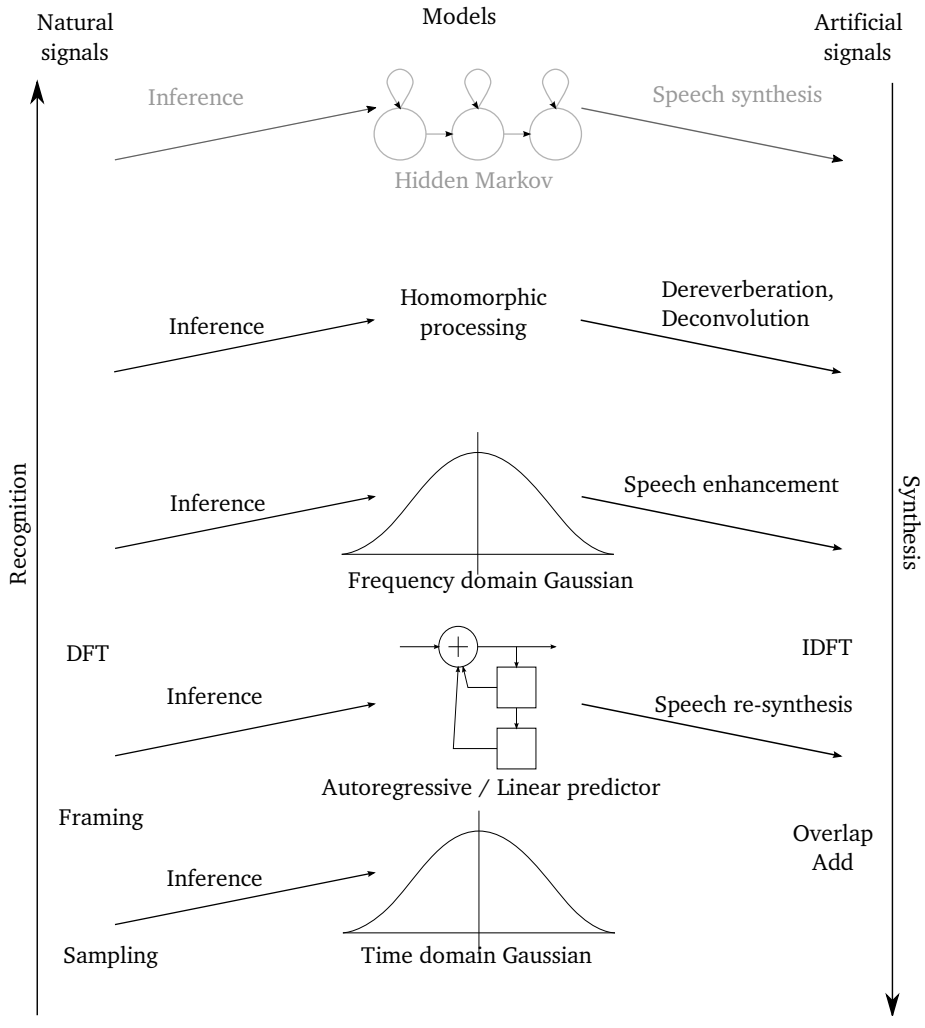
Linear Prediction

A map of speech signal processing

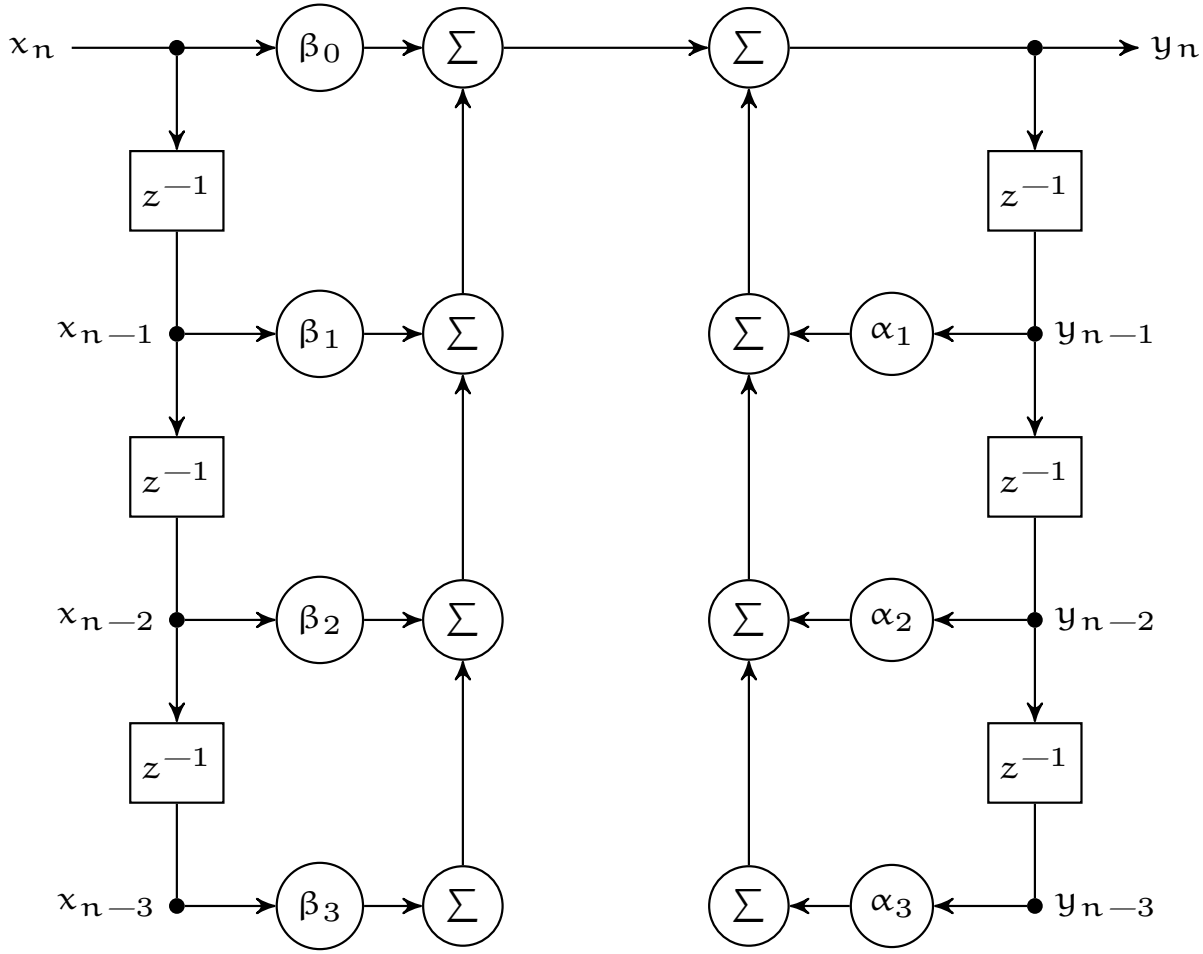


Linear Prediction

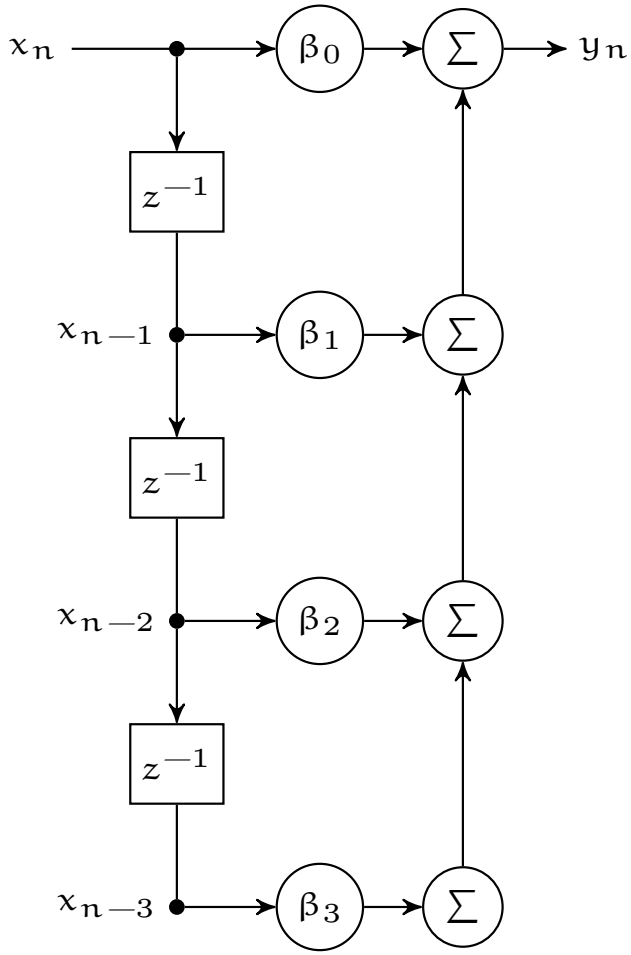
Linear Prediction is:

- A production model.
- A tractable filter.
- A spectral smoothing technique.

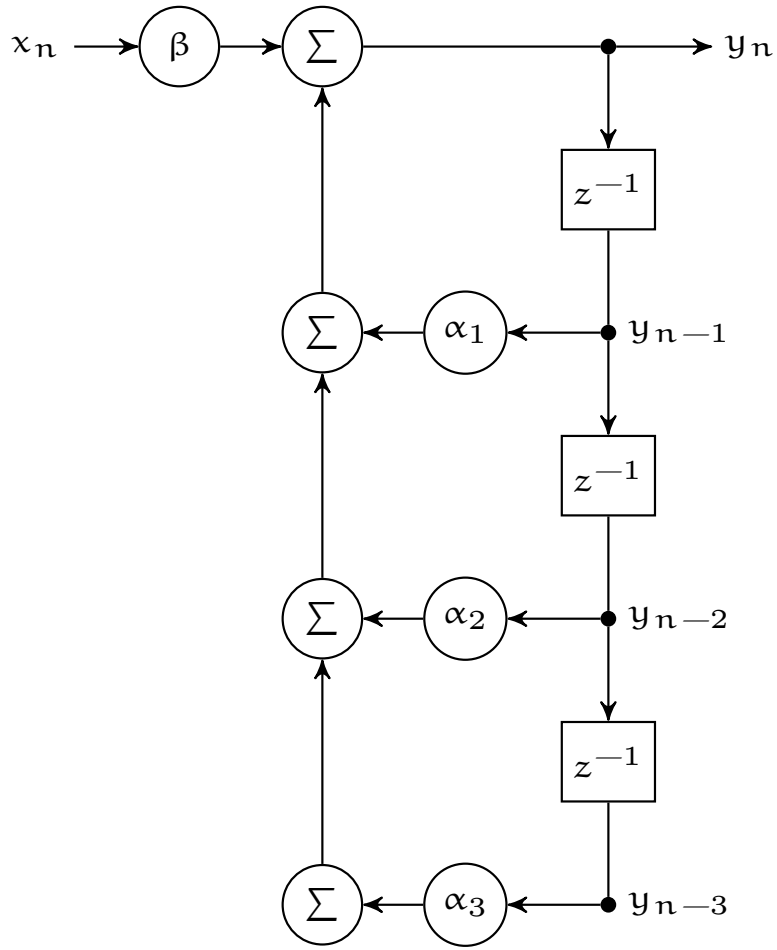
General digital filter



All zero filter



AR Process


x_n
β
$\sum$
y_n
z^{-1}
$\sum$
α_1
