

LING 490 - SPECIAL TOPICS IN LINGUISTICS

Fundamentals of Digital Signal Processing

Yan Tang

Department of Linguistics, UIUC

Week 15

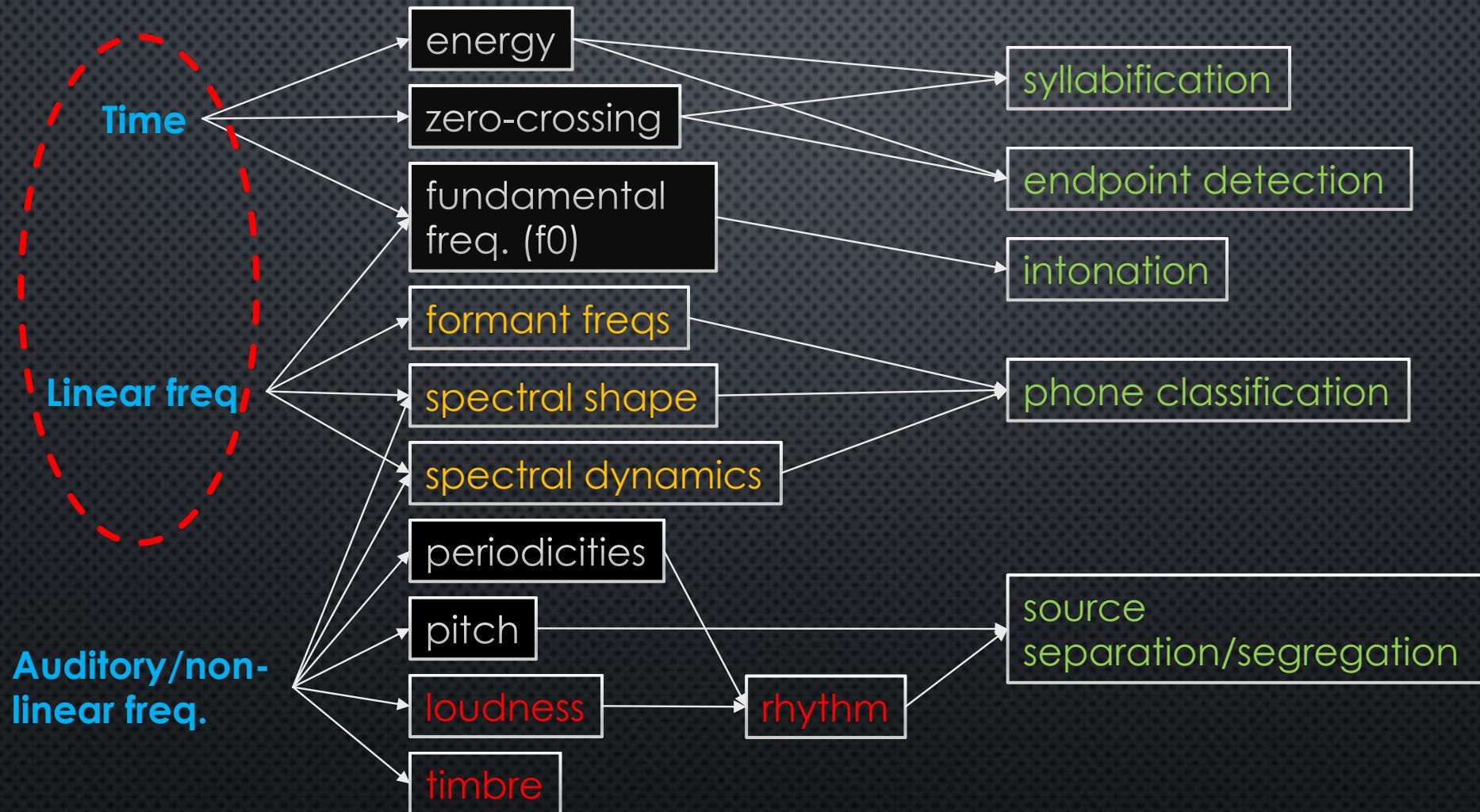
Last week...

- Pitch and fundamental frequency (f_0)
 - Pitch is an attribute of sensation
 - F_0 is a physical property of a periodic sound

Last week...

- F0 estimation using **autocorrelation**:
 - The correlation of a signal with itself being delayed
 - $\phi(\tau) = \sum_{n=0}^{N-1} x(n)x(n + \tau)$
 - The effect of window type
 - The effect of window size
 - Centre clipping: enhance periodicity
- F0 estimation using the average magnitude difference function (**AMDF**)
 - Look for dips instead of peaks
 - $M(\tau) = \sum_{n=0}^{N-1} |x(n) - x(n + \tau)|$

Useful parameters of speech signals



Spectral Centre of Gravity

- Also known as Spectral Centroid
- A measure of the distribution of energy at different frequencies in a spectrum.
- Midpoint of the spectral energy distribution of that sound.
- It can be considered a kind of "balancing point" of the frequency spectrum.

