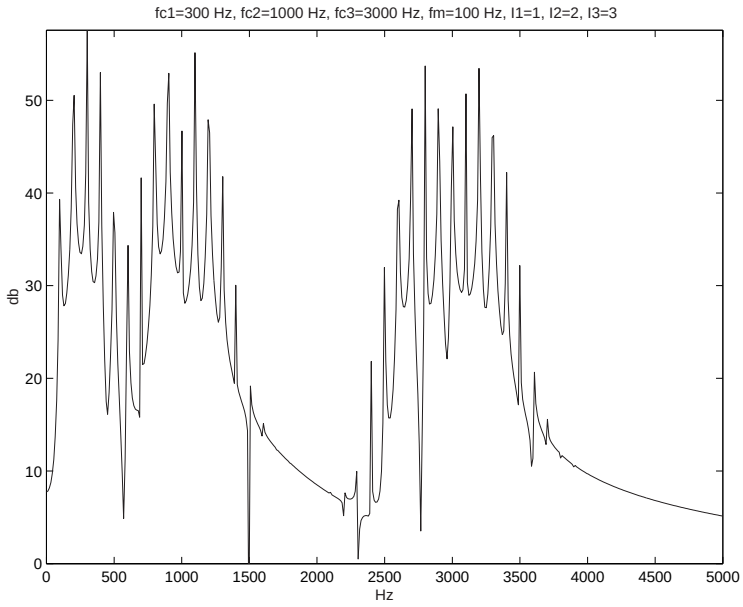


Figure 5.23: Spettro relativo a modulazione di frequenza con tre portanti e una modulante

5.5.1.3 Modulante composta

Esaminiamo ora il caso di modulante composta da due sinusoidi (fig. 5.24)(a), ciascuna con il suo indice di modulazione

$$\phi(t) = I_1 sin(2\pi f_1 t) + I_2 sin(2\pi f_2 t)$$

Sostituendo in (5.27) risulta:

$$s(t) = \sin[2\pi f_c t + I_1 \sin(2\pi f_1 t) + I_2 \sin(2\pi f_2 t]$$

Sviluppando la prima modulante si ha:

$$s(t) = \sum_k J_k(I_1) \sin[2\pi(f_c + k f_1)t + I_2 \sin(2\pi f_2 t))]$$

e poi la seconda modulante si arriva a:

$$s(t) = \sum_k \sum_n J_k(I_1) \cdot J_n(I_2) \sin[2\pi(f_c + k f_1 + n f_2)t]$$

Lo spettro risultante e' molto piu' complicato di quello del caso di una modulante semplice. Sono presenti tutte le parziali a frequenza $|f_c \pm k f_1 \pm n f_2|$ e con ampiezza $J_k(I_1) \cdot J_n(I_2)$. Per interpretare l'effetto si consideri $f_1 > f_2$. Se fosse presente solo la modulante a frequenza f_1, lo spettro risultante avrebbe un certo numero di componenti di ampiezza $J_k(I_1)$ e frequenza $f_c \pm k f_1$. Quando viene applicato anche la modulante a frequenza f_2, queste componenti diventano a loro volta portanti con bande laterali prodotte da f_2. Attorno a ciascuna delle componenti prodotte da f_1 si avranno cioe' righe spaziate di f_2. La banda risultante e' approssimativamente uguale alla somma delle due bande.

Se le frequenze hanno rapporti semplici tra loro, lo spettro e' del tipo $|f_c \pm k f_m|$ dove ora f_m e' il massimo comun divisore tra f_1 e f_2. Per esempio se $f_c = 700$ Hz, $f_1 = 300$ Hz 2e $f_2 = 200$ Hz, le componenti sono $700 \pm k100$ e la fondamentale 100 Hz. Pertanto scegliendo f_1 e f_2 multipli di f_m si

