Chapter 9

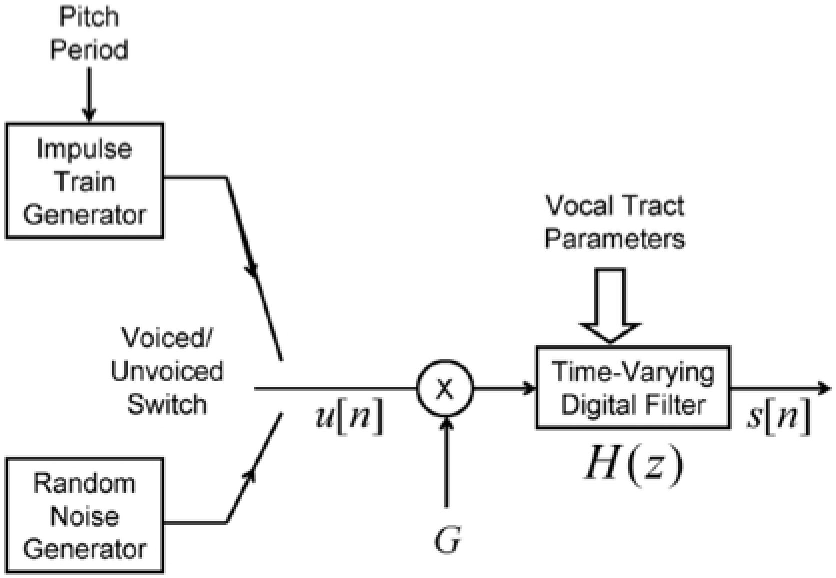
Linear Predictive Analysis of Speech Signals
语音信号的线性预测分析

LPC Methods

- LPC methods are the most widely used in speech coding, speech synthesis, speech recognition, speaker recognition and verification and for speech storage
  - LPC methods provide extremely accurate estimates of speech parameters, and does it extremely efficiently
  - basic idea of Linear Prediction: current speech sample can be closely approximated as a linear combination of past samples, i.e.,

$$\boxed{s(n) = \sum_{k=1}^{p} \alpha_k \, s(n-k) \text{ for some value of } p, \alpha_k\text{'s}}$$

Speech Production Model



$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + G u(n)$$

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

- the time-varying digital filter represents the effects of the glottal pulse shape, the vocal tract IR, and radiation at the lips
- the system is excited by an impulse train for voiced speech, or a random noise sequence for unvoiced speech
- this ‘all-pole’ model is a natural representation for non-nasal voiced speech—but it also works reasonably well for nasals and unvoiced sounds

LPC Methods

- for periodic signals with N_p period , it is obvious that

$$s(n) \approx s(n - N_p)$$

  but that is not what LP is doing; it is estimating $s(n)$ from the p (p<< N_p) most recent values of $s(n)$ by linearly predicting its value
- for LP, the predictor coefficients (the α_k's) are determined (computed) by minimizing the sum of squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones

Linear Prediction Model

- a *p*-th order linear predictor is a system of the form

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k) \Leftrightarrow P(z) = \sum_{k=1}^{p} \alpha_k z^{-k} = \frac{\tilde{S}(z)}{S(z)}$$

- the prediction error, *e*(*n*), is of the form

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

- the prediction error is the output of a system with transfer function

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

LP Estimation Issues

- need to determine $\{\alpha_k\}$ directly from speech such that they give good estimates of the time-varying spectrum
- need to estimate $\{\alpha_k\}$ from short segments of speech
- minimize mean-squared prediction error over short segments of speech
  - if the speech signal obeys the production model exactly, then
  - $\alpha_k = a_k$
  - $e(n) = Gu(n)$
  - $A(z)$ is an inverse filter for $H(z)$

Solution for $\{\alpha_k\}$

- short-time average prediction squared-error is defined as

$$E_{\hat{n}} = \sum_m e_{\hat{n}}^2(m) = \sum_m \left(s_{\hat{n}}(m) - \tilde{s}_{\hat{n}}(m)\right)^2$$

$$= \sum_m \left(s_{\hat{n}}(m) - \sum_{k=1}^{p} \alpha_k s_{\hat{n}}(m-k) \right)^2$$

- select segment of speech $s_{\hat{n}}(m) = s(m+\hat{n})$ in the vicinity of sample $\hat{n}$
- the key issue to resolve is the range of *m* for summation (to be discussed later)

Solution for $\{\alpha_k\}$

- can find values of α_k that minimize $E_{\hat{n}}$ by setting

$$\frac{\partial E_{\hat{n}}}{\partial \alpha_i} = 0, \qquad i = 1, 2, \ldots, p$$

- giving the set of equations

$$-2\sum_m s_{\hat{n}}(m-i)[s_{\hat{n}}(m) - \sum_{k=1}^{p} \hat{\alpha}_k s_{\hat{n}}(m-k)] = 0, \;\; 1 \le i \le p$$

$$-2\sum_m s_{\hat{n}}(m-i)\; e_{\hat{n}}(m) = 0, \;\; 1 \le i \le p$$

where $\hat{\alpha}_k$ are the values of α_k that minimize $E_{\hat{n}}$ (from now on just use α_k rather than $\hat{\alpha}_k$ for the optimum values)

- prediction error $(e_{\hat{n}}(m))$ is orthogonal to signal $(s_{\hat{n}}(m-i))$ for delays (i) of 1 to p

Solution for $\{\alpha_k\}$

- defining

$$\phi_{\hat{n}}(i,k) = \sum_{m} s_{\hat{n}}(m-i)s_{\hat{n}}(m-k)$$

- we get

$$\sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(i,k) = \phi_{\hat{n}}(i,0), \quad i = 1,2,\dots,p$$

- leading to a set of *p* equations in *p* unknowns that can be solved in an efficient manner for the $\{\alpha_k\}$

Solution for {α_k}

- minimum mean-squared prediction error has the form

$$E_{\hat{n}} = \sum_{m} s_{\hat{n}}^2(m) - \sum_{k=1}^{p} \alpha_k \sum_{m} s_{\hat{n}}(m) s_{\hat{n}}(m-k)$$

- which can be written in the form

$$E_{\hat{n}} = \phi_{\hat{n}}(0,0) - \sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(0,k)$$

- Process
  - Compute $\phi_{\hat{n}}(i,k)$ for $1 \le i \le p,\ 0 \le k \le p$
  - Solve matrix equation for α_k
- need to specify range of *m* to compute $\phi_{\hat{n}}(i,k)$
- need to specify $s_{\hat{n}}(m)$

minimum mean-squared prediction error:

$$E_{\hat{n}} = \sum_{m} [e_{\hat{n}}(m)]^2$$

$$= \sum_{m} e_{\hat{n}}(m) \left\{ s_{\hat{n}}(m) - \sum_{k=1}^{p} a_k s_{\hat{n}}(m-k) \right\}$$

$$= \sum_{m} e_{\hat{n}}(m) s_{\hat{n}}(m) - \sum_{k=1}^{p} a_k \sum_{m} e_{\hat{n}}(m) s_{\hat{n}}(m-k)$$

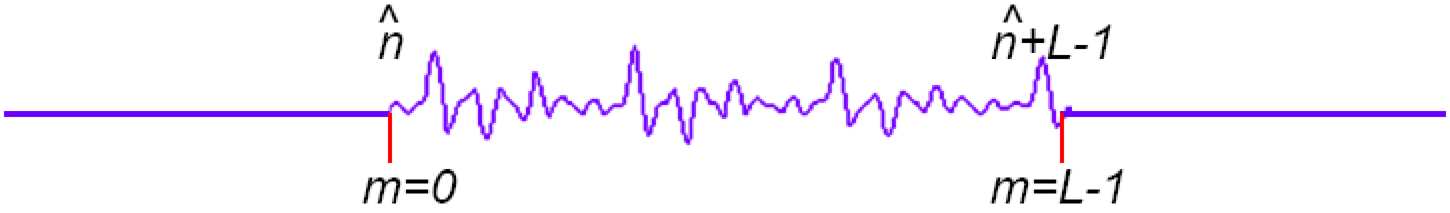
Autocorrelation Method

- assume $s_{\hat{n}}(m)$ exists for $0 \le m \le L-1$ and is exactly zero everywhere else (i.e., window of length *L* samples)

