11

Time and frequency-warping musical signals

G. Evangelista

11.1 Introduction

In this chapter we describe interesting audio effects that can be obtained by deforming the time and/or the frequency axis. Whilst discrete-time warping techniques were introduced in 1965 [Bro65], their interest in musical applications is fairly recent. Time warping aims at deforming the waveform or the envelope of the signal, while frequency warping modifies its spectral content, e.g., by transforming a harmonic signal into an inharmonic one or vice versa. The effects obtained by warping often increase the richness of the signal by introducing detuning or fluctuation of the waveform. The sounds from natural instruments like piano and drums already possess this property. The wave propagation in stiff strings and membranes can actually be explained in terms of frequency warping. By warping these sounds one can enhance or reduce their natural features. Even uninteresting synthetic sounds such as pulse trains may be transformed into interesting sounds by warping. Frequency warping is amenable to a time-varying version that allows us to introduce dynamic effects such as vibrato, tremolo and Flatterzunge in flute.

The quality of warping ultimately depends on the warping map, i.e., on the function describing the deformation of the time or frequency axis. Time and frequency warping are flexible techniques that give rise to a tremendous amount of possibilities, most of which are at present still unexplored from a musical point of view. By choosing the proper map one can actually morph the sound of an instrument into that produced by another instrument.

This chapter is divided into two main sections. In the first section we describe the time- and frequency-warping operations and derive algorithms for computing these effects, including recent advances in approximation. In the second section we illustrate some of their musical applications based on examples and case studies.

DAFX: Digital Audio Effects, Second Edition. Edited by Udo Zölzer.
© 2011 John Wiley & Sons, Ltd. Published 2011 by John Wiley & Sons, Ltd. ISBN: 978-0-470-66599-2

11.2 Warping

11.2.1 Time warping

Suppose that we want to change the shape of a periodic waveform $s(t)$ by moving the amplitude values attained by the signal to other time instants. One can achieve this by plotting the signal on an elastic sheet and by stretching and/or compressing the sheet at different points along its horizontal direction. The waveshape appears as if the original time axis had been deformed. Instants of time that were equidistant now have a different time distribution. This deformation of the time axis called time warping is characterized by a warping map $\theta(t)$, mapping points of the original t-axis onto points of the transformed axis. An example of time warping a sinewave is shown in Figure 11.1. The figure is obtained by plotting the original signal along the ordinates and transforming time instants into new time instants via the warping map, obtaining the signal plotted along the abscissa axis. Notice that to one point in the original signal there can correspond more points in the warped signal. These points are obtained by joining time instants of the original signal to points on the warping characteristic $\theta(t)$ using horizontal lines. The corresponding warped time instants are the value(s) of the abscissa corresponding to these intersection point(s). The time-warped signal is obtained by plotting the corresponding amplitude values at the new time instants along the abscissa. In this example the signal $\sin(\theta(t))$ may be interpreted as a phase modulation of the original sinewave. Time warping a signal composed of a superposition of sinewaves is equivalent to phase modulating each of the component sinewaves and adding them together. By time warping we alter not only the waveshape, but also the period of the signal. Clearly, the map is an effective modulo of the period of the signal, that is, the map $\theta(t)$ and the map

$$\theta_0(t) = \mathrm{rem}\,(\theta(t), T)\,,$$

where $\mathrm{rem}(x, T)$ denotes the remainder of the integer division of x by T, have the same net effect on a T-periodic signal. More generally, we can time warp an arbitrary, aperiodic signal $s(t)$ via an arbitrary map, obtaining a signal

$$s_{tw}(t) = s(\theta(t))$$

whose waveshape and envelope may be completely different from the starting signal. If the map is invertible, i.e., one-to-one, then

$$s_{tw}(\theta^{-1}(t)) = s(t),$$

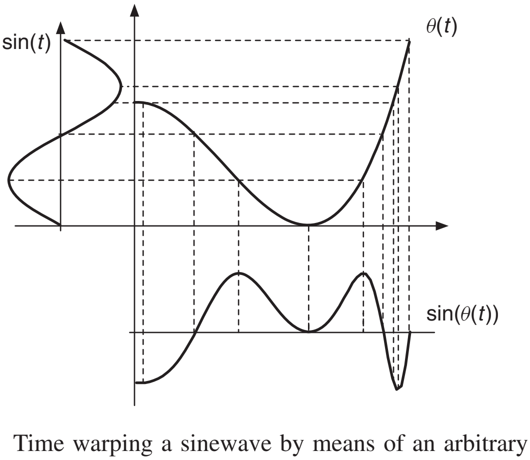

Figure 11.1 Time warping a sinewave by means of an arbitrary map $\theta(t)$.

that is, at time $\tau = \theta^{-1}(t)$ the time-warped signal attains the same amplitude value as that attained by the starting signal at time t.

Time-warping transformations are useful for musical applications, e.g., for morphing a sound into a new one in the time domain.

11.2.2 Frequency warping

Frequency warping is the frequency-domain counterpart of time warping. Given a signal whose discrete-time Fourier transform (DTFT) is $S(\omega)$, we form the signal $s_{fw}(t)$ whose DTFT is

$$S_{fw}(\omega) = S(\theta(\omega)).$$

That is, the frequency spectrum of the frequency-warped signal agrees with that of the starting signal at frequencies that are displaced by the map $\theta(\omega)$. If the map is invertible, then

$$S_{fw}(\theta^{-1}(\omega)) = S(\omega).$$

The frequency-warped signal is obtained by computing the inverse DTFT of the warped frequency spectrum. In order to obtain a real warped signal from a real signal, the warping map must have odd parity, i.e.,

$$\theta(-\omega) = -\theta(\omega).$$

In order to illustrate the features of frequency warping, consider a periodic signal $s(t)$ whose frequency spectrum peaks at integer multiples of the fundamental frequency ω_0. The frequency spectrum of the warped signal will peak at frequencies

$$\widehat{\omega}_k = \theta^{-1}(k\omega_0).$$

The situation is illustrated in Figure 11.2, where the original harmonic frequencies are represented by dots along the ordinate axis. The warped frequencies are obtained by drawing horizontal lines from the original set of frequencies to the graph of $\theta(\omega)$ and by reading the corresponding values of the abscissa. As a result, harmonically related partials are mapped onto non-harmonically related ones. Furthermore, if the frequency-warping map is not monotonically increasing, one obtains effects analogous to the foldover of frequencies. This is similar to that which is obtained from a phase vocoder in which the frequency bands are scrambled in the synthesis of the signal. However,

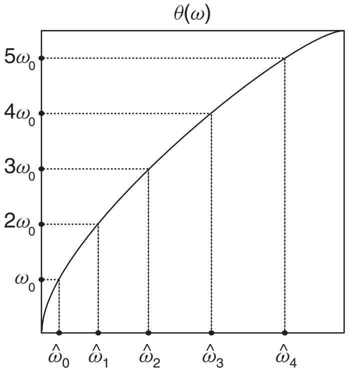

Figure 11.2 Frequency warping of a periodic signal: transformation of the harmonics into inharmonic partials.

the resolution and flexibility of the frequency-warping method are generally much higher than that of the scrambled phase vocoder.

Energy preservation and unitary frequency warping

By frequency warping a signal one dilates or shrinks portions of its frequency spectrum. As a result, the areas under the spectral characteristics are affected. Perceptually this results in an amplification of certain bands and an attenuation of other bands. This is depicted in Figure 11.3 where the original narrow band spectrum of Figure 11.3(a) is dilated, obtaining the dotted curve shown in Figure 11.3(b). In order to circumvent this problem, which causes an alteration of the relative energy levels of the spectrum, one should perform an equalization aimed at reducing the amplitude of dilated portions and increasing that of shrunk portions of the spectrum. Mathematically this is simply achieved, in the case where the warping map is increasing, by scaling the magnitude square of the DTFT of the warped signal by the derivative of the warping map. In fact, the energy in an arbitrary band $[\omega_0, \omega_1]$ is

$$E_{[\omega_0,\omega_1]} = \frac{1}{2\pi}\int_{\omega_0}^{\omega_1} |S(\omega)|^2 \, \mathrm{d}\omega.$$

By the simple change of variable $\omega = \theta(\Omega)$ in the last integral we obtain

$$E_{[\omega_0,\omega_1]} = \frac{1}{2\pi}\int_{\Omega_0=\theta^{-1}(\omega_0)}^{\Omega_1=\theta^{-1}(\omega_1)} |S(\theta(\Omega))|^2 \frac{\mathrm{d}\theta}{\mathrm{d}\Omega}\mathrm{d}\Omega = \frac{1}{2\pi}\int_{\Omega_0}^{\Omega_1} \left|\widetilde{S}_{fw}(\Omega)\right|^2 \mathrm{d}\Omega, \tag{11.1}$$

where

$$\widetilde{S}_{fw}(\omega) = \sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}S(\theta(\omega)) \tag{11.2}$$

is the DTFT of the scaled frequency warped signal. Equation (11.1) states the energy preservation property of the scaled warped signal in any band of the spectrum: the energy in any band $[\omega_0, \omega_1]$ of the original signal equals the energy of the warped signal in the warped band $[\theta^{-1}(\omega_0), \theta^{-1}(\omega_1)]$. Thus, the scaled frequency warping is a unitary operation on signals.

