

## LECTURE 15: LINEAR PREDICTION

[Return to Main](#)

[Objectives](#)

### Basic Theory:

[Simple Case](#)

[General Case](#)

[Covariance](#)

[Autocorrelation](#)

[Error](#)

### Spectral Matching:

[Spectrum](#)

[Noise-Weighting](#)

[Preemphasis](#)

[Typical Front End](#)

### On-Line Resources:

[Signal Modeling](#)

[Linear Prediction](#)

[MAD Speech](#)

- Objectives:
  - Introduce the theory of linear prediction
  - Develop autocorrelation and covariance techniques for solution
  - Understand similarities with regression
  - Explain the relationship to windowing and maximum entropy
  - Add a new technique to our signal modeling block diagram

There is a classic textbook on this subject:

J.D. Markel and A.H. Gray, *Linear Prediction of Speech*, Springer-Verlag, New York, New York, USA, ISBN: 0-13-007444-6, 1976.

This lecture also includes material from two other textbooks:

J. Deller, et. al., *Discrete-Time Processing of Speech Signals*, MacMillan Publishing Co., ISBN: 0-7803-5386-2, 2000.

and,

L.R. Rabiner and B.W. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Upper Saddle River, New Jersey, USA, ISBN: 0-13-015157-2, 1993.



## Introduction:

- 01: Organization  
([html](#), [pdf](#))

## Speech Signals:

- 02: Production  
([html](#), [pdf](#))
- 03: Digital Models  
([html](#), [pdf](#))
- 04: Perception  
([html](#), [pdf](#))
- 05: Masking  
([html](#), [pdf](#))
- 06: Phonetics and Phonology  
([html](#), [pdf](#))
- 07: Syntax and Semantics  
([html](#), [pdf](#))

## Signal Processing:

- 08: Sampling  
([html](#), [pdf](#))
- 09: Resampling  
([html](#), [pdf](#))
- 10: Acoustic Transducers  
([html](#), [pdf](#))
- 11: Temporal Analysis  
([html](#), [pdf](#))
- 12: Frequency Domain Analysis  
([html](#), [pdf](#))
- 13: Cepstral Analysis  
([html](#), [pdf](#))
- 14: **Exam No. 1**  
([html](#), [pdf](#))
- 15: Linear Prediction  
([html](#), [pdf](#))
- 16: LP-Based Representations  
([html](#), [pdf](#))

## Parameterization:

- 17: Differentiation  
([html](#), [pdf](#))
- 18: Principal Components  
([html](#), [pdf](#))

# ECE 8463: FUNDAMENTALS OF SPEECH RECOGNITION

Professor Joseph Picone  
Department of Electrical and Computer Engineering  
Mississippi State University

email: [picone@isip.msstate.edu](mailto:picone@isip.msstate.edu)  
phone/fax: 601-325-3149; office: 413 Simrall  
URL: [http://www.isip.msstate.edu/resources/courses/ece\\_8463](http://www.isip.msstate.edu/resources/courses/ece_8463)

Modern speech understanding systems merge interdisciplinary technologies from Signal Processing, Pattern Recognition, Natural Language, and Linguistics into a unified statistical framework. These systems, which have applications in a wide range of signal processing problems, represent a revolution in Digital Signal Processing (DSP). Once a field dominated by vector-oriented processors and linear algebra-based mathematics, the current generation of DSP-based systems rely on sophisticated statistical models implemented using a complex software paradigm. Such systems are now capable of understanding continuous speech input for vocabularies of hundreds of thousands of words in operational environments.

In this course, we will explore the core components of modern statistically-based speech recognition systems. We will view speech recognition problem in terms of three tasks: signal modeling, network searching, and language understanding. We will conclude our discussion with an overview of state-of-the-art systems, and a review of available resources to support further research and technology development.

Tar files containing a compilation of all the notes are available. However, these files are large and will require a substantial amount of time to download. A tar file of the html version of the notes is available [here](#). These were generated using wget:

```
wget -np -k -m http://www.isip.msstate.edu/publications/courses/ece_8463/lectures/current
```

A pdf file containing the entire set of lecture notes is available [here](#). These were generated using Adobe Acrobat.

Questions or comments about the material presented here can be directed to [help@isip.msstate.edu](mailto:help@isip.msstate.edu).



