The z-transform

Augustin-Louis

Augustin-Louis

Augustin-Louis Cauchy
1789–1857

Laurent series

I should have used Pierre Alphonse Laurent, but he died before he was famous enough to have his photo taken

- (Probably) started with (something like) Cauchy's formula

$$x_n = \frac{1}{2\pi j} \oint_\gamma \frac{F(z)}{(z-c)^{n+1}}\, dz$$

- Showed that it evaluates to

$$F(z) = \sum_{n=-\infty}^{\infty} x_n (z-c)^n$$

- Or if $c = 0$,

$$F(z) = \sum_{n=-\infty}^{\infty} x_n z^n$$

It's called a Laurent series[1]

[1] `https://en.wikipedia.org/wiki/Laurent_series`

Taylor series

- For z real, only sum from 0

$$F(z) = \sum_{n=0}^{\infty} x_n (z - c)^n$$

  is a Taylor series

- If $c = 0$

$$F(z) = \sum_{n=0}^{\infty} x_n z^n$$

  is a Maclaurin series (after Colin Maclaurin)

so, the Laurent series is not nearly as scary as it looks.

Definition

The z-transform is a Laurent series with $c = 0$ and negative n

$$F(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n},$$

$$x_n = \frac{1}{2\pi j} \oint_C dz\, F(z) z^{n-1}.$$

It has all the properties of the Taylor and Maclaurin series

- It's unique
- It decays quickly (infinite sum converges)

Why?

Sample a continuous signal

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$$
$$= \sum_{n=-\infty}^{\infty} x_n \delta(t - nT),$$

which is a sampled form of $x(t)$.
Now stick that in the Laplace transform...

So...

$$\begin{aligned} F(s) &= \int_{-\infty}^{\infty} dt \sum_{n=-\infty}^{\infty} x_n \delta(t - nT) \exp(-st), \\ &= \sum_{n=-\infty}^{\infty} x_n \int_{-\infty}^{\infty} dt\, \delta(t - nT) \exp(-st), \\ &= \sum_{n=-\infty}^{\infty} x_n \exp(-snT). \end{aligned}$$

Define

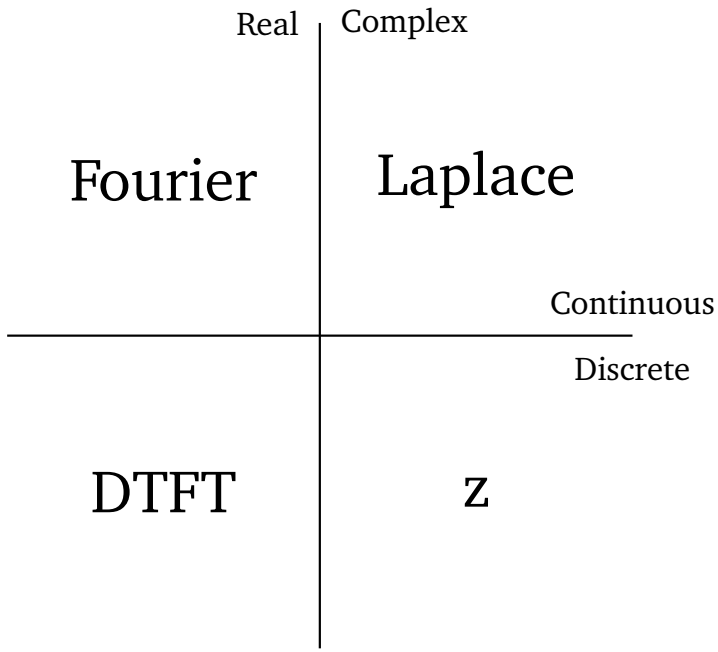
$$z = \exp(sT),$$

and

$$F(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n}.$$

The Laplace transform of a sampled signal is the z-transform

Summary



Homomorphic signal processing

Alan

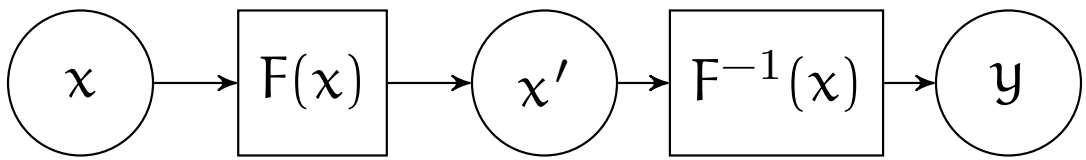
Alan

Alan Oppenheim
1937–

Homomorphic signal processing

The term homomorphic comes from Oppenheim.

- Refers to the idea of transforming a signal such that the components are linearly combined.
- Allows non-linear processing in a linear environment.



i.e., x is a non-linear combination of signals, x' is a linear combination.

Convolutional case

Convolution is a very common situation for homomorphic processing.

$$x = s * h$$

After DFT,

$$X = S \times H.$$

And then after $\log$,

$$\log X = \log S + \log H.$$

Cepstrum 1: The easy way

The Bogert *et al* paper in 1963:

- Give the paper a silly name.
- Make up new words.
- Fourier transform of log power spectrum.
- Since the power spectrum is symmetric, it's the DCT.
- And since

$$\log(x^2) = 2\log(x),$$

you can use magnitude intead. It's just a factor of 2 smaller.

Earthquakes

Say there is a signal $x(t)$ added to a delayed and scaled version of itself

$$y(t) = x(t) + \alpha x(t - \tau).$$

The Fourier transform and power spectrum of that signal are then

$$Y(f) = X(f) + \alpha X(f) e^{j2\pi f\tau}$$
$$Y(f)^2 = X(f)^2(1 + 2\alpha \cos 2\pi f\tau + \alpha^2).$$

Given that $\alpha < 1$, and $\log(1 + x) = \sum_{n=1}^{\infty} \frac{x^n}{n}$ the log power spectrum can be approximated as

$$2 \log Y(f) = 2 \log X(f) + 2\alpha \cos 2\pi f\tau.$$

So, the echo manifests itself as ripple in the log spectrum.

Cepstrum 2: The tricky way

Oppenheim and Schafer in 1968:

- Use difficult words.
- Use lots of equations, contour integrals.
- Make up new functions (e.g., complex log).
- Inverse Fourier transform of (complex) log of complex spectrum.