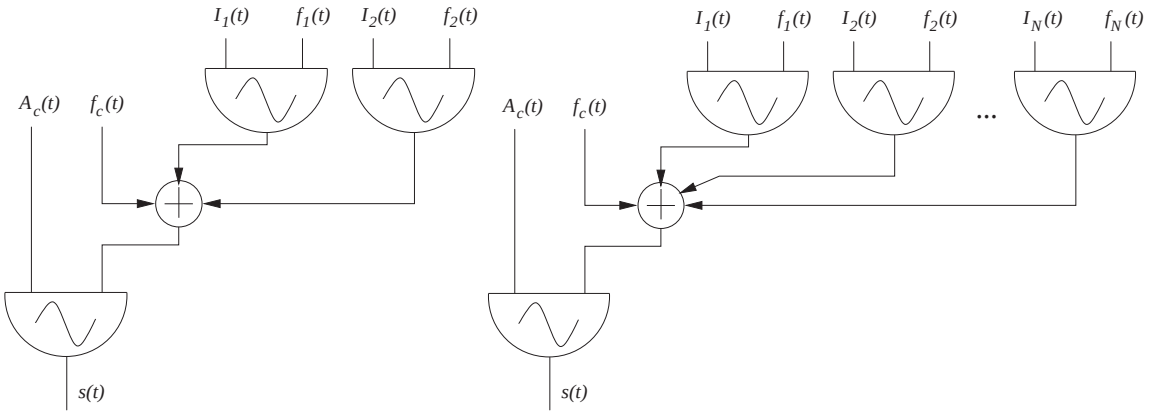


Figure 5.24: Modulazione di frequenza (a) con due modulanti e (b) con N modulanti

ottengono suoni dello stesso tipo di quelli ottenuti con la modulazione semplice ma con una dinamica spettrale piu' ricca.

In generale se il segnale modulante e' composto da N sinusoidi (fig. 5.24)(b) risulteranno componenti di frequenza $|f_c \pm k_1 f_1 \pm ... \pm k_N f_N|$ con ampiezze date dal prodotto di N funzioni di Bessel. Anche qui se i rapporti sono semplici risulta uno spettro del tipo $|f_c \pm k f_m|$ dove f_m e' il massimo comun divisore tra le frequenze modulanti. Se i rapporti non sono semplici le righe risultanti saranno sparse dando luogo a suoni inarmonici o anche rumorosi per alti valori degli indici.

Ad esempio Schottstaedt usa la doppia modulante per simulare il suono del piano, ponendo $f_1 \simeq f_c$ e $f_2 \simeq 4f_c$. In questo modo cerca di simulare la leggera inarmonicita' delle corde del piano. Inoltre fa diminuire gli indici di modulazione al crescere di f_c e quindi della fondamentale della nota. In questo modo le note basse sono piu' ricche di armoniche di quelle alte.

M-5.19

Synthesize a frequency modulated sinusoid in the case of composite modulation. Plot the signal spectrum.

M-5.19 Solution

```
%%% headers %%%
%[...]

%%% define controls %%%
fc1=700;     %carrier freq.
fm=700;      %modulating freq. 1
fm=2800;     %modulating freq. 2
I1=1;        %modulation index 1
I2=1;        %modulation index 2
t=0:(1/Fs):3;%time vector (in s)

%%% compute sound %%%
```

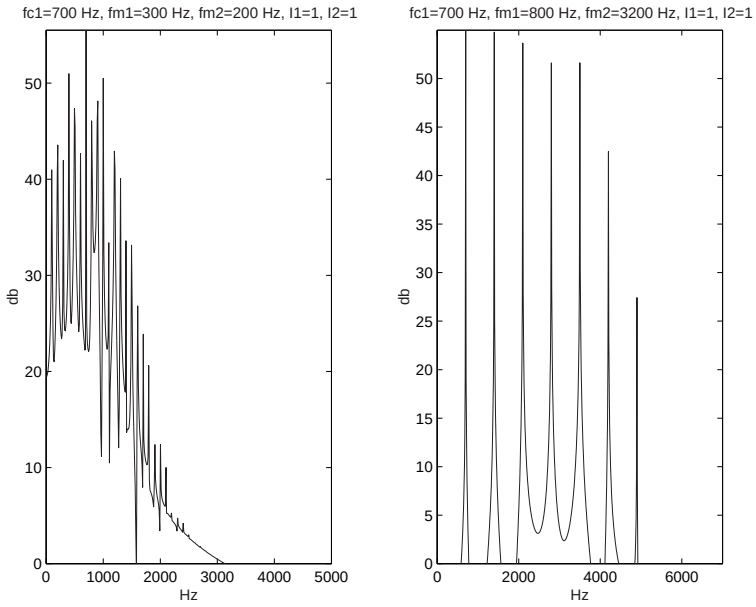


Figure 5.25: Due esempi di modulazione con portante semplice e modulante composta

```
s=sin(2*pi*fc*t+...      %sound signal
     I1*sin(2*pi*fm1*t)+I2*sin(2*pi*fm2*t));
```