## LECTURE 15: LINEAR PREDICTION

- Objectives:
  - Introduce the theory of linear prediction
  - Develop autocorrelation and covariance techniques for solution
  - Understand similarities with regression
  - Explain the relationship to windowing and maximum entropy
  - Add a new technique to our signal modeling block diagram

There is a classic textbook on this subject:

J.D. Markel and A.H. Gray, *Linear Prediction of Speech*, Springer-Verlag, New York, New York, USA, ISBN: 0-13-007444-6, 1976.

This lecture also includes material from two other textbooks:

J. Deller, et. al., *Discrete-Time Processing of Speech Signals*, MacMillan Publishing Co., ISBN: 0-7803-5386-2, 2000.

and,

L.R. Rabiner and B.W. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Upper Saddle River, New Jersey, USA, ISBN: 0-13-015157-2, 1993.

## LINEAR PREDICTION - A SIMPLE DERIVATION

Let us define a speech signal,  $s(n)$ . Consider the problem of predicting the current value from the previous value,  $\tilde{s}(n) = \alpha s(n-1)$ . This prediction will be in error by some amount:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \alpha s(n-1)$$

We would like to minimize the error by finding the best, or optimal, value of  $\alpha$ . Let us define the short-time average prediction error:

$$\begin{aligned} E &= \sum_n e^2(n) \\ &= \sum_n \{s(n) - \alpha s(n-1)\}^2 \end{aligned}$$

We can minimize the error w.r.t  $\alpha$  by differentiating  $E$  and setting the result equal to zero:

$$E = \sum_n (s^2(n) - 2\alpha s(n)s(n-1) + \alpha^2 s^2(n-1))$$

Differentiating w.r.t  $\alpha$ ,

$$\frac{\partial E}{\partial \alpha} = 0 = \sum_n -2s(n)s(n-1) + 2\alpha s^2(n-1)$$

or,

$$\sum_n s(n)s(n-1) = \alpha \sum_n s^2(n-1)$$

which implies:

$$\alpha = \frac{\sum_n s(n)s(n-1)}{\sum_n s^2(n-1)} = \frac{c(1, 0)}{c(1, 1)} \approx \frac{r(1)}{r(0)}$$

Notes:

- related to the correlation structure of the signal ( $\alpha < 1$ )
- independent of the energy level of the signal

What short-term technique do we use to compute correlation/covariance?

## LINEAR PREDICTION - GENERAL CASE

Let us define a speech signal,  $s(n)$ , and a predicted value:  $\tilde{s}(n) = \sum_{k=1}^p \alpha_k s(n-k)$ .

Why the added terms? This prediction error is given by:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^p \alpha_k s(n-k)$$

We would like to minimize the error by finding the best, or optimal, value of  $\{\alpha_k\}$ .

Let us define the short-time average prediction error:

$$\begin{aligned} E &= \sum_n e^2(n) \\ &= \sum_n \left\{ s(n) - \sum_{k=1}^p \alpha_k s(n-k) \right\}^2 \\ &= \sum_n s^2(n) - \sum_n \left\{ 2s(n) \sum_{k=1}^p \alpha_k s(n-k) \right\} + \sum_n \left\{ \sum_{k=1}^p \alpha_k s(n-k) \right\}^2 \\ &= \sum_n s^2(n) - 2 \sum_{k=1}^p \alpha_k \sum_n s(n)s(n-k) + \sum_n \left\{ \sum_{k=1}^p \alpha_k s(n-k) \right\}^2 \end{aligned}$$

We can minimize the error w.r.t  $\alpha_l$  for each  $1 \leq l \leq p$  by differentiating  $E$  and setting the result equal to zero:

$$\frac{\partial E}{\partial \alpha_l} = 0 = -2 \sum_n s(n)s(n-l) + 2 \sum_n \left\{ \sum_{k=1}^p \alpha_k s(n-k) \right\} s(n-l)$$

## THE COVARIANCE METHOD

Rearranging terms:

$$\sum_n s(n)s(n-l) = \sum_{k=1}^p \alpha_k \left( \sum_n s(n-k)s(n-l) \right)$$

or,

$$c(l, 0) = \sum_{k=1}^p \alpha_k c(k, l)$$

This equation is known as the linear prediction (Yule-Walker) equation.  $\{\alpha_k\}$  are known as linear prediction coefficients, or *predictor coefficients*.

