Chapter 4

Linear predictive analysis of speech

Basanta Joshi, PhD
basanta@ioe.edu.np

Lecture notes can be downloaded from
www.basantajoshi.com.np

Contents

4.1. Basic Principles of linear predictive analysis
4.2. Auto correlation method
4.3. Covariance method
4.4. Solution of LPC equations
4.5. Cholesky method
4.6. Durbin's Recursive algorithm
4.7. Application of LPC parameters
4.7.1. Pitch detection using LPC parameters
4.7.2. Formant analysis
4.7.3. VELP
4.7.4. CELP

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 <u>linear</u> combination of past samples, i.e.,

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

LPC Methods

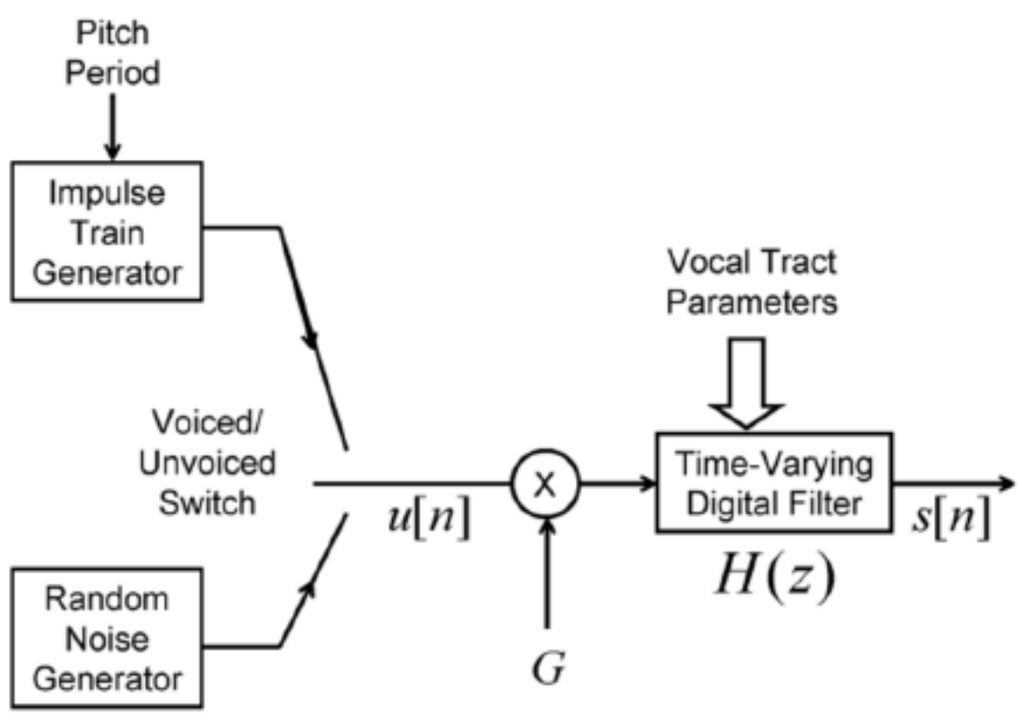
- for periodic signals with period N_p, it is obvious that