Figure 5.25 shows the spectrum of an FM oscillator with sinusoidal carrier and a composite modulation made of two sinusoids. Note that in the first case the simple ratio between f_{m1} and f_{m2} produces a spectrum of the form $|f_c \mp kf_m|$ (in which $f_m = 100$ Hz, max. common divisor between f_{m1} and f_{m2}, is the fundamental). In the second case, the values $f_{m1} = f_c$ and $f_{m2} = 4f_c$ are chosen in such a way that the fundamental coincides with f_c and that upper partials are harmonic (since $f_{m1} = f_c$ coincides with the max. common divisor between f_{m1} and f_{m2}).

I rapporti semplici dell'ultimo esempio visto determinano uno spettro esattamente armonico. E' possibile sperimentare l'effetto dell'inarmonicita' variando i valori di f_1 e f_2 in modo che siano solo approssimativamente pari a f_c e a $4f_c$ rispettivamente. La figura 5.26 mostra lo spettro risultante per scostamenti progressivi di f_1 e f_2 dai valori proporzionali a f_c.

Anche per gli algoritmi di modulazione di frequenza e' possibile pensare ad una interfaccia che renda semplice controllare la sintesi con inviluppi di ampiezza e frequenza al frame rate. Un oscillatore FM a portante e modulante composta, ad esempio, avrebbe interfaccia `FMoper(t0,a,[fc1 fc2 ... fcN],[fm1 fm2 ... fmM],[I1 I2 ... IM])` in cui tutti i parametri di ingresso possono essere rappresentati con inviluppi temporali. La realizzazione di questo operatore e' lasciata come esercizio.

5.5.1.4 Modulanti in cascata e in feedback

Consideriamo ora il caso di modulante sinusoidale a sua volta modulata da un'altra sinusoide (fig. 5.27)

$$\phi(t) = I_1 sin(2\pi f_1 t + I_2 \sin(2\pi f_2 t))$$

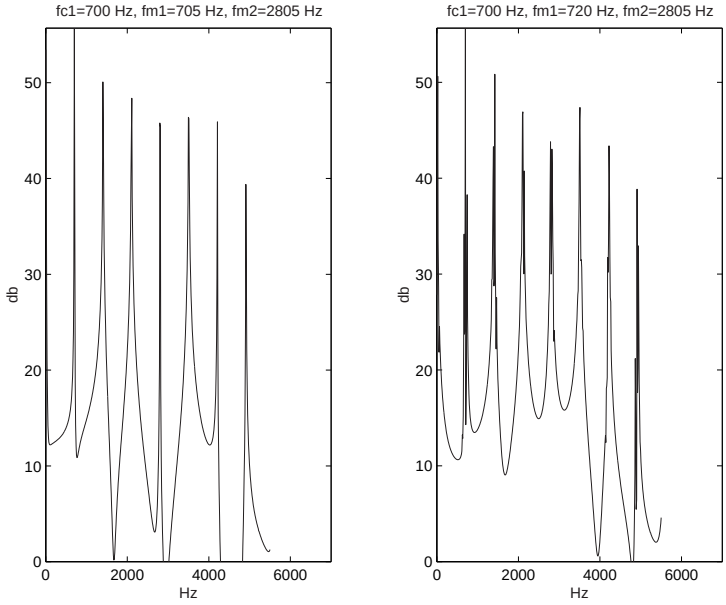


Figure 5.26: Suoni inarmonici dovuti a rapporti non semplici tra frequenze

Il segnale e' quindi definito da:

$$\begin{aligned} s(t) &= \sin[2\pi f_c t + I_1 \sin(2\pi f_1 t + I_2 \sin(2\pi f_2 t))] \\ &= \sum_k J_k(I_1) \sin[2\pi(f_c + k f_1)t + k I_2 \sin(2\pi f_2 t)] \\ &= \sum_k \sum_n J_k(I_1) \cdot J_n(k I_2) \sin[2\pi(f_c + k f_1 + n f_2)t] \end{aligned}$$

Il risultato puo' venire interpretato come se ciascuna parziale prodotta dal modulatore f_1 sia a sua volta modulata da f_2 con indice di modulazione kI_2. Pertanto risulteranno le componenti di frequenza $|f_c \pm k f_1 \pm n f_2|$ con approssimativamente $0 \leq k \leq I_1$ e $0 \leq n \leq I_1 \cdot I_2$. La frequenza massima e' $f_c + I_1(f_1 + I_2 f_2)$. La struttura dello spettro e' simile a quella prodotta da due modulanti sinusoidali, ma con banda maggiore. Anche qui se i rapporti sono semplici lo spettro sara' del tipo $|f_c \pm k f_m|$ dove f_m e' il massimo comun divisore tra f_1 e f_2.