By enumerating the equations for each value of  $l$ , we can express this in matrix form:

$$\underline{\bar{c}} = \underline{C} \underline{\bar{\alpha}}$$

where,

$$\underline{\bar{\alpha}} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_p \end{bmatrix} \quad \underline{C} = \begin{bmatrix} c(1,1) & c(1,2) & \dots & c(1,p) \\ c(2,1) & c(2,2) & \dots & c(2,p) \\ \dots & \dots & \dots & \dots \\ c(p,1) & c(p,2) & \dots & c(p,p) \end{bmatrix} \quad \underline{\bar{c}} = \begin{bmatrix} c(1,0) \\ c(2,0) \\ \dots \\ c(p,0) \end{bmatrix}$$

The solution to this equation involves a matrix inversion:

$$\underline{\bar{\alpha}} = \underline{C}^{-1} \underline{\bar{c}}$$

and is known as the *covariance method*. Under what conditions does  $\underline{C}^{-1}$  exist?

Note that the covariance matrix is symmetric. A fast algorithm to find the solution to this equation is known as the Cholesky decomposition (a  $\underline{VDV}^T$  approach in which the covariance matrix is factored into lower and upper triangular matrices).

## THE AUTOCORRELATION METHOD

Using a different interpretation of the limits on the error minimization — forcing data only within the frame to be used — we can compute the solution to the linear prediction equation using the autocorrelation method:

$$\bar{\alpha} = \underline{R}^{-1} \bar{r}$$

where,

$$\bar{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_p \end{bmatrix} \quad \underline{R} = \begin{bmatrix} r(0) & r(1) & \dots & r(p-1) \\ r(1) & r(0) & \dots & r(p-2) \\ \dots & \dots & \dots & \dots \\ r(p-1) & r(p-2) & \dots & r(0) \end{bmatrix} \quad \bar{r} = \begin{bmatrix} r(1) \\ r(2) \\ \dots \\ r(p) \end{bmatrix}$$

Note that  $\underline{R}$  is symmetric, and all of the elements along the diagonal are equal, which means (1) an inverse always exists; (2) the roots are in the left-half plane.

The linear prediction process can be viewed as a filter by noting:

$$e(n) = s(n) - \sum_{k=1}^p \alpha_k s(n-k)$$

and

$$E(z) = S(z)A(z)$$

where

$$A(z) = 1 - \sum_{k=1}^p \alpha_k z^{-k}$$

$A(z)$  is called the analyzer; what type of filter is it? (pole/zero? phase?)

$\frac{1}{A(z)}$  is called the synthesizer; under what conditions is it stable?

## LINEAR PREDICTION ERROR

We can return to our expression for the error:

$$E = \sum_n e^2(n) = \sum_n \left\{ s(n) - \sum_{k=1}^p \alpha_k s(n-k) \right\}^2$$

and substitute our expression for  $\{\alpha_k\}$  and show that:

Autocorrelation Method:

$$E = r(0) - \sum_{k=1}^p \alpha_k r(k)$$

Covariance Method:

$$E = c(0, 0) - \sum_{k=1}^p \alpha_k c(0, k)$$

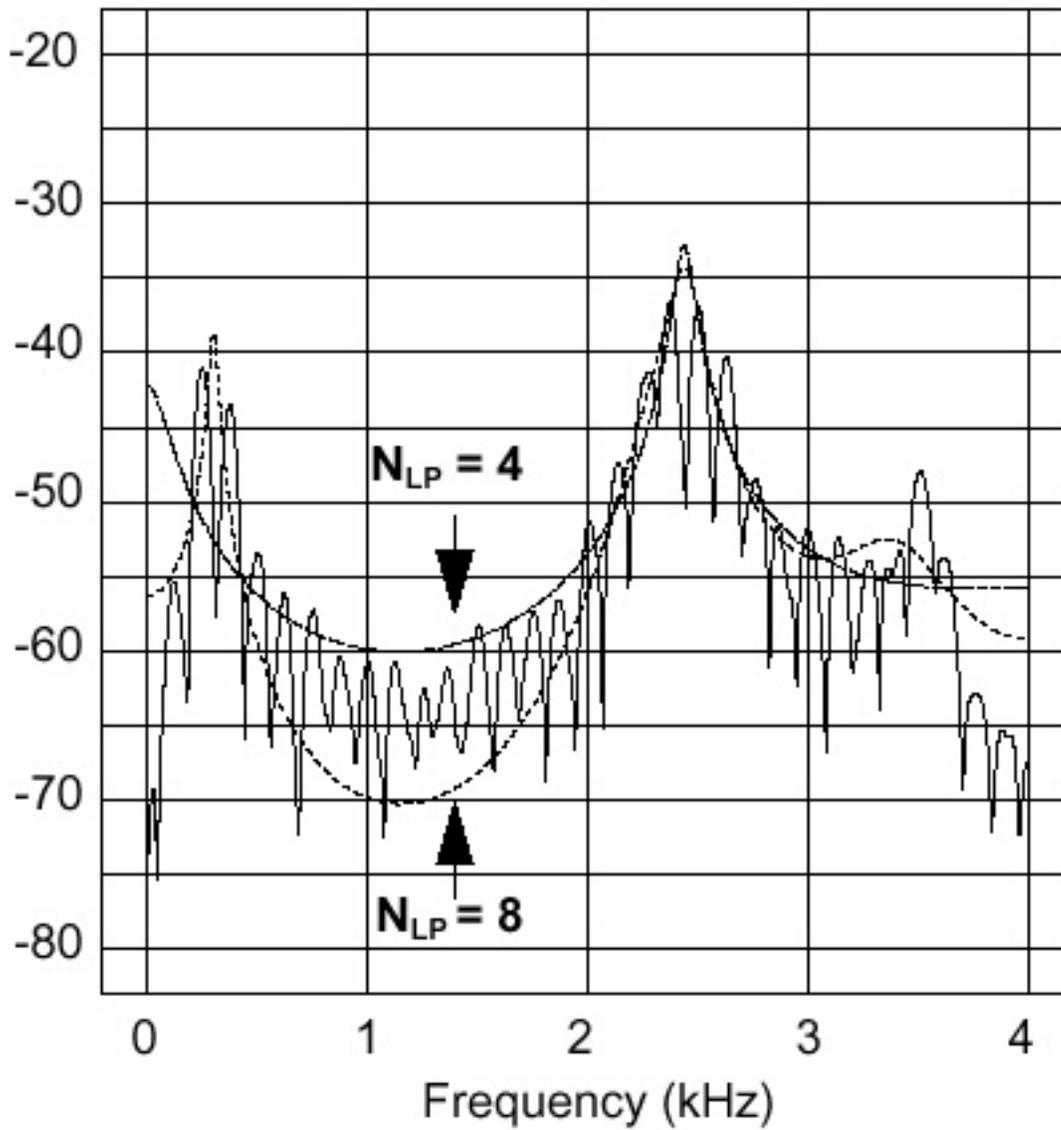
Later, we will discuss the properties of these equations as they relate to the magnitude of the error. For now, note that the same linear prediction equation that applied to the signal applies to the autocorrelation function, except that samples are replaced by the autocorrelation lag (and hence the delay term is replaced by a lag index).

Since the same coefficients satisfy both equations, this confirms our hypothesis that this is a model of the minimum-phase version of the input signal.

Linear prediction has numerous formulations including the covariance method, autocorrelation formulation, lattice method, inverse filter formulation, spectral estimation formulation, maximum likelihood formulation, and inner product formulation. Discussions are found in disciplines ranging from system identification, econometrics, signal processing, probability, statistical mechanics, and operations research.

