Bilinear transforms

Henri

Henri

Henri Padé
1863–1953

Bilinear transform

A way of finding a z-transform given a Laplace transform.

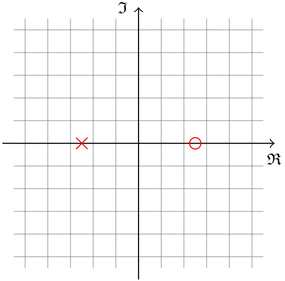
$$z = \exp(sT) = \frac{\exp\left(\frac{sT}{2}\right)}{\exp\left(-\frac{sT}{2}\right)} \approx \frac{1 + \frac{sT}{2}}{1 - \frac{sT}{2}}.$$

$$s \approx \frac{2}{T}\frac{z-1}{z+1}.$$

i.e.,

$$H(z) = H(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}$$

All pass filter, s-plane

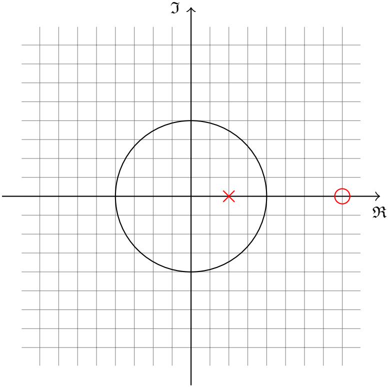


$$H(s) = \frac{a - s}{a + s}.$$

Consider the s-plane with one pole and one zero mirrored.

- These are real; they could be complex.
- There can be any number of pole zero pairs
- The frequency response is flat.

All pass filter, z-plane



This is from applying the bilinear transform to the all-pass function.
Notice that
if $\alpha = 0$, $H(z) = z^{-1}$.

$$H(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \qquad |\alpha| < 1.$$

Mapping

Say we have an input sequence $x_1, x_2, \ldots$ and an output sequence $y_1, y_2, \ldots$, related by

$$x_n = \sum_{k=-\infty}^{\infty} y_k \psi_{k,n}.$$

We want to find the output sequence.
It can be shown[1] that to map a discrete sequence to another discrete sequence in a way that preserves convolution,

$$\Psi_k(z) = [\Psi_1(z)]^k \, .$$

[1]This one I'm not going to prove

Delay line

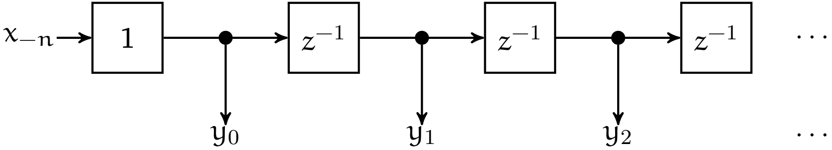
Take the trivial case, just a delay,

$$\Psi_k(z) = \left[z^{-1}\right]^k .$$

It's clear that

$$H_k(z) = z^{-k},$$

which can be expressed as a lattice like this:

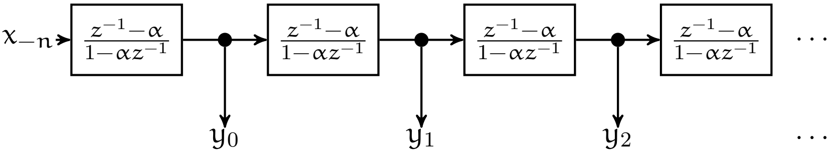


Delay line with all-pass

What happens if

$$\Psi_k(z) = \left[\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}\right]^k$$

You might expect something like this:

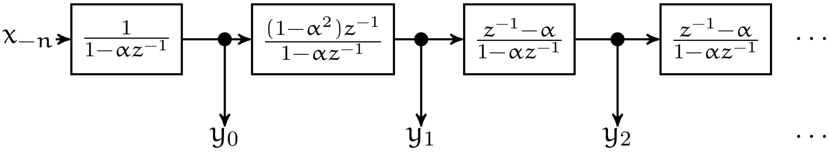


Oppenheim's lattice

In fact, it's less clear. Oppenheim showed that

$$H_k(z) = \frac{(1-\alpha^2)z^{-1}}{(1-\alpha z^{-1})^2}\left[\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}\right]^{k-1}, \qquad H_0(z) = \frac{1}{1-\alpha z^{-1}}.$$

which is

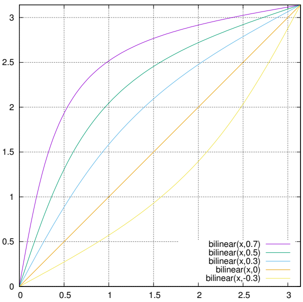


i.e., there is a small edge effect.

So what?