y_{n-1}
z^{-1}
$\sum$
α_2
y_{n-2}
z^{-1}
$\sum$
α_3
y_{n-3}

Filter estimation

We can't in general estimate both poles and zeros.

- There are ways, but it's non-linear.

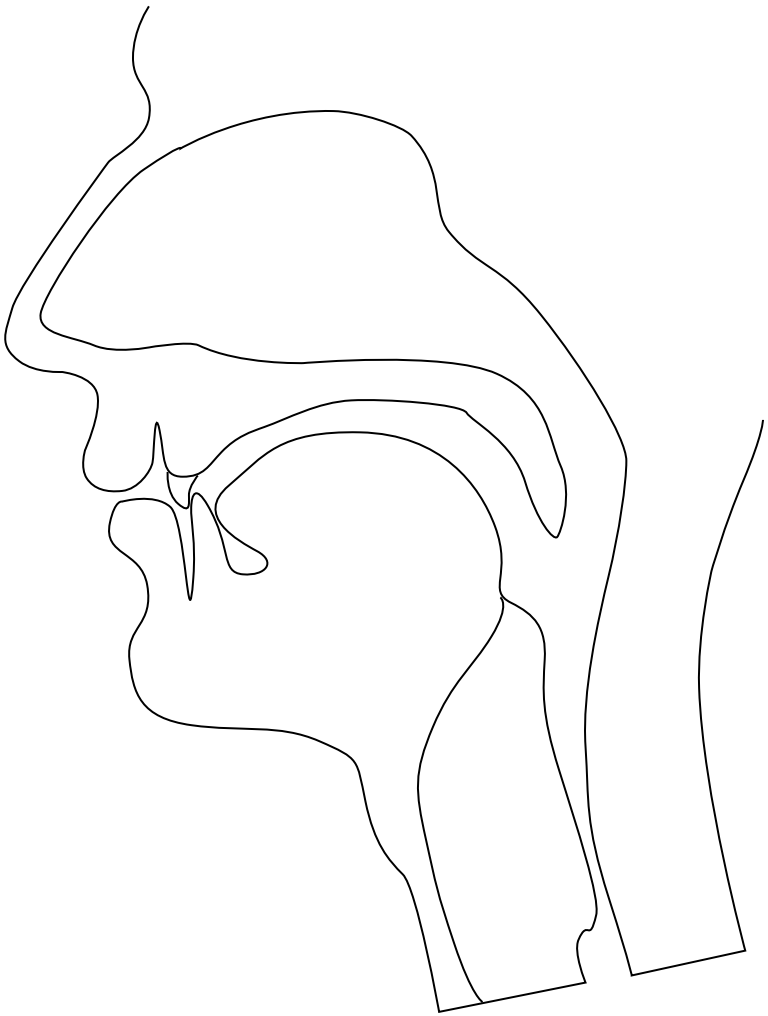
One solution is to choose either all pole or all zero.

- The vocal tract certainly contains poles.
- It also contains zeros, but:
  - A zero can be estimated by some poles.
  - The ear is more sensitive to poles than zeros.