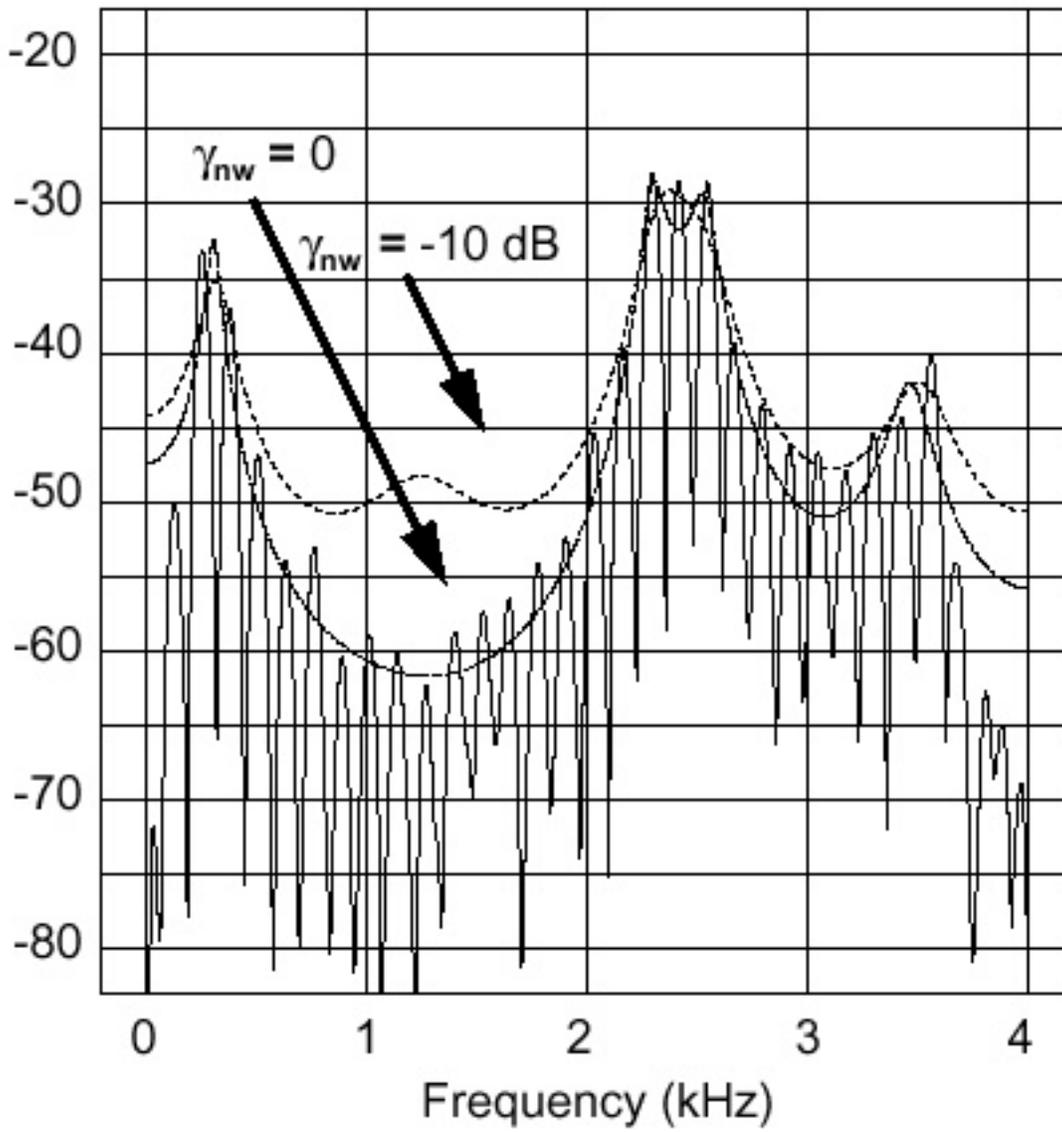
# SPECTRAL MATCHING INTERPRETATION

Magnitude(dB)