- The all-pass maps magnitude to magnitude.
  $\Longrightarrow$ The unit circle is mapped to the unit circle.
- The all pass messes with the phase.
  $\Longrightarrow$ The lattice of Oppenheim implements a frequency warp.

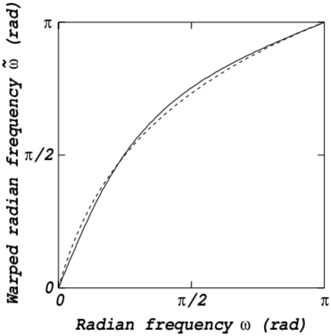
Frequency warp



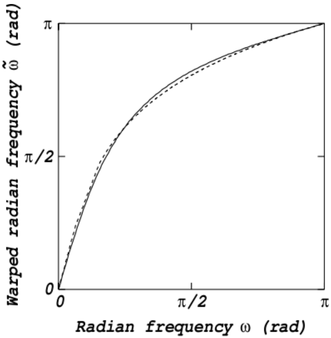
The frequency warp function is the phase response of the all-pass.

$$\tilde{\omega} = \omega + 2\tan^{-1}\left(\frac{\alpha \sin\omega}{1 - \alpha\cos\omega}\right).$$

Uses for bilinear transform



(a) mel scale ($\alpha = 0.42$)



(b) Bark scale ($\alpha = 0.55$)

Most common is as an approximation to Bark or mel.

Warpable things

It's possible to warp anything that has a network with z^{-n}:

- Obviously, time domain samples.
- Cepstra.
- Autoregression coefficients.
- Linear prediction coefficients.

Beware: A finite sequence is warped to an infinite sequence.

- The (truncated) warped sequence is not a complete representation.
- It's not necessarily invertible.
  But it is if you use (say) enough cepstra

Inversion

However, the most useful aspect of the bilinear transform is that it is invertible.

- No binning.
- Can be used for recognition and synthesis.

It's a linear transform

The lattice amounts to multiplication by this matrix:

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{M-1} \\ 0 & 1-\alpha^2 & 2\alpha(1-\alpha^2) & \dots & (M-1)\alpha^{M-2}(1-\alpha^2) \\ 0 & -\alpha(1-\alpha^2) & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -1^M\alpha^{M-2}(1-\alpha^2) & \dots & \dots & \dots \end{pmatrix}$$

i.e., A frequency warp is a linear transform of the cepstrum!

VTLN

Vocal Tract Length Normalisation.

- Alter the warping a bit depending on VTL.
- Effectively warp more for males, less for females.

but more later...

Root cepstrum

Spectral compression

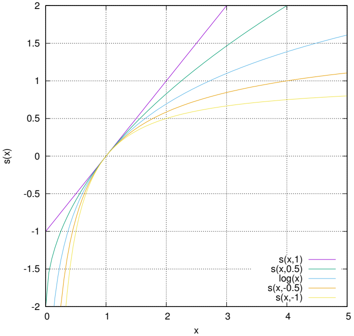
Oppenheim's homomorphic approach suggests using the (complex) logarithm.

- Useful to linearise convolutional effects.
- Fairly "nice" framework.
- Sensitive to small spectral values.

Hermansky's PLP uses the cube root of power spectrum.

- Somewhat closer to the form of human hearing.
- Naturally floored at zero.
- Is cube root really the right root?

Generalised logarithm



Say we define a function

$$s_\gamma(x) = \frac{x^\gamma - 1}{\gamma}.$$

How does it behave?

Inverse

The inverse function is

$$s_\gamma^{-1} = (1 + \gamma x)^{1/\gamma}.$$

Apply this to the z-transform representation of the cepstrum

$$\hat{H}(z) = \sum_{m=-\infty}^{\infty} c_m z^{-m} \approx \sum_{m=0}^{M} c_m z^{-m}.$$

Gives

$$H(z) = \begin{cases} \left(1 + \gamma \sum_{m=0}^{M} c_m z^{-m}\right)^{\frac{1}{\gamma}}, & 0 < |\gamma| \leqslant 1, \\ \exp \sum_{m=0}^{M} c_m z^{-m}, & \gamma = 0. \end{cases}$$

A generalised analysis

Different values of γ correspond to different traditional analyses:

$\gamma = 0$ Cepstrum. The generalised log reverts to the usual log.

$\gamma = -1$ All pole. But note that no optimisation criterion is specified yet.

$\gamma = 1$ All zero. In practice not that useful.

The most useful is $\gamma = -1$:

- The generalised logarithm has the same z-transform as an all pole model
- It's not a maximum likelihood fit though!

Optimality & UELS

So far it's defined backwards

- If we calculate a root-cepstrum, it can be thought of as an LP model.
- It's not optimal in any LP sense!

To make it optimal, you have to define an optimisation criterion.