Rationale

Oppenheim realised that Bogert et al. were doing a homomorphic transformation

- The logarithm (obviously) separates convolutions
- The second DFT separates maximum and minimum phase
  This is not obvious at all!

Say we have a signal that is all poles and zeros

$$F(z) = \frac{\prod_{k=1}^{M_i}(1 - a_k z^{-1}) \prod_{k=1}^{M_o}(1 - b_k z)}{\prod_{k=1}^{N_i}(1 - c_k z^{-1}) \prod_{k=1}^{N_o}(1 - d_k z)} A z^r$$

Rationale

Do a bit of algebra

$$\log F(z) = \sum_{n=-\infty}^{-1} \left[\sum_{k=1}^{M_o} \frac{b_k^{-n}}{n} z^{-n} - \sum_{k=1}^{N_o} \frac{d_k^{-n}}{n} z^{-n} \right] + \log(A) + \sum_{n=1}^{\infty} \left[- \sum_{k=1}^{M_i} \frac{a_k^n}{n} z^{-n} + \sum_{k=1}^{N_i} \frac{c_k^n}{n} z^{-n} \right].$$

That has the form of a z-transform, the inverse of which is

$$x_n = \begin{cases} \sum_{k=1}^{M_o} \frac{b_k^{-n}}{n} - \sum_{k=1}^{N_o} \frac{d_k^{-n}}{n} & n < 0, \\ \log(A) & n = 0, \\ - \sum_{k=1}^{M_i} \frac{a_k^n}{n} + \sum_{k=1}^{N_i} \frac{c_k^n}{n} & n > 0. \end{cases}$$

Complex cepstrum

The complex cepstrum is defined in terms of z-transform

- It's defined as the inverse z-transform of the logarithm of the z-transform

$$c_n = \frac{1}{2\pi j} \oint_C dz \, \log(X(z)) z^{n-1}.$$

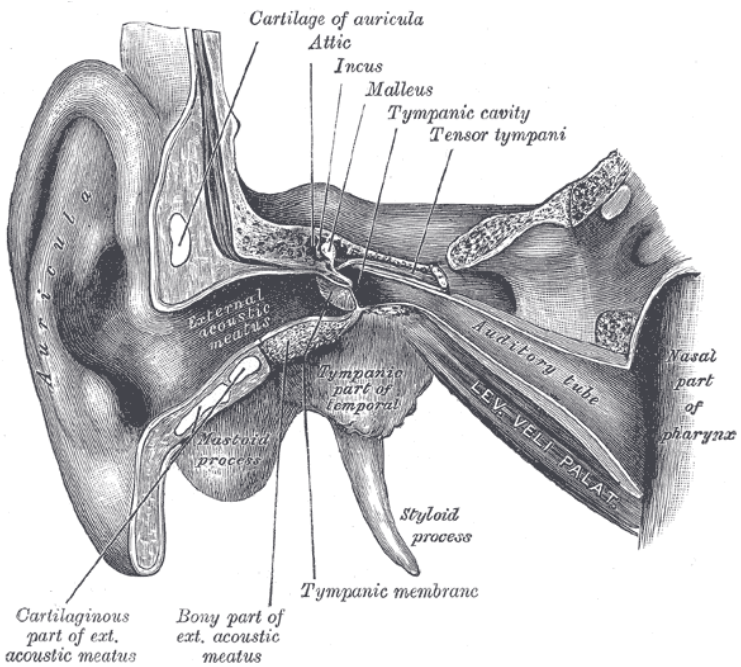
- The unit circle is in the convergence region, so

$$c_n = \underbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega \, \log \overbrace{\left(X(e^{j\omega})\right)}^{\text{Fourier t'form}} e^{j\omega n}}_{\text{Inverse Fourier transform}}$$

i.e., we can get away with the forward and inverse discrete time Fourier transforms

The MFCC approach to ASR

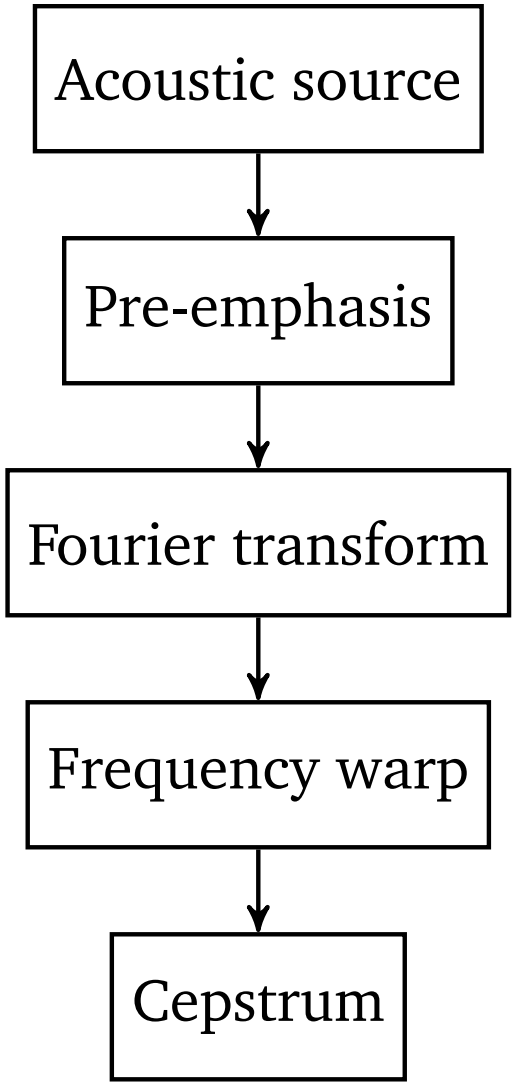
MFCC



MFCC: Mel Frequency Cepstral Coefficients

- MFCCs are a rough perception model.
- They model the path of sound through the human ear.

Signal flow



MFCCs are also a rough homomorphic model

- It's not presented that way mathematically.
- It's not justified that way conceptually.
- See later.

