Time and frequency warping musical signals

DAFX Presentation

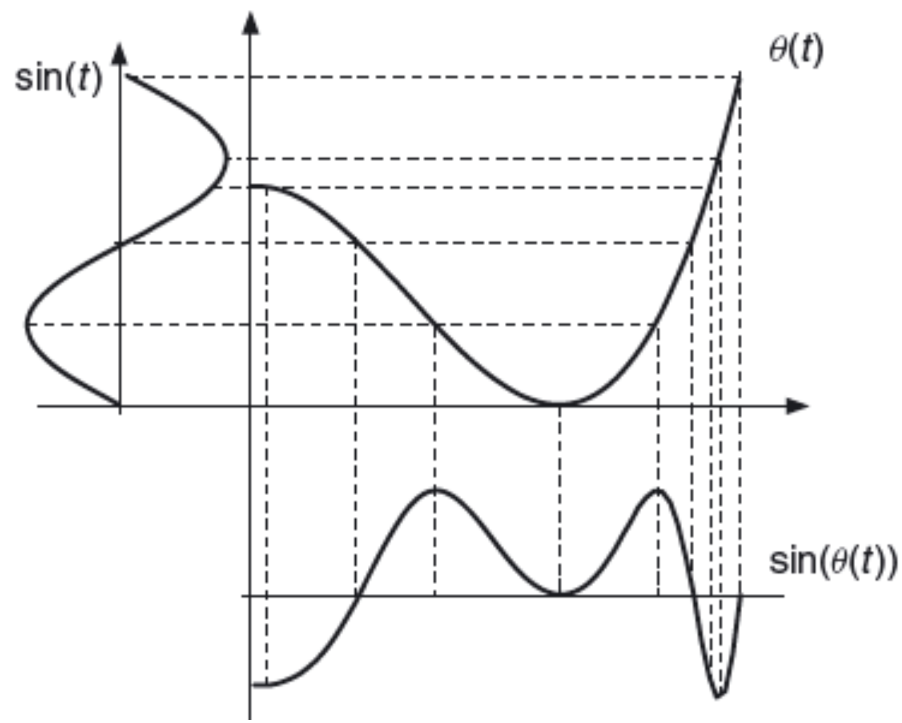
Time Warping

-Warping map θ(t): maps points of the original t-axis onto points of the transformed axis

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

-Map only invertible if one-to-one

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



Frequency Warping

-DTFT of signal is :

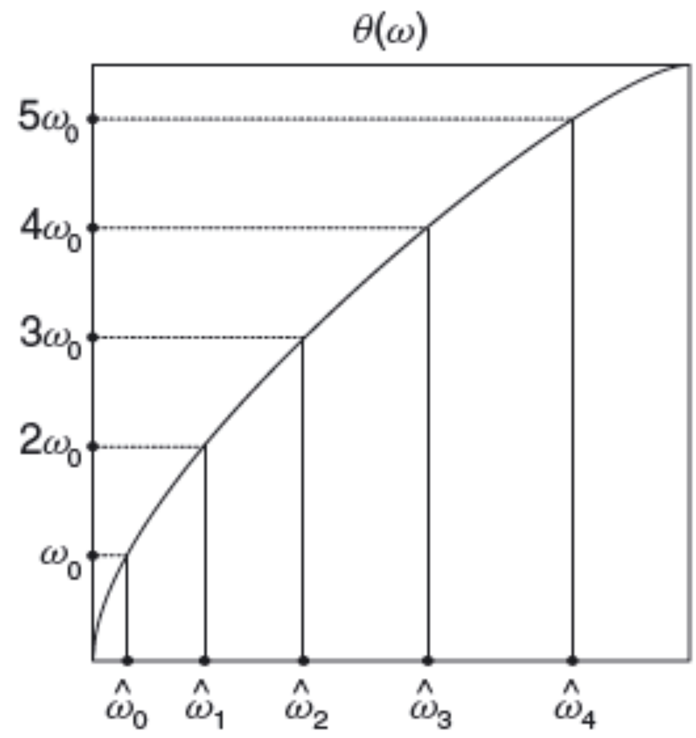
$$S_{fw}(\omega) = S(\theta(\omega))$$

-Warping map must have odd parity meaning:

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

-Applied to a signal with spectrum peaks at multiples of ω0:

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



Energy Preservation

-Warping can result in attenuation or amplification of different frequency bands

-Energy in given band is: $$E_{[\omega_0,\omega_1]} = \frac{1}{2\pi}\int_{\omega_0}^{\omega_1} |S(\omega)|^2\,\mathrm{d}\omega$$

-Substitute $\omega = \theta(\Omega)$ yields:

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

-Meaning: the energy in any band $[\omega_0, \omega_1]$ of the original signal equals the energy of the warped signal in the warped band $[\theta^{-1}(\omega_0), \theta^{-1}(\omega_1)]$.