Spectral Centre of Gravity

- The spectral centroid measured at one instant in time probably will not prove to be very useful: sounds tend to constantly change.
- If it is measured at several points during the course of the signal, it could reveal how dynamic the signal

Spectral Centre of Gravity

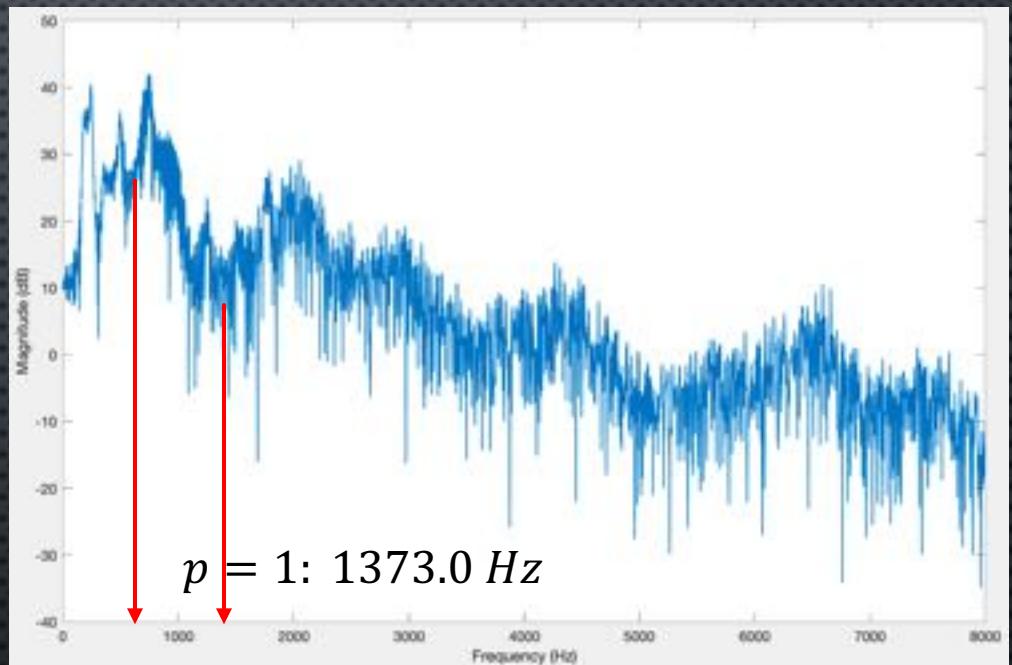
- Spectral Centre of Gravity in Hz is defined as:

$$CoG = \frac{\sum_{k=0}^{K-1} f(k)X(k)^p}{\sum_{k=0}^{K-1} X(k)^p}$$

$f(k)$: centre frequency of spectral bin k

$X(k)$: magnitude of spectral bin k

p : 1 – magnitude; 2 - power

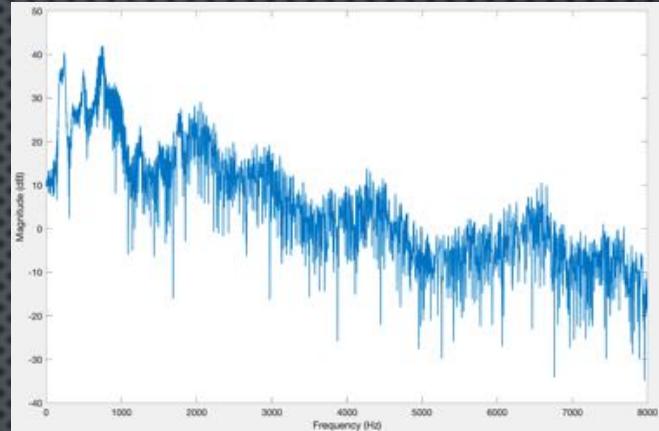


Why is CoG interesting?

Amongst other things:

- Related to impression of "brightness" of a sound
- Essential information for computing spectral moments
- Related to spectral tilt
- Lombard effect: higher spectral CoG – a flattening of spectral tilt
- For English (and Dutch), a more level spectral slope (decreased spectral tilt), i.e., a higher COG, strongly correlates with perceived sentence accent

Freq-domain speech processing



Why not simply use the (log) magnitude spectrum?