Synonyms:

- All pole.
- Linear Prediction (LP).
- Linear Predictive coding (LPC).
- Auto-Regressive modelling (AR).

Source model



LPC is a model of the vocal tract

- It is a generative model.

The speech is seen as the output of the system.

- The input is the excitation
- An impulse train for voiced sounds.
- white noise for unvoiced sounds.

Choose an analysis that doesn't distinguish voiced and unvoiced!

Definition

Say we have:

- N acoustic observations, $\mathbf{y} = (y_{t-N+1}, y_{t-N+2}, \ldots, y_t)^T$. They tend to be time indexed, in which case the most recent is time t.
- A vector of P linear prediction coefficients, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_P)^T$.

Define a model where the observation is a function of a current input, or excitation, and previous observations (outputs):

$$y_n = \beta x_n + \sum_{p=1}^{P} \alpha_p y_{n-p}.$$

z-transform

In the z domain, this becomes

$$y(z) = \beta x(z) + \sum_{p=1}^{P} \alpha_p y(z) z^{-p}$$

so,

$$\frac{y(z)}{x(z)} = H(z) = \frac{\beta}{1 - \sum_{p=1}^{P} \alpha_p z^{-p}}.$$

So, we have an all pole system.[1]
If you stick white noise in, what comes out has resonances.

[1] And I could have done this backwards: start with the all pole and derive the difference equation.

Prediction

If the excitation, x_n, is just a white Gaussian process, we can re-interpret the difference equation:

$$y_n = \underbrace{\sum_{p=1}^{P} \alpha_p y_{n-p}}_{\text{Prediction}} + \underbrace{\beta x_n}_{\text{Error}}.$$

This is a predictor. It's predicting the current output given the previous outputs, with some error.

Matrix notation

Ideally, $N \gg P$, so there are lots of samples. It's more easily expressed in matrix form:

$$\underbrace{\begin{pmatrix} y_{t-N+P+1} \\ y_{t-N+P+2} \\ \vdots \\ y_t \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} y_{t-N+1} & y_{t-N+2} & \cdots & y_{t-N+P} \\ y_{t-N+2} & y_{t-N+3} & \cdots & y_{t-N+P+1} \\ \vdots & \vdots & & \vdots \\ y_{t-P} & y_{t-P+1} & \cdots & y_{t-1} \end{pmatrix}}_{\mathbf{Y}\text{ (Tall and thin, overdetermined)}} \underbrace{\begin{pmatrix} \alpha_P \\ \alpha_{P-1} \\ \vdots \\ \alpha_1 \end{pmatrix}}_{\boldsymbol{\alpha}} + \beta \mathbf{x}$$

MMSE

The usual (classical) approach is Minimum Mean Squared Error. It goes like this:

1. Define an error function being the difference between prediction and reality

$$e_n = \underbrace{y_n}_{\text{Reality}} - \underbrace{\sum_{p=1}^{P} \alpha_p y_{n-p}}_{\text{Prediction}} .$$

2. Minimise $\mathbb{E}\left(|e_n|^2\right)$.

Notice that the error here is the same as the excitation.

MMSE Solution

- Blah blah Wiener-Hopf equations.
- Blah blah Principle of orthogonality.
- Blah blah Second order dependence of cost function.

Maximum Likelihood

Say the excitation is a Gaussian with zero mean and unit variance:

$$p(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^{N-P}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x}\right).$$

Make a change of variable

$$\mathbf{x} \to \mathbf{y}$$

Basically you can just substitute, but persuade yourself that the Jacobian is $1/\beta$ for each of the $N - P$ equations.

Maximum Likelihood solution

After substitution, we get

$$p\left(\mathbf{y} \mid \boldsymbol{\alpha}\right) = \frac{1}{\sqrt{2\pi\beta^2}^{N-P}} \exp\left(-\frac{1}{2\beta^2}(\mathbf{y} - \mathbf{Y}\boldsymbol{\alpha})^T(\mathbf{y} - \mathbf{Y}\boldsymbol{\alpha})\right).$$

Finally, differentiate w.r.t. $\boldsymbol{\alpha}$ and equate to zero[2]:

$$\hat{\boldsymbol{\alpha}} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{y}.$$

This is not specific to LPC; it's a standard statistical result.

[2] You can sort of see the result is going to be a rearrangement of $\mathbf{y} = \mathbf{Y}\alpha$

ML solution for LPC

This bit is specific to LPC:

- Look carefully at $\mathbf{Y}^\top \mathbf{Y}$
  It is basically the autocorrelation, with a few edge effects.
- Look carefully at $\mathbf{Y}^\top \mathbf{y}$
  It is also basically the autocorrelation, with a few edge effects.

$$\hat{\boldsymbol{\alpha}} = \begin{pmatrix} r_0 & r_1 & \dots & r_{P-1} \\ r_1 & r_0 & \dots & r_{P-2} \\ \vdots & \vdots & & \vdots \\ r_{P-1} & r_{P-2} & \dots & r_0 \end{pmatrix}^{-1} \begin{pmatrix} r_P \\ r_{P-1} \\ \vdots \\ r_1 \end{pmatrix}$$

Gain

The gain is just the variance:

$$
\begin{aligned}
\hat{\beta}^2 &= \frac{1}{N-P}(\mathbf{y}-\mathbf{Y}\boldsymbol{\alpha})^T(\mathbf{y}-\mathbf{Y}\boldsymbol{\alpha}) \\
&= \frac{1}{N-P}(\mathbf{y}-\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{y})^T(\mathbf{y}-\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{y}) \\
&= \dots \\
&= \frac{1}{N-P}\left(\mathbf{y}^T\mathbf{y}-\mathbf{y}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{y}\right) \\
&= \frac{1}{N-P}\left(\mathbf{y}^T\mathbf{y}-\boldsymbol{\alpha}^T\mathbf{Y}^T\mathbf{y}\right) \\
&= r_0 - \boldsymbol{\alpha}^T\mathbf{r}_1
\end{aligned}
$$

where $\mathbf{r}_1$ denotes $(r_1, r_2, \dots, r_P)^T$.

Practicalities

You can blindly stick this into Matlab and it will work. But don't tell a future employer that.

- The matrix to invert is symmetric and Toeplitz. It's structured. This is what the Levinson-Durbin[3] recursion is for.
- The fastest way to get the autocorrelation is via the DFT, periodogram and DCT.
- Iff you have a good initial guess for the parameters, Newton's method (or another iterative method) can be faster.

That DFT will be important when using frequency warping.

[3] I won't teach it; look it up.

The LP spectrum

Now we have parameters, go back to the z-transform

$$H(z) = \frac{\beta}{1 - \sum_{p=1}^{P} \alpha_p z^{-p}}.$$

Writing

$$z = e^{j\omega},$$

and squaring, we get

$$|H(\omega)|^2 = \frac{\beta^2}{|1 - \sum_{p=1}^{P} \alpha_p e^{-j\omega p}|^2}.$$

This is the LP power spectrum.

LP cepstrum

You could

1. Sample the LP spectrum.
2. Calculate logarithms and DCT.

But, there is a trick. Most texts say “It can be shown that”, and then gloss over it. I’ll show you.