  (Assumption #1)

$$s_{\hat{n}}(m) = s(m+\hat{n})w(m), \quad 0 \le m \le L-1$$

where *w*(*m*) is a finite length window of length *L* samples



Autocorrelation Method

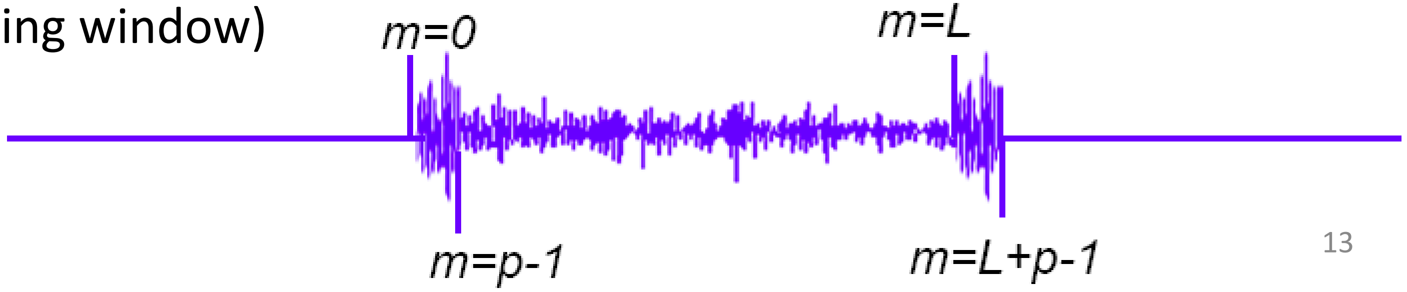
- if $s_{\hat{n}}(m)$ is non-zero only for $0 \le m \le L-1$, then

$$e_{\hat{n}}(m) = s_{\hat{n}}(m) - \sum_{k=1}^{p} \alpha_k s_{\hat{n}}(m-k)$$

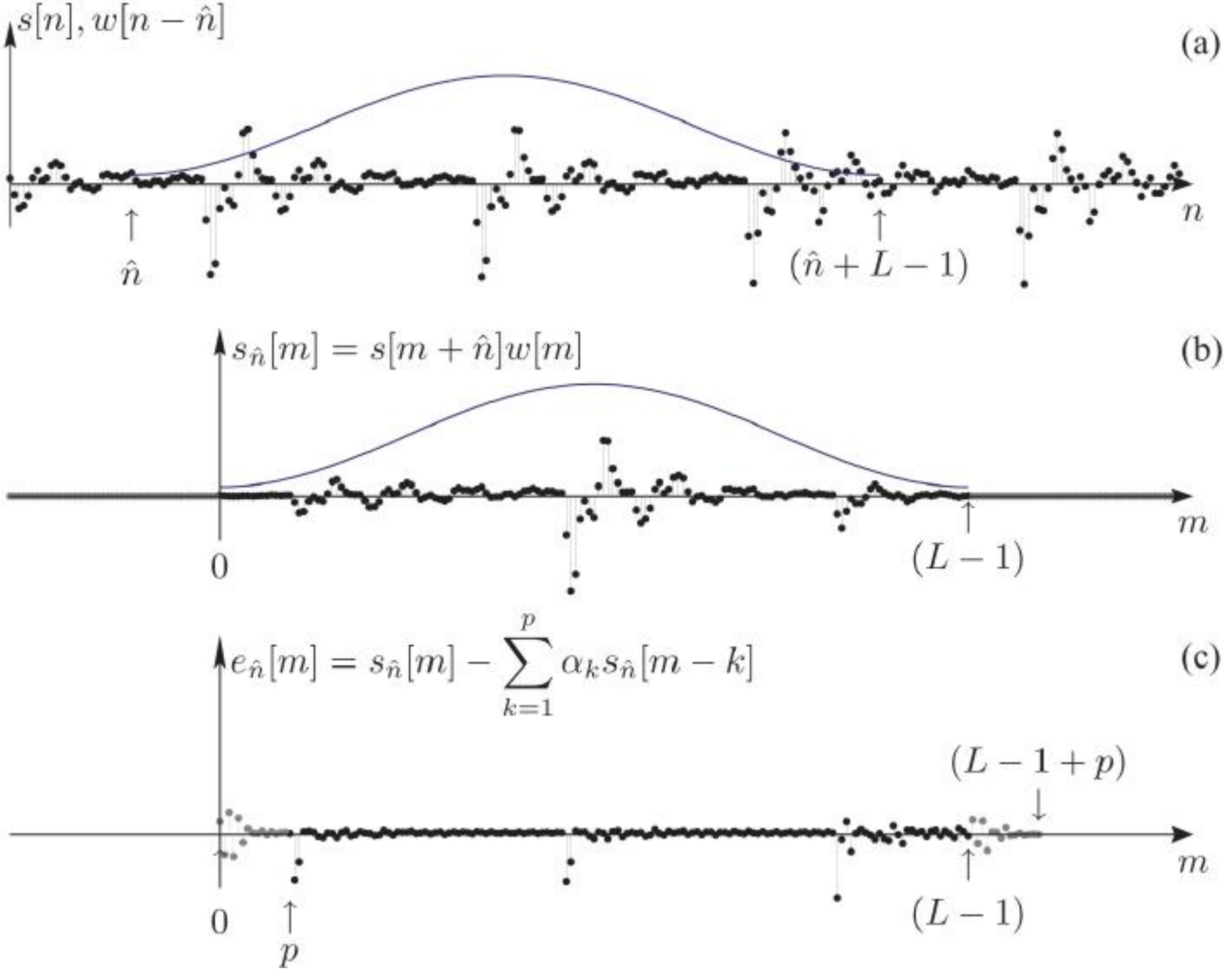
is non-zero only over the interval $0 \le m \le L-1+p$, giving

$$E_{\hat{n}} = \sum_{m=-\infty}^{\infty} e_{\hat{n}}^2(m) = \sum_{m=0}^{L-1+p} e_{\hat{n}}^2(m)$$

- at values of m near 0 (i.e. m = 0,1,…,p-1) we are predicting signal from zero-valued samples outside the window range => $e_{\hat{n}}(m)$ will be (relatively) large
- at values near m=L (i.e. m = L,L+1,…,L+p-1) we are predicting zero-valued samples (outside window range) from non-zero values => $e_{\hat{n}}(m)$ will be (relatively) large
- for these reasons, normally use windows that taper the segment to zero (e.g., Hamming window)



Autocorrelation Method



Autocorrelation Method

- for calculation of $\phi_{\hat{n}}(i,k)$ since $s_{\hat{n}}(m)=0$ outside the range $0 \le m \le L-1$, then

$$\phi_{\hat{n}}(i,k)=\sum_{m=0}^{L-1+p} s_{\hat{n}}(m-i)s_{\hat{n}}(m-k), \quad 1 \le i \le p, 0 \le k \le p$$

- which is equivalent to the form

$$\phi_{\hat{n}}(i,k)=\sum_{m=0}^{L-1-(i-k)} s_{\hat{n}}(m)s_{\hat{n}}(m+i-k), \quad 1 \le i \le p, 0 \le k \le p$$

- can easily show that

$$\phi_{\hat{n}}(i,k)=f(i-k)=R_{\hat{n}}(i-k), \qquad 1 \le i \le p, 0 \le k \le p$$

where $R_{\hat{n}}(i-k)$ is the shot-time autocorrelation of $s_{\hat{n}}(m)$ evaluated at *i-k,* where

$$R_{\hat{n}}(k)=\sum_{m=0}^{L-1-k} s_{\hat{n}}(m)s_{\hat{n}}(m+k)$$

Autocorrelation Method

- since $R_{\hat{n}}(k)$ is even, then

$$\phi_{\hat{n}}(i,k) = R_{\hat{n}}(|i-k|),\ 1 \le i \le p,\ 0 \le k \le p$$

- thus the basic equation becomes

$$\sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(i-k) = \phi_{\hat{n}}(i,0), \quad 1 \le i \le p$$

$$\sum_{k=1}^{p} \alpha_k R_{\hat{n}}(|i-k|) = R_{\hat{n}}(i), \quad 1 \le i \le p$$

with the minimum mean-squared prediction error of the form

$$E_{\hat{n}} = \phi_{\hat{n}}(0,0) - \sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(0,k)$$

$$= R_{\hat{n}}(0) - \sum_{k=1}^{p} \alpha_k R_{\hat{n}}(k)$$

Autocorrelation Method

- as expressed in matrix form

$$\begin{bmatrix} R_{\hat{n}}(0) & R_{\hat{n}}(1) & . & . & R_{\hat{n}}(p-1) \\ R_{\hat{n}}(1) & R_{\hat{n}}(0) & . & . & R_{\hat{n}}(p-2) \\ . & . & . & . & . \\ . & . & . & . & . \\ R_{\hat{n}}(p-1) & R_{\hat{n}}(p-2) & . & . & R_{\hat{n}}(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_{\hat{n}}(1) \\ R_{\hat{n}}(2) \\ . \\ . \\ R_{\hat{n}}(p) \end{bmatrix}$$

$$\Re \boldsymbol{\alpha} = \boldsymbol{r}$$

with solution

$$\boldsymbol{\alpha} = \Re^{-1} \boldsymbol{r}$$

- $\Re$ is a *pxp* Toeplitz Matrix => symmetric with all diagonal elements equal => there exist more efficient algorithms to solve for $\{\alpha_k\}$ than simple matrix inversion

Covariance Method

- there is a second basic approach to defining the speech segment $s_{\hat{n}}(m)$ and the limits on the sums, namely fix the interval over which the mean-squared error is computed, giving (Assumption #2)

$$E_{\hat{n}} = \sum_{m=0}^{L-1} e_{\hat{n}}^2(m) = \sum_{m=0}^{L-1} \left[s_{\hat{n}}(m) - \sum_{k=1}^{p} \alpha_k s_{\hat{n}}(m-k) \right]^2$$

$$\phi_{\hat{n}}(i,k) = \sum_{m=0}^{L-1} s_{\hat{n}}(m-i) s_{\hat{n}}(m-k), \;\; 1 \le i \le p, \;\; 0 \le k \le p$$