Energy Preservation Cont'd

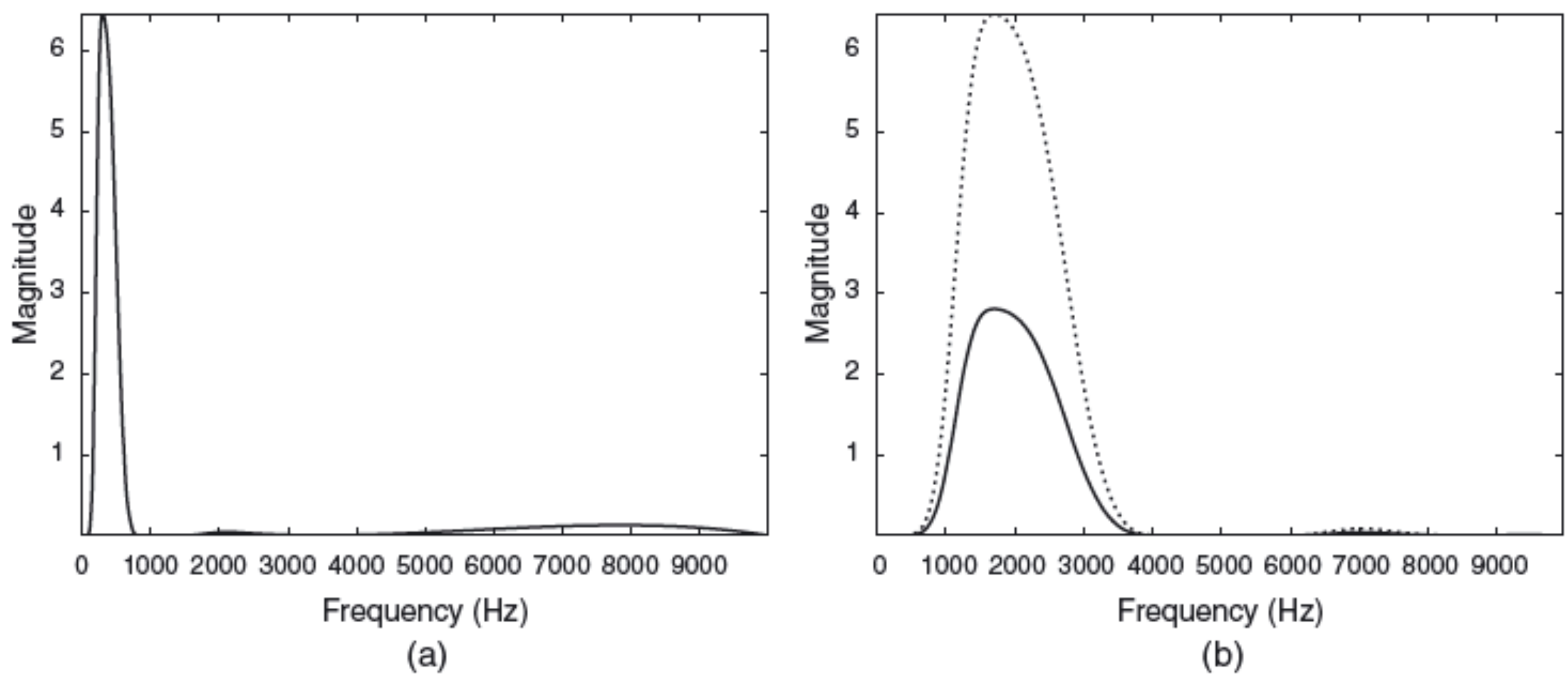


Figure 11.3 Frequency warping a narrow-band signal: (a) original frequency spectrum; (b) frequency-warped spectrum (dotted line) and scaled frequency-warped spectrum (solid line).