- source and filter are mixed
- difficult to choose time-frequency resolution to always get formants
 - spurious peaks
 - merged peaks
 - can be solved using pitch-synchronous analysis, but that is difficult too
- redundant for clean speech
- DFT is a general tool: not specialised for speech

Linear Predictive Coding (LPC)

- LPC is a powerful speech analysis technique
- One of the most useful methods for encoding good quality speech at a low bit rate.
- Provides extremely accurate estimates of speech parameters, and is relatively efficient computationally

Why LPC



- The voiced speech signal is quasi-periodic
- Anything that is periodic is, in principle, predictable.
- Can we take a number of speech samples and try to predict what comes next?
 - ... Yes, and with reasonable results too!

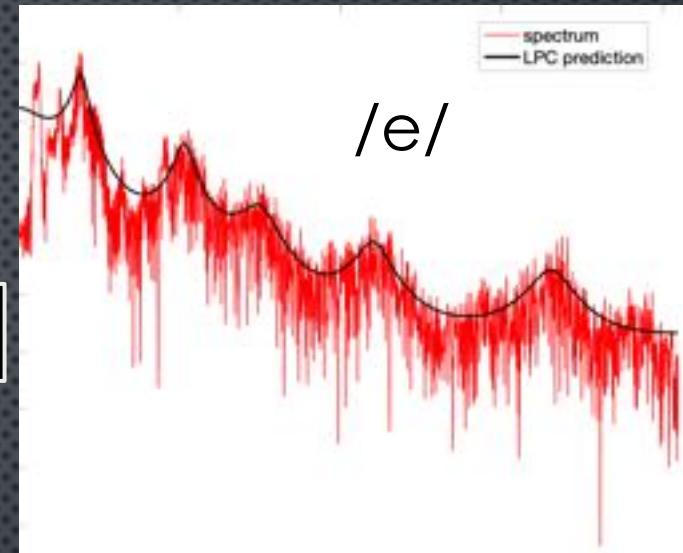
LPC: Basic principles



- LPC starts with the assumption that the speech signal is produced by a buzzer (producing pulse train) at the end of a tube.
source
- The glottis (the space between the vocal cords) produces the buzz, which is characterized by its intensity (loudness) and frequency (pitch)
- The vocal tract (the throat and mouth) forms the tube, which is characterized by its resonances, which are called formants
filter

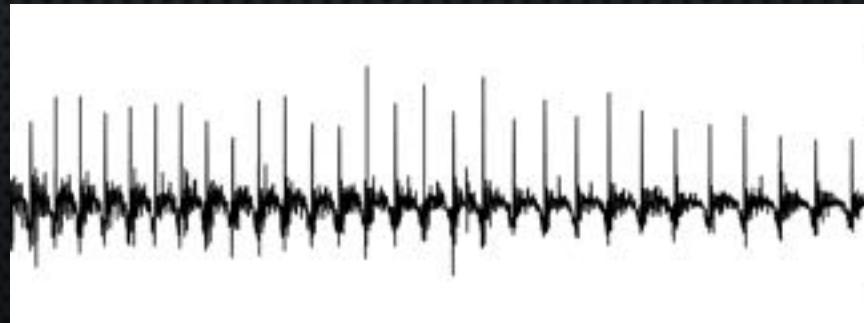
How does LPC do it?

filter



- LPC analyses the speech signal by:
 - estimating the formants and removing their effects from the speech signal
 - estimating the intensity and frequency of the remaining buzz sound
- The process of removing the formants is called *inverse filtering*, and the remaining signal is called the *residue*

source



The solution

- The basic solution is a *difference equation*, which expresses each sample of the signal as a linear combination of previous samples
- Hence ‘Linear’ and ‘Prediction’ in LPC

So how to predict?

$$y(t) = \sum_{j=0}^M b_j x(t-j) - \sum_i^N a_i y(t-i)$$

- We assume that whatever comes next in $y(n)$ is some linear combination of what's gone before:

$$\hat{y}(n) = a_1 y(n-1) + a_1 y(n-2) + \cdots + a_N y(n-N)$$

a_1, a_2, \dots, a_N : the prediction coefficients

Prediction coefficients

$$\hat{y}(n) = a_1y(n - 1) + a_2y(n - 2) + \cdots + a_Ny(n - N)$$

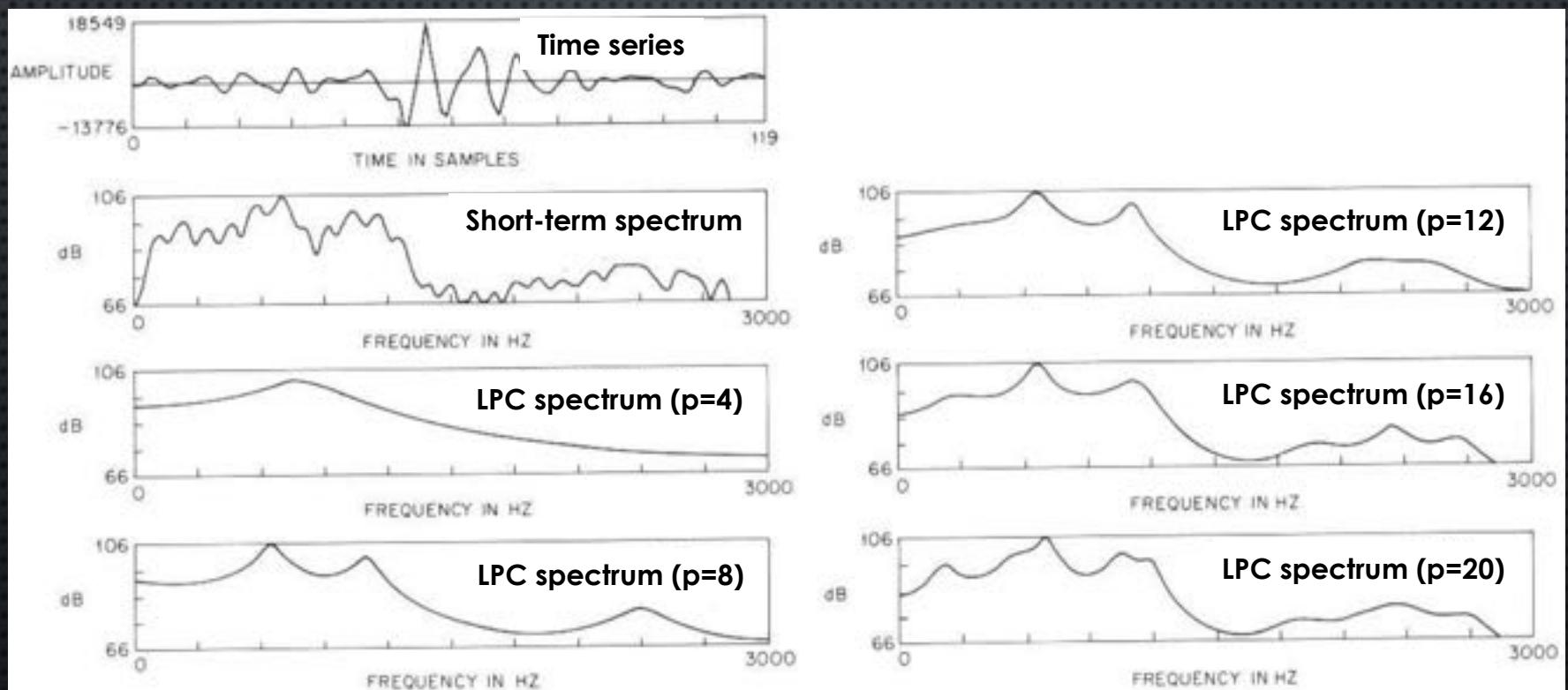
- LPC system needs to estimate these coefficients.
- In other words, what values of a_1, a_2, \dots, a_N give us the closest approximation to the true signal?

How's it done? (technical aside)

- This is a straightforward problem, in principle.
- In practice, it involves
 - 1. the computation of a set of coefficient values
 - 2. the solution of a set of linear equations.
- These can be solved using a variety of numerical analysis techniques (we're not going to cover this here)

LPC spectra and formant estimation

- Frequency response can be determined by the LPC coefficients



Have we done a good job?

- Given the predicted value and the actual value of $y(n)$, we can measure the error (difference):