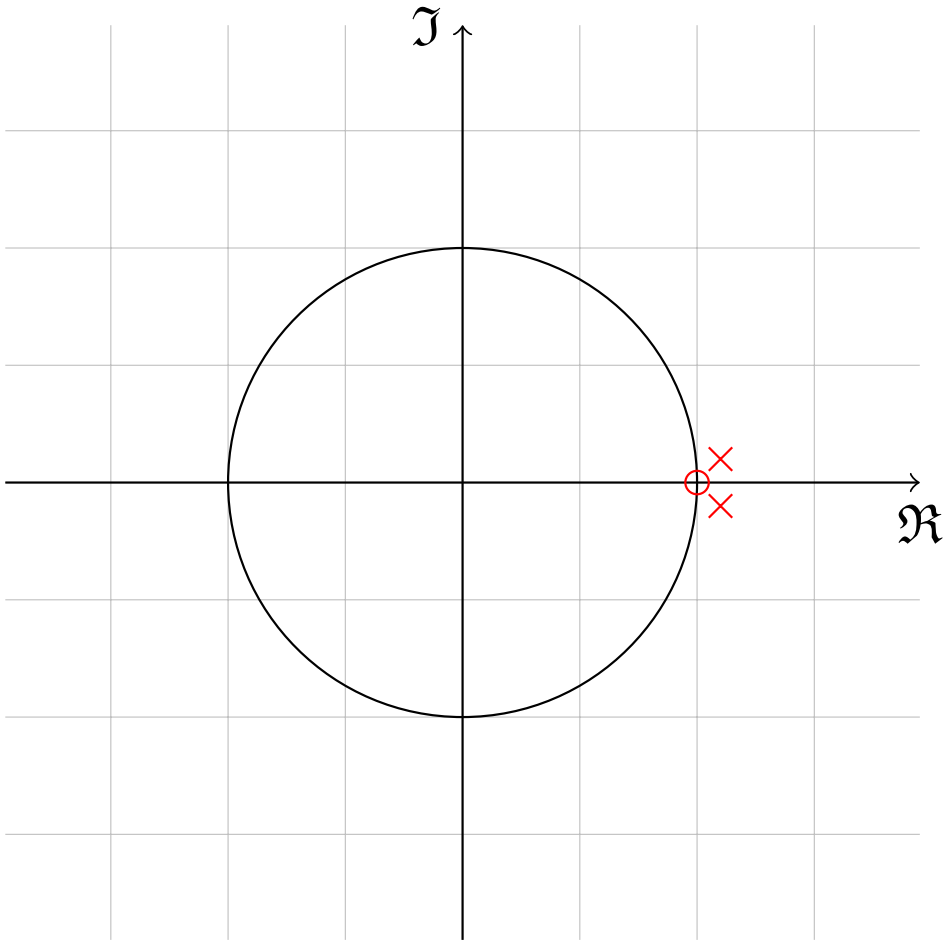
Source

There're few restrictions on the audio source

- Sampling rate 8 kHz or greater.
  16 kHz is typical.
- 8 or 12 bit resolution is OK.
  16 is typical.

Anything higher than 40 dB SNR.

Pre-emphasis



Glottal formant & lip radiation

The official story:

- Mimics human hearing response.
- Rough inverse of human production.

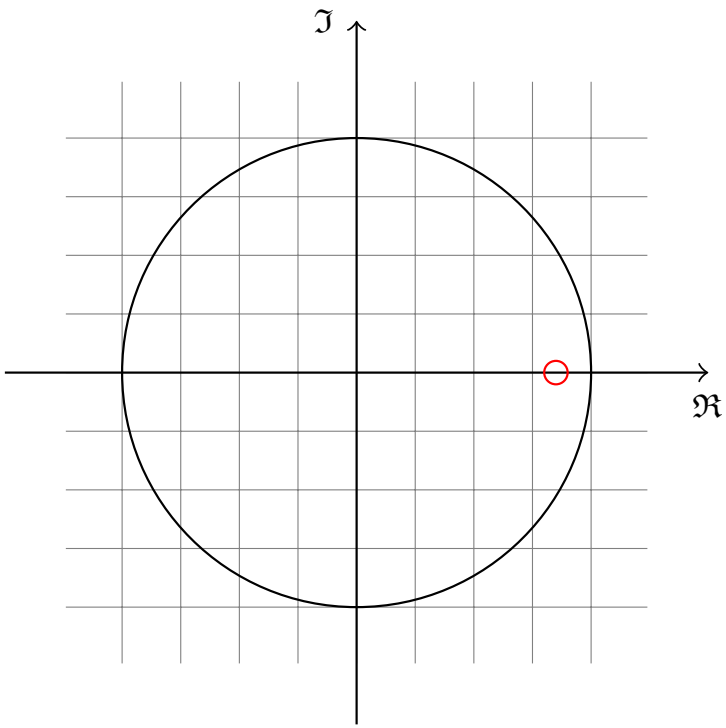
The un-official story:

- For LPC, it's necessary.
- For MFCCs, it doesn't make a blind bit of difference!

But, see later. Especially cepstral normalisation.

Single zero filter

Cancel out the second pole of the Glottal formant



z transform

$$H(z) = 1 - \rho z^{-1}.$$

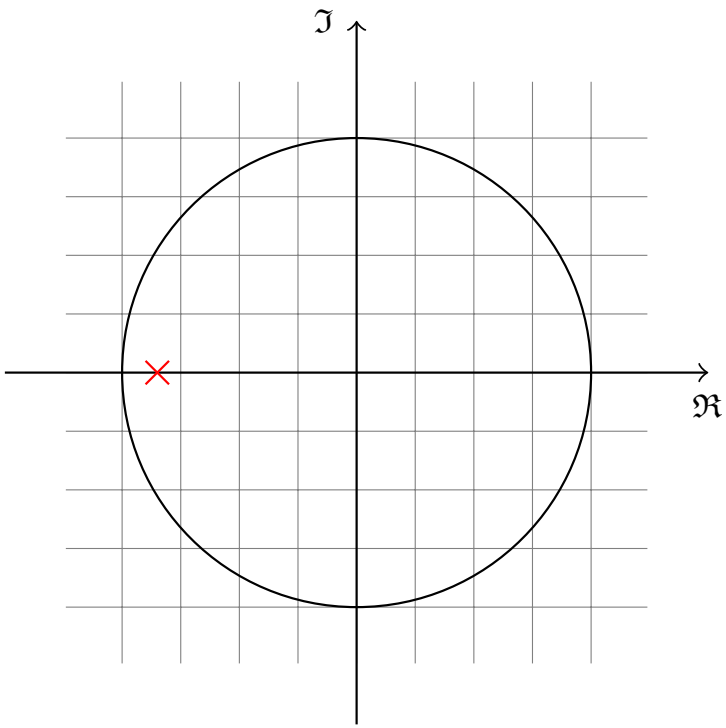
Difference equation

$$y_t = x_t - \rho x_{t-1}.$$

Typically, $\rho = 0.97$.