Consideriamo infine il caso in cui si usi come modulante il valore precedente del segnale generato. Si ha cosi' la cosiddetta feedback FM. Essa e' descritta in termini digitali da queste relazioni:

$$\begin{aligned} \phi(n) &= \beta s(n-1) \\ s(n) &= \sin(2\pi \frac{f_c}{F_s} n + \phi(n)) \end{aligned}$$

dove β e' il fattore di feedback e agisce come fattore di scala o indice di modulazione per il feedback. Al crescere di β il segnale passa da sinusoidale verso la forma d'onda a dente di sega in modo continuo. Lo spettro e' armonico di frequenza f_c con aumento graduale del numero di armoniche. In termini di funzioni di Bessel risulta

$$s(t) = \sum_k \frac{2}{k\beta} J_k(k\beta) \sin(2\pi k f_c t)$$

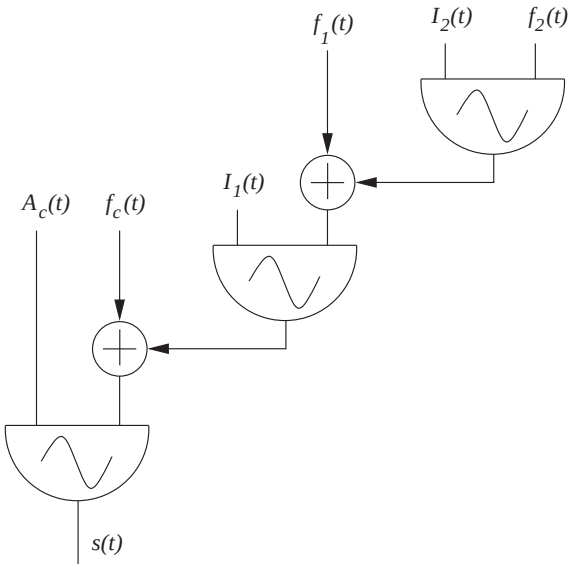


Figure 5.27: Modulazione di frequenza con due modulanti in cascata

5.5.1.5 Cenni storici

I due metodi classici per l'arricchimento spettrale e per la traslazione dello spettro, ovverosia distorsione non lineare (vedi sezione 5.5.2.3) e modulazione ad anello (vedi sezione 5.5.2.1) , hanno perso progressivamente di interesse in favore del metodo della modulazione in frequenza, il quale unisce i due effetti in un'unica formulazione molto versatile per produrre molti tipi di suono, e d'altra parte evita alcuni difetti di questi metodi. Per questa ragione, la sintesi per modulazione di frequenza e' il piu' usato fra i metodi non lineari. Questo metodo e' diventato molto popolare da quando fu usato nei sintetizzatori Yamaha tipo DX7 ed e' tuttora usato anche nelle schede audio.

Non evocando questa tecnica di sintesi nessuna esperienza musicale nell'esecutore, il controllo parametrico risulta poco intuitivo e richiede quindi una rilevante dose di esperienza specifica, caratteristica questa degli strumenti innovativi. Inoltre non ci sono metodi chiusi che consentano di derivare in modo preciso i parametri di un modello FM analizzando un suono dato. Per riprodurre dei suoni di strumenti musicali e' quindi preferibile usare altre tecniche come la sintesi additiva o la sintesi per modelli fisici.

Il principale punto di forza della sintesi FM, ossia l'elevata dinamica timbrica legata a pochi parametri e a basso costo computazionale, ha quindi perso progressivamente terreno nei confronti di altre tecniche di sintesi, piu' costose, ma controllabili in maniera piu' naturale e intuitiva. Il metodo conserva comunque la particolarita' di definire un suo spazio timbrico peculiare. Esso pur non prestandosi particolarmente alla simulazione di qualita' di suoni naturali, offre comunque un grande ventaglio di sonorita' originali di notevole interesse per la computer music.