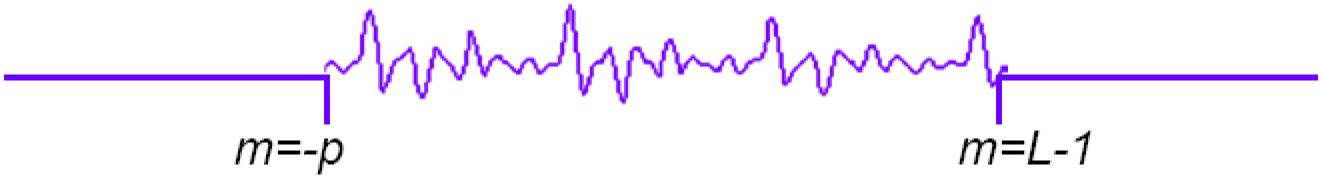
Covariance Method

- changing the summation index gives

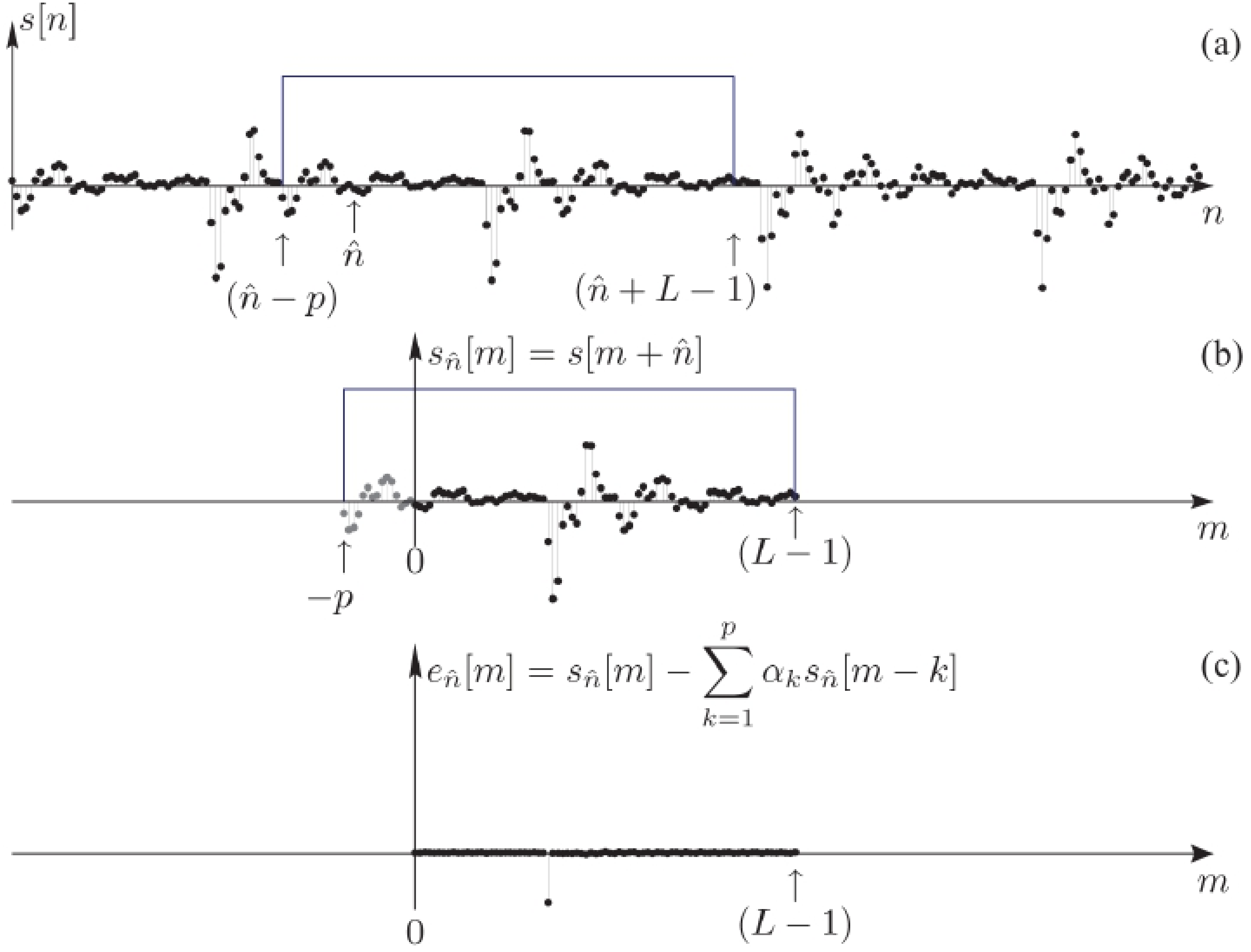
$$\phi_{\hat{n}}(i,k) = \sum_{m=-i}^{L-i-1} s_{\hat{n}}(m) s_{\hat{n}}(m+i-k), \ 1 \le i \le p, \ 0 \le k \le p$$

$$\phi_{\hat{n}}(i,k) = \sum_{m=-k}^{L-k-1} s_{\hat{n}}(m) s_{\hat{n}}(m+k-i), \ 1 \le i \le p, \ 0 \le k \le p$$

- key difference from Autocorrelation Method is that limits of summation include terms before *m* = 0 => window extends p samples backwards from $s(\hat{n} - p)$ to $s(\hat{n} + L - 1)$
- since we are extending window backwards, don't need to taper it using a HW- since there is no transition at window edges



Covariance Method



Covariance Method

- cannot use autocorrelation formulation => this is a true cross correlation
- need to solve set of equations of the form

$$\sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(i,k) = \phi_{\hat{n}}(i,0), \quad i = 1,2,\ldots,p,$$

$$E_{\hat{n}} = \phi_{\hat{n}}(0,0) - \sum_{k=1}^{p} \alpha_k \, \phi_{\hat{n}}(0,k)$$

$$\begin{bmatrix} \phi_{\hat{n}}(1,1) & \phi_{\hat{n}}(1,2) & . & . & \phi_{\hat{n}}(1,p) \\ \phi_{\hat{n}}(2,1) & \phi_{\hat{n}}(2,2) & . & . & \phi_{\hat{n}}(2,p) \\ . & . & . & . & . \\ . & . & . & . & . \\ \phi_{\hat{n}}(p,1) & \phi_{\hat{n}}(p,2) & . & . & \phi_{\hat{n}}(p,p) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ \alpha_p \end{bmatrix} = \begin{bmatrix} \phi_{\hat{n}}(1,0) \\ \phi_{\hat{n}}(2,0) \\ . \\ . \\ \phi_{\hat{n}}(p,0) \end{bmatrix}$$

$$\boldsymbol{\phi\alpha} = \psi \quad \text{or} \quad \boldsymbol{\alpha} = \boldsymbol{\phi}^{-1}\psi$$

Covariance Method

- we have $\phi_{\hat{n}}(i,k) = \phi_{\hat{n}}(k,i)$ => symmetric but not Toeplitz matrix
- all terms $\phi_{\hat{n}}(i,k)$ have a fixed number of terms contributing to the computed values (*L* terms)
- $\phi_{\hat{n}}(i,k)$ is a covariance matrix => specialized solution for $\{\alpha_k\}$ called the Covariance Method

LPC Summary

1. Speech Production Model

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n)$$

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

2. Linear Prediction Model

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

$$P(z) = \frac{\tilde{S}(z)}{S(z)} = \sum_{k=1}^{p} \alpha_k z^{-k}$$

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

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

LPC Summary

3. LPC Minimization

$$E_{\hat{n}} = \sum_m e_{\hat{n}}^2(m) = \sum_m \left[s_{\hat{n}}(m) - \tilde{s}_{\hat{n}}(m) \right]^2$$

$$= \sum_m \left[s_{\hat{n}}(m) - \sum_{k=1}^{p} \alpha_k s_{\hat{n}}(m-k) \right]^2$$

$$\frac{\partial E_{\hat{n}}}{\partial \alpha_i} = 0, \quad i = 1, 2, \ldots, p$$

$$\sum_m s_{\hat{n}}(m-i) s_{\hat{n}}(m) = \sum_{k=1}^{p} \alpha_k \sum_m s_{\hat{n}}(m-i) s_{\hat{n}}(m-k)$$

$$\phi_{\hat{n}}(i,k) = \sum_m s_{\hat{n}}(m-i) s_{\hat{n}}(m-k)$$

$$\sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(i,k) = \phi_{\hat{n}}(i,0), \qquad i = 1, 2, \ldots, p$$

$$E_{\hat{n}} = \phi_{\hat{n}}(0,0) - \sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(0,k)$$