Unbiased Estimation of Log Spectrum[2]

- An iterative algorithm to estimate parameters c_m.
- If $\gamma = -1$, amounts to the same LMS criterion as LPC.

For $0 < |\gamma| \leqslant 1$, UELS will give an optimal representation in terms of c_m.

- You still need to convert the result to the cepstrum if $\gamma \neq 1$.

[2]Which I ain't going to explain.

Uses

Root cepstrum is not quite the same as the homomorphic logarithm

- Doesn't quite linearise convolutional noise.
- The result is not quite a cepstrum.

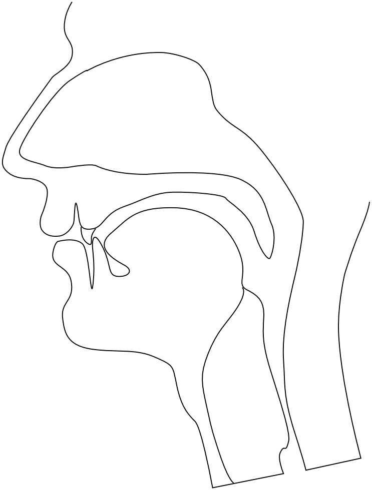
However,

- It does tend to give good results sometimes.
- Especially in noise.

Root cepstrum is most useful as a theoretical tool to unify several different approaches.

Vocal Tract Length Normalisation

VTLN



VTLN is inspired by the vocal tract.

- Vocal tract length is different for different people.
- Men have longer tracts than women.

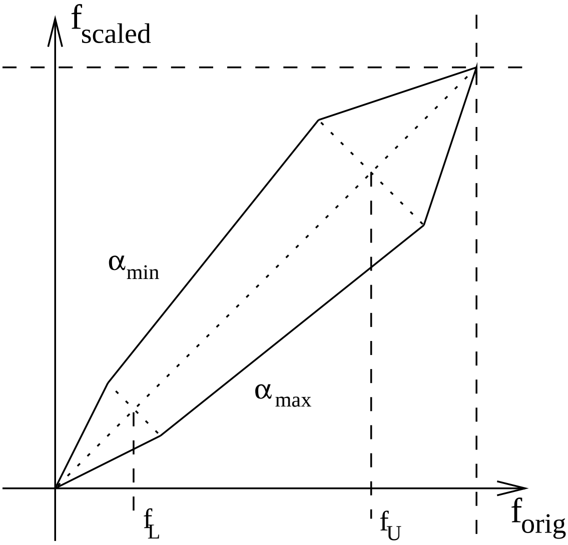
Principle

The general idea is to warp spectra to compensate for vocal tract length. Two ways to do it:

1. Build the warping into the warping that is done for perceptual reasons.
2. Add another warp on top of that warping.

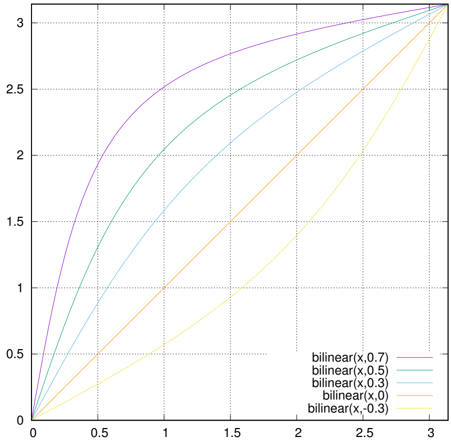
In practice, these blur into one…

The HTK kite



(From the HTK book)

The bilinear warp



Pros and cons

Kite

- Easy to implement.
- Too many parameters.
- Not clear what the final transformation is.

Bilinear

- Easy to implement, but a little slow.
- only one parameter.
- Can be applied to cepstra

...and in fact the bilinear transform is a linear transform in cepstral space.

Concatenating two bilinear warps

- Say we have a value, $\alpha_1 \approx 0.42$, for the perceptual warping.
- Concatenate a second warp, $\alpha_2 \approx 0.05$, for the VTLN.
- The total warp is actually a single warp with

$$\alpha = \frac{\alpha_1 + \alpha_2}{1 + \alpha_1 \alpha_2}.$$

How to calculate the warp factor?

In a theoretically uninspiring way!

- Calculate cepstra for several values of the warp factor.
- Use the one that “works best”

“Works best” can be

- Highest likelihood given a (speech recognition) model.
- Good speech recognition performance.
- Sounds nicest.

Is it worth it?

VTLN can respond to small amounts of training data

- It's useful for both recognition and synthesis
- This is what Lakshmi's thesis was about

However, remember that VTLN is a linear transform in the cepstrum!

- It will be subsumed by adaptation linear transforms.
- For moderate amounts of training data, feature space MLLR is better