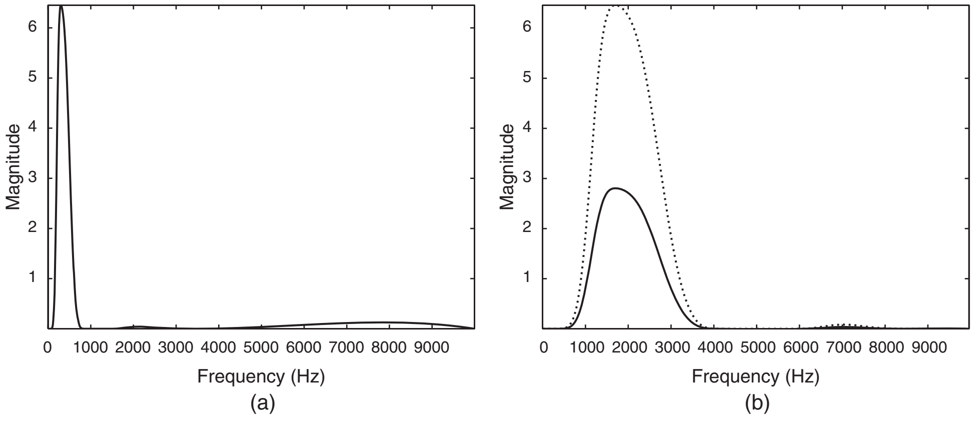

Figure 11.3 Frequency warping a narrow-band signal: (a) original frequency spectrum; (b) frequency-warped spectrum (dotted line) and scaled frequency-warped spectrum (solid line).

11.2.3 Algorithms for warping

In Sections 11.2.1 and 11.2.2 we explored basic methods for time warping in the time domain and frequency warping in the frequency domain, respectively. However, one can derive time- and frequency-warping algorithms in crossed domains. It is easy to realize that time and frequency warping are dual operations. Once a time-domain algorithm for frequency warping is determined, then a frequency-domain algorithm for time warping will work the same way. This section contains an overview of techniques for computing frequency warping. The same techniques can be used for time warping in the dual domain. We start from the basic maps using the Fourier transform and end up with time-varying warping using allpass chains in dispersive delay lines.

Frequency warping by means of FFT

A simple way to implement the frequency-warping operation on finite length discrete-time signals is via the FFT algorithm. Let

$$S\left(\frac{2\pi m}{N}\right) = \sum_{n=0}^{N-1} s(n)\mathrm{e}^{-j\frac{2\pi nm}{N}}$$

denote the DFT of a length N signal $s(n)$. Consider a map $\theta(\omega)$ mapping the interval $[-\pi, \pi]$ onto itself and extend $\theta(\omega)$ outside this interval by letting

$$\theta(\omega + 2k\pi) = \theta(\omega) + 2k\pi,\ k \text{ integer.}$$

The last requirement is necessary in order to guarantee that the warped discrete-time signal has a 2π-periodic Fourier transform

$$S_{fw}(\omega + 2k\pi) = S(\theta(\omega + 2k\pi)) = S(\theta(\omega) + 2k\pi) = S(\theta(\omega)) = S_{fw}(\omega),$$

i.e., $S_{fw}(\omega)$ is the Fourier transform of a discrete-time signal. In order to obtain the frequency-warped signal we would need to compute $S\left(\theta\left(\frac{2\pi m}{N}\right)\right)$ and then perform the inverse Fourier transform. However, from the DFT we only know $S(\omega)$ at integer multiples of $\frac{2\pi}{N}$. The map $\theta(\omega)$ is arbitrary and $\theta\left(\frac{2\pi m}{N}\right)$ is not necessarily a multiple of $\frac{2\pi}{N}$. However, we may approximate $\theta\left(\frac{2\pi m}{N}\right)$ with the nearest integer multiple of $\frac{2\pi}{N}$, i.e., we can define the quantized map

$$\theta_q\left(\frac{2\pi m}{N}\right) = \frac{2\pi}{N}\mathrm{round}\left[\theta\left(\frac{2\pi m}{N}\right)\frac{N}{2\pi}\right].$$

The values $S\left(\theta_q\left(\frac{2\pi m}{N}\right)\right)$ are known from the DFT of the signal and we can compute the approximated frequency-warped signal by means of the inverse DFT:

$$s_{fw}(n) \approx \frac{1}{N}\sum_{m=0}^{N-1} S\left(\theta_q\left(\frac{2\pi m}{N}\right)\right)\mathrm{e}^{j\frac{2\pi nm}{N}}.$$

The diagram of the frequency-warping algorithm via FFT is shown in Figure 11.4. If the warping map is an increasing function, one can introduce the equalization factor as in (11.2) simply

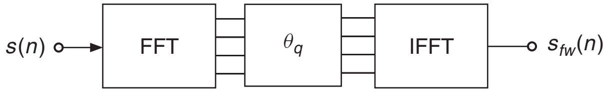

Figure 11.4 Frequency warping by means of FFT: schematic diagram.

by multiplying $S\left(\theta_q\left(\frac{2\pi m}{N}\right)\right)$ by the factor $\left.\sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}\right|_{\omega=\frac{2\pi m}{N}}$ before processing with the IFFT block. The FFT algorithm for warping is rather efficient, with a complexity proportional to $N \log N$. However, it has some drawbacks. The quantization of the map introduces distortion in the desired frequency spectrum, given by repetitions of the same value in phase and magnitude at near frequencies. These sum almost coherently and are perceived as beating components that have a slow amplitude decay. In frequency-warping signals one must pay attention to the fact that the warped version of a finite-length signal is not necessarily finite length. In the FFT-warping algorithm, components that should lie outside the analysis interval are folded back into it causing some echo artifacts. Furthermore, even if the original warping map is one-to-one, the quantized map is not and the warping effect cannot be undone without losses. The influence of the artifacts introduced by the FFT-warping algorithms may be reduced by zero-padding the original signal in order to obtain a larger value of N and, at the same time, a smaller quantization step for θ, at the expense of an increased computational cost.

Dispersive delay lines

In order to derive alternate algorithms for frequency warping [Bro65, OJ72], consider the DTFT (11.2) of the scaled frequency-warped version of a causal signal $s(n)$,

$$\widetilde{S}_{fw}(\omega) = \sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}} S(\theta(\omega)) = \sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}} \sum_{n=0}^{\infty} s(n)\mathrm{e}^{-jn\theta(\omega)}. \tag{11.3}$$

The last formula is obtained by considering the DTFT of the signal $s(n)$, replacing ω with $\theta(\omega)$ and multiplying by $\sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}$. The warped signal $\widetilde{s}_{fw}(k)$ is obtained from the inverse DTFT of $\widetilde{S}_{fw}(\omega)$:

$$\widetilde{s}_{fw}(k) = \mathrm{IDTFT}\left[\widetilde{S}_{fw}(\omega)\right](k) = \sum_{n=0}^{\infty} s(n)\mathrm{IDTFT}\left[\sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}\mathrm{e}^{-jn\theta(\omega)}\right](k). \tag{11.4}$$

Defining the sequences $\lambda_n(k)$ as follows,

$$\lambda_n(k) = \mathrm{IDTFT}\left[\sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}\mathrm{e}^{-jn\theta(\omega)}\right](k) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} \sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}\mathrm{e}^{j[k\omega - n\theta(\omega)]}\mathrm{d}\omega \tag{11.5}$$

we can put (11.4) in the form

$$\widetilde{s}_{fw}(k) = \sum_{n=0}^{\infty} s(n)\lambda_n(k). \tag{11.6}$$

If we find a way of generating the sequences $\lambda_n(k)$, then we have a new algorithm for frequency warping, which consists of multiplying these sequences by the signal samples and adding the result. From (11.5) we have an easy way for accomplishing this since

$$\Lambda_n(\omega) = \mathrm{DTFT}[\lambda_n](\omega) = \Lambda_{n-1}(\omega)\mathrm{e}^{-j\theta(\omega)}, \tag{11.7}$$

with

$$\Lambda_0(\omega) = \sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}.$$

Notice that the term $\mathrm{e}^{-j\theta(\omega)}$ has magnitude 1 and corresponds to an allpass filter. The sequence $\lambda_0(k)$ may be generated as the impulse response of the filter $\sqrt{\frac{\mathrm{d}\theta}{\mathrm{d}\omega}}$. The sequence $\lambda_n(k)$ is obtained

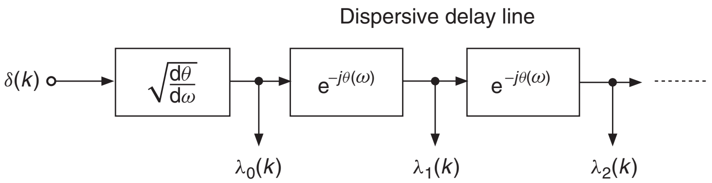

Figure 11.5 Dispersive delay line for generating the sequences $\lambda_n(k)$.

by filtering $\lambda_{n-1}(k)$ through the allpass filter $e^{-j\theta(\omega)}$. This can be realized in the structure of Figure 11.5 for computing the sequences $\lambda_n(k)$ as the impulse responses of a chain of filters. In order to perform warping it suffices to multiply each of the outputs by the corresponding signal sample and sum these terms together. The structure is essentially a delay line in which the elementary delays are replaced by allpass filters. Each of these filters introduces a frequency dependent group delay

$$\tau_G(\omega) = \frac{d\theta}{d\omega}.$$

The result is reminiscent of propagation of light in dispersive media where speed depends on frequency. For this reason this structure is called a dispersive delay line. What happens if we input a generic signal $y(k)$ to the dispersive delay line? The outputs $\widehat{y}_n(k)$ are computed as the convolution of the input signal by the sequences $\lambda_n(k)$,

$$y(k) * \lambda_n(k) = \sum_r y(k-r)\lambda_n(r).$$

As a special case, for $k = 0$ and choosing as input the signal $s(k) = y(-k)$, which is the time-reversed version of $y(k)$, we obtain

$$\widehat{y}_n(0) = \sum_r s(r)\lambda_n(r).$$

The last equation should be compared with (11.6) to notice that the summation is now over the argument $\lambda_n(r)$. However, we can define the transposed sequences

$$\lambda_r^T(n) \equiv \lambda_n(r),$$

and write

$$\widehat{y}_n(0) = \sum_r s(r)\lambda_r^T(n). \tag{11.8}$$

From (11.5) we have

$$\lambda_r^T(n) = \text{IDTFT}\left[\sqrt{\frac{d\theta}{d\omega}}e^{-jn\theta(\omega)}\right](r) = \frac{1}{2\pi}\int_{-\pi}^{+\pi}\sqrt{\frac{d\theta}{d\omega}}e^{j[r\omega-n\theta(\omega)]}d\omega. \tag{11.9}$$