Warping using FFT

-Denotes DFT of signal s(n) of length N:

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

-Warping map $\theta(\omega)$ must have periodic Fourier transform:

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

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

-To find $S_{fw}(\omega)$ compute $S\left(\theta\left(\frac{2\pi m}{N}\right)\right)$ and then inverse fourier transform:

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

-Only compute DFT at integer multiples of $\frac{2\pi}{N}$ so must quantize values:

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

-Preserve energy multiplying by $\sqrt{\frac{d\theta}{d\omega}}\Big|_{\omega=\frac{2\pi m}{N}}$ before the IDFT

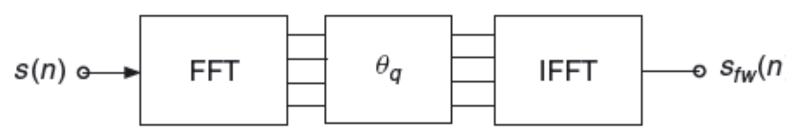


Figure 11.4 Frequency warping by means of FFT: schematic diagram.

-FFT method warping cannot be undone without losses. Quantization prevents mapping from being one-to-one

Dispersive Delay Lines

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

- $\widetilde{S}_{fw}(\omega)$ is the DTFT of a scaled frequency warped causal signal. Taking the IDTFT yields the warped signal:

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

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

-Defining $\lambda_n(k)$ and substituting reduces $\widetilde{s}_{fw}(k)$ to a simple convolution:

$$\widetilde{s}_{fw}(k) = \sum^{\infty} s(n)\lambda_n(k).$$

-Use the impulse response of a cascade of all pass filters to find series of lambda values.

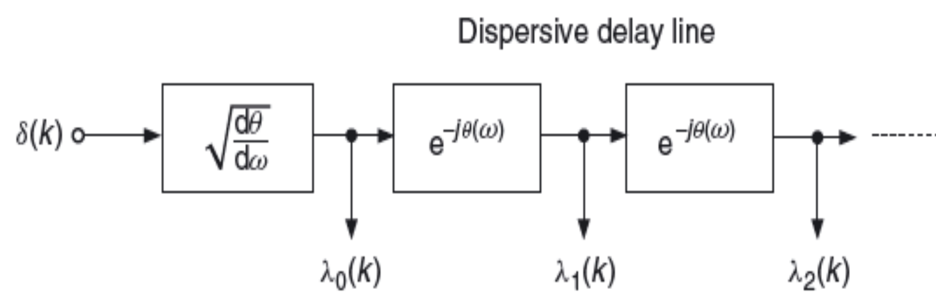


Figure 11.5 Dispersive delay line for generating the sequences $\lambda_n(k)$.

Dispersive Delays Cont'd

-This method is invertible. Use transpose of lambda in IDTFT:

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

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

-Convolution of $\lambda_r^T(n)$ with the warped signal will result in unwarping:

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

-A problem with this method is that is requires an infinite number of delays and most transfer functions involved are not rational

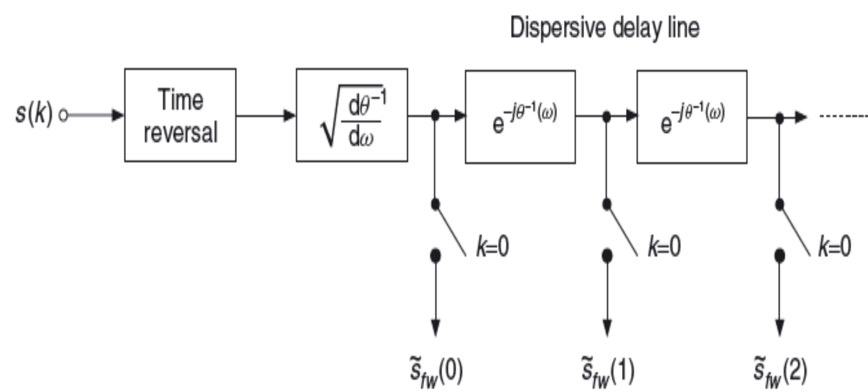


Figure 11.6 Computational structure for frequency warping.

Laguerre Transform

-The only one-to-one map implementable by a rational transfer functions is given by the phase of a first order all pass filter:

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

-Varying the b parameter from -1 to 1 yields the family of Laguerre curves:

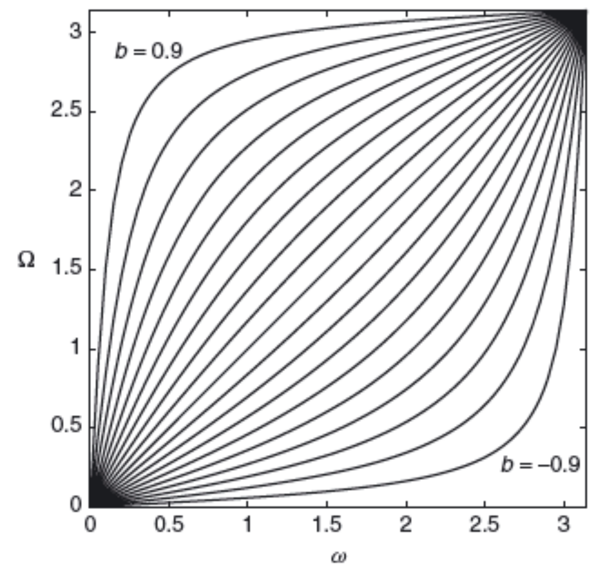


Figure 11.7 The family of Laguerre warping maps.