5.5.2 Altre sintesi non lineari

5.5.2.1 Sintesi moltiplicativa

La trasformazione non lineare piu' semplice consiste nella moltiplicazione di due segnali. Nel campo analogico e' chiamata modulazione ad anello (*ring modulation*) o RM ed e' piuttosto difficile da

produrre in modo preciso. Nel campo numerico invece consiste in una semplice operazione di moltiplicazione. Se $x_1(t)$ e $x_2(t)$ sono due segnali il segnale di uscita e' dato da

$$s(t) = x_1(t) \cdot x_2(t) \tag{5.30}$$

Lo spettro risultante e' dato dalla convoluzione tra gli spettri dei due segnali.

Normalmente uno dei due segnali e' sinusoidale di frequenza f_c ed e' chiamato portante $c(t)$ (*carrier*) e l'altro e' un segnale in ingresso alla trasformazione ed e' chiamato modulante $m(t)$. Si ha pertanto

$$s(t) = m(t) \cdot c(t) = m(t)\cos(2\pi f_c t + \phi_c)$$

e lo spettro risultante e'

$$S(f) = \frac{1}{2}\left[M(f - f_c)e^{j\phi_c} + M(f + f_c)e^{-j\phi_c}\right]$$

Lo spettro di s(t) e' composto da due copie dello spettro di $m(t)$: una banda laterale inferiore (LSB) e la banda laterale superiore (USB). La LSB e' rovesciata in frequenza e entrambe le bande sono centrate attorno a f_c. A seconda della larghezza di banda di $m(t)$ e della frequenza della portante f_c, le bande laterali possono essere parzialmente riflesse attorno all'origine dell'asse di frequenza. Se la portante ha diverse componenti spettrali, lo stesso effetto si ripete per ogni componente. L'effetto acustico della modulazione ad anello e' relativamente facile da capire per segnali semplici. Diventa pero' piuttosto complicato da immaginare per segnali con numerose parziali. Se sia la portante che la modulante sono sinusoidali di frequenza rispettivamente f_c e f_m, si sente la somma di due differenti parziali a frequenza $f_c + f_m$ e $f_c - f_m$. Ad esempio se $f_c = 500$ Hz e $f_m = 400$ Hz, la modulazione ad anello produce due parziali a frequenza 900 Hz e 100 Hz. Se invece se $f_c = 100$ Hz e di nuovo $f_m = 400$ Hz, si producono due parziali a frequenza 500 Hz e -300 Hz. Quest'ultima ha frequenza negativa; si ha quindi una riflessione (foldunder) attorno allo 0 con cambio di segno della fase. Infatti $cos(-2\pi 100t + \phi) = cos(2\pi 100t - \phi)$. In definitiva si sentiranno due componenti a frequenza 500 Hz e 300 Hz.

Se la portante e' sinusoidale e la modulante e' periodica di frequenza f_m,

$$m(t) = \sum_{k=1}^{N} b_k \cos(2\pi k f_m t + \phi)$$

risulta

$$s(t) = \sum_{k=1}^{N} \frac{b_k}{2}\left[\cos\left[2\pi(f_c + kf_m)t + \phi_k\right] + \cos\left[2\pi(f_c - kf_m)t - \phi_k\right]\right] \tag{5.31}$$

L'armonica k-esima dara' luogo a due righe, una nella LSB e l'altra nella USB, a frequenza $f_c - kf_m$ e $f_c + kf_m$. Lo spettro risultante ha quindi righe a frequenza $|f_c \pm kf_m|$ con $k = 1, 2, ..$, dove si e' usato il valore assoluto per tenere conto delle possibili riflessioni attorno allo 0. Valgono per questi spettri le considerazioni fatte sopra sulle famiglie di spettri $|f_c \pm kf_m|$.

5.5.2.2 Modulazione di ampiezza

La modulazione di ampiezza era piu' facile da realizzare nel campo analogico e pertanto e' stata usata per molto tempo. Essa puo' essere implementata come

$$s(t) = [1 + \delta m(t)]c(t) \tag{5.32}$$

dove si e' assunto che l'ampiezza di picco di $m(t)$ sia 1. Il coefficiente δ determina la profondita' di modulazione. L'effetto e' massimo quando $\delta = 1$ e viene disattivato quando $\delta = 0$.