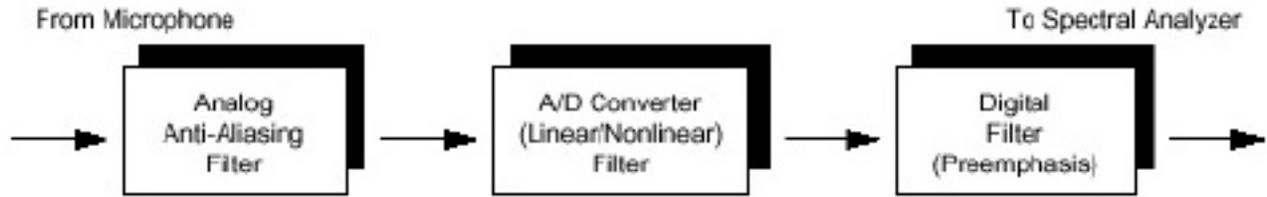


NOISE FLOORING  
VIA THE AUTOCORRELATION FUNCTION

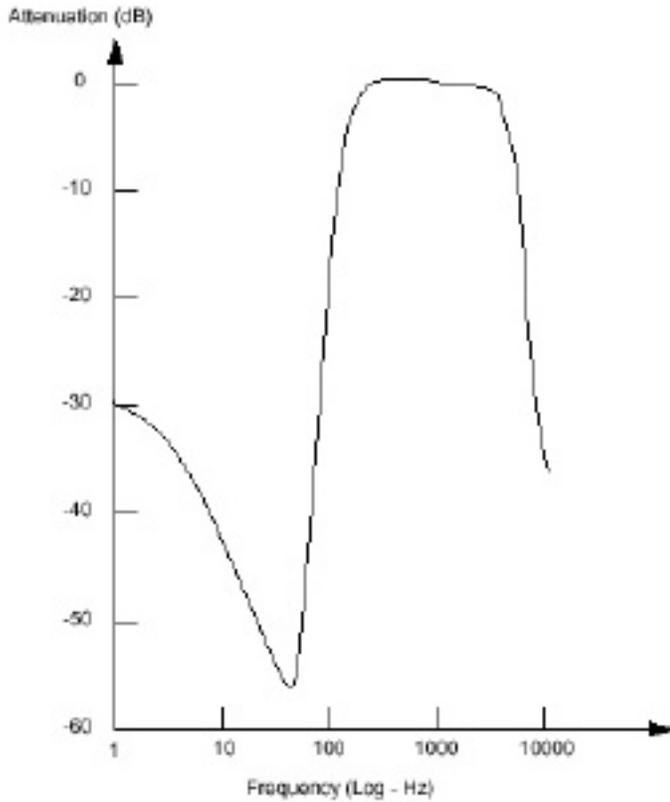
Magnitude(dB)



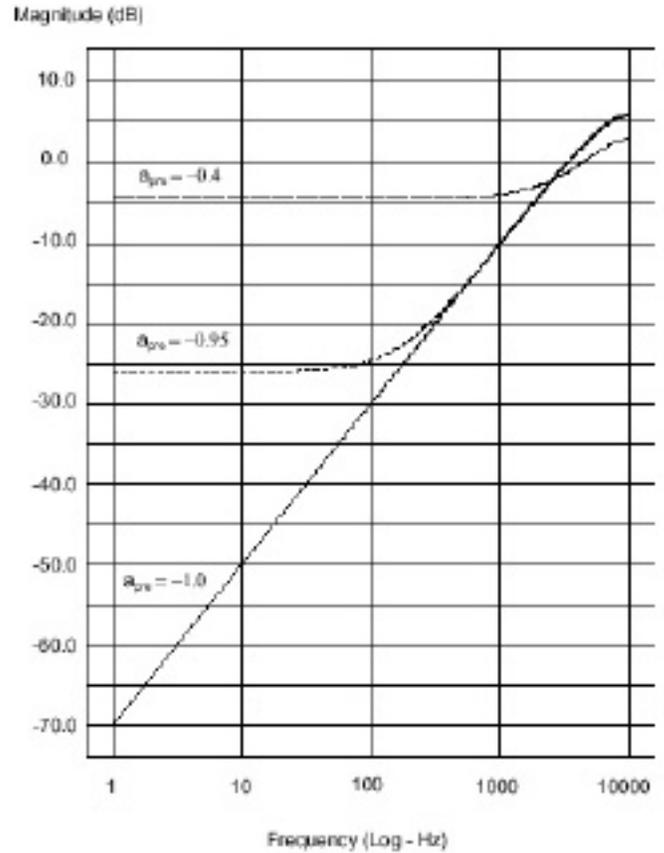
# SIGNAL CONDITIONING COMPENSATES FOR MICROPHONE AND CHANNEL CHARACTERISTICS



## Frequency Response of a CODEC



## Preemphasis Filter



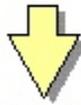
# A TYPICAL SPEECH RECOGNITION FRONT END



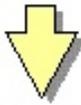
Input Speech



Fourier Transform



Cepstral Analysis



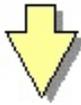
Perceptual Weighting



Time Derivative



Time Derivative



Energy  
+

Mel-Spaced Cepstrum



Delta Energy  
+

Delta Cepstrum



Delta-Delta Energy  
+

Delta-Delta Cepstrum

- Measure features 100 times per sec.
- Use a 25 msec window for frequency domain analysis.
- Include absolute energy and 12 spectral measurements.
- Time derivatives to model spectral change.

- Incorporate knowledge of the nature of speech sounds in measurement of the features.
- Utilize rudimentary models of human perception.