LPC Summary

4. Autocorrelation Method

$$s_{\hat{n}}(m) = s(m+\hat{n})w(m), \quad 0 \le m \le L-1$$

$$e_{\hat{n}}(m) = s_{\hat{n}}(m) - \sum_{k=1}^{p} \alpha_k s_{\hat{n}}(m-k), \quad 0 \le m \le L-1+p$$

$s_{\hat{n}}(m)$ defined for $0 \le m \le L-1$; $e_{\hat{n}}(m)$ defined for $0 \le m \le L-1+p$

$\Rightarrow$ large errors for $0 \le m \le p-1$ and for $L \le m \le L+p-1$

$$E_{\hat{n}} = \sum_{m=0}^{L-1+p} e_{\hat{n}}^2(m)$$

$$\phi_{\hat{n}}(i,k) = R_{\hat{n}}(i-k) = \sum_{m=0}^{L-1-(i-k)} s_{\hat{n}}(m) s_{\hat{n}}(m+i-k) = R_{\hat{n}}\left(|i-k|\right)$$

$$\sum_{k=1}^{p} \alpha_k R_{\hat{n}}(|i-k|) = R_{\hat{n}}(i), \quad 1 \le i \le p$$

$$E_{\hat{n}} = R_{\hat{n}}(0) - \sum_{k=1}^{p} \alpha_k R_{\hat{n}}(k)$$

LPC Summary

4. Autocorrelation Method

- resulting matrix equation

$$\Re\alpha = r \text{ or } \alpha = \Re^{-1} r$$

$$\begin{bmatrix} R_{\hat{n}}(0) & R_{\hat{n}}(1) & . & . & R_{\hat{n}}(p-1) \\ R_{\hat{n}}(1) & R_{\hat{n}}(0) & . & . & R_{\hat{n}}(p-2) \\ . & . & . & . & . \\ . & . & . & . & . \\ R_{\hat{n}}(p-1) & R_{\hat{n}}(p-2) & . & . & R_{\hat{n}}(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_{\hat{n}}(1) \\ R_{\hat{n}}(2) \\ . \\ . \\ R_{\hat{n}}(p) \end{bmatrix}$$

- matrix equation solved using Levinsn-Durbin method

LPC Summary

5. Covariance Method

- fix interval for error signal

$$E_{\hat{n}} = \sum_{m=0}^{L-1} e_{\hat{n}}^2(m) = \sum_{m=0}^{L-1} \left[s_{\hat{n}}(m) - \sum_{k=1}^{p} \alpha_k s_{\hat{n}}(m-k) \right]^2$$

- need signal for from $s(\hat{n}-p)$ to $s(\hat{n}+L-1)$ => *L*+*p* samples

$$\sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(i,k) = \phi_{\hat{n}}(i,0),\ i = 1,2,\ldots,p$$

$$E_{\hat{n}} = \phi_{\hat{n}}(0,0) - \sum_{k=1}^{p} \alpha_k \phi_{\hat{n}}(0,k)$$

- expressed as a matrix equation

$\phi\alpha = \psi$ or $\alpha = \phi^{-1}\psi$, ϕ symmetric matrix

$$\begin{bmatrix} \phi_{\hat{n}}(1,1) & \phi_{\hat{n}}(1,2) & . & . & \phi_{\hat{n}}(1,p) \\ \phi_{\hat{n}}(2,1) & \phi_{\hat{n}}(2,2) & . & . & \phi_{\hat{n}}(2,p) \\ . & . & . & . & . \\ . & . & . & . & . \\ \phi_{\hat{n}}(p,1) & \phi_{\hat{n}}(p,2) & . & . & \phi_{\hat{n}}(p,p) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ \alpha_p \end{bmatrix} = \begin{bmatrix} \phi_{\hat{n}}(1,0) \\ \phi_{\hat{n}}(2,0) \\ . \\ . \\ \phi_{\hat{n}}(p,0) \end{bmatrix}$$

Frequency Domain Interpretations of Linear Predictive Analysis

The Resulting LPC Model

- The final LPC model consists of the LPC parameters, $\{\alpha_k\}$, k=1,2,…,p, and the gain, G, which together define the system function

$$\tilde{H}(z) = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}}$$

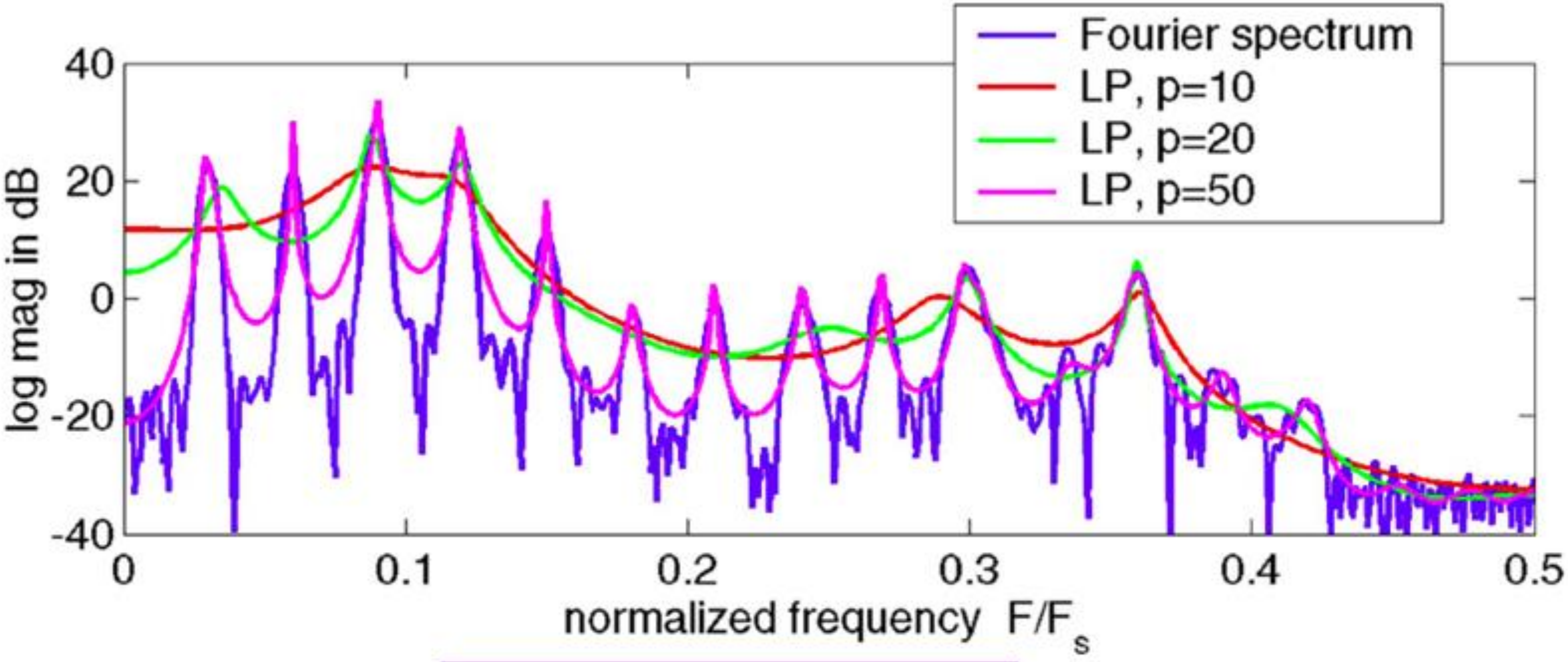
with frequency response

$$\tilde{H}(e^{j\omega}) = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k e^{-j\omega k}} = \frac{G}{A(e^{j\omega})}$$

with the gain determined by matching the energy of the model to the short-time energy of the speech signal, i.e.,

$$G^2 = E_{\hat{n}} = \sum_{m} \left(e_{\hat{n}}(m)\right)^2 = R_{\hat{n}}(0) - \sum_{k=1}^{p} \alpha_k R_{\hat{n}}(k)$$