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

  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***

LPC Methods

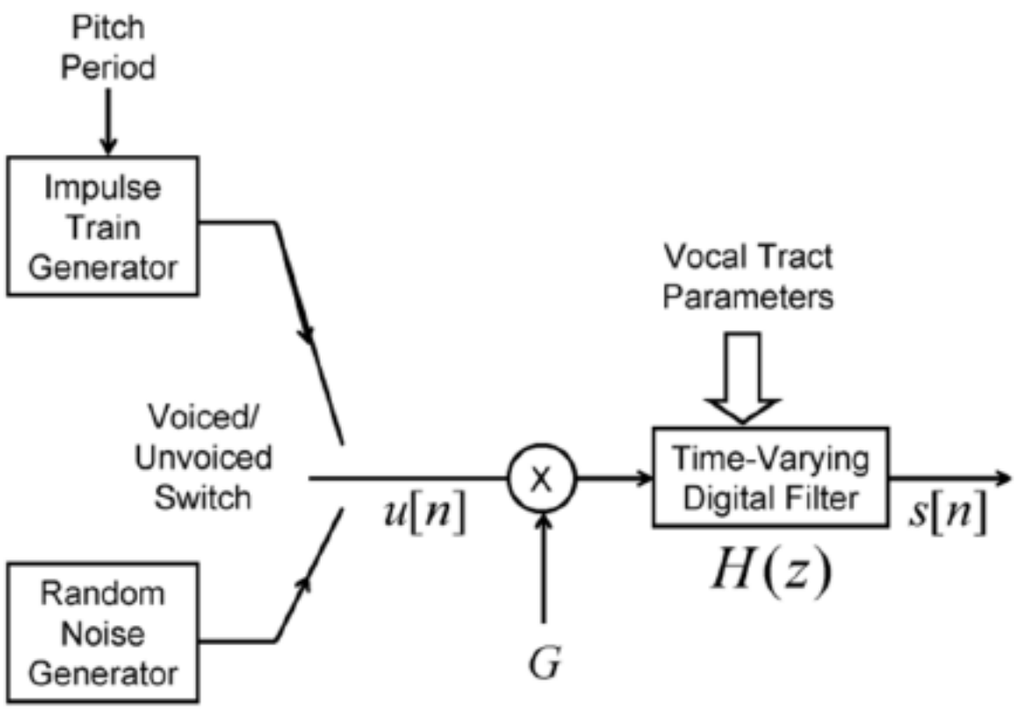
- LP is based on speech production and synthesis models
  - speech can be modeled as the output of a linear, time-varying system, excited by either quasi-periodic pulses or noise;
  - assume that the model parameters remain constant over speech analysis interval
- LP provides a ***robust, reliable and accurate method*** for estimating the parameters of the linear system (the combined vocal tract, glottal pulse, and radiation characteristic for voiced speech)



LPC Methods

- LP methods have been used in control and information theory—called methods of system estimation and system identification
  - used extensively in speech under group of names including
    1. covariance method
    2. autocorrelation method
    3. lattice method
    4. inverse filter formulation
    5. spectral estimation formulation
    6. maximum likelihood method
    7. inner product method

Basic Principles of LP

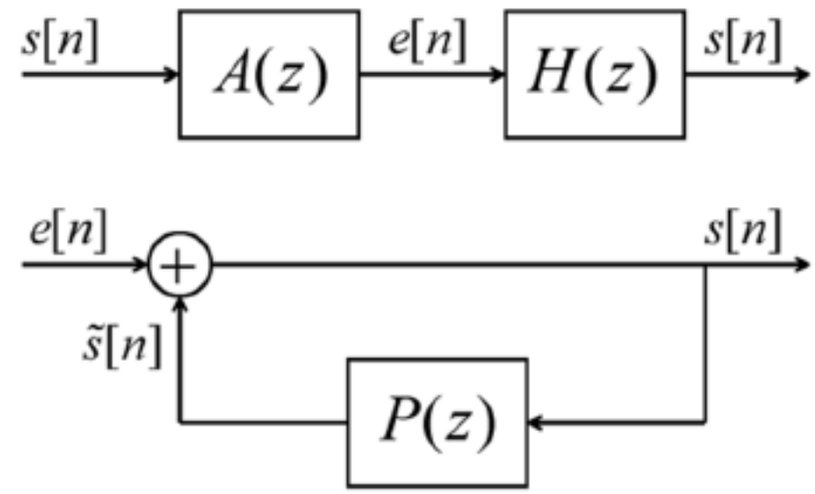


$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + G\,u(n)$$

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

- 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

LP Basic Equations



- a pth 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

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

- if the speech signal obeys the production model exactly, and if $\alpha_k = a_k, 1 \le k \le p$

$\Rightarrow e(n) = Gu(n)$ and <u>$A(z)$ is an inverse filter for $H(z)$</u>, i.e.,

$$H(z) = \frac{1}{A(z)}$$

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 <u>short segments of speech</u>
- need to <u>minimize mean-squared prediction error</u> over short segments of speech
- resulting $\{\alpha_k\}$ assumed to be the actual $\{a_k\}$ in the speech production model

=> intend to show that all of this can be done ***efficiently, reliably, and accurately*** for speech

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 $\{\alpha_k\}$

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

- defining

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

- we get

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

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

Solution for $\{\alpha_k\}$

- 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:

1. compute $\phi_{\hat{n}}(i,k)$ for $1 \le i \le p$, $0 \le k \le p$
2. solve matrix equation for α_k

- need to specify <u>range of m</u> to compute $\phi_{\hat{n}}(i,k)$
- need to specify $s_{\hat{n}}(m)$