Suppose that the map $\theta(\omega)$ has odd parity, is increasing and maps π onto π. Then we can perform in (11.9) the same change of variable $\Omega = \theta(\omega)$ as in (11.1) to obtain

$$\lambda_r^T(n) = \frac{1}{2\pi}\int_{-\pi}^{+\pi}\sqrt{\frac{d\theta^{-1}}{d\omega}}e^{j[n\omega-r\theta^{-1}(\omega)]}d\omega.$$

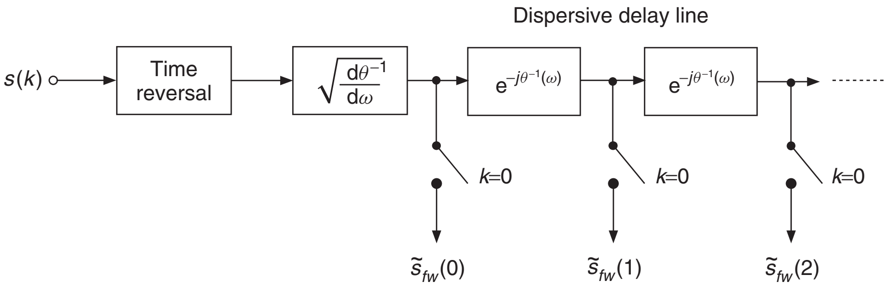

Figure 11.6 Computational structure for frequency warping.

As a result,

$$\Lambda_r^T(\omega) = \sqrt{\frac{d\theta^{-1}}{d\omega}} e^{jr\theta^{-1}(\omega)},$$

hence the transposed sequences $\lambda_r^T(n)$ have the same form as the sequences $\lambda_r(n)$ except that they are based on the inverse map $\theta^{-1}(\omega)$. Consequently, (11.8) is a formula for unwarping the signal. Furthermore, by exchanging the roles of $\theta(\omega)$ and $\theta^{-1}(\omega)$, (11.8) is also a valid algorithm for warping. The corresponding structure is shown in Figure 11.6. The input signal is time reversed, then fed to the $\sqrt{\frac{d\theta^{-1}}{d\omega}}$ filter and to the dispersive delay line. The output of each filter is collected at time $k = 0$ by means of switches closing at that instant to form the scaled frequency-warped sequence $\tilde{s}_{fw}(n)$. The structures in Figures 11.5 and 11.6 still present some computational problems. In general, the transfer functions involved are not rational. Furthermore, an infinite number of filters is needed for computing the transform. One can show that the only one-to-one map implementable by a rational transfer function is given by the phase of the first-order allpass filter

$$A(z) = \frac{z^{-1} - b}{1 - bz^{-1}}, \tag{11.10}$$

where $-1 < b < 1$. By varying the real parameter b in the allowed range, one obtains the family of Laguerre curves shown in Figure 11.7. The curves with a negative value of the parameter are the inverses of those with a positive value, i.e., the inverse mapping $\theta^{-1}(\omega)$ corresponds to a sign reversal of the parameter. One can show that for causal signals the derivative $\sqrt{\frac{d\theta^{-1}}{d\omega}}$ can be replaced by the filter

$$\Lambda_0^T(z) = \frac{\sqrt{1 - b^2}}{1 + bz^{-1}}.$$

The structure in Figure 11.6 includes a time-reversal block and switches closing at time zero. It is clear that for a finite-length N signal one can equivalently form the signal $s(N - n)$ and close the switches at time $k = N$. Furthermore, by inspection of the structure, the required number M of allpass filters is approximately given by N times the maximum group delay, i.e.,

$$M \approx N \frac{1 + |b|}{1 - |b|}.$$

A larger number of sections would contribute little or nothing to the output signal.

The main advantage of the time-domain algorithm for warping is that the family of warping curves is smooth and does not introduce artifacts, as opposed to the FFT-based algorithm illustrated in the above. Furthermore, the effect can be undone and structures for unwarping signals are obtained by the identical structure for warping provided that we reverse the sign of the parameter.

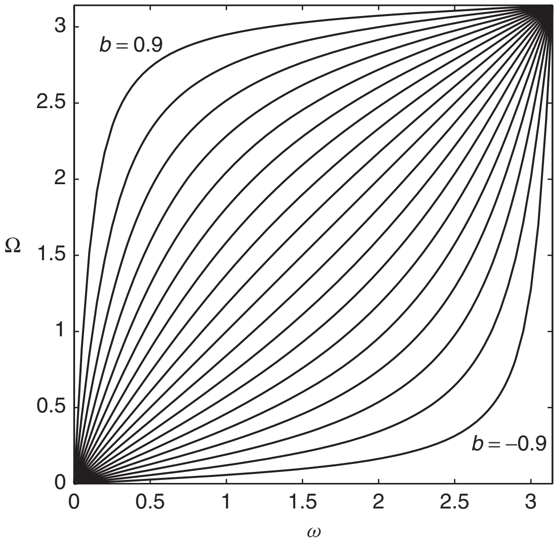

Figure 11.7 The family of Laguerre warping maps.

In fact, the frequency-warping algorithm corresponds to the computation of an expansion over an orthogonal basis, giving rise to the Laguerre transform. Next we provide a simple MATLAB® function implementing the structure of Figure 11.6. The following M-file 11.1 gives a simple implementation of the Laguerre transform.

M-file 11.1 `(lagt.m)`

```
function y=lagt(x,b,M)
% Author: G. Evangelista
% computes M terms of the Laguerre transform y of the input x
% with Laguerre parameter b
N=length(x);
x=x(N:-1:1);     % time reverse input
% filter by normalizing filter lambda_0
yy=filter(sqrt(1-b^2),[1,b],x);
y(1)=yy(N);      % retain the last sample only
for k=2:M
% filter the previous output by allpass
 yy=filter([b,1],[1,b],yy);
 y(k)=yy(N);     % retain the last sample only
end
```

11.2.4 Short-time warping and real-time implementation

The frequency-warping algorithm based on the Laguerre transform illustrated in Section 11.2.3 is not ideally suited to real-time implementation. Besides the computational cost, which is of the order of N^2, each output sample depends on every input sample. Another drawback is that with a long signal the frequency-dependent delays cumulate to introduce large delay differences between high and low frequencies. As a result, the time organization of the input signal is destroyed by frequency warping. This can also be seen from the computational structure in Figure 11.6, where subsignals pertaining to different frequency regions of the spectrum travel with different speeds along the dispersive delay line. At sampling time some of these signals have reached the end of the line, whilst other are left behind. For example, consider the Laguerre transform of a signal $s(n)$ windowed by a length N window $h(n)$ shifted on the interval $rM, ..., rM + N - 1$. According to (11.6) we obtain

$$\widetilde{s}_{fw}^{(r)}(k) = \sum_{n=rM}^{rM+N-1} h(n - rM)s(n)\lambda_n(k) = \sum_{n=0}^{N-1} x^{(r)}(n)\lambda_{n+rM}(k), \tag{11.11}$$

where

$$x^{(r)}(n) = h(n)s(n + rM).$$

The DTFT of (11.11) yields

$$\widetilde{S}_{fw}^{(r)}(\omega) = \mathrm{e}^{-jrM\theta(\omega)} \sum_{n=0}^{N-1} x^{(r)}(n)\Lambda_n(\omega) = \mathrm{e}^{-jrM\theta(\omega)}\Lambda_0(\omega)X^{(r)}(\theta(\omega)). \tag{11.12}$$

From this we can see that the spectral contribution of the signal supported on $rM, ..., rM + N - 1$ is delayed, in the warped signal, by the term $\mathrm{e}^{-jM\theta(\omega)}$, which introduces a largely dispersed group delay $M\tau_G(\omega)$. Approximations of the warping algorithm are possible in which windowing is applied in order to compute a short-time Laguerre transform (STLT) and, at the same time, large frequency-dependent delay terms are replaced by constant delays. In order to derive the STLT algorithm, consider a window $w(n)$ satisfying the perfect overlap-add condition

$$\sum_{r=-\infty}^{+\infty} w(n - rL) = 1, \tag{11.13}$$

where $L \leq N$ is an integer. This condition says that the superposition of shifted windows adds up to one. If $\widetilde{s}_{fw}(n)$ denotes the Laguerre transform (11.6) of the signal $s(n)$, then we have identically,

$$\widetilde{s}_{fw}(k) = \sum_{r=-\infty}^{+\infty} w(k - rL)\widetilde{s}_{fw}(k) = \sum_{r=-\infty}^{+\infty} \sum_{n=0}^{+\infty} s(n)w(k - rL)\lambda_n(k). \tag{11.14}$$

By taking the DTFT of both sides of (11.14) one can show that

$$\widetilde{S}_{fw}(\omega) = \sum_{r=-\infty}^{+\infty} \mathrm{e}^{-jrL\omega} \frac{1}{2\pi} \int_{-\pi}^{+\pi} \Lambda_0(\Omega)S\left(\theta(\Omega)\right) W(\omega - \Omega)\mathrm{e}^{jrL\Omega}\mathrm{d}\Omega. \tag{11.15}$$