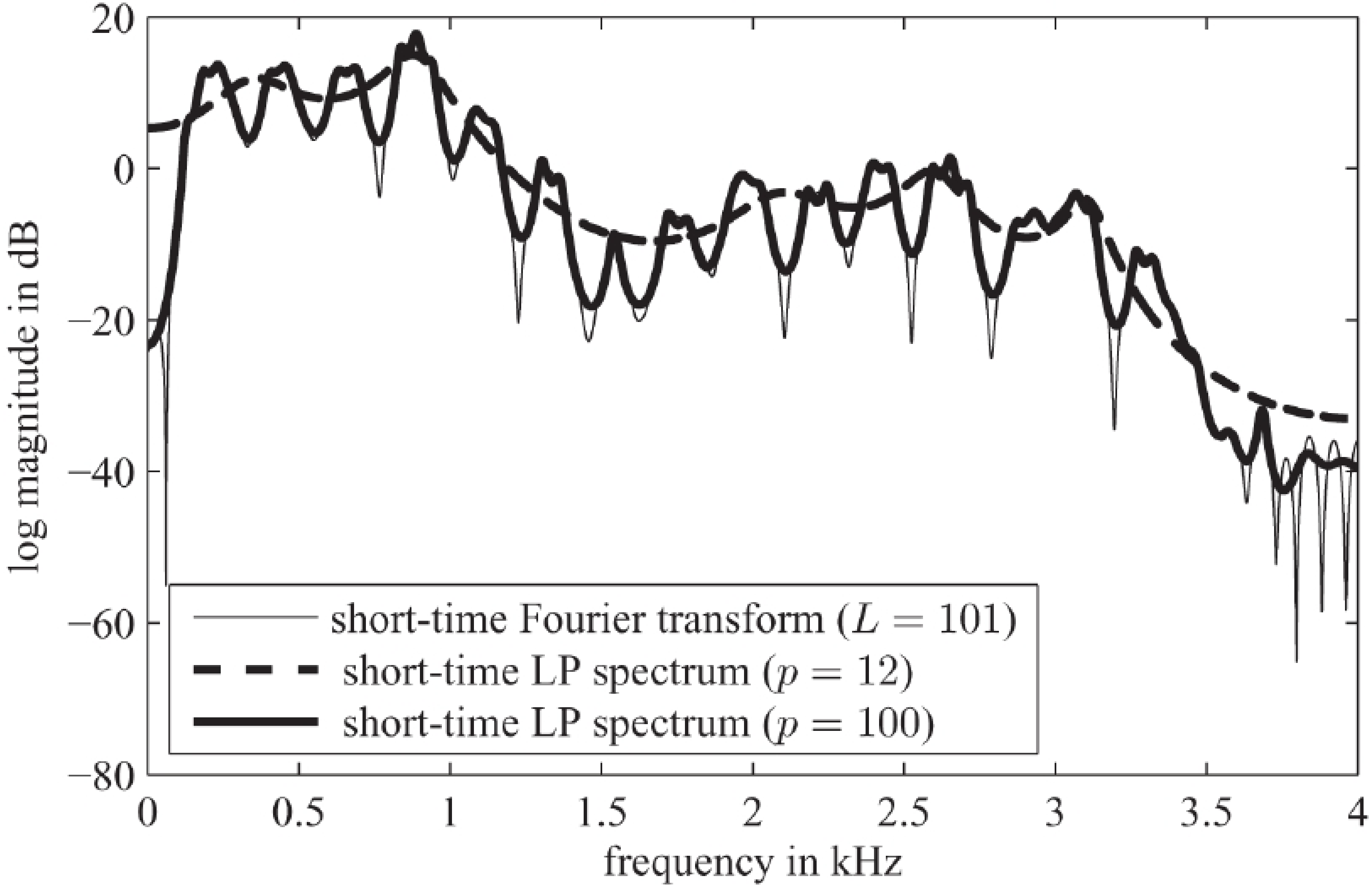
LPC Spectrum



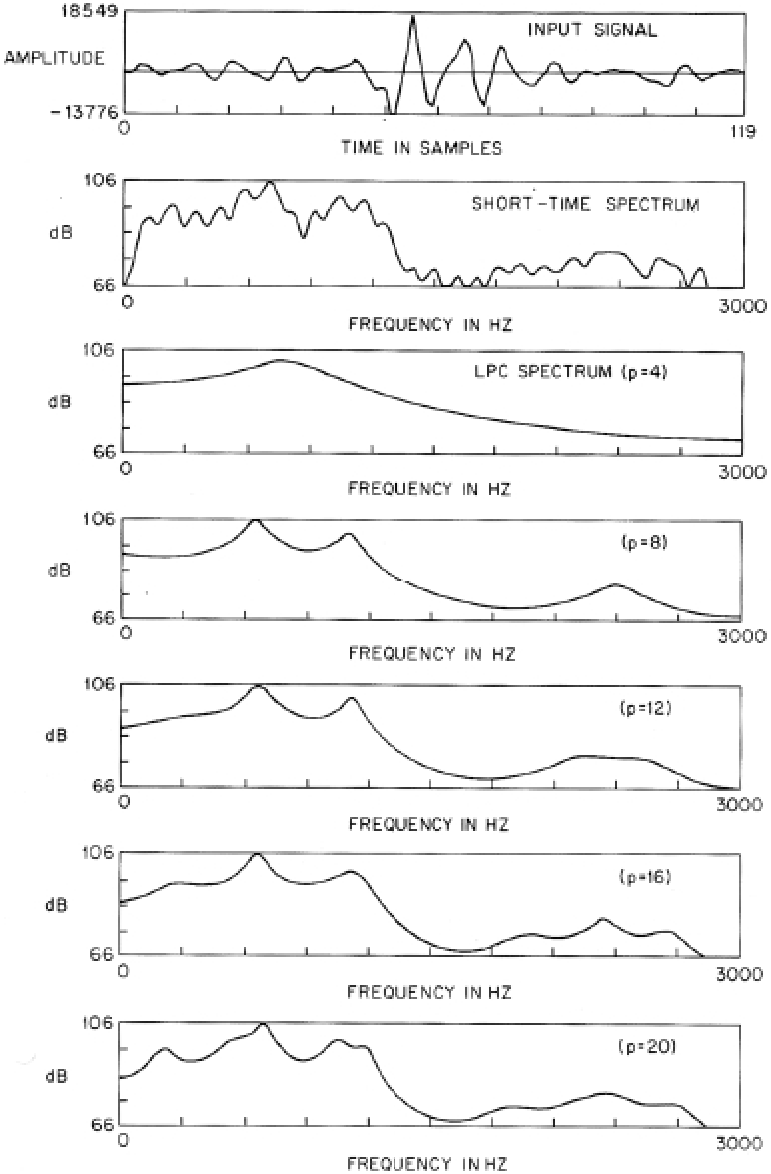
$$\tilde{H}(e^{j\omega}) = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k e^{-j\omega k}}$$

LP Analysis is seen to be a method of short-time spectrum estimation with removal of excitation fine structure (a form of wideband spectrum analysis)

Effects of Model Order



Effects of Model Order



- plots show Fourier transform of segment and LP spectra for various orders
  - as *p* increases, more details of the spectrum are preserved
  - need to choose a value of *p* that represents the spectral effects of the glottal pulse, vocal tract and radiation--nothing else

Linear Prediction Spectrogram

- Speech spectrogram previously defined as:

$$20\log|S_r[k]| = 20\log|\sum_{m=0}^{L-1} s[rR+m]w[m]e^{-j(2\pi/N)km}|$$

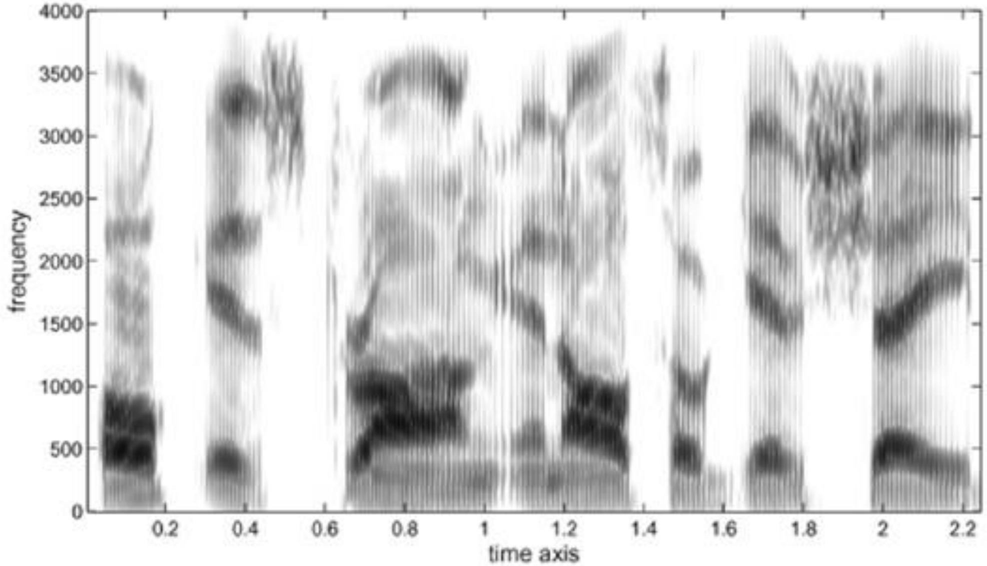
for set of times, $t_r = rRT$, and set of frequencies, $F_k = kF_S / N, \quad k = 1, 2, \dots, N/2$ where *R* is the time shift (in samples) between adjacent STFTS, *T* is the sampling period, F_S = 1 / *T* is the sampling frequency, and *N* is the size of the discrete Fourier transform used to computed each STFT estimate.

- Similarly we can define the LP spectrogram as an image plot of:

$$20\log|\tilde{H}_r[k]| = 20\log\left|\frac{G_r}{A_r(e^{j(2\pi/N)k})}\right|$$

where G_r and $A_r(e^{j(2\pi/N)k})$ are the gain and prediction error polynomial at analysis time *rR*.