Single pole filter

Single pole is not so common, but I've seen it.



z transform

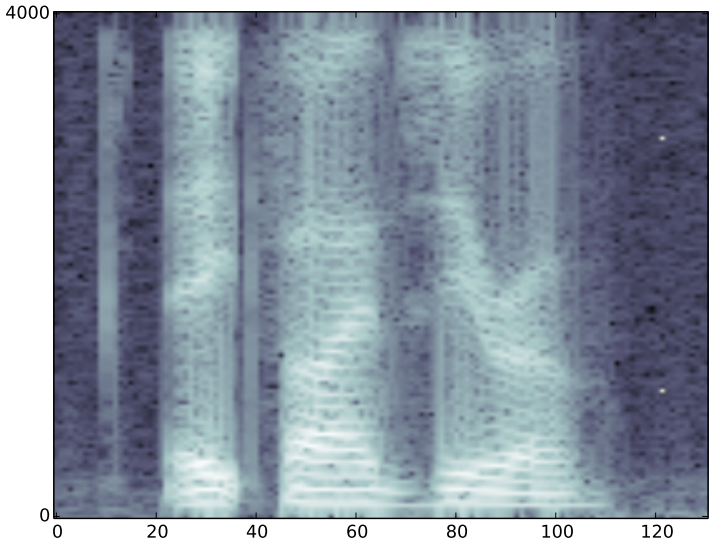
$$H(z) = \frac{1}{1 + \rho z^{-1}}.$$

Difference equation

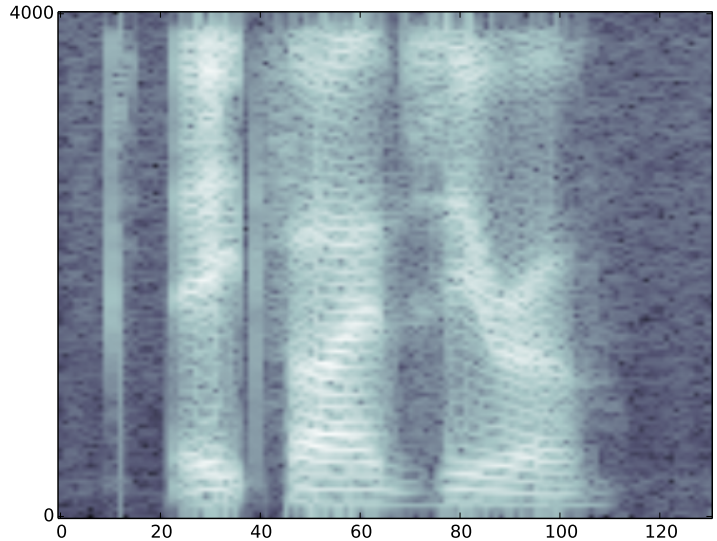
$$y_t = x_t - \rho y_{t-1}.$$

8 kHz spectrogram

No pre-emphasis



Pre-emphasis



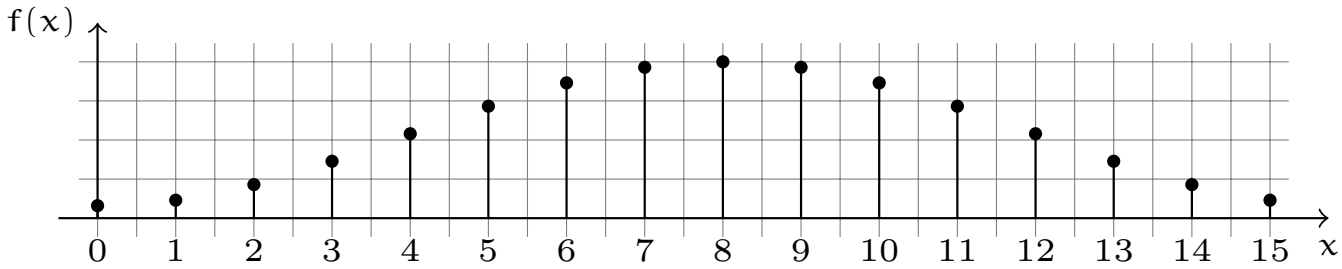
“Eight five zero”

DFT

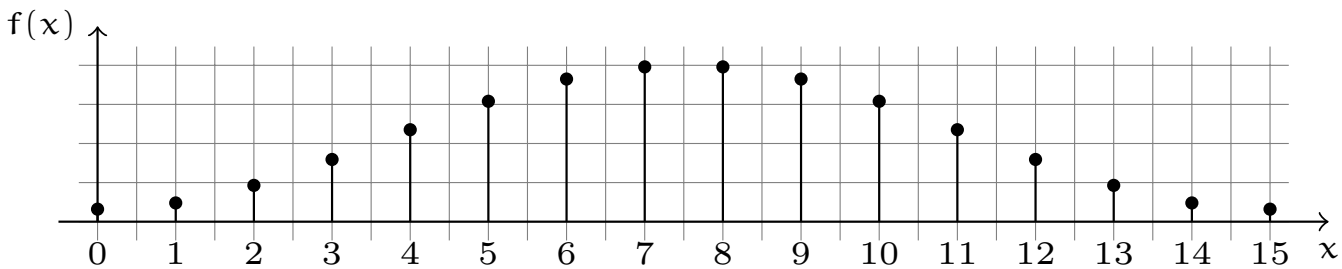
The DFT is just the fastest way to get a frequency representation.
Motivated by the frequency dependent response of the cochlea.
A few notes on windows...
`http://en.wikipedia.org/wiki/Window_function`

Hamming Window

$$f(x) = 0.54 - 0.46 \cos\left(2\pi \frac{x}{N}\right), \quad 0 \leqslant x \leqslant N - 1.$$