On the other hand, from (11.12) a delay compensated version of $\widetilde{S}_{fw}^{(r)}(\omega)$ is

$$\widehat{S}_{fw}^{(r)}(\omega) = \mathrm{e}^{-jr(L\omega - M\theta(\omega))}\widetilde{S}_{fw}^{(r)}(\omega) = \mathrm{e}^{-jrL\omega}\Lambda_0(\omega)X^{(r)}(\theta(\omega)), \tag{11.16}$$

which is the DTFT of the sequence

$$\widehat{s}_{fw}^{(r)}(k) = \sum_{n=0}^{N-1} h(n)s(n + rM)\lambda_n(k - rL). \tag{11.17}$$

This equation defines the short-time Laguerre transform (STLT) of the signal $s(n)$. In order to select the proper integer M we need to study the term $X^{(r)}(\theta(\omega))$. One can show that

$$X^{(r)}(\theta(\omega)) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} S\,(\Omega)\, H(\theta(\omega)-\Omega)\mathrm{e}^{jrM\Omega}\mathrm{d}\Omega. \tag{11.18}$$

We would like to approximate the integral in (11.15) by $\Lambda_0(\omega)X^{(r)}(\theta(\omega))$. Suppose that $H(\omega)$ is an unwarped version of $W(\omega)$, i.e., that

$$H(\omega) = \frac{\mathrm{d}\theta^{-1}(\omega)}{\mathrm{d}\omega}W(\theta^{-1}(\omega)) = \left|\Lambda_0^T(\omega)\right|^2 W(\theta^{-1}(\omega)). \tag{11.19}$$

By performing in (11.18) the change of variable $\Omega = \theta(\omega) + \theta(\alpha - \omega)$ we obtain

$$X^{(r)}(\theta(\omega)) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} S\,(\theta(\omega) - \theta(\omega-\alpha))\, W(\omega-\alpha)\mathrm{e}^{jrM(\theta(\omega)-\theta(\omega-\alpha))}\mathrm{d}\alpha. \tag{11.20}$$

Since $W(\omega)$ is a lowpass function, only the terms for $\alpha \approx \omega$ contribute to the last integral. Therefore, from (11.16) and (11.20) we conclude that the superposition of delay-compensated versions of $\widetilde{S}^{(r)}_{fw}(\omega)$ can be approximated as follows,

$$\widehat{S}_{fw}(\omega) = \sum_{r=-\infty}^{+\infty}\widehat{S}^{(r)}_{fw}(\omega) \approx \sum_{r=-\infty}^{+\infty}\mathrm{e}^{-jrL\omega}\frac{1}{2\pi}\int_{-\pi}^{+\pi}\Lambda_0(\alpha)S\,(\theta(\alpha))\, W(\omega-\alpha)\mathrm{e}^{jrM\theta(\alpha)}\mathrm{d}\alpha. \tag{11.21}$$

Equation (11.21) should be compared with (11.15). A linear approximation of $\theta(\alpha)$ is

$$\theta(\alpha) = \theta'(0)\alpha + O\left(\alpha^3\right) = \frac{1+b}{1-b}\alpha + O\left(\alpha^3\right). \tag{11.22}$$

One can show that this is a fairly good approximation for $|\alpha| < \frac{1-b}{2}\pi$. In this frequency range, if we select

$$M \approx \frac{1-b}{1+b}L \tag{11.23}$$

then

$$\widehat{S}_{fw}(\omega) \approx \widetilde{S}_{fw}(\omega),$$

i.e., the overlap-add of STLT components well approximates the Laguerre transform. In other words, an approximate scheme for computing the Laguerre transform consists of taking the Laguerre transform of overlapping signal frames windowed by the unwarped window $h(n)$ and overlap-adding the result, as shown in Figure 11.9. This method allows for a real-time implementation of frequency warping via the Laguerre transform. It relies on the linear approximation (11.22) of the phase of the allpass, valid for the low-frequency range. An important issue is the choice of the window $w(n)$. Many classical windows, e.g., rectangular, triangular, etc., satisfy condition (11.13). However, (11.21) is a close approximation of the Laguerre transform only if the window sidelobes are sufficiently attenuated. Furthermore, the unwarped version (11.19) of the window can be computed via a Laguerre transform with the normalizing filter $\Lambda_0^T(\omega)$ removed. In principle $h(n)$ has infinite length. However, the inverse Laguerre transform of a lowpass window $w(n)$ has essential length

$$N \approx \left.\frac{\mathrm{d}\theta^{-1}}{\mathrm{d}\omega}\right|_{\omega=0} N_w = \frac{1-b}{1+b}N_w.$$

In order to avoid artifacts in the multiplication of the signal by the window, we are interested in windows whose Laguerre transform essentially is a dilated or stretched version of the window itself. This property turns out to be approximately well satisfied by the Hanning window

$$w(n) = \begin{cases} \frac{1}{2}\left(1 - \cos\frac{2\pi n}{N_w}\right) & n = 0, 1, ..., N_w - 1 \\ 0 & \text{otherwise} \end{cases} .$$

The choice of the length N_w is arbitrary. Furthermore, the Hanning window satisfies (11.13) for any L integer submultiple of N_w. Long windows tend to better approximate pure frequency warping. However, both response time and computational complexity increase with the length of the window. Moreover, the time-organization destruction effect is more audible using extremely long windows. The integer L controls the overlap $N_w - L$ of the output warped frames. When warping with a positive value of the parameter b one should select a considerable overlap, e.g., $N_w = 5L$, in order to avoid amplitude distortion of the high-frequency components, which, in this case, are more concentrated in the Laguerre domain, as shown in Figure 11.8. Finally, the integer M fixing the input frames overlap is obtained by rounding the right-hand side of (11.23). Next we provide a simple M-file 11.2 implementing frequency warping by means of STLT overlap-add. The function gives a simple implementation of frequency warping via short-time Laguerre transform.[1]

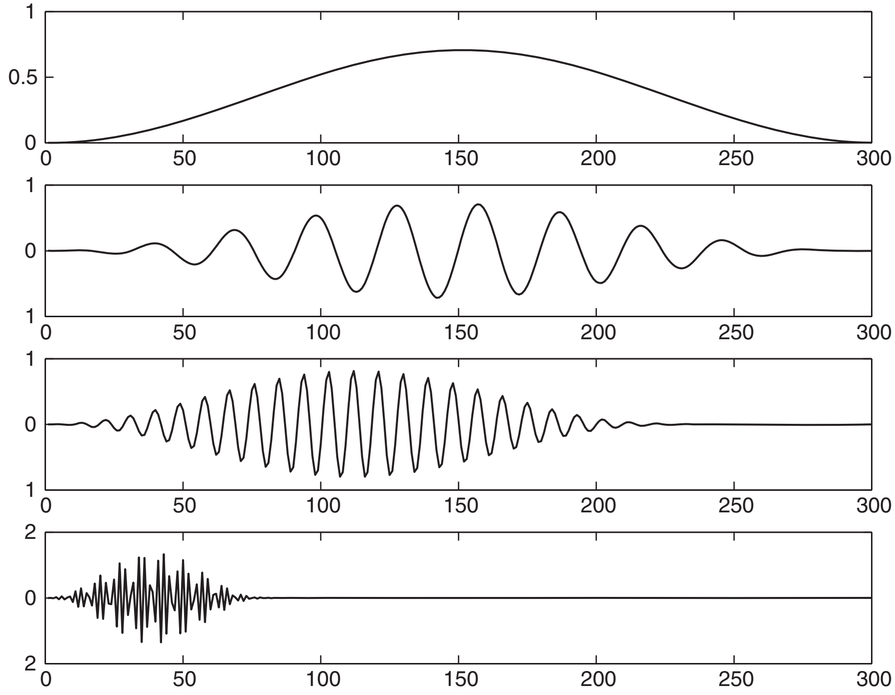

Figure 11.8 Short-time warping: different length of warped signals from low to high frequencies (top to bottom).

[1] The function lugtun is the same as lagt reported in Section 11.2.3, except that the line yy = filter (sqrt(1-b ˆ 2),[1,b],x); is replaced by the line yy = x; in order to compute the non-normalized Laguerre transform.

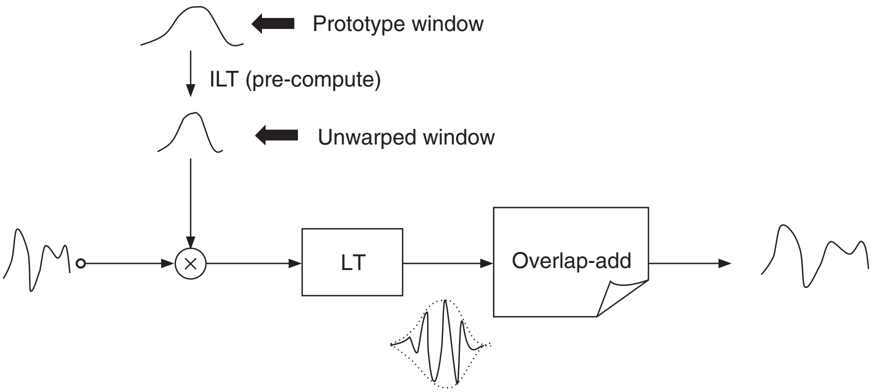

Figure 11.9 Block diagram of the approximate algorithm for frequency warping via overlap-add of the STLT components. The block LT denotes the Laguerre transform and ILT its inverse.