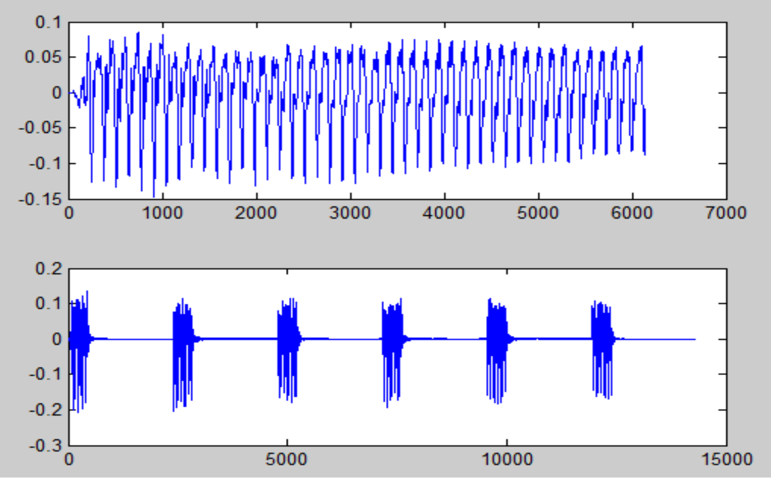
-The required number of all pass filters, M, is given by N times the maximum group delay:

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

Laguerre Transform Matlab Implementation

```
function y=lagt(x,b)

N=length(x);
M=N*(1+abs(b))/(1-abs(b))
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
```




```
x=wavread('sound.wav');

xx=[];
yy=[];
start=1;
finish=1024;

for i=(0:1:50)
    xTemp=x(start+i*1024:1024+i*1024);
    y=lagt(xTemp,-.4);

    xx=horzcat(xx,xTemp);
    yy=horzcat(yy,y);
end

 subplot(2,1,1)
 plot(xx)
 subplot(2,1,2)
 plot(yy)

wavwrite(yy,44100,'out.wav')
```

Short Time Warping

-The laguerre transform's computational cost is of the order N^2

$$\tilde{s}_{fw}^{(r)}(k) = \sum_{n=rM}^{rM+N-1} h(n-rM)s(n)\lambda_n(k)$$

-Windowing and then overlap overlap adding provides a good short time estimation

-Many window types will work, but the hanning window works particularly well due to the fact that it has a Laguerre transform that is essentially a dialated version of itself.

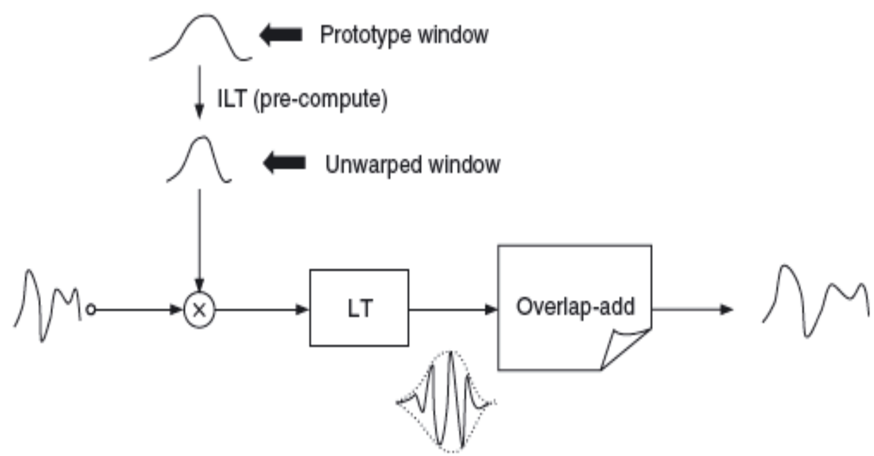


Figure 11.9 Block diagram of the approximate algorithm for frequency warping via overlap-add of the STLT components. The block LT denotes the Laguerre transform and ILT its inverse.

Short Time Laguerre Transform Implementation

```
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
```