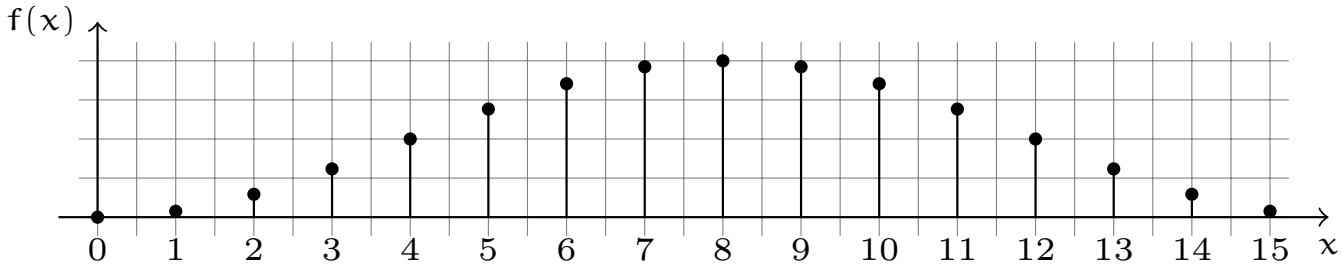


$$f(x) = 0.54 - 0.46 \cos\left(2\pi \frac{x}{N-1}\right), \quad 0 \leqslant x \leqslant N - 1.$$

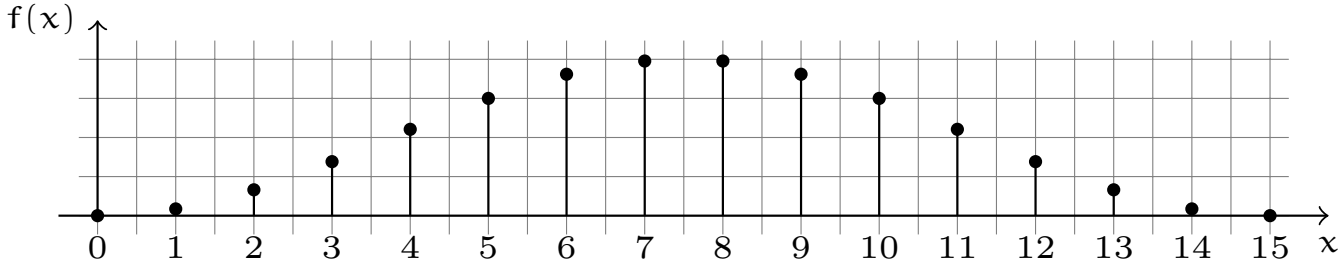


Hann Window

$$f(x) = 0.5 - 0.5\cos\left(2\pi\frac{x}{N}\right), \quad 0 \leqslant x \leqslant N-1.$$



$$f(x) = 0.5 - 0.5\cos\left(2\pi\frac{x}{N-1}\right), \quad 0 \leqslant x \leqslant N-1.$$



Which one?

For ASR

- It doesn't really matter.
- You could even skip the window completely.

For TTS (Text to Speech, speech synthesis) and enhancement

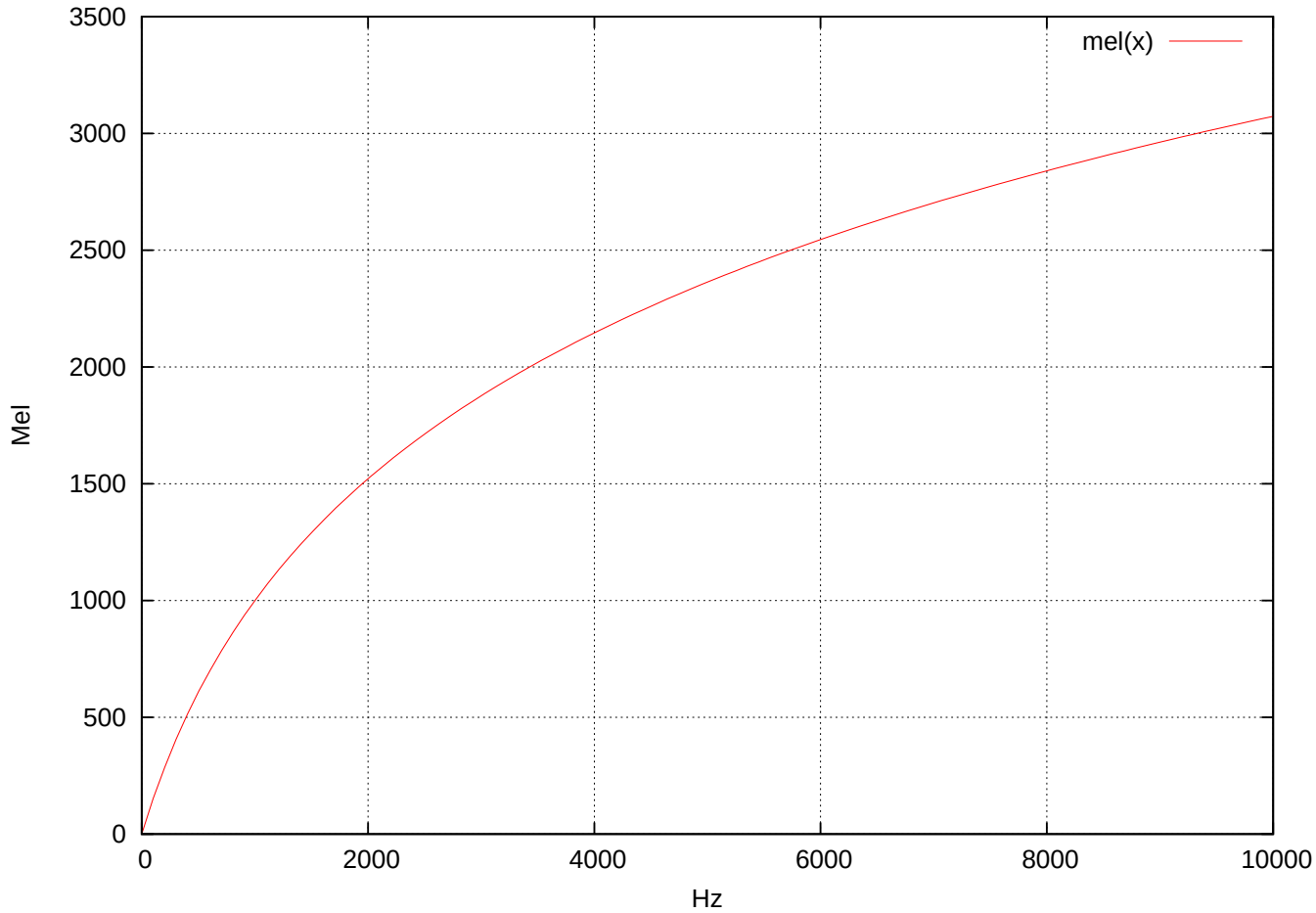
- You should use a window that works with overlap-add.
- Hann is OK, Hamming is not.
- Make sure the total power sums to unity.