M-file 11.2 (winlagt.m)

```
function sfw=winlagt(s,b,Nw,L)
% Author: G. Evangelista
% Frequency warping via STLT of the signal s with parameter b,
% output window length Nw and time-shift L
w=L*(1-cos(2*pi*(0:Nw-1)/Nw))/Nw;      % normalized Hanning window
N=ceil(Nw*(1-b)/(1+b));                % length of unwarped window h
M=round(L*(1-b)/(1+b));                % time-domain window shift
h=lagtun(w,-b,N); h=h(:)               % unwarped window
Ls=length(s);                          % pad signal with zeros
K=ceil((Ls-N)/M);                      % to fit an entire number
s=s(:); s=[s ; zeros(N+K\ast M-Ls,1)]; % of windows
Ti=1; To=1;                            % initialize I/O pointers
Q=ceil(N*(1+abs(b))/(1-abs(b)));       % length of Laguerre transform
sfw=zeros(Q,1);                        % initialize output signal
for k=1:K
 yy=lagt(s(Ti:Ti+N-1).*h,b,Q);         % Short-time Laguerre transf.
 sfw(To:end)=sfw(To:end)+yy;           % overlap-add STLT
 Ti=Ti+M;To=To+L;                      % advance I/O signal pointers
 sfw=[sfw; zeros(L,1)];                % zero pad for overlap-add
end
```

11.2.5 Vocoder-based approximation of frequency warping

A major drawback of the short-time Laguerre-transform-based algorithm for warping described in Section 11.2.4 is that the warping map approximation (11.22) is only valid in the low-frequency range. This results in a choice of window length (11.23) that is satisfactory only in this range, generating artifacts for wideband signals. Moreover, the warping map $\theta(\omega)$ is still constrained to follow one of the Laguerre curves in Figure 11.7.

An alternate approximate algorithm for frequency warping [EC07] can be derived from the STFT based time-frequency representation of the discrete time signal $s(n)$,

$$s(n) = \sum_{q=0}^{M-1} \sum_{r} S_{q,r} g_{q,r}(n), \tag{11.24}$$

where

$$g_{q,r}(n) = g(n - rN) \mathrm{e}^{j\frac{2\pi}{M}qn}, \tag{11.25}$$

for $q = 0, 1, \ldots, M-1$ and $r \in \mathbf{Z}$, are overlapping time-shifted and frequency-modulated windows and

$$S_{q,r} = \sum_{n} x(n) g^{*}_{q,r}(n) \tag{11.26}$$

are the STFT coefficients of the signal. The integer M controls the frequency resolution and is here constrained to an integer multiple $M = KN$ of the time-shift integer N, for some integer K. If the window g has finite length M, then the factor K determines the number of windows that have non-zero overlap in each length N segment. Intuitively, each narrow-band component of the time-frequency representation intercepts only a small portion of the warping map θ, which makes a local linear approximation possible.

The analysis and synthesis real windows $g(n)$ in (11.26) and (11.24) are identical as we assume that

$$\sum_{r} g^2(n - rN) = \frac{1}{M} \tag{11.27}$$

for perfect reconstruction. For our purposes, $g(n)$ can be identified with the sine window

$$g(n) = \sqrt{\frac{2}{KM}} \sin\left(\frac{\pi n}{M}\right), \tag{11.28}$$

in which case the following Parseval relationship holds true,

$$\sum_{q=0}^{M-1} \sum_{r} \left|S_{q,r}\right|^2 = \|s\|^2. \tag{11.29}$$

Taking the DTFT of both sides of (11.24) yields

$$S(\omega) = \sum_{q=0}^{M-1} \sum_{r} S_{q,r} G_{q,r}(\omega), \tag{11.30}$$

where

$$G_{q,r}(\omega) = \mathrm{e}^{-jrN\left(\omega - \frac{2\pi q}{M}\right)} G\left(\omega - \tfrac{2\pi q}{M}\right). \tag{11.31}$$

As a result, the DTFT of the warped signal $S_{fw}(\omega) = S(\theta(\omega))$ can be obtained from (11.30) by replacing the DTFT of the shifted-modulated windows $G_{q,r}(\omega)$ with the warped windows

$$\tilde{G}_{q,r}(\omega) = \mathrm{e}^{-jrN\left(\theta(\omega) - \frac{2\pi q}{M}\right)} G\left(\theta(\omega) - \tfrac{2\pi q}{M}\right). \tag{11.32}$$

Since the window is a low pass function, the warped windows (11.32) are essentially non-zero only for ω in the neighborhood of the roots of the equation $\theta(\omega) = \frac{2\pi q}{M}$. If the map θ is invertible, a first order Taylor series approximation about the unique roots $\omega_q = \theta^{-1}\left(\frac{2\pi q}{M}\right)$ is then justified:

$$\theta(\omega) \approx \theta(\omega_q) + \theta'(\omega_q)(\omega - \omega_q) = \tfrac{2\pi q}{M} + \beta_q(\omega - \omega_q), \tag{11.33}$$

where

$$\beta_q = \theta'\left(\omega_q\right) = \theta'\left(\theta^{-1}\left(\tfrac{2\pi q}{M}\right)\right) = \left(\frac{\mathrm{d}\theta^{-1}}{\mathrm{d}\omega}\bigg|_{\omega=\frac{2\pi q}{M}}\right)^{-1}. \tag{11.34}$$

Considering (11.33) for each synthesis band results in the locally linear approximation of the warping map shown in Figure 11.10.

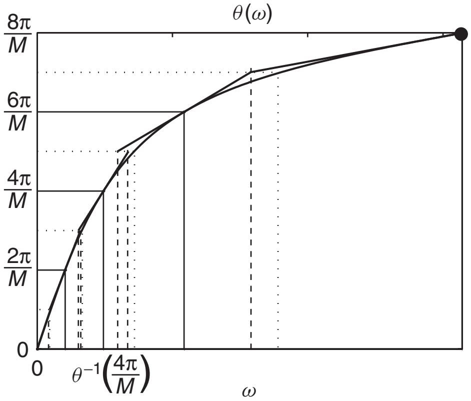

Figure 11.10 Locally linear approximation of warping map: center bands (solid lines) and band edges (dotted lines).

In the same approximation we have

$$\tilde{G}_{q,r}(\omega) \approx \mathrm{e}^{-jrN\beta_q(\omega-\omega_q)} G(\beta_q(\omega - \omega_q)). \tag{11.35}$$

Like several other windows, the sine window $g(n)$ is obtained by sampling a continuous time function $g^{(a)}(t)$, so that $g(n) = g^{(a)}(n)$. In that case, one can show [EC07] that

$$\tilde{g}_{q,r}(n) \approx \frac{\mathrm{e}^{j\omega_q n}}{\beta_q} g^{(a)}\left(\frac{n - r\beta_q N}{\beta_q}\right), \tag{11.36}$$

which is a scaled, shifted and modulated version of the original window. For a unitary warping operation (see Section 11.2.2) the window (11.36) must be further multiplied by a factor $\sqrt{\beta_q}$, which is the square root of the derivative of the map evaluated at the point ω_q.

Since the shift factor $\beta_q N$ is not necessarily an integer, the computation is greatly simplified if $\beta_q N$ is rounded to the nearest integer N_q. The resulting quantization error of the time-shift steps

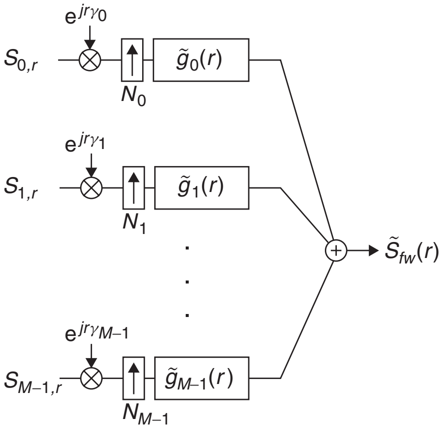

Figure 11.11 Synthesis filter bank structure for computing approximate frequency warping from STFT analysis coefficients.

is negligible for sufficiently large N. Similarly, the window length $\beta_q M = \beta_q KN$ can be rounded to the integer $M_q = KN_q$. In this case, the approximated computation of frequency warping from the STFT $S_{q,r}$ of the signal $s(n)$ can be performed by means of the multirate filter bank shown in Figure 11.11, where

$$\tilde{g}_q(n) = \tilde{g}_{q,0}(n), \tag{11.37}$$

with phase factors $\gamma_q = N_q\omega_q$.

The modulating frequencies of the impulse responses of the various channels of the filter bank are not harmonically related and the window length is channel dependent. Therefore, while the analysis coefficients can be obtained by frame-by-frame FFT, the approximate warping synthesis structure cannot be implemented in an efficient FFT-based computational scheme. In the operation count per number of samples, the synthesis filter bank has linear complexity. If the signal is real and if the warping map has odd symmetry, then only the first half of the complex filter-bank sections need to be computed, the output being formed by taking the real part.

With the given time-frequency approximate warping algorithm, real-time operation is only possible if $N_q \geq N$ for any q. This constraint is satisfied if $\theta'(\omega) \geq 1$ for any ω. In fact, in the multirate structure in Figure 11.11 each channel operates at a different rate regulated by the upsampling factor N_q. When $N_q < N$ the synthesis structure produces fewer samples than those required by the STFT analysis block. In this case, the output lags behind the input and real-time operation is ruled out. In synthesis applications the map and the input pitch can both be suitably scaled to guarantee real-time operation. In audio effects applications, in order to compute warping in real-time, interpolation or prediction of the STFT data must be introduced in those channels where the warping map has $\theta'(\omega) < 1$, similarly to that often introduced in real-time pitch-shifters when the pitch is shifted upwards.