Atal's solution

This is from Atal, but I guess the technique is somewhat older. The key is equate the z transforms of

- the log magnitude spectrum.
- the cepstrum.

$$\frac{d}{dz^{-1}} \log \left[\frac{\beta}{1 - \sum_{p=1}^{P} \alpha_p z^{-p}} \right] = \frac{d}{dz^{-1}} \left[\sum_{n=1}^{\infty} c_n z^{-n} \right]$$

Differentiating gets rid of the logarithm.

...

Differentiate and re-arrange.

$$\sum_{p=1}^{P} p\alpha_p z^{-p+1} = \left(1 - \sum_{p=1}^{P} \alpha_p z^{-p}\right) \sum_{n=1}^{\infty} n c_n z^{-n+1}.$$

Equate terms in z^{-1}

$$c_n = \begin{cases} \alpha_1 & n = 1, \\ \sum_{p=1}^{n-1} \left(1 - \frac{p}{n}\right) \alpha_p c_{n-p} + \alpha_n & 1 < n \leqslant P, \\ \sum_{p=1}^{n-1} \left(1 - \frac{p}{n}\right) \alpha_p c_{n-p} & n > P. \end{cases}$$

This works because the z-transform is a Laurent series: It's unique

Notation

Notation does tend to vary with this stuff, so beware.

- The notation for the gain often changes; G is common.
- Sometimes there is a α_0, which is then set to 1.
- Often, people use a parameter equivalent to $-\alpha_n$.

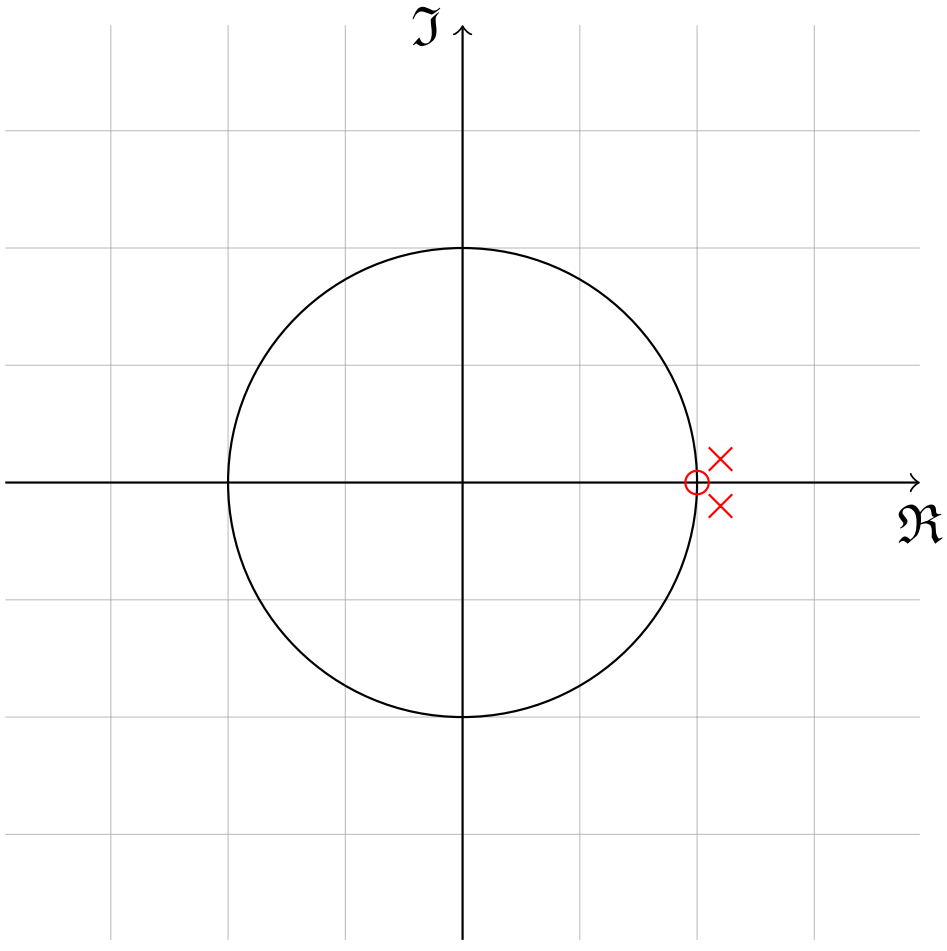
$$
\begin{aligned}
a_0 &= 1, \\
a_1 &= -\alpha_1, \\
&\vdots \\
a_P &= -\alpha_P.
\end{aligned}
$$

i.e., something closer to this is common:

$$H(z) = \frac{G}{\sum_{p=0}^{P} a_p z^{-p}}.$$

I'm using Greek for parameters.

Pre-emphasis



Glottal formant & lip radiation

LP assumes excitation is Gaussian (in time domain).

- Implies white Gaussian in the frequency domain.
- Not true of the human vocal tract.

Before doing LP, use pre-emphasis

- Exact shape of filter is not so important.
- Makes the signal more consistent with white excitation.

Perceptual Linear Prediction

Perceptual Linear Prediction (PLP)

PLP in this lecture has two angles

Specifically a particular combination of techniques, based on linear prediction, developed by Hynek Hermansky.

Generally a way of using perceptual or "bio-inspired" phenomina in speech processing.

Background

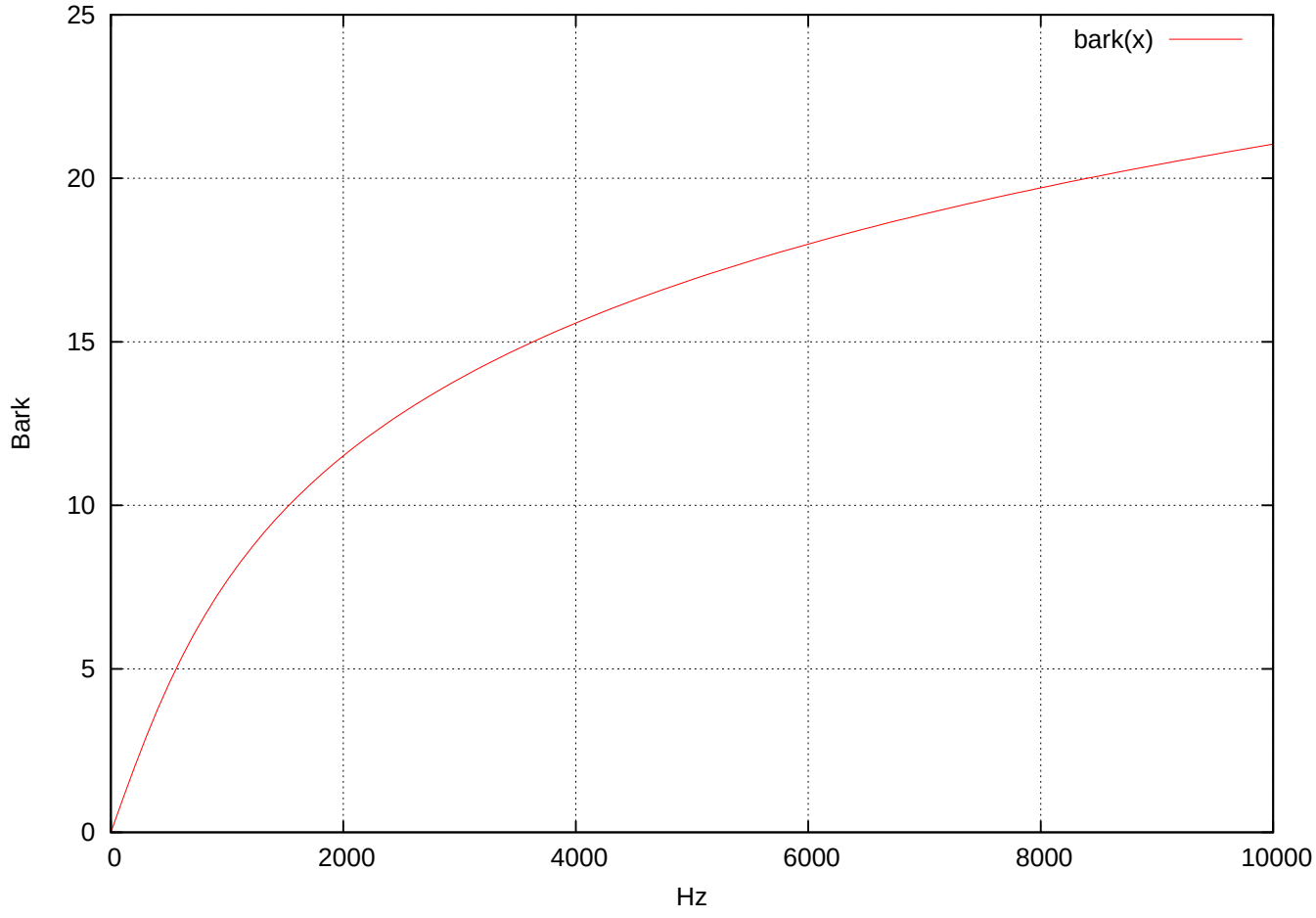
Speech has:

- a production mechanism (vocal tract).
- a perception mechanism (ear).

The two have undoubtedly evolved together to be mutually optimal.

> *Using knowledge of human auditory processing in computational speech processing should improve performance.*

Bark scale



$$\mathrm{b} = 6 \log \left(\frac{\mathrm{h}}{600} + \sqrt{\left(\frac{\mathrm{h}}{600} \right)^2 + 1} \right) .$$

Note on Bark scale

Bark is actually a bunch of bandwidths, not really a scale.
There are several approximations to the sequence of points.
Compare Hynek's with

$$\sinh^{-1} x = \log\left(x + \sqrt{x^2 + 1}\right).$$

The scale actually has 24 bands. Critical bands are 1 Bark wide.
1 bark $\approx$ 100 mels.
Named after Barkhausen.

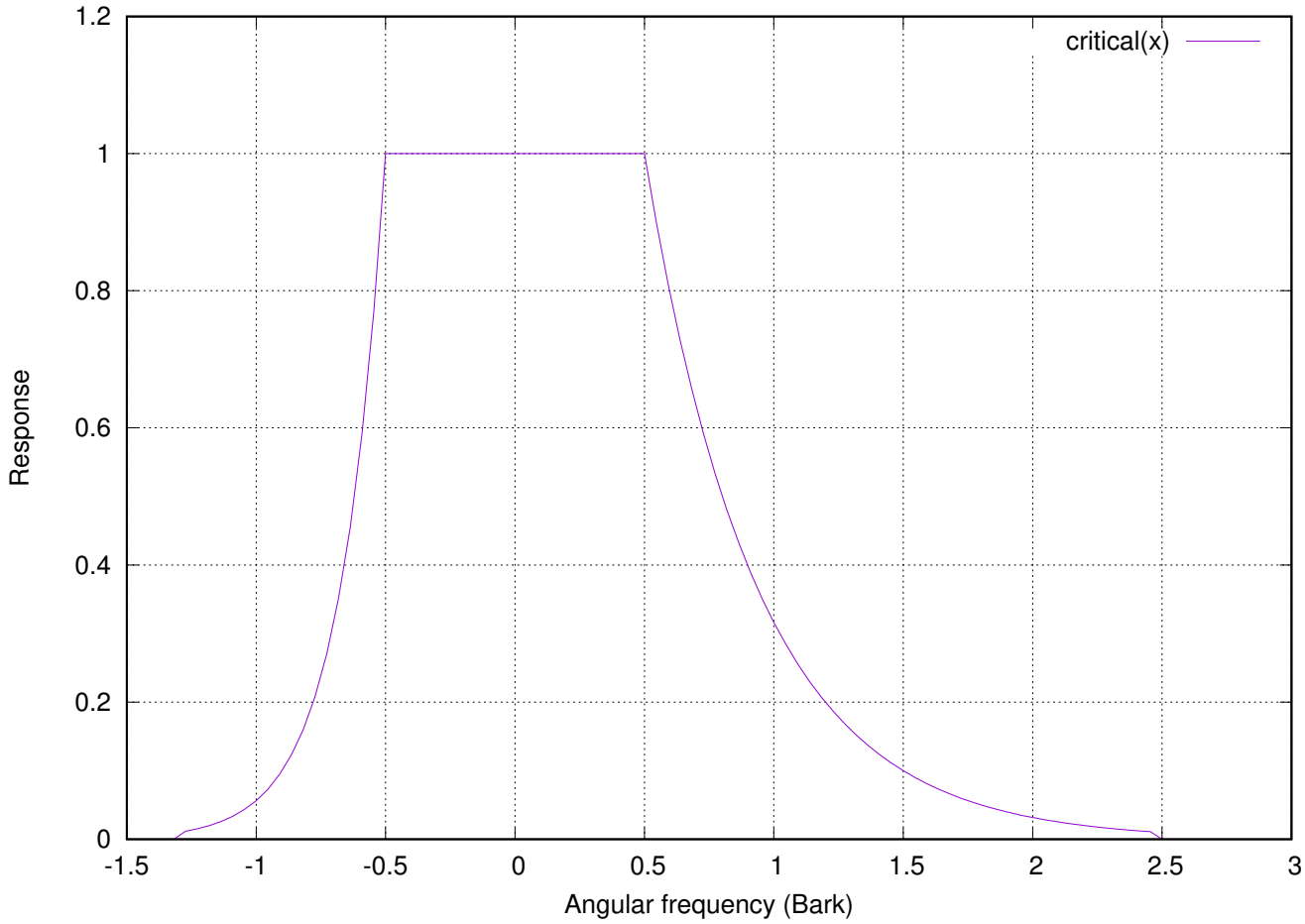
Critical bands

Critical bands are defined by sinusoids in noise:

1. Generate bandlimited white noise.
2. Generate a sinusoid with the same center frequency.
3. Increase the sinusoid amplitude until you hear it.
4. Repeat from 1 with wider bandwidth.