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

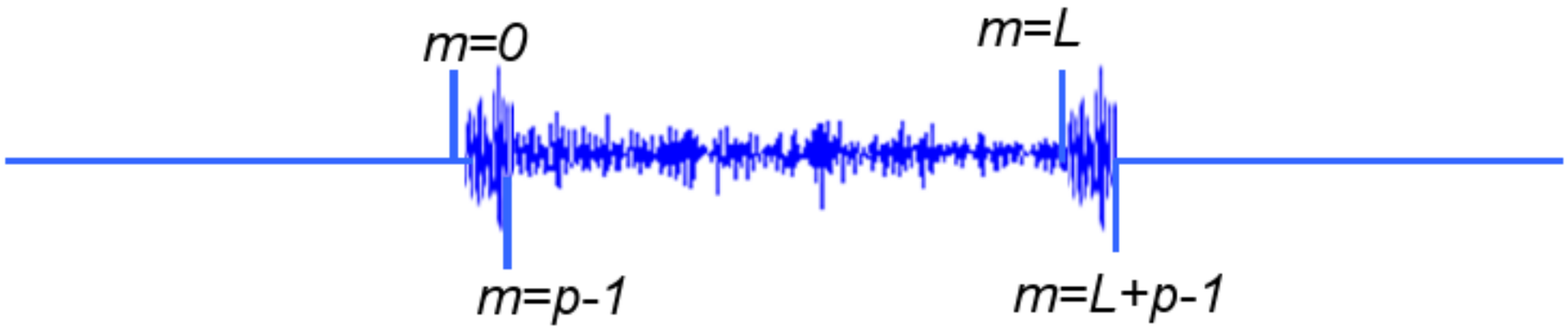
$$e_{\hat{n}}(m) = s_{\hat{n}}(m) - \sum_{k=1}^{p} \alpha_k s_{\hat{n}}(m-k)$$

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

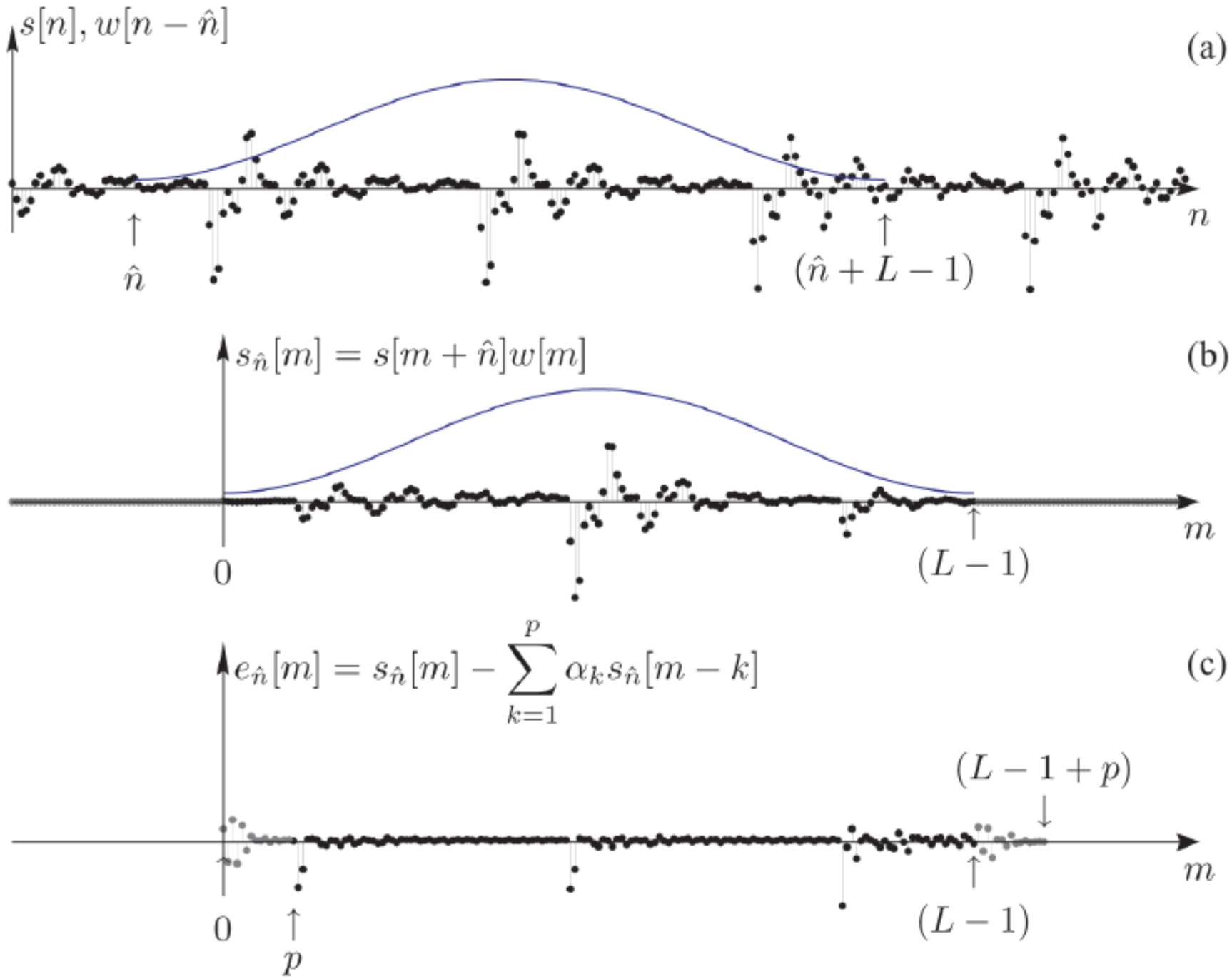
$$E_{\hat{n}} = \sum_{m=-\infty}^{\infty} e_{\hat{n}}^2(m) = \sum_{m=0}^{L-1+p} e_{\hat{n}}^2(m)$$