The artifacts introduced by the vocoder approximation of frequency warping are much less severe than the ones introduced with short-time Laguerre transform, both numerically and perceptually, as measured with PEAQ quality index. The main problems arise from the quantization of time-shift step and window length, especially in conjunction with flatter portions of the warping map (small derivative). These problems are mitigated by using proportionally larger values of M and N. Increasing the overlap does not improve quality.

11.2.6 Time-varying frequency warping

Suppose that each frequency-dependent delay element in the structure of Figure 11.5 has its own phase characteristics $\theta_k(\omega)$ and suppose that we remove the scaling filter. Accordingly, the outputs of the structure are the sequences

$$\psi_n(k) = a_1(k) * a_2(k) * \dots * a_n(k),$$

with $\psi_0(k) = \delta(k)$, obtained by convolving the impulse responses $a_m(k)$ of the allpass filters

$$A_m(z) = \frac{z^{-1} - b_m}{1 - b_m z^{-1}}.$$

Hence the z-transforms of the sequences $\psi_n(k)$ are

$$\Phi_n(z) = \prod_{k=1}^{n} \frac{z^{-1} - b_m}{1 - b_m z^{-1}}$$

and their DTFT is

$$\Phi_n(\omega) = \prod_{k=1}^{n} e^{-j\theta_k(\omega)} = \mathrm{e}^{-j\Theta_n(\omega)},$$

where

$$\Theta_n(\omega) = \sum_{k=1}^{n} \theta_k(\omega)$$

is the sign-reversed cumulative phase of the first n delay elements. By multiplying each signal sample $s(n)$ by the corresponding sequence $\varphi_n(k)$ we obtain the signal

$$s_{tvfw}(k) = \sum_{n=0}^{+\infty} s(n)\varphi_n(k),$$

whose DTFT is

$$S_{tvfw}(\omega) = \sum_{n=0}^{+\infty} s(n)e^{-j\Theta_n(\omega)}.$$

Note that this is an important generalization of (11.3) in which the phase terms are not integer multiples of each other. In the special case where all the delays are equal we have $\theta_k(\omega) = \theta(\omega)$ and $\Theta_n(\omega) = n\theta(\omega)$. If we suppose that the delays are equal in runs of N, then signals of finite length N, supported on the intervals $(r-1)N, \dots, rN-1$ are frequency warped according to distinct characteristics. For the same reason, signal samples falling in these intervals are differently warped. Portions of the signal falling in two adjacent intervals are warped in a mixed way. More generally, one can have a different delay for each signal sample. This results in a time-varying frequency warping [EC99, EC00]. From a musical point of view one is often interested in slow and oscillatory variations of the Laguerre parameter, as we will discuss in Section 11.3. It is possible to derive a computational structure for time-varying warping analogous to that reported in Figure 11.6. This is obtained by considering the sequences $\psi_n(k)$ whose z-transforms satisfy the following recurrence:

$$\Psi_0(z) = \frac{1}{1 - b_1 z^{-1}}$$

$$\Psi_n(z) = H_n(z)\Psi_{n-1}(z),$$

where

$$H_n(z) = \frac{1 - b_n b_{n+1}}{1 - b_{n-1} b_n} \frac{z^{-1} - b_{n-1}}{1 - b_{n+1} z^{-1}}$$

and $b_0 = 0$. This set of sequences plays the same role as the transposed sequences (11.9) in the Laguerre expansion. However, the sequences $\varphi_n(k)$ and $\psi_n(k)$ are not orthogonal, rather, they are biorthogonal, i.e.,

$$\sum_{k=0}^{+\infty} \varphi_n(k)\psi_m(k) = \delta(n - m).$$

Consequently, our time-varying frequency-warping scheme is not a unitary transform of the signal, hence it does not verify the energy preservation property (11.1). However, one can show that this is a complete representation of signals. Hence the time-varying frequency warping is an effect that can be undone without storing the original signal. The modified structure for computing time-varying frequency warping is reported in Figure 11.12. In order to preserve the same direction of warping as in the fixed parameter Laguerre transform, the sign of the parameter sequence must be reversed, which is equivalent to exchanging the roles of $\theta_n(\omega)$ and $\theta_n^{-1}(\omega)$. The inverse structure can be derived in terms of a tapped dispersive delay line based on $\Phi_n(\omega)$ with the warped signal samples used as tap weights. Next we provide a simple M-file 11.3 implementing the structure of Figure 11.12. The function gives a simple implementation of the variable parameter generalized Laguerre transform.

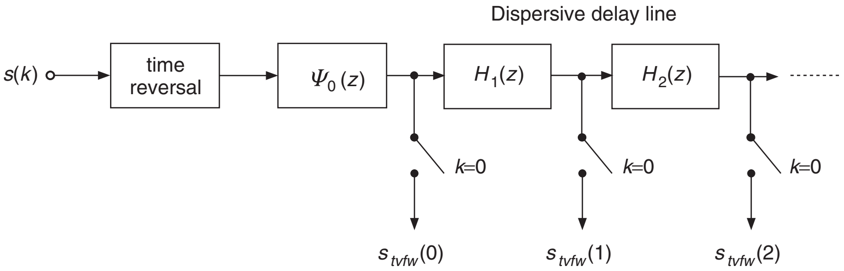

Figure 11.12 Structure for computing time-varying frequency warping via generalized Laguerre transform with variable parameter.

M-file 11.3 (lagtbvar.m)

```
function y=lagtbvar(x,b,M)
% Author: G. Evangelista
% computes coefficients y of biorthogonal Laguerre expansion of x
% using variable parameters b(k) where b is a length M array
N=length(x);
yy=x(N:-1:1);                    % time reverse input
y=zeros(1,M);
yy=filter(1,[1, b(1)],yy);      % filter by psi_0(z)
y(1)=yy(N);                      % retain the last sample only
% filter by H_1(z)(unscaled, b to -b)
yy=filter([0,1],[1, b(2)],yy);
y(2)=yy(N)*(1-b(1)*b(2));       % retain the last sample only and scale
for k=3:M
 % filter by H_(k-1)(z)(unscaled, b to -b)
 yy=filter([b(k-2),1],[1, b(k)],yy);
 y(k)=yy(N)*(1-b(k-1)*b(k));   % retain the last sample only and scale
end
```

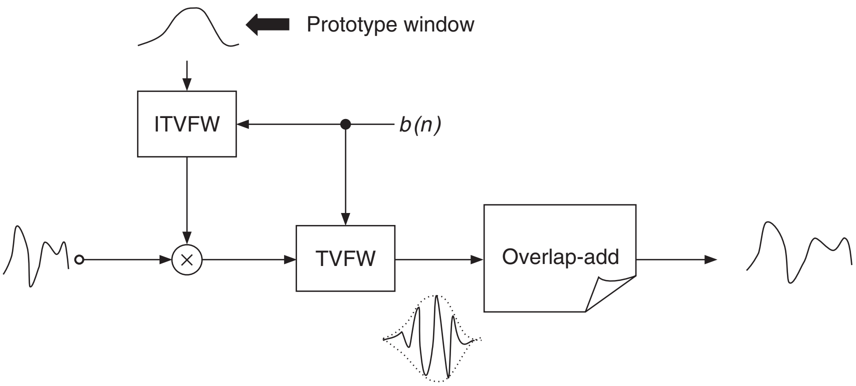

Figure 11.13 Block diagram of the approximate algorithm for time-varying frequency warping. The blocks TVFW and ITVFW respectively denote time-varying frequency warping and its inverse.

Time-varying frequency warping has a fast approximate algorithm whose block diagram is reported in Figure 11.13. The scheme is similar to the overlap-add method derived for the Laguerre transform and is shown in Figure 11.9. However, due to the time-varying aspect, the inverse time-varying warping of the prototype window must be computed for each input frame.

An extension of the vocoder-based frequency-warping algorithm discussed in Section 11.2.5 for handling the time-varying case is available [E08], which works under the assumption that the warping map is constant in each time interval of N samples. Since changes in the warping map are audible only over time intervals larger than 25 ms, this constraint on the update rate is not too severe.

11.3 Musical uses of warping

In this section we describe a few applications of warping in music. As already pointed out, many aspects and properties of warping musical signals are still to be explored and many results of this section may be deemed as experimental. Applications that will be discussed range from accurate pitch-shifting of inharmonic sources and inharmonization of harmonic sources, to feature and transient extraction, vibrato editing and morphing.