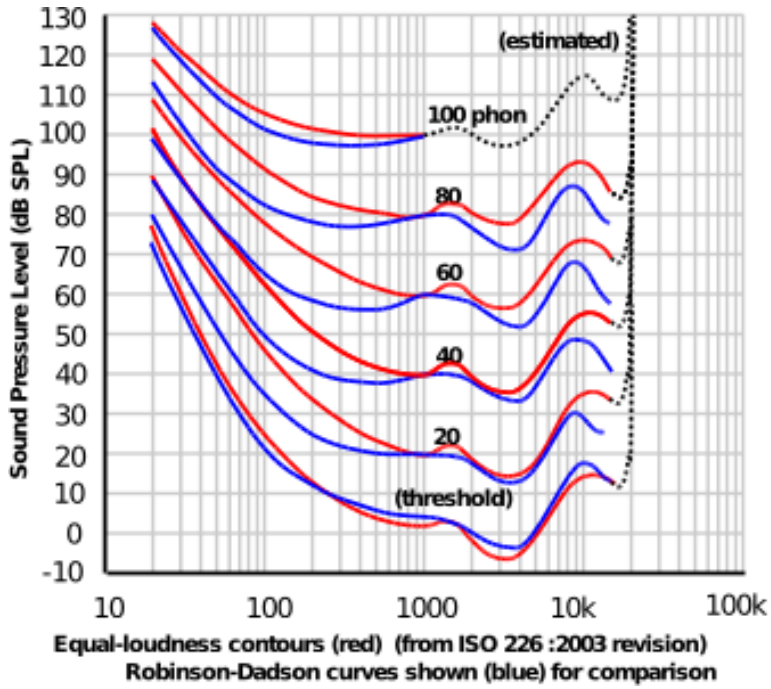
After a while, you don't need to increase the sinusoid amplitude as the bandwidth increases. This is the critical bandwidth.

The PLP critical band filter



This gets time-reversed when applied in the convolution.
Notice that the flat part is about 1 bark wide. That is significant.

ISO equal loudness



- Play someone a tone at 1000 Hz followed by one at the measurement frequency.
- Get them to adjust the reference to be the same loudness as the test.

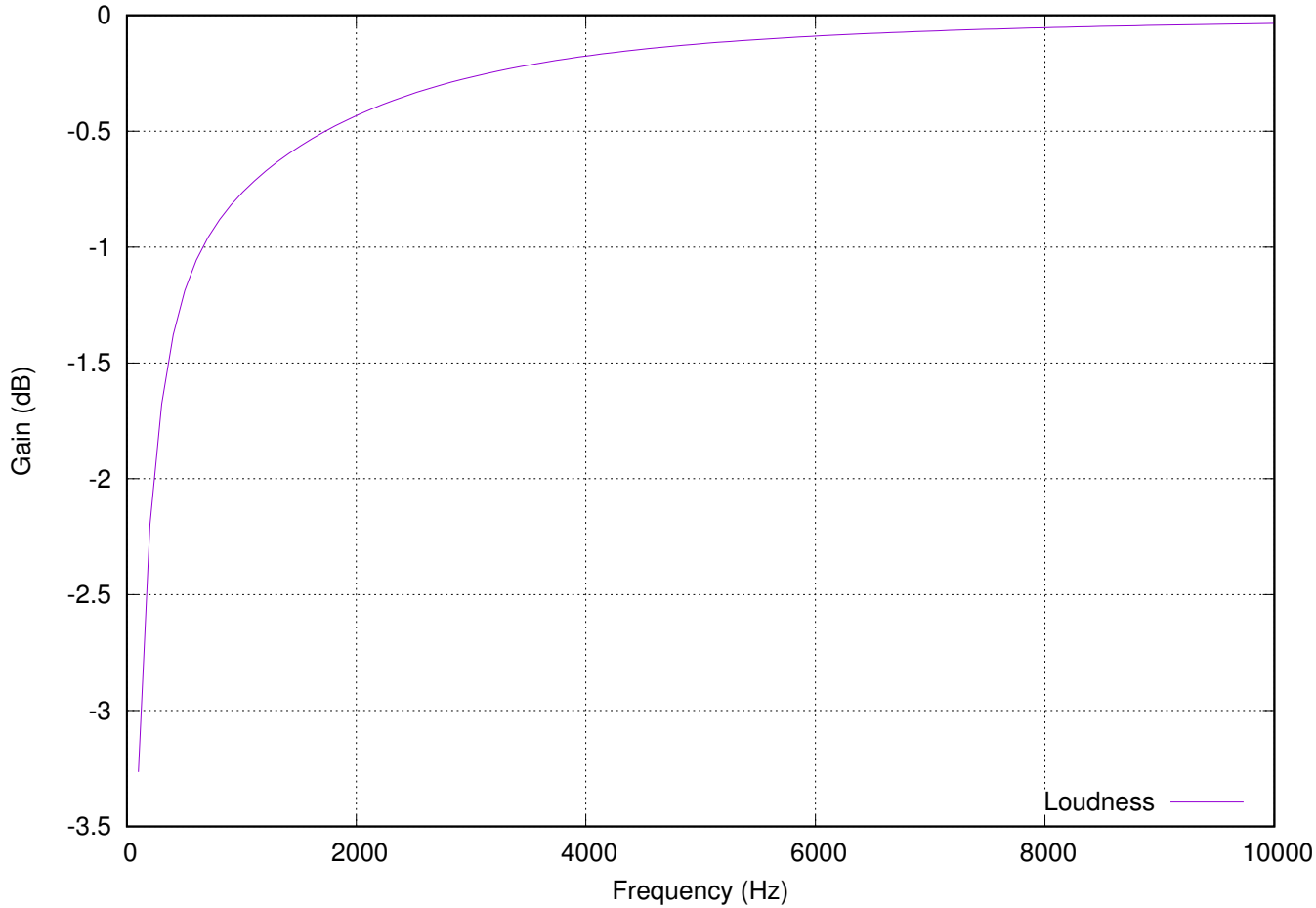
Hermansky's equal loudness

$$E(\omega) = \frac{(\omega^2 + 56.8 \times 10^6)\omega^4}{(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)}$$