- 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., $m = L, L+1, \ldots, L+p-1$) we are predicting zero-valued samples (outside window range) from non-zero samples => $e_{\hat{n}}(m)$ will be (relatively) large

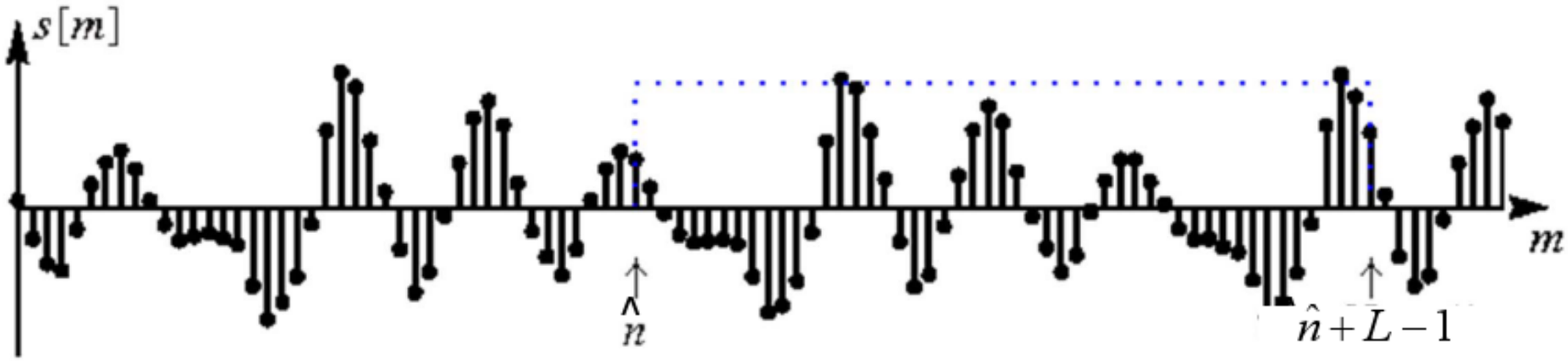
• for these reasons, normally use windows that taper the segment to zero (e.g., Hamming window)