Tipiche applicazioni sono l'uso di un segnale audio come portante $c(t)$ e un oscillatore a bassa frequenza (LFO) come modulatore $m(t)$. L'ampiezza del segnale audio varia seguendo l'ampiezza di $m(t)$ e cosi' verra' sentita. Se il modulatore e' un segnale udibile e la portante una sinusoide di frequenza f_c, l'effetto e' simile a quello visto per il modulatore ad anello, solo che in uscita si sentira' anche la frequenza della portante f_c.

Si noti che a causa del tempo di integrazione del nostro sistema uditivo, l'effetto e' percepito differentemente in dipendenza del campo di frequenza dei segnali considerati. Una modulazione con frequenza sotto 20 Hz sara' sentita nel dominio del tempo (variazione di ampiezza), mentre modulazioni con frequenza superiori verranno sentite come componenti spettrali distinte (banda laterale inferiore, portante, banda laterale superiore).

5.5.2.3 Sintesi per distorsione non lineare

L'idea fondamentale della sintesi per distorsione non lineare, conosciuta anche sotto il nome di *wave-shaping* e' quella di passare una sinusoide per un blocco distorcente. E' noto infatti che se una sinusoide passa per filtro lineare viene modificata la sua ampiezza e fase, ma non la forma d'onda. Se invece l'amplificatore e' non lineare la forma d'onda del segnale viene modificata e vengono create altre componenti spettrali. Questo fatto e' ben noto nei segnali analogici, dove si cerca di evitarlo o usarlo per creare effetti tipo amplificazione con tubi elettronici. Nel campo digitale si e' pensato di sfruttarlo per produrre suoni periodici di spettro variabile. Il blocco distorcente e' realizzato mediante una funzione non lineare $F(x)$ chiamata funzione distorcente o *shaping function* memorizzata su tabella. Piu' raramente la funzione viene calcolata direttamente. La funzione distorcente dipende solo dal valore istantaneo dell'ingresso. Pertanto in corrispondenza ad un segnale di ingresso $x(t)$ il metodo calcola

$$s(t) = F[x(t)] \tag{5.33}$$

cercando in tabella ad ogni campione il valore all'ascissa $x(t)$.

Questa tecnica puo' essere usata come effetto audio che' oer la sintesi. Nel primo caso si usa una leggera distorsione, spesso sotto forma di saturazione, su un segnale qualsiasi per arricchire un po' lo spettro e simulare l'effetto che si verifica sovente in strumenti meccanici o elettronici analogici.

Per la sintesi dei suoni normalmente si usa un ingresso sinusoidale di ampiezza I (che puo' essere variata)

$$x(t) = I \cdot \cos(2\pi f t)$$

per cui la formula di sintesi diviene:

$$s(t) = F[x(t)] = F[I \cdot \cos(2\pi f t)]$$

In figura 5.28 e' riportato lo schema a blocchi della sintesi per distorsione non lineare. Con il parallelogramma viene indicato il modulo che effettua la distorsione mediante lettura da tabella del valore di $F(x)$.

In generale nella sintesi, se $F(x) = F_1(x) + F_2(x)$, la distorsione prodotta da $F(.)$ e' uguale alla somma di quelle prodotte da $F_1(\cdot)$ e $F_2(\cdot)$ separatamente. In particolare una funzione pari, cioe' simmetrica rispetto all'asse y genera solo armoniche pari e una funzione dispari (antisimmetrica) genera solo armoniche dispari. Normalmente una funzione distorcente produce infinite armoniche. Se pero' la funzione e' un polinomio $p(x)$ di grado N, vengono prodotte solo le prime N armoniche. In questo modo si puo' controllare il foldover. Se la funzione e' polinomiale e' anche facile calcolare