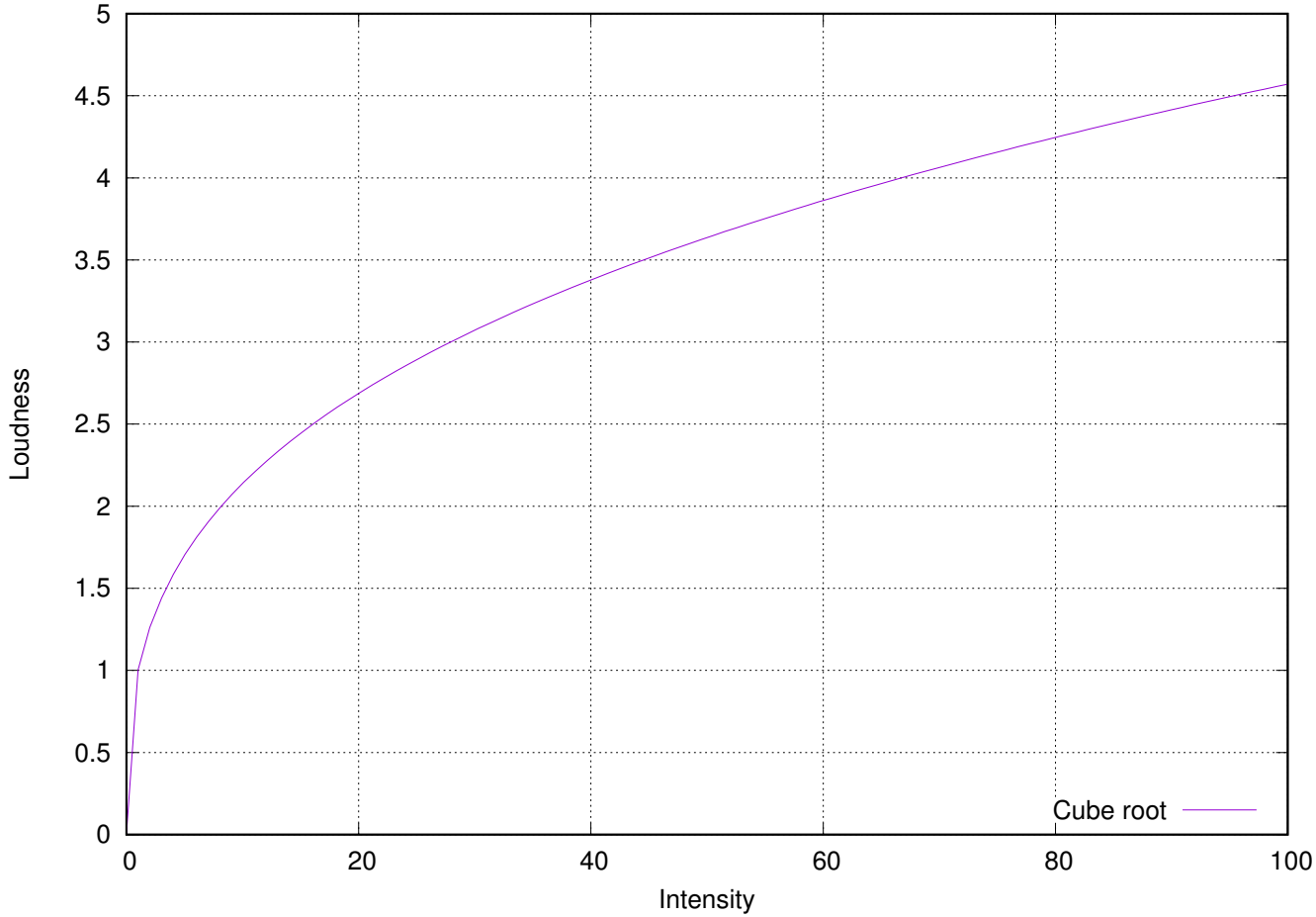
or even

$$E(\omega) = \frac{(\omega^2 + 56.8 \times 10^6)\omega^4}{(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)(\omega^6 + 9.58 \times 10^{26})}$$

Which is:


0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
Gain (dB)
0
2000
4000
6000
8000
10000
Frequency (Hz)
Loudness

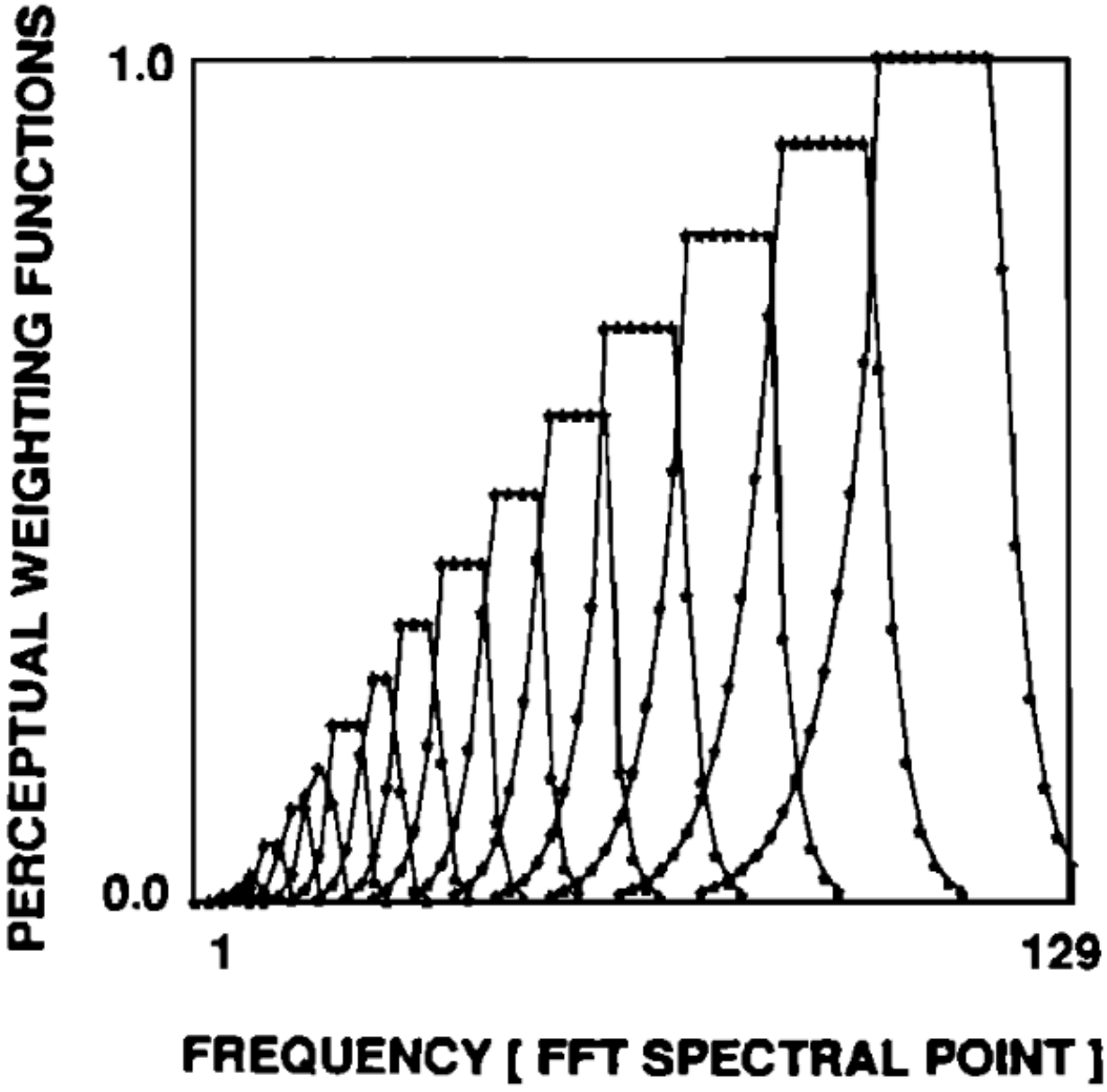
Amplitude compression



An approximation to the power law of hearing.

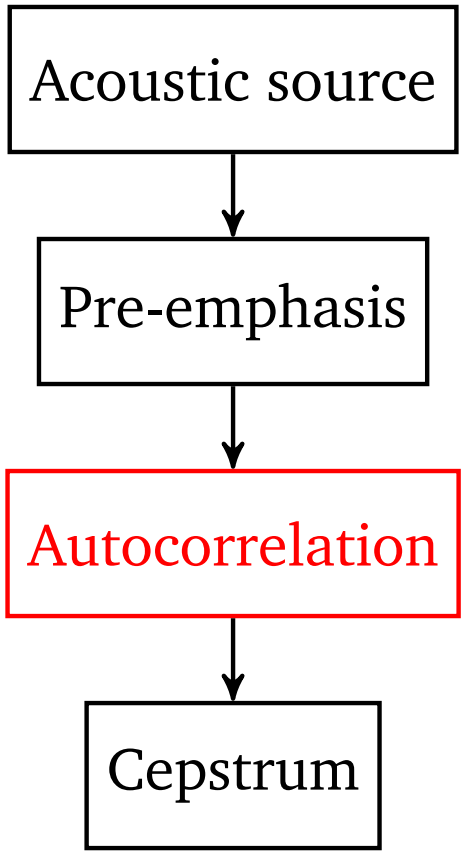
> *Perceived loudness is proportional to the cube root of intensity.*

All taken together...