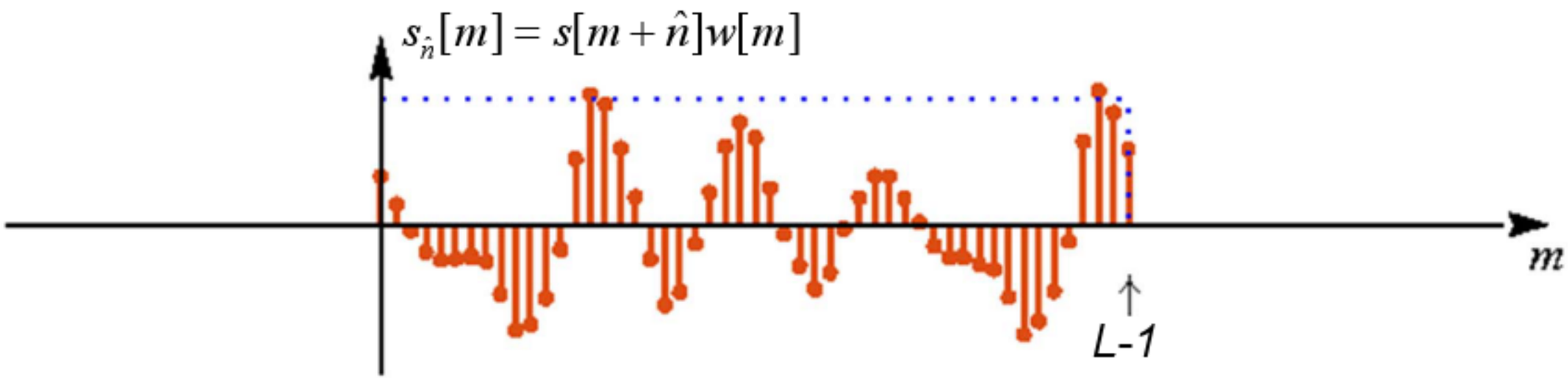


Autocorrelation Method



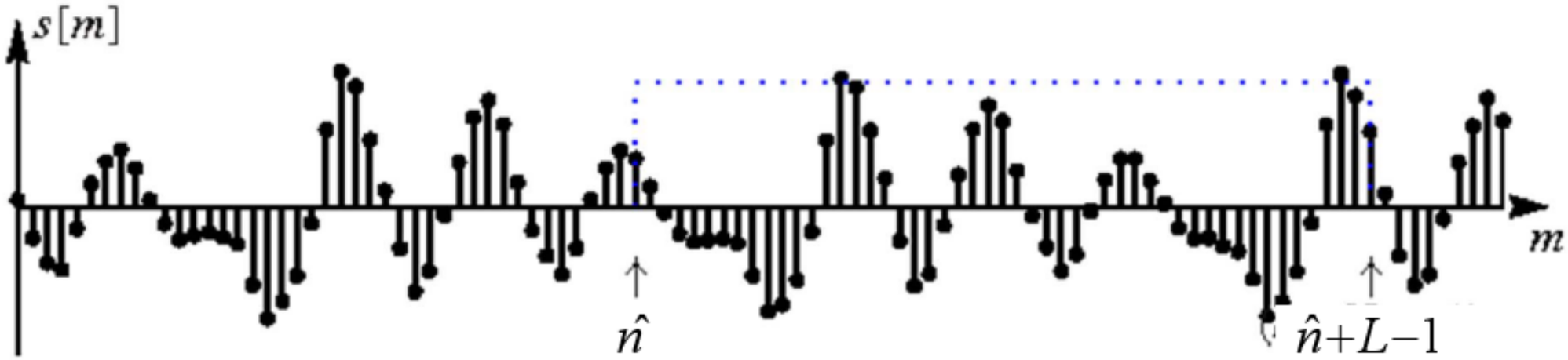
The "Autocorrelation Method"



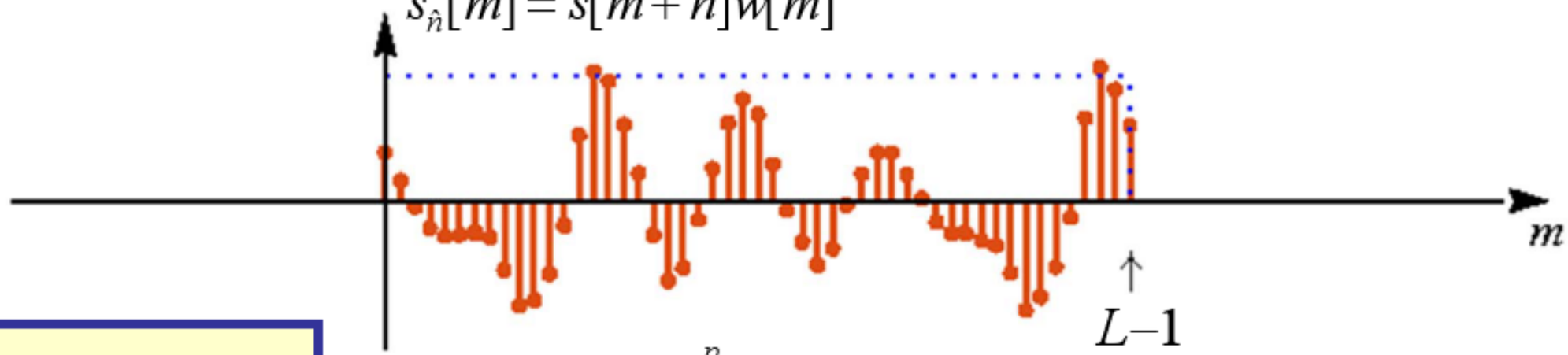


$$R_{\hat{n}}[k] = \sum_{m=0}^{L-1-k} s_{\hat{n}}[m]s_{\hat{n}}[m+k] \quad k = 1,2,\ldots,p$$

The "Autocorrelation Method"