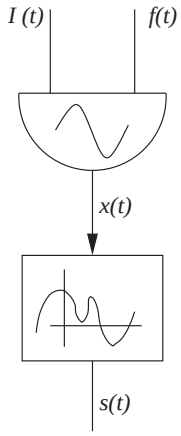


Figure 5.28: Sintesi per distorsione non lineare

le ampiezze delle armoniche generate dalla distorsione di una sinusoide. Infatti, tenendo conto che il polinomio di Chebyshev di ordine k e' definito come $T_k(\cos\theta) = \cos(k \cdot \theta)$, ne deriva che usandolo come funzione distorcente di una sinusoide di ampiezza unitaria si ha

$$s(t) = F[I \cdot \cos(2\pi f t)] = T_k[\cos(2\pi f t)] = \cos(2\pi k f t)$$

Viene cioe' generata l'armonica k-esima. Pertanto, sviluppando il polinomio distorcente $p(x)$ in serie di polinomi di Chebyshev

$$F(x) = p(x) = \sum_{i=0}^{N} d_i x^i = \sum_{k=0}^{N} h_k T_k(x)$$

si ottengono le ampiezze delle armoniche, dove h_k sara' l'ampiezza dell'armonica k-esima. Usando quindi il polinomio $p(x)$ cosi' determinato come funzione distorcente di una sinusoide ad ampiezza $I = 1$, si ottiene

$$s(t) = F[\cos(2\pi f t)] = \sum_{k=0}^{N} h_k \cos(2\pi k f t)$$

Se varia l'ampiezza dell'ingresso I, varia anche la distorsione e lo spettro prodotto. Questo e' simile ad un'espansione o contrazione della funzione, in quanto viene usato una parte maggiore o minore della funzione. L'ampiezza e l'intensita' del suono prodotto varia quindi con l'ampiezza dell'ingresso e deve essere pertanto compensata con un'opportuna funzione di normalizzazione. Nel caso invece che la distorsione sia usata come effetto, questi cambiamenti spesso sono in accordo con il fenomeno acustico che si vuole imitare e quindi non devono essere compensati.

Un'altra variazione dinamica del waveshaping che e' facile implementare consiste nell'aggiungere una costante all'ingresso. In questo caso la funzione viene traslata orizzontalmente. Anche in questo caso lo spettro varia, ma non e' piu' separabile l'effetto della parte pari da quella dispari della funzione originaria.

5.6 Key concepts

⇝ Sound models

Definition of sound model. Sound synthesis algorithms as instrument families. Control parameters and playability of a model. Approaches to sound model classification: structure (direct generation, feed-forward models, feed-back models), or level of representation (signal models, source models).

⇝ Signal generators

Table look-up oscillators, resampling and sampling increment. Recursive generators of sinusoidal signals. Definition of *control-rate*, and oscillators controlled in amplitude/frequency; in particular, computation of the signal phase from the frame-rate frequency. Envelope generators and the ADSR amplitude envelope. Noise generators: definitions of white noise and pink ($1/f$) noise.

Sampling techniques: advantages and limitations. Sample processing: pitch shifting and looping.

⇝ Granular synthesis

Definition of "grain". Granulation of real signals, and the Overlap and Add (OLA) method. Related problems (discontinuities, control). Synthetic grains: frequency modulated gaussian grains, and visualization in the time-frequency plane. Synchronous and asyncronous granular synthesis, time-frequency masks.

⇝ Additive synthesis

Basic formulation (equation (5.9)) and its relation to Fourier analysis. The problem of control signals. Additive synthesis by analysis: extraction of frequency and amplitude envelopes through *STFT*, sinusoid tracking procedure (figure 5.7.

"Sines-plus-noise" models: analysis and modeling of the stochastic component, resynthesis and transformations. "Sines-plus-transients-plus-noise" models: DCT representation, analysis and modeling of transients.

⇝ Source-filter models

Basic block scheme (figure 5.13), and characteristics of the two blocks. Possible (physical) interpretations of this structure. Notable source blocks: noise generators and pulse generators. Notable filtering blocks: 2nd order resonators (equation (5.16)), center frequency and bandwidth. Time-varying filters and related problems.

⇝ LPC voice modeling

Functioning of the vocal system, schematization as a source-filter system. Vocalized versus non-vocalized sounds, and corresponding source signals. A simple subtractive model for the voice (equation (5.18, glottis, vocal tract, and lip radiation blocks. LPC analysis, equivalence between extraction of the residual and reconstruction of the source signal.

⇝ FM synthesis

Definition of carrier and modulating signals. Computation of the signal phase from the sample-rate frequency. Equivalent formulations: frequency modulation and phase modulation. The simplest case: sinusoidal carrier and sinusoidal modulation. Analysis of $f_c \pm k f_m$ spectrums (equation 5.28), classification in terms of the ratio f_c/f_m. Other FM structures: composite modulation, composite carrier, cascade and feedback modulation.