The non-linear ear

The ear is not equally sensitive to different frequencies (in Hertz).
Broadly speaking

- There is a lower limit; we can't hear DC.
- There is an upper limit of about 20 kHz (think CD sampling).
- The response is approximately logarithmic.

Mel scale



$$\mathrm{m} = 2595 \log_{10}\left(1 + \frac{\mathrm{h}}{700}\right).$$

Other warps

Mel is not the only scale.

- There is also the Bark scale.
- Bark is actually a bunch of points rather than a scale, but you can interpolate them.

Then there is the bilinear transform

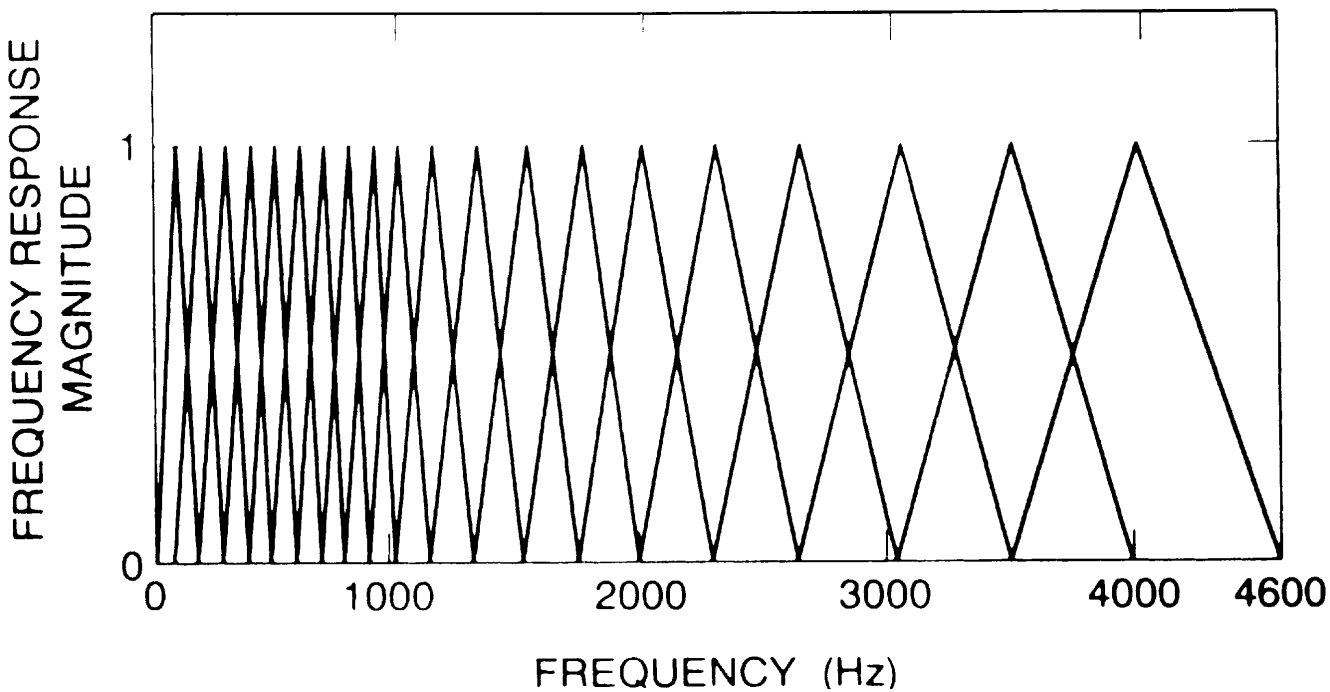
- Which is another lecture in itself.

Filter-bank

The DFT gives us a scale linear in Hertz

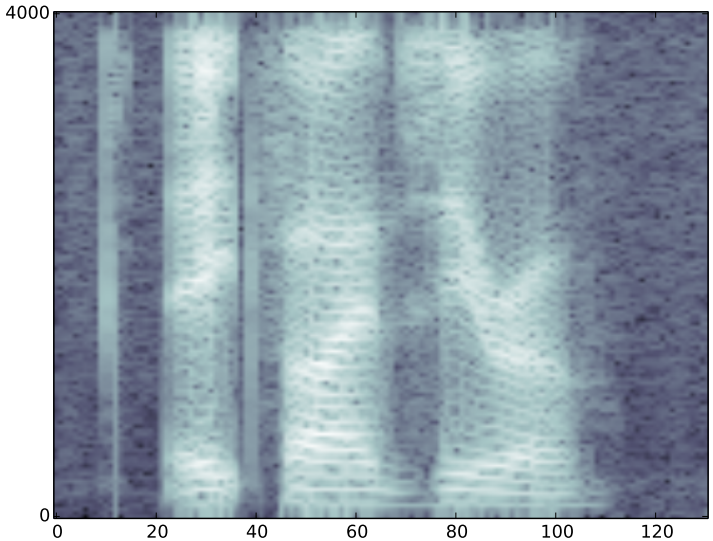
- It's the wrong scaling.
- There are way too many bins.

A very common approach is use a filter-bank.