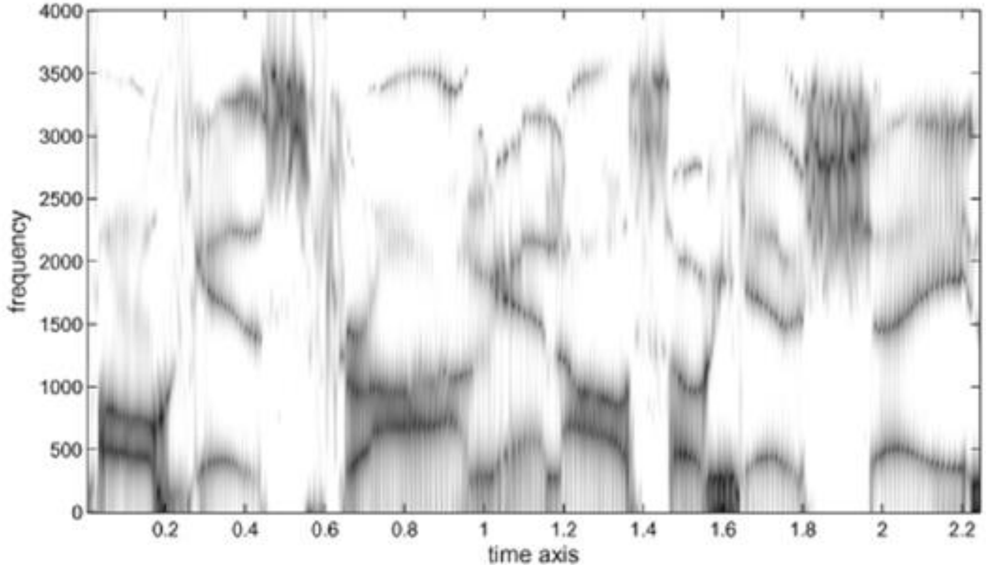
Linear Prediction Spectrogram



Wideband Fourier spectrogram

(L=81, R=3, N=1000,

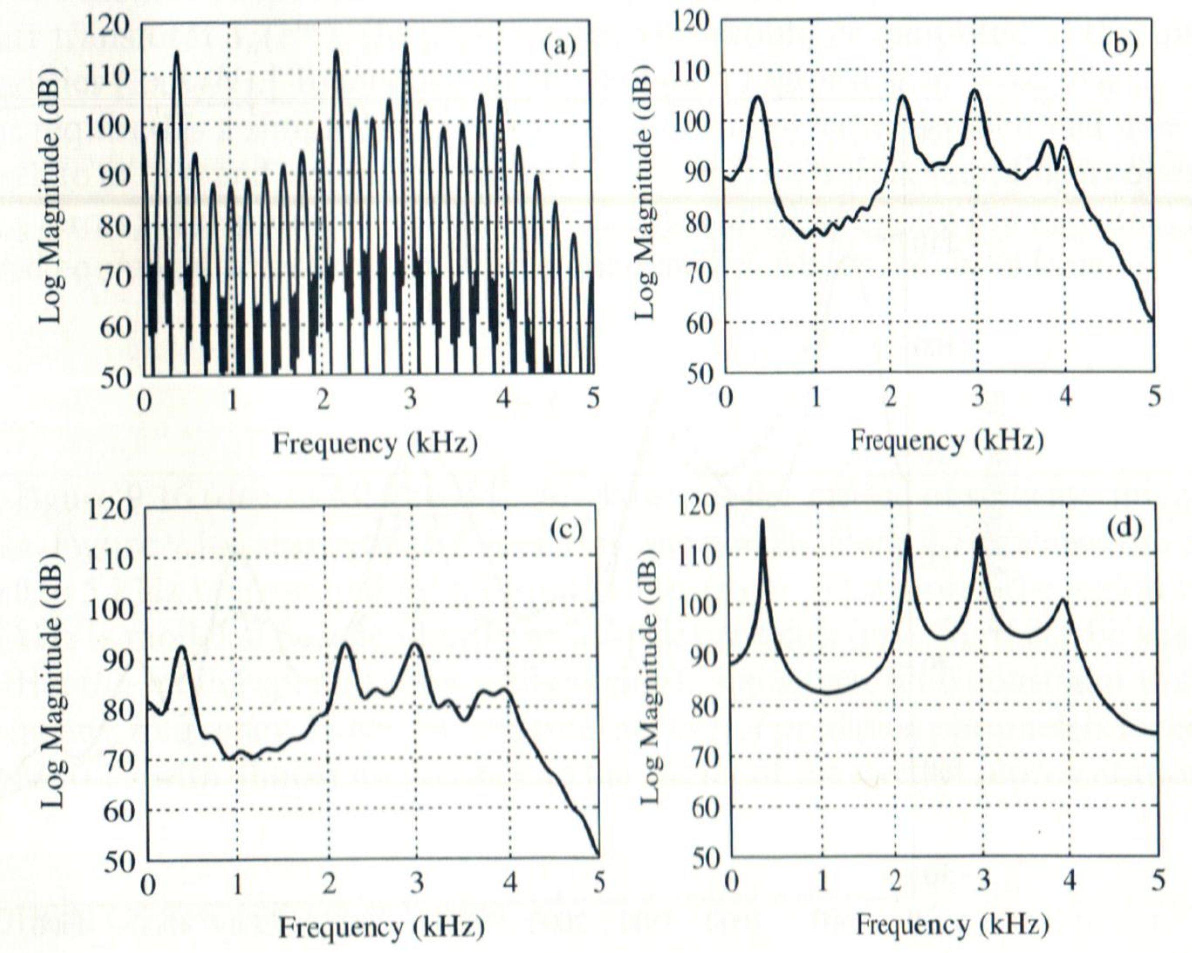
40 db dynamic range)



Linear predictive spectrogram

(p=12)

Comparison to Other Spectrum Analysis Methods



Spectra of synthetic vowel /IY/

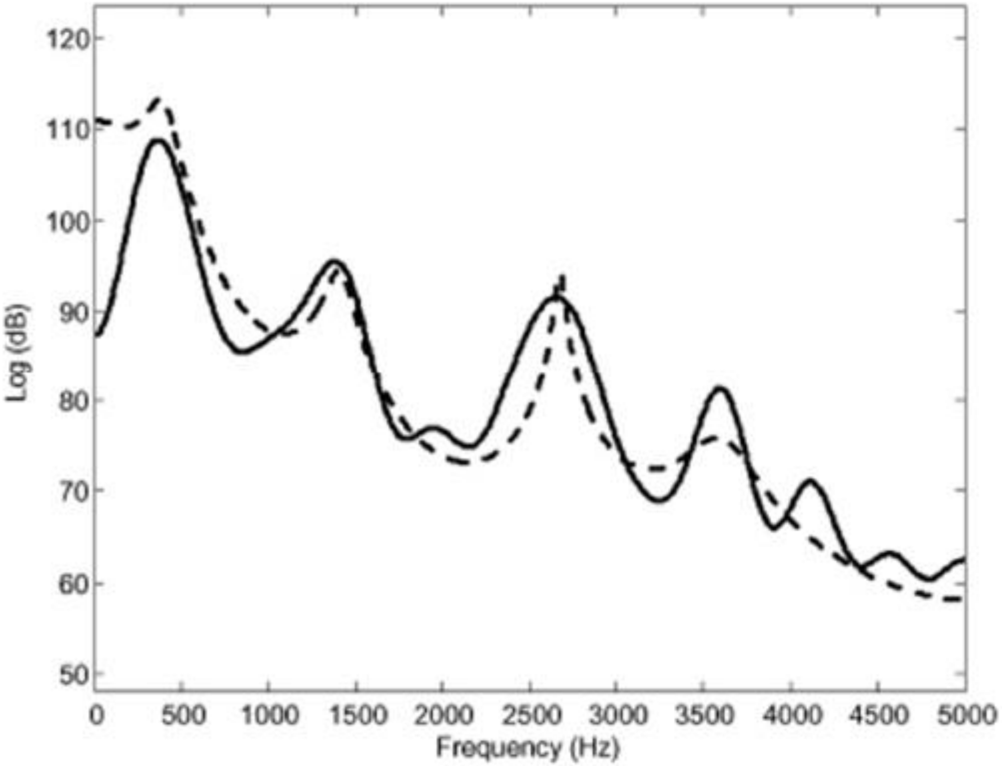
(a) Narrowband spectrum using 40 msec window

(b) Wideband spectrum using a 10 msec window

(c) Cepstrally smoothed spectrum

(d) LPC spectrum from a 40 msec section using a *p*=12 order LPC analysis

Comparison to Other Spectrum Analysis Methods



- Natural speech spectral estimates using cepstral smoothing (solid line) and linear prediction analysis (dashed line).
- Note the fewer (spurious) peaks in the LP analysis spectrum since LP used *p*=12 which restricted the spectral match to a maximum of 6 resonance peaks.
- Note the narrow bandwidths of the LP resonances versus the cepstrally smoothed resonances.