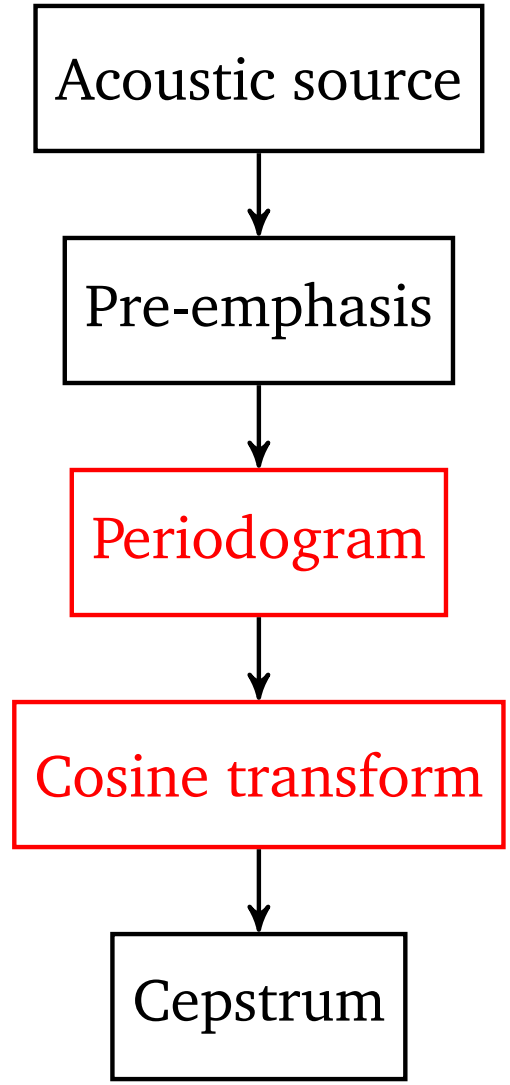


LPC uses DFT/DCT for the autocorrelation

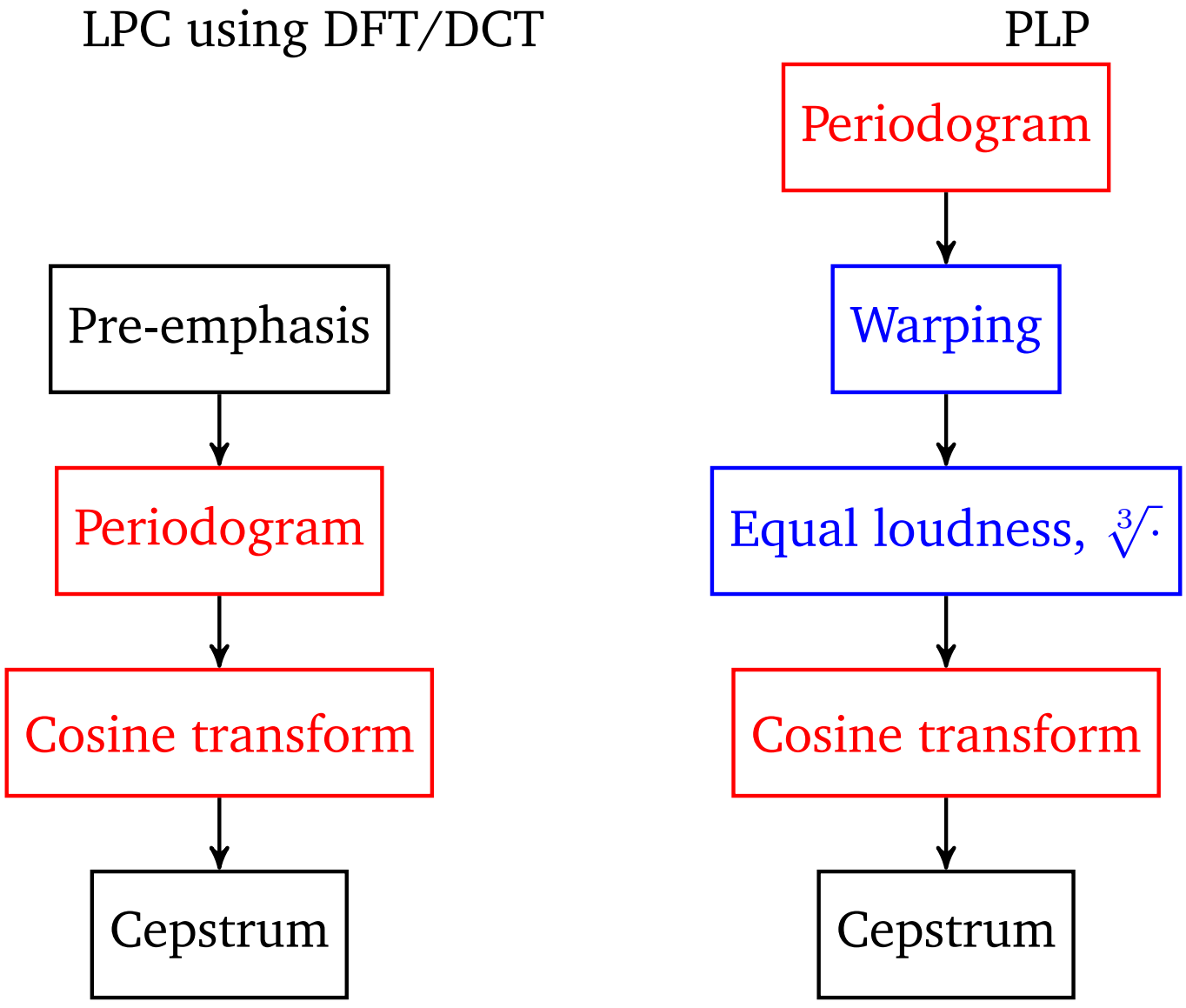
Raw LPC



LPC using DFT/DCT



LPC vs. PLP


LPC using DFT/DCT
Pre-emphasis
Periodogram
Cosine transform
Cepstrum
PLP
Periodogram
Warping
Equal loudness, $\sqrt[3]{\cdot}$
Cosine transform
Cepstrum

Conclusions about PLP

PLP gives quite tight specifications for some parameters

- In general, the actual numbers don't matter.
- The trends of curves are important, not the values.

e.g.,

- Triangular bins work as well as trapezoidal ones.
- Mel scale is as good as Bark.
- Pre-emphasis is OK for loudness.
- 12 LPC coefficients tend to be better than 5 if you have the data.

Contribution of PLP

Equal loudness is basically pre-emphasis.

- And it's quicker to do it in the time domain.
- Also more numerically stable in fixed point.

PLP's main contribution is frequency warping of the LP spectrum.