11.3.1 Pitch-shifting inharmonic sounds

The sounds from a large class of instruments are inherently inharmonic. The spacing of the frequencies of the partials is not uniform. In piano sounds, in the low register, the displacement of the partials from the harmonics becomes more and more apparent as we move towards the lower end of the keyboard. In Figure 11.14 we report data (× marks) extracted from a low-pitch piano tone ($\approx$ 27 Hz). These represent the differences between the frequency of a partial and that of the next one. If the sound were harmonic, one should observe a flat distribution of points aligned on the pitch frequency. On the contrary, one observes that the spacing between the partials increases with the order of the overtones. The distribution of the partials can be closely matched to the derivative of a Laguerre curve. This can be obtained by means of an optimization of the parameter b in (11.10). It turns out that the absolute value of the optimum Laguerre parameter decreases as we move from lower to higher tones. This means that the warping curve becomes more and more linear, as can be seen from Figure 11.7. By frequency warping the original piano tone with the inverse of the fitted Laguerre map one transforms the originally inharmonic partials into a set of

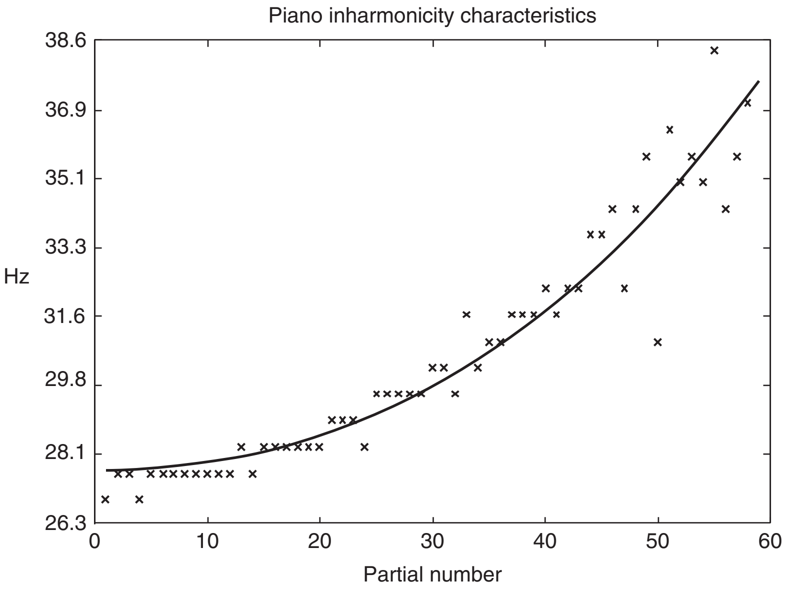

Figure 11.14 Inharmonicity characteristics of a 27 Hz piano tone: data are marked by x and the solid curve represents the optimum Laguerre difference curve fitting the data.

harmonic partials. As a result of warping the fundamental frequency, the pitch of the resulting tone will be higher. Vice versa, by warping by a Laguerre map with a small positive value of the parameter one decreases pitch and increases the degree of inharmonicity. This gives us a method for pitch-shifting piano tones that is particularly accurate in matching the inharmonicity of lower tones. Given a piano tone one can determine the value and the sign of the warping parameter in order to transform it to a lower or higher tone. Specifically, suppose that the fundamental frequency is f_0 and that the desired frequency is $\widehat{f_0}$. In terms of the normalized frequency ω, with a sampling rate f_S, we have, respectively, $\omega_0 = \frac{2\pi f_0}{f_S}$ and $\widehat{\omega}_0 = \frac{2\pi \widehat{f_0}}{f_S}$. As remarked in Section 11.2.2 the new normalized fundamental frequency after warping is $\widehat{\omega}_0 = \theta^{-1}(\omega_0)$. One can show that

$$\theta^{-1}(\omega) = 2\arctan\left(\frac{1-b}{1+b}\tan\frac{\omega}{2}\right),$$

hence we can determine the required value of b as follows:

$$b = \frac{\tan\frac{\pi f_0}{f_S} - \tan\frac{\pi \widehat{f_0}}{f_S}}{\tan\frac{\pi f_0}{f_S} + \tan\frac{\pi \widehat{f_0}}{f_S}}. \tag{11.38}$$

For inharmonic sounds, pitch-shifting by frequency warping is more accurate than conventional algorithms based on proportional scaling of fundamental frequency and overtones. In fact, the warping characteristics can be ultimately justified by means of a physical model of stiff strings or membranes [VS94,TEC97]. It is quite striking that the Laguerre characteristics match those of piano tones for a large range. Therefore one obtains accurate pitch-shifting and the inharmonicity law by pure frequency warping. Otherwise one should resort to a combination of conventional

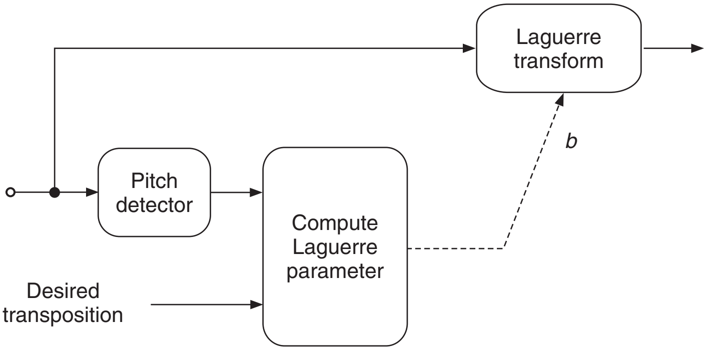

Figure 11.15 Block diagram of inharmonic sounds pitch shifter.

pitch-shifting and warping. The block diagram of a pitch shifter for inharmonic sounds based on the Laguerre transform is shown in Figure 11.15. Frequency warping can also be used in conjunction with proportional pitch-shifting algorithms to pitch-shift harmonic signals. These techniques usually yield a rational alteration of the pitch and one needs to improve their resolution. Also, ratios other than simple ratios with small integers in both numerator and denominator are costly from a computational point of view. By frequency warping the proportionally pitch-shifted signal with a small absolute value of the warping parameter one can introduce a small incremental pitch-shifting operation, which, when added to the rational pitch-shifting operation, provides a pitch closer or equal to the desired pitch. At the same time, the inharmonicity introduced is not perceptually relevant due to the small value of the warping parameter.

11.3.2 Inharmonizer

Among the new effects introduced by frequency warping is the inharmonizer. This effect is obtained by frequency warping an original harmonic sound with a large absolute value ($\simeq$0.5) of the parameter. The resulting sound is enriched by inharmonic partials, maps of the original harmonic partials, as discussed in Section 11.2.2. Notice that both pitch and duration of the original sound are altered by warping. In fact, frequency warping stretches or shrinks the width of the peaks centered on the partial frequencies. As a result, the amplitude envelopes of the partials are altered. In the first approximation they are simply time-scaled. In order to restore the original pitch and duration one can resort to resampling techniques. At conventional sampling rates (20–44 kHz) the fundamental frequency of a large class of sounds from natural instruments falls into the low-frequency portion of the axis. In that region the warping map is approximately linear with coefficients which are the derivative of the map in $\omega = 0$. This is also the amount by which the duration of the signal is scaled. This makes it possible to achieve pitch and duration rescaling by a single resampling operation. In many cases the inharmonizer effect introduces interesting detuning of the higher partials, transforming, for example, a trumpet sound into a bell-like sound or a guitar sound into a piano-like sound.

The inharmonizer can also be used in physical model synthesis, e.g., as a Karplus–Strong post-processing block, in order to model inharmonicity due to dispersive propagation in stiff media [TEC04]. In that case, the synthesis block can be implemented with elementary delays instead of a long chain of allpass filters. Moreover, using the vocoder-based warping algorithm of Section 11.2.5, one can achieve higher flexibility in the choice of the warping map in order to match physical dispersion characteristics.

11.3.3 Comb filtering + warping and extraction of excitation signals in inharmonic sounds

As previously pointed out, by frequency warping the original piano tone with the inverse of the fitted Laguerre map, one transforms the originally inharmonic partials into a set of harmonic partials. This property can be exploited in order to extract the hammer noise from piano sounds. In fact, the audible effect of the hammer noise lies in areas of the frequency spectrum that are not masked by the partials, i.e., in between the partials. It is easy to build a comb filter based on the harmonics of the transformed piano sound. In fact, given a narrow-band lowpass filter with frequency response $H(\omega)$, the frequency response $H(\omega P)$, where P is the period of the signal expressed in number of samples, is a comb filter adjusted to the harmonics. This filter is obtained by inserting $P - 1$ zeros in the filter coefficients. Likewise, if $G(\omega)$ is a high pass filter, the filter $G(\omega P)$ will select all the frequency bands that lie in between the harmonics. In order to obtain the piano hammer noise it suffices to unwarp the signal in order to regularize the partials into harmonics, determine the transformed pitch, filter with $G(\omega P)$ and apply frequency warping to re-obtain the inharmonic distribution. In the present case it is more convenient to pre-warp the filters rather than the signals. However, in a more general setting where the inharmonic signal components are analyzed by means of pitch-synchronous wavelets [Eva93, Eva94], which include downsampling operations, it can be shown that it is more convenient to warp the signal [EC97, EC98a, EC98b]. The block diagram of a tuned warped comb filter is shown in Figure 11.16.

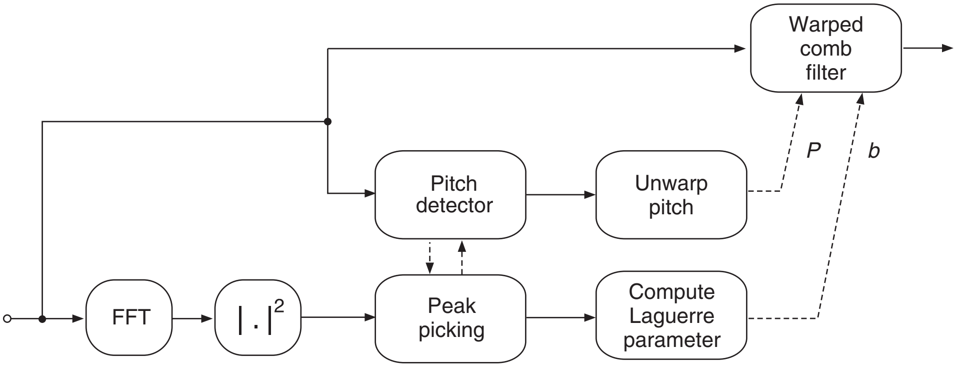

Figure 11.16 Block diagram of tuned warped comb structure for extracting partials or excitation noise from inharmonic sounds.