Solutions of LPC Equations

Autocorrelation Method

(Levinson-Durbin Algorithm)

Levinson-Durbin Algorithm 1

- Autocorrelation equations (at each frame$\hat{n}$)

$$\sum_{k=1}^{p} \alpha_k R[|i-k|] = R[i] \quad 1 \le i \le p$$

$$\mathbf{R\alpha = r}$$

- *R* is a positive definite symmetric Toeplitz matrix
- The set of optimum predictor coefficients satisfy

$$R[i] - \sum_{k=1}^{p} \alpha_k R[|i-k|] = 0, \quad 1 \le i \le p$$

- with minimum mean-squared prediction error of

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

Levinson-Durbin Algorithm 2

- By combining the last two equations we get a larger matrix equation of the form:

$$\underbrace{\begin{bmatrix} R[0] & R[1] & R[2] & \dots & R[p] \\ R[1] & R[0] & R[1] & \dots & R[p-1] \\ R[2] & R[1] & R[0] & \dots & R[p-2] \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ R[p] & R[p-1] & R[p-2] & \dots & R[0] \end{bmatrix}}_{\boldsymbol{R}^{(p)}} \underbrace{\begin{bmatrix} 1 \\ -\alpha_1^{(p)} \\ -\alpha_2^{(p)} \\ \cdot \\ -\alpha_p^{(p)} \end{bmatrix}}_{\boldsymbol{a}^{(p)}} = \begin{bmatrix} E^{(p)} \\ 0 \\ 0 \\ \cdot \\ 0 \end{bmatrix} = \boldsymbol{E}^{(p)}$$

- expanded (*p*+1)x(*p*+1) matrix is still Toeplitz and can be solved iteratively by incorporating new correlation value at each iteration and solving for higher order predictor in terms of new correlation value and previous predictor

Levinson-Durbin Algorithm 3

- Show how *i*-th order solution can be derived from (*i*-1)-st order solution; i.e., given $\alpha^{(i-1)}$ the solution to $R^{(i-1)}\alpha^{(i-1)} = E^{(i-1)}$ we derive solution to $R^{(i)}\alpha^{(i)} = E^{(i)}$
- The (*i*-1)-st solution can be expressed as

$$\begin{bmatrix} R[0] & R[1] & R[2] & \dots & R[i-1] \\ R[1] & R[0] & R[1] & \dots & R[i-2] \\ R[2] & R[1] & R[0] & \dots & R[i-3] \\ . & . & . & . & . \\ R[i-1] & R[i-2] & R[i-3] & \dots & R[0] \end{bmatrix} \begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ . \\ -\alpha_{i-1}^{(i-1)} \end{bmatrix} = \begin{bmatrix} E^{(i-1)} \\ 0 \\ 0 \\ . \\ 0 \end{bmatrix}$$

Levinson-Durbin Algorithm 4

- Appending a 0 to vector $\alpha^{(i-1)}$ and multiplying by the matrix $R^{(i)}$ gives a new set of (*i*+1) equations of the form:

$$\begin{bmatrix} R[0] & R[1] & R[2] & \dots & R[i] \\ R[1] & R[0] & R[1] & \dots & R[i-1] \\ R[2] & R[1] & R[0] & \dots & R[i-2] \\ . & . & . & \dots & . \\ R[i-1] & R[i-2] & R[i-3] & \dots & R[1] \\ R[i] & R[i-1] & R[i-2] & \dots & R[0] \end{bmatrix} \begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ . \\ -\alpha_{i-1}^{(i-1)} \\ 0 \end{bmatrix} = \begin{bmatrix} E^{(i-1)} \\ 0 \\ 0 \\ . \\ 0 \\ \gamma^{(i-1)} \end{bmatrix}$$

- where $\gamma^{(i-1)} = R[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R[i-j]$ and *R*[*i*] are introduced

Levinson-Durbin Algorithm 5

- Key step is that since Toeplitz matrix has special symmetry we can reverse the order of the equations (first equation last, last equation first), giving:

$$\begin{bmatrix} R[0] & R[1] & R[2] & \dots & R[i] \\ R[1] & R[0] & R[1] & \dots & R[i-1] \\ R[2] & R[1] & R[0] & \dots & R[i-2] \\ . & . & . & \dots & . \\ R[i-1] & R[i-2] & R[i-3] & \dots & R[1] \\ R[i] & R[i-1] & R[i-2] & \dots & R[0] \end{bmatrix} \begin{bmatrix} 0 \\ -\alpha_{i-1}^{(i-1)} \\ -\alpha_{i-2}^{(i-1)} \\ . \\ -\alpha_{1}^{(i-1)} \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma^{(i-1)} \\ 0 \\ 0 \\ . \\ 0 \\ E^{(i-1)} \end{bmatrix}$$

Levinson-Durbin Algorithm 6

- To get the equation into the desired form (a single component in the vector $E^{(i)}$) we combine the two sets of matrices (with a multiplicative factor k_i) giving:

$$R^{(i)}\left[\begin{bmatrix}1\\-\alpha_1^{(i-1)}\\-\alpha_2^{(i-1)}\\.\\-\alpha_{i-1}^{(i-1)}\\0\end{bmatrix}-k_i\begin{bmatrix}0\\-\alpha_{i-1}^{(i-1)}\\-\alpha_{i-2}^{(i-1)}\\.\\-\alpha_1^{(i-1)}\\1\end{bmatrix}\right]=\left[\begin{bmatrix}E^{(i-1)}\\0\\0\\.\\0\\\gamma^{(i-1)}\end{bmatrix}-k_i\begin{bmatrix}\gamma^{(i-1)}\\0\\0\\.\\0\\E^{(i-1)}\end{bmatrix}\right]$$

- Choose k_i so that vector on right has only a single non-zero entry, i.e.,

$$k_i=\frac{\gamma^{(i-1)}}{E^{(i-1)}}=\frac{R[i]-\sum_{j=1}^{i-1}\alpha_j^{(i-1)}R[i-j]}{E^{(i-1)}}$$

Levinson-Durbin Algorithm 7

- The first element of the right hand side vector is now:

$$E^{(i)} = E^{(i-1)} - k_i \gamma^{(i-1)} = E^{(i-1)}(1-k_i^2)$$

- The k_i parameters are called PARCOR (partial correlation) coefficients
- With this choice of k_i, the vector of *i*-th order predictor coefficients is:

$$\begin{bmatrix} 1 \\ -\alpha_1^{(i)} \\ -\alpha_2^{(i)} \\ . \\ -\alpha_{i-1}^{(i)} \\ -\alpha_i^{(i)} \end{bmatrix} = \begin{bmatrix} 1 \\ -\alpha_1^{(i-1)} \\ -\alpha_2^{(i-1)} \\ . \\ -\alpha_{i-1}^{(i-1)} \\ 0 \end{bmatrix} - k_i \begin{bmatrix} 0 \\ -\alpha_{i-1}^{(i-1)} \\ -\alpha_{i-2}^{(i-1)} \\ . \\ -\alpha_1^{(i-1)} \\ 1 \end{bmatrix}$$

- yielding the updating procedure

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}, \quad j = 1, 2, \ldots, i$$

$$\alpha_i^{(i)} = k_i$$

Levinson-Durbin Algorithm 8

- The final solution for order *p* is:

$$\alpha_j = \alpha_j^{(p)} \qquad 1 \le j \le p$$

- with prediction error

$$E^{(p)} = E[0]\prod_{m=1}^{p}(1-k_m^2) = R[0]\prod_{m=1}^{p}(1-k_m^2)$$

- If we use normalized autocorrelation coefficients:

$$r[k] = R[k] / R[0]$$

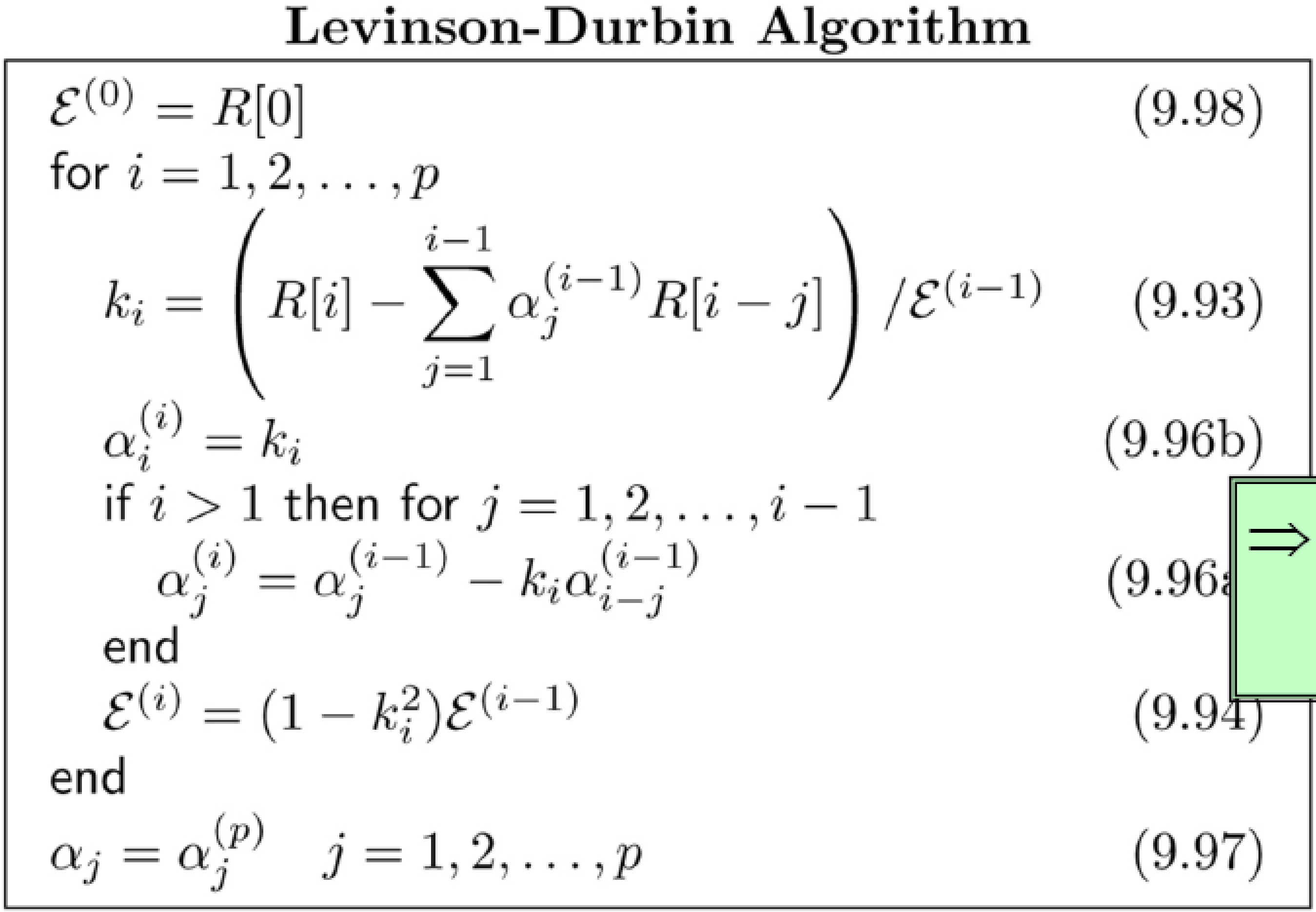
- we get normalized errors of the form:

$$\nu^{(i)} = \frac{E^{(i)}}{R[0]} = 1 - \sum_{k=1}^{i}\alpha_k^{(i)} r[k] = \prod_{m=1}^{i}(1-k_m^2)$$

where

$$0 < \nu^{(i)} \le 1 \text{ or } -1 < k_i < 1$$

Levinson-Durbin Algorithm



Levinson-Durbin Algorithm

```
$\mathcal{E}^{(0)} = R[0]$                                                      (9.98)
for $i = 1, 2, \ldots, p$
    $k_i = \left( R[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R[i-j] \right) / \mathcal{E}^{(i-1)}$      (9.93)
    $\alpha_i^{(i)} = k_i$                                                      (9.96b)
    if $i > 1$ then for $j = 1, 2, \ldots, i-1$
        $\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$                        (9.96a)
    end
    $\mathcal{E}^{(i)} = (1 - k_i^2)\mathcal{E}^{(i-1)}$                                   (9.94)
end
$\alpha_j = \alpha_j^{(p)} \quad j = 1, 2, \ldots, p$                                  (9.97)
```

$$\Rightarrow A^{(i)}(z) = A^{(i-1)}(z) - k_i z^{-i} A^{(i-1)}(z^{-1})$$

Autocorrelation Example

- consider a simple *p* = 2 solution of the form

$$\begin{bmatrix} R(0) & R(1) \\ R(1) & R(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \end{bmatrix}$$

- with solution

$$E^{(0)} = R(0)$$

$$k_1 = R(1)/R(0)$$

$$\alpha_1^{(1)} = R(1)/R(0)$$

$$E^{(1)} = \frac{R^2(0) - R^2(1)}{R(0)}$$

Autocorrelation Example

$$k_2 = \frac{R(2)R(0) - R^2(1)}{R^2(0) - R^2(1)}$$

$$\alpha_2^{(2)} = \frac{R(2)R(0) - R^2(1)}{R^2(0) - R^2(1)}$$

$$\alpha_1^{(2)} = \frac{R(1)R(0) - R(1)R(2)}{R^2(0) - R^2(1)}$$

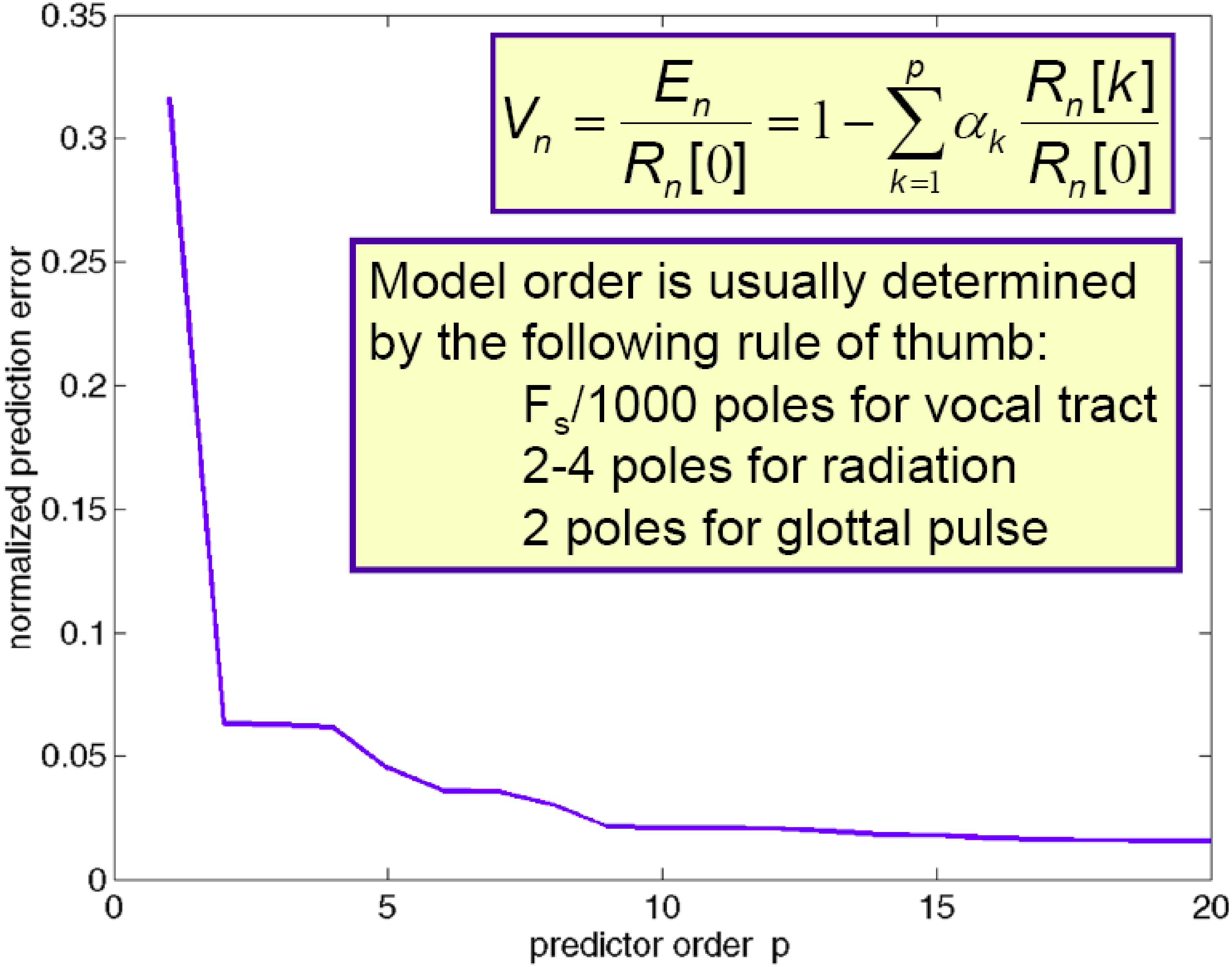
- with final coefficients

$$\alpha_1 = \alpha_1^{(2)}$$

$$\alpha_2 = \alpha_2^{(2)}$$

$E^{(i)}$ = prediction error for predictor of order i

Prediction Error as a Function of *p*



$$V_n = \frac{E_n}{R_n[0]} = 1 - \sum_{k=1}^{p} \alpha_k \frac{R_n[k]}{R_n[0]}$$

Model order is usually determined by the following rule of thumb:

F_s/1000 poles for vocal tract
2-4 poles for radiation
2 poles for glottal pulse

Autocorrelation Method Properties

- mean-squared prediction error always non-zero
  - decreases monotonically with increasing model order
- autocorrelation matching property
  - model and data match up to order *p*
- spectrum matching property
  - favors peaks of short-time FT
- minimum-phase property
  - zeros of *A(z)* are inside the unit circle
- Levinson-Durbin recursion
  - efficient algorithm for finding prediction coefficients
  - PARCOR coefficients and MSE are by-products