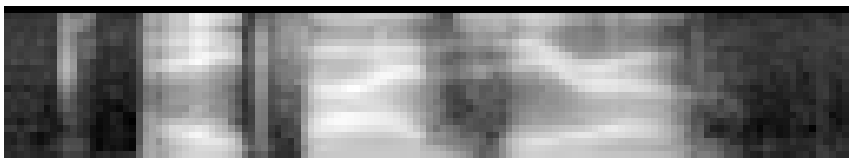


23 bin mel-spectrogram

Spectrogram



Mel-o-gram



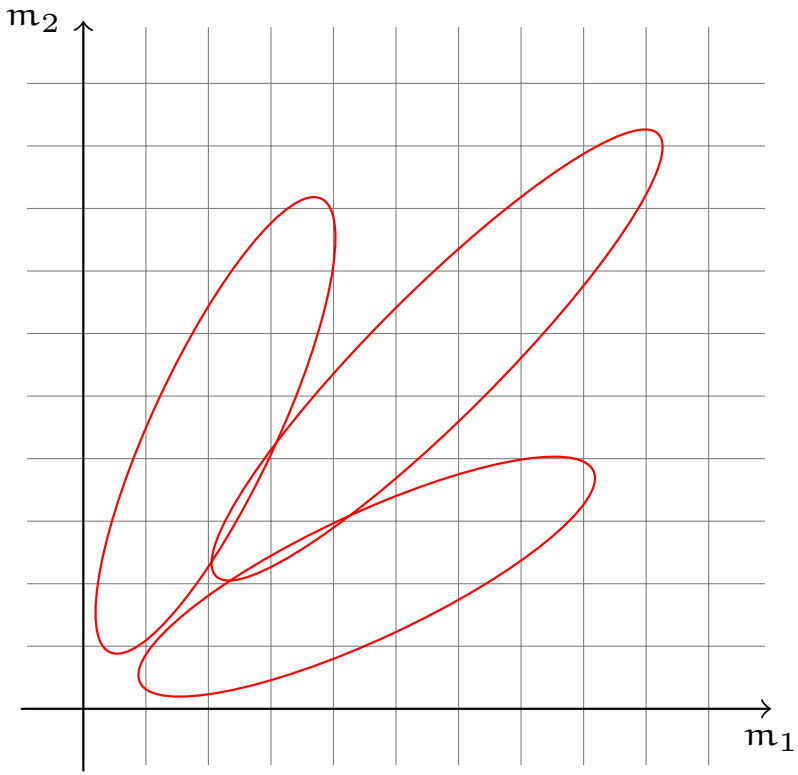
“Eight five zero”

Cepstrum

In ASR, the cepstrum has two distinct components:

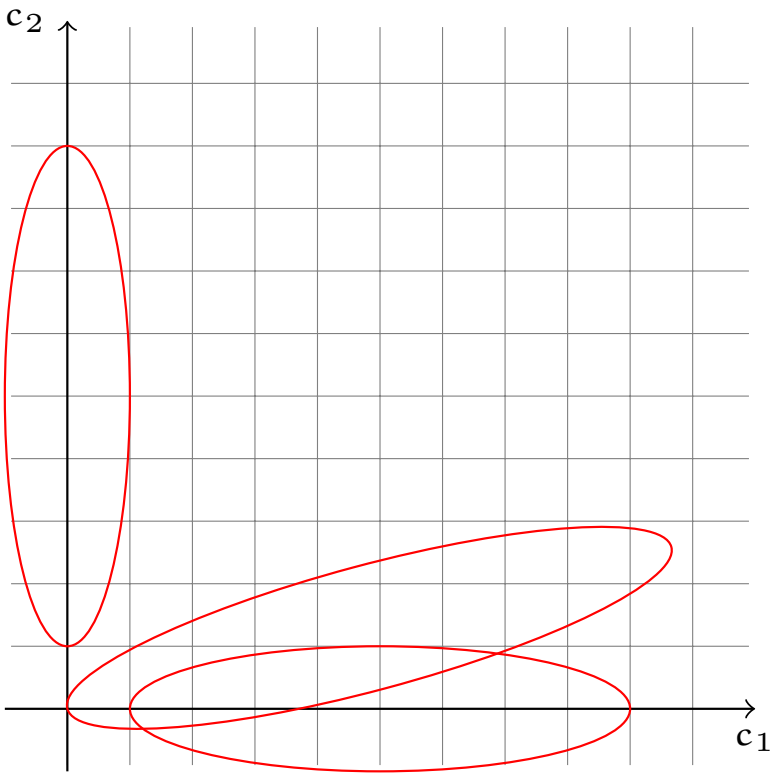
1. The logarithm is to do compression.
   The ear has been shown to have (approximately) logarithmic sensitivity.
2. The DCT is to do decorrelation and dimensionality reduction.

Correlated data



Filterbank output is highly correlated. The distributions do not align with the axes.
Higher level processing requires multivariate full covariance distributions.

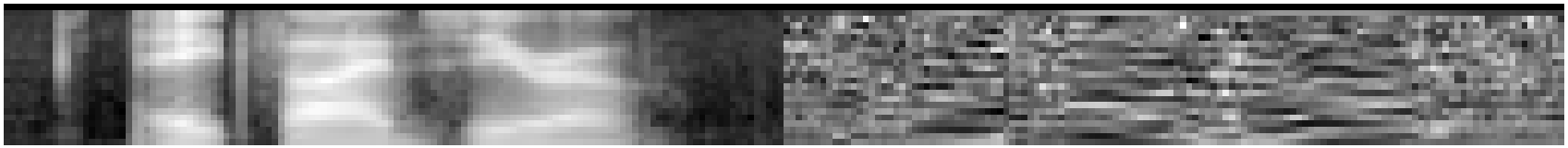
Decorrelated data



Cepstral data is largely uncorrelated. The distributions align with the axes.
Higher level processing can use diagonal covariance.

23 bin cepstrogram

Mel-o-gram Cepstr-o-gram



"Eight five zero"

Note

- This cepstrum is both mean and variance normalised.
- Otherwise, the different bins have way different dynamic ranges.