Advantages (compact structure, low number of control parameters) and drawbacks (non-intuitive control, lack of general model identification techniques) of FM.

5.7 Commented bibliography

Among the plethora of available sound synthesis languages, one of the most widely used (and one of the most important historically) is *Csound*, developed by Barry Vercoe at the Massachusetts Institute of Technology. Csound descends from the family of Music-N languages created by Max Mathews at Bell Laboratories. See [11].

A second influential sound synthesis programming paradigm was developed starting from the early 1980's, mainly by Miller Puckette, and is today represented in three main software implementation: Max/MSP, jmax, and Pd. The "Max paradigm" (so named in honor of Max Mathews) is described by Puckette [5] as a way of combining pre-designed building blocks into sound-processing "patches", to be used in real-time settings. This includes a scheduling protocol for both control- and audio-rate computations, modularization and component intercommunication, and a graphical environment to represent and edit patches.

A discussion on recursive generators of sinusoidal signals is found e.g. in [4]. Models for fractal signals are also partially discussed in [4].

About granular synthesis: the most widely treated case is (*asynchronous granular synthesis*), where simple grains are distributed irregularly. A classic introduction to the topic is [7]. In particular, figure 5.3 in this chapter is based on an analogous figure in [7]. In another classic work, Truax [10] describe the granulation of recorded waveforms.

Additive synthesis was one of the first sound modeling techniques adopted in computer music and has been extensively used in speech applications as well. The main ideas of the synthesis by analysis techniques that we have reviewed date back to the work by McAulay and Quatieri [3]. In the same period, Smith and Serra started working on "sines-plus-noise" representations, usually termed *SMS* (Spectral Modeling Synthesis) by Serra. A very complete coverage of the topic is provided in [9]. The extension of the additive approach to a "sines-plus-transients-plus-noise" representation is more recent, and has been proposed by Verma and Meng [12].

About subtractive synthesis: a tutorial about filter design techniques, including normalization approaches that use L^1, L^2, and L^∞ norms of the amplitude response, is [2] Introductions to LPC analysis techniques and their applications in speech technology can be found in many textbook. See e.g. [6].

The first paper about applications of FM techniques to sound synthesis was [1]. It was later reprinted in [8].

Bibliography

[1] J. Chowning. The synthesis of complex audio spectra by means of Frequency Modulation. *J. Audio Engin. Soc.*, 21(7), 1973. Reprinted in [8].

[2] P. Dutilleux. Filters, Delays, Modulations and Demodulations: A Tutorial. In *Proc. COST-G6 Conf. Digital Audio Effects (DAFx-98)*, pages 4–11, Barcelona, 1998.

[3] R. McAulay and T. F. Quatieri. Speech Analysis/Synthesis Based on a Sinusoidal Speech Model. *IEEE Trans. Acoustics, Speech, and Signal Process.*, 34:744–754, 1986.

[4] S. J. Orfanidis. *Introduction to Signal Processing*. Prentice Hall, 1996.

[5] M. Puckette. Max at seventeen. *Computer Music J.*, 26(4):31–43, 2002.

[6] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ, 1978.

[7] C. Roads. Asynchronous granular synthesis. In G. De Poli, A. Piccialli, and C. Roads, editors, *Representations of Musical Signals*, pages 143–186. MIT Press, 1991.

[8] C. Roads and J. Strawn, editors. *Foundations of Computer Music*. MIT Press, 1985.

[9] X. Serra. Musical sound modeling with sinusoids plus noise. In C. Roads, S. Pope, A. Piccialli, and G. De Poli, editors, *Musical Signal Processing*, pages 91–122. Swets & Zeitlinger, 1997. http://www.iua.upf.es/∼xserra/articles/msm/.

[10] B. Truax. Real-time granular synthesis with a digital signal processor. *Computer Music J.*, 12(2):14–26, 1988.

[11] B. Vercoe. Csound: A manual for the audio processing system and supporting programs with tutorials. Technical report, Media Lab, M.I.T., Cambridge, Massachusetts, 1993. Software and Manuals available from ftp://ftp.maths.bath.ac.uk/pub/dream/.

[12] T. S. Verma and T. H. Y. Meng. Extending Spectral Modeling Synthesis with Transient Modeling Synthesis. *Computer Music J.*, 24(2):47–59, 2000.

Contents