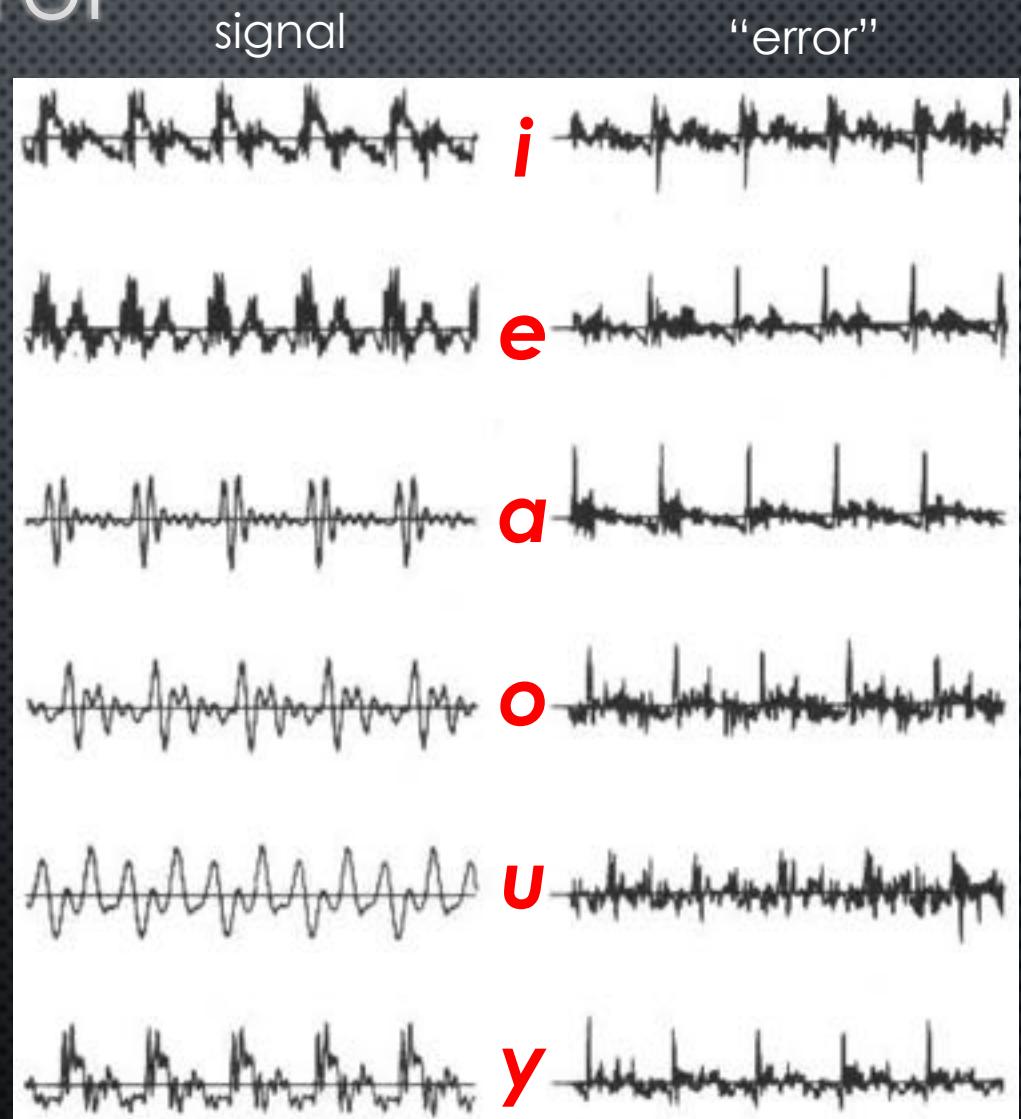
$$e(n) = y(n) - \hat{y}(n)$$

- Ideally, we want $e(n)$ to be as small as possible i.e. our prediction is near perfect

Prediction error

- Even if we have the best set of coefficients, the error may still be non-zero. Why?

Prediction error $e(n)$ covaries with signal

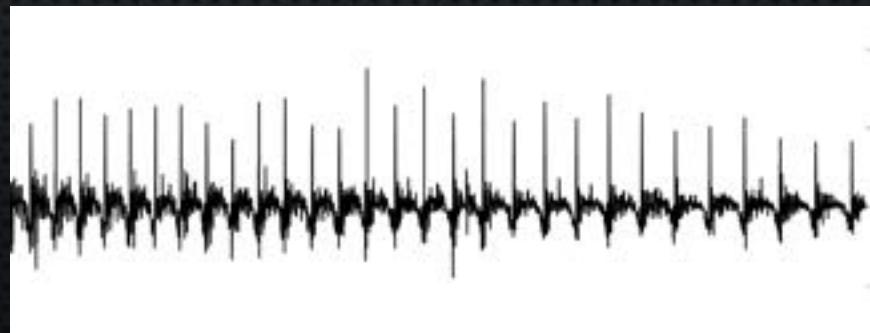


The source

- LPC models the vocal tract filter
- The source is accounted for by the error $e(n)$
 - $e(n)$ is not really an error at all!
 - $e(n)$ can be used as the basis for F0 estimation

filter

source



LPC remarks

- Not good for sounds like nasals
- Need to choose number of coefficients
- C in LPC stands for ‘coding’ – coefficients can be used as a more efficient means of representing a signal.
- Still need to code the error signal, or we could use a different one all together