# Index of /publications/journals/ieee\_proceedings/1993/signal\_modeling

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>	01-Jan-1999 12:07	-	
 <a href="#">paper_v2.pdf</a>	22-Jun-1999 16:14	452k	

Apache/1.3.9 Server at www.isip.msstate.edu Port 80

# Linear Prediction analysis

Linear prediction analysis of speech is historically one of the most important speech analysis techniques. The basis is the source-filter model where the filter is constrained to be an all-pole linear filter. This amounts to performing a linear prediction of the next sample as a weighted sum of past samples:

$$\hat{s}_n = \sum_{i=1}^p \alpha_i s_{n-i} \quad (61)$$

This linear filter has the transfer function:

$$H(z) = \frac{1}{1 - \sum_{i=1}^p \alpha_i z^{-i}} \quad (62)$$

A good introductory article is [\[8\]](#), and this subject is also covered well in [\[1, 2, 3\]](#).

- 
- [Motivation from lossless tubes](#)
  - [Parameter estimation](#)
  - [The autocorrelation method](#)
  - [The covariance method](#)
  - [Pre-emphasis](#)
  - [The LP spectrum](#)
  - [Gain computation](#)
  - [The lattice filter implementation](#)
  - [The Itakura distance measure](#)
  - [The LP cepstrum](#)
  - [Log area ratios](#)
  - [The roots of the predictor polynomial](#)
  - [Line spectral pairs](#)
-



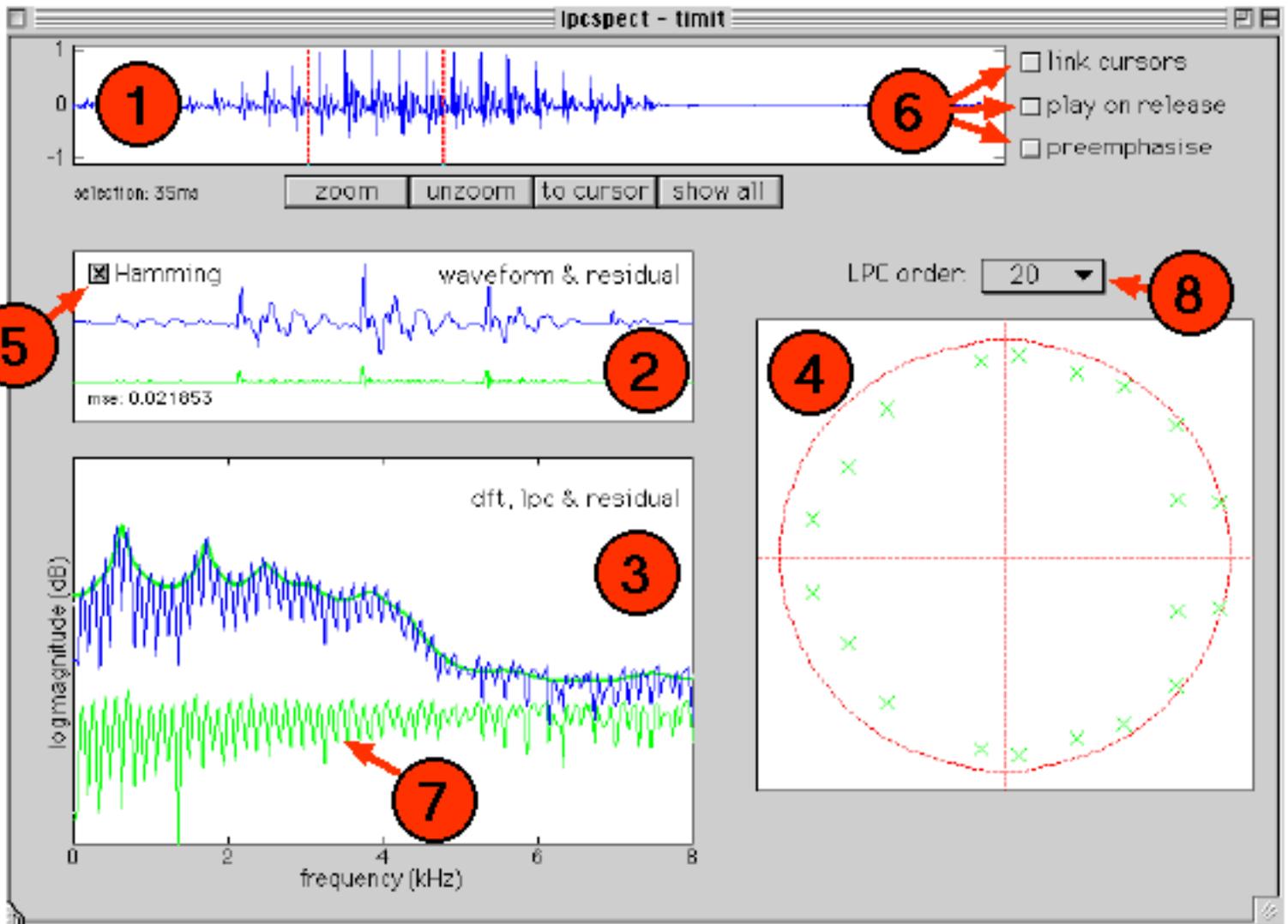
---

[introduction](#) | [demonstration](#) | [investigate](#) | [reading](#) | [credits](#) | [downloading](#) | [home](#)

## Introduction

This tool allows users to explore linear prediction of speech and other signals. It also reinforces the concepts of windowing, preemphasis and frame size selection in speech analysis. Users can select regions for analysis and see the results as lpc-smoothed spectra and poles. The residual signal and its spectrum is also displayed. This tool is not a substitute for learning about linear prediction. To understand what the various panels mean, you should read one of the texts listed below. The demo requires the MATLAB signal processing toolbox.

## The tool



Type 'lpcspect' to launch the demo. The *file* menu is used to load an existing signal file or to create a new one using the [createsig](#) tool. Supported formats currently include .wav, .snd and .au sound files. The signal will appear in panel (1), and associated linear prediction signals for the region between the cursors will be displayed in the other panels. Use the cursors to select any segment. On cursor release, the segment appears in panel (2) along with linear prediction residual (error signal). Panel (3) shows the DFT spectrum of this segment, overlaid with the LPC-smoothed spectrum and the DFT of the residual (7). The poles corresponding to the linear predictor are shown in panel (4).

The LPC order can be changed using menu (8). The signal can be preemphasised by checking the relevant box in cluster (6). Cursors can be linked (so that a constant segment length is maintained). The waveform segment chosen will be played on release if the associated checkbox is checked.

Clicking on any signal waveform (including the residual) causes it to be played. Holding the mouse button down in the z-plane panel (4) will cause the frequency at the selected z-plane angle to be displayed in the top left corner of the panel.

The chosen waveform segment can be Hamming-windowed (5). Unchecked, a rectangular window is applied.

# Things to investigate

1. Choose a vowel segment of speech. Examine what happens to the residual signal (and its mean square error) as you increase the number of poles. What happens to the LPC-smoothed spectrum? How many poles provide a comfortable fit to a vowel segment.
2. Now repeat this analysis for unvoiced sounds such as fricatives. How does the error compare?
3. Again, perform this analysis for a nasal sound.
4. Can you predict where the poles will appear for a given analysis order, and how they will change when you modify the order?
5. Explore the effect of preemphasis on the LPC spectrum and the number of poles required to obtain a given residual error.
6. Listen to the waveform segment and the residual signal in sequence. Can you detect a change in timbre? How does this change manifest itself as the LPC order increases?
7. What is the effect of changing LPC order on the spectrum of the residual signal?
8. Choose a vowel sound and a relatively low LPC order (10, say). Read off the pole locations (frequencies) by holding the mouse down in panel (4). Now use the [polezero](#) tool to recreate just these pole locations and listen to the resulting sound.
9. Use the *create* option on the file menu to create a harmonic series with, say, 5 components. Do you expect to be able to fit it well with a 10th order model? Try it.
10. Now add some noise (using *create*) to your harmonic series. Do the pole locations change? (Read about the peak-hugging properties of LPC).

## Further reading

- A good introduction to LPC is provided by Makhoul, J. (19??)
- This tool was written partly as an example of the interface issues in speech and hearing demonstrations, and is described further in Cooke et al (1999) The interactive auditory demonstrations project, Eurospeech'99, Budapest, September.

---

## Credits etc

**Produced by:** Martin Cooke, April 1999.

**Permissions:** This demonstration may be used and modified freely by anyone. It may be distributed in unmodified form.