Cepstrum truncation

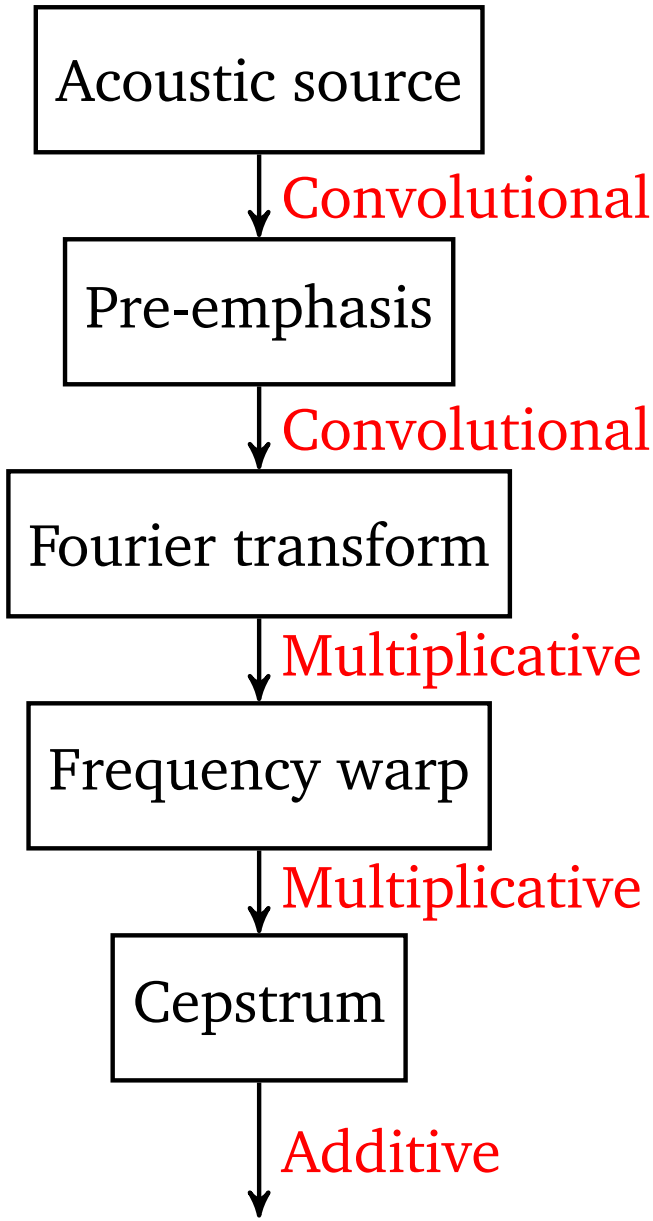
Typical numbers:

- DFT is 256 point and gives 129 bins.
- Filterbank gives 20–30 bins.
- So, the cepstrum is 20–30 bins, but...
  We can get away with the first 12 or so.

Generally

- Low order cepstra have speaker-independent information
  = Good for ASR
- High order cepstra have speaker-dependent information.
  = Good for Synthesis

Convolutional noise



Consider the flow of convolutional noise through the front-end.
It's really quite simple, no maths necessary.
The cepstrum is a very convenient parameterisation in which to do convolutional noise removal.

Cepstral Mean Normalisation

Make two assumptions:

1. The convolutional noise is constant.
   Actually quite a reasonable assumption.
2. The mean of speech cepstra is zero.
   This is more of a leap of faith.

$\Longrightarrow$ The cepstral mean is the convolutional noise.

Removing the cepstral mean is equivalent to removing convolutional noise.

In practice, in the context of ASR, regardless of whether or not the assumptions are valid, CMN works very well, the vast majority of the time!

Static CMN

Say we have cepstral samples $\mathbf{c}_{t-N+1}, \mathbf{c}_{t-N+2}, \ldots, \mathbf{c}_t$,
The ML estimate of the mean, $\hat{\boldsymbol{\mu}}_t$, is

$$\hat{\boldsymbol{\mu}}_t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{c}_{t-N+i} = \bar{\mathbf{c}}_t.$$

Adaptive CMN

Now say we find another sample, c_{t+1}. The new mean is

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_{t+1} &= \frac{1}{N+1} \sum_{i=1}^{N+1} \mathbf{c}_{t-N+i} \\
&= \frac{1}{N+1} \left(\sum_{i=1}^{N} \mathbf{c}_{t-N+i} + \mathbf{c}_{t+1} \right) \\
&= \frac{1}{N+1} \left(N \bar{\mathbf{c}}_t + \mathbf{c}_{t+1} \right) \\
&= \frac{N}{N+1} \hat{\boldsymbol{\mu}}_t + \frac{1}{N+1} \mathbf{c}_{t+1}.
\end{aligned}$$

Adaptive CMN 2

Transforming $N + 1 \to N$ and $t + 1 \to t$, we have two cases

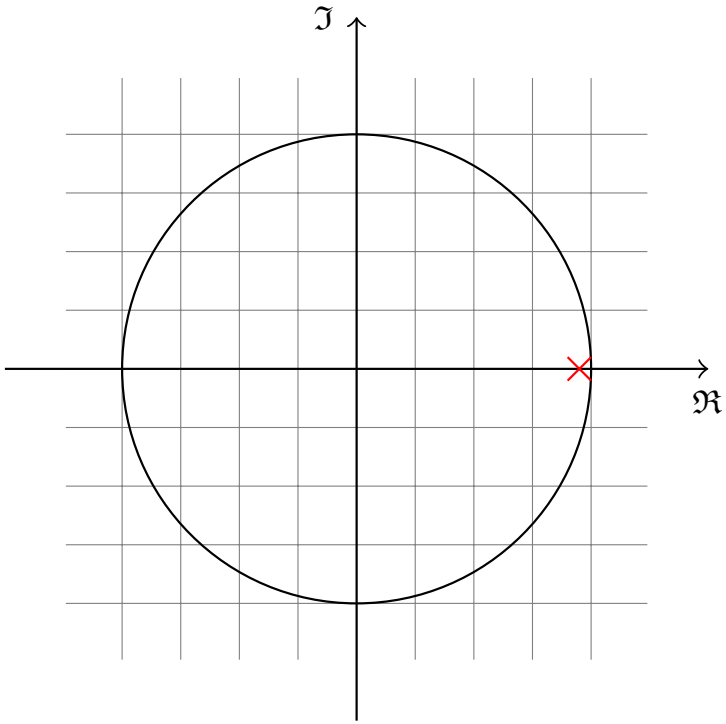
1. If we update N with each new observation

$$\hat{\boldsymbol{\mu}}_t = \frac{N-1}{N}\hat{\boldsymbol{\mu}}_{t-1} + \frac{1}{N}\mathbf{c}_t.$$

2. If we fix N to some value

$$\hat{\boldsymbol{\mu}}_t = \rho\hat{\boldsymbol{\mu}}_{t-1} + (1-\rho)\mathbf{c}_t, \quad \rho = \frac{N-1}{N}.$$

Adaptive CMN as a filter



$$H(z) = \frac{1-\rho}{1-\rho z^{-1}}.$$

Typically

- 1 second time constant.
- 100 frames per second.
- $\rho = 99/100 = 0.99$.

Summary

The MFCC representation is an analogue of the human ear.

- It is non-linear in frequency.
- It has a logarithmic sensitivity.

It is also a homomorphic system for convolutional noise

- The logarithm separates out the distortion.

Notice that the homomorphic representation arises also from the perception (human ear) point of view.
It's not completely accidental, but rarely presented like this.