11.3.4 Vibrato, glissando, trill and flatterzunge

Vibrato can be generated by means of time-varying frequency warping, by using an oscillating sequence of parameters b with low amplitude and frequency. For small values of the warping parameter, the warping curve only slightly deviates from the linear map and the harmonic structure of the signal is essentially preserved, while pitch-shifting is the only perceptually relevant effect. This is especially true when the parameter law is oscillatory so that the harmonics fluctuate around their original frequency. This allows us to introduce dynamic pitch fluctuations in natural or synthetic sounds, which can be directly controlled by the warping parameter sequence according to Equation (11.38). In particular, one can use a sinusoidal LFO as a control parameter generator to insert very natural vibrato. Both the frequency and amplitude of the oscillator can be changed at will, i.e., to synchronize the effect on the amplitude envelope of the signal or to include random fluctuations. Trill and rapid fluctuations of the pitch can be obtained by means of a square wave

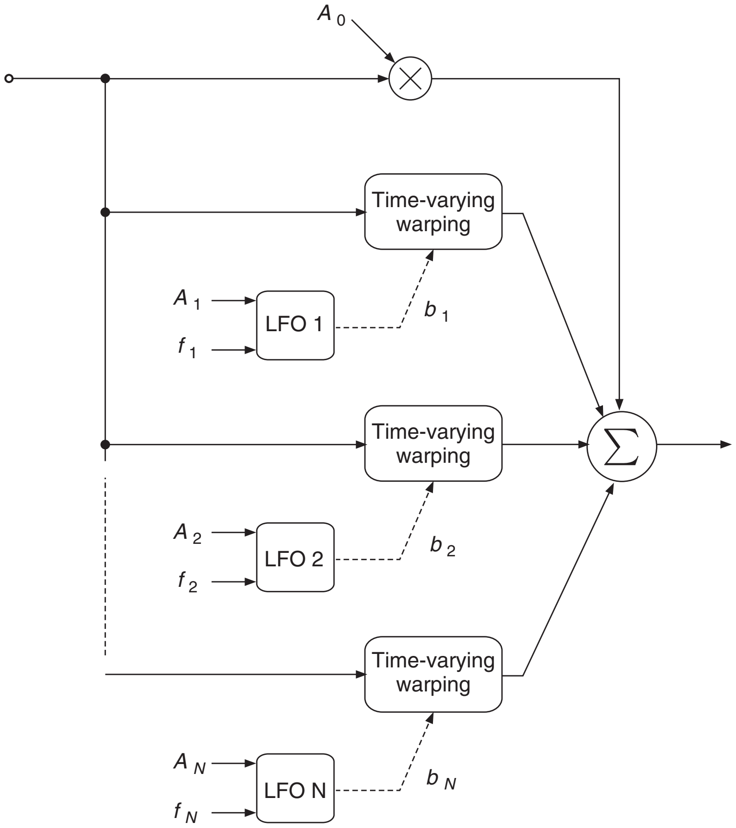

Figure 11.17 Block diagram for computing vibrato, trill, chorus-like, phasing-like or flange-like effects. For Flatterzunge we add random noise to the LFOs. For glissando the LFOs are replaced by envelope generators.

LFO. By mixing pitch-modulated versions of the sound with the original signal one can obtain effects similar to phasing, flanging and chorusing. By frequency warping a flute sound using random noise or random amplitude square wave as parameter sequences one obtains interesting effects typical of Flatterzunge. As another example, glissando can be inserted by means of an increasing or decreasing sequence of parameters. A general structure based on mixed independent time-varying warping channels for computing the above effects is shown in Figure 11.17. In much the same way, one can edit sounds containing vibrato or any pitch modulation in order to reduce or remove this effect. It suffices to extract the pitch fluctuation law from the sound by means of a pitch detection algorithm or by tracking the partials in the spectrogram of the sound. From this law one can obtain the law of variation of the parameter b and by applying the time-varying frequency-warping algorithm with a reversed sign sequence of parameters, one can counteract the pitch modulation effect [EC00].

11.3.5 Morphing

Accurate spectral morphing requires arbitrary maps of the frequency axis in order to transform the partials of one sound into the partials of another sound. The FFT warping algorithm illustrated in Section 11.2.3 can be employed with simplicity to perform this task. However, since invertibility

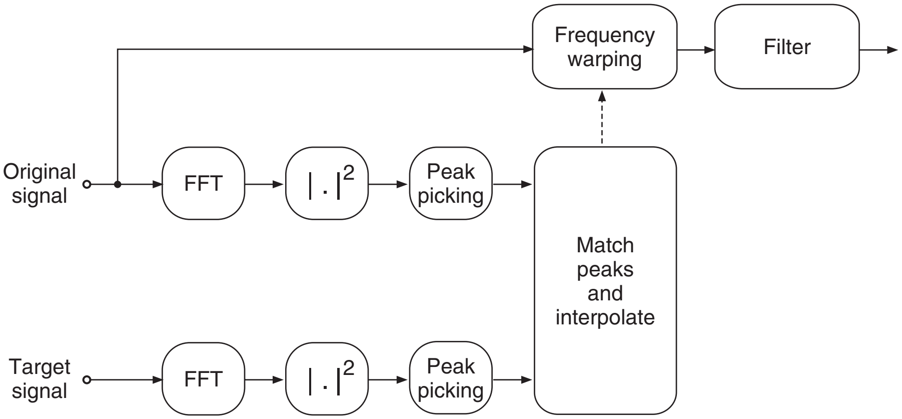

Figure 11.18 Simple diagram for computing morphing via frequency warping.

is not an issue, versions of the Laguerre transform based on higher-order allpass filters can be employed as well. In order to determine the suitable warping map one can use a peak-picking algorithm in the frequency domain to detect the partials of both the original and desired sound. Simple morphing examples can be computed using the structure shown in Figure 11.18. A set of points on an initial-final frequency plane is determined, which can be interpolated to produce a smooth warping curve. As an example one can eliminate the even harmonics in a voiced sound by mapping these onto odd harmonics. Realistic morphing also requires amplitude scaling of the partials. This corresponds to a simple filtering operation on the signal. Morphing can also be performed as a dynamic operation by means of time-varying frequency warping using a sequence of maps.

11.4 Conclusion

In this chapter we introduced a class of digital audio effects based on frequency-warping techniques of recent interest in musical applications. The deformation of the frequency axis, whether static or dynamic, introduces a new point of view and new tools for processing sounds. This transformation allows us to insert or edit vibrato, trill, Flatterzunge and glissando, adding controlled expression to static sounds. Harmonic sounds can be mapped into inharmonic sounds, introducing fine partial detuning to color them. Frequency warping also provides a concerned or model-based method for pitch-shifting inherently inharmonic sounds such as piano and drums sounds. Mixing independent time-varying warping channels achieves interesting generalizations of flanging, chorusing and phasing effects. An efficient algorithm based on the short-time Laguerre transform makes frequency warping computable in real-time. Since frequency warping is at present fairly unexploited in musical contexts we encourage musicians and sound engineers to experiment with this appealing technique.

References

[Bro65] P. W. Broome. Discrete orthonormal sequences. *J. Assoc. Comput. Machinery*, 12(2): 151–168, 1965.

[E08] G. Evangelista. Modified phase vocoder scheme for dynamic frequency warping. In *Proc. IEEE 3rd Int. Symp. Commun. Control Signal Process. (ISCCSP 2008)*, pp. 1291–1296, 2008.

[EC97] G. Evangelista and S. Cavaliere. Analysis and regularization of inharmonic sounds via pitch-synchronous frequency warped wavelets. In *Proc. Int. Comp. Music Conf*., pp. 51–54, 1997.

[EC98a] G. Evangelista and S. Cavaliere. Discrete frequency warped wavelets: theory and applications. *IEEE Trans. Signal Process.*, special issue on Theory and Applications of Filter Banks and Wavelets, 46(4): 874–885, 1998.

[EC98b] G. Evangelista and S. Cavaliere. Frequency warped filter banks and wavelet transform: a discrete-time approach via Laguerre expansions. *IEEE Trans. Signal Proc.*, 46(10): 2638–2650, 1998.

[EC99] G. Evangelista and S. Cavaliere. Time-varying frequency warping: results and experiments. In *Proc. DAFX-99 Digital Audio Effects Workshop*, pp. 13–16, 1999.

[EC07] G. Evangelista and S. Cavaliere. Real-time and efficient algorithms for frequency warping based on local approximations of warping operators. In *Proc. Digital Audio Effects Conf. (DAFx '07)*, pp. 269–276, 2007.

[EC00] G. Evangelista and S. Cavaliere. Audio effects based on biorthogonal time-varying frequency warping. *EURASIP J. Appl. Signal Process.*, 1(1): 27–35, 2001.

[Eva93] G. Evangelista. Pitch synchronous wavelet representations of speech and music signals. *IEEE Trans. Signal Process.*, special issue on Wavelets and Signal Processing, 41(12): 3313–3330, 1993.

[Eva94] G. Evangelista. Comb and multiplexed wavelet transforms and their applications to signal processing. *IEEE Trans. Signal Process.*, 42(2): 292–303, 1994.

[OJ72] A. V. Oppenheim and D. H. Johnson. Discrete representation of signals. *Proc. IEEE*, 60: 681–691, 1972.

[TEC04] I. Testa, G. Evangelista and S. Cavaliere. Physically inspired models for the synthesis of stiff strings with dispersive waveguides. *EURASIP J. Appl. Signal Process.* special issue on Model-Based Sound Synthesis. 2004(7): 964–977, 2004.

[TEC97] I. Testa, G. Evangelista and S. Cavaliere. A physical model of stiff strings. In *Proc. Inst. Acoust. (Internat. Symp. on Music and Acoustics, ISMA'97)*, vol. 19: Part 5 (1997) Book 1, pp. 219–224, Edinburgh, 1997.

[VS94] S. A. Van Duyne and J. O. Smith. A simplified approach to modeling dispersion caused by stiffness in strings and plates. In *Proc. Int. Comp. Music Conf.*, pp. 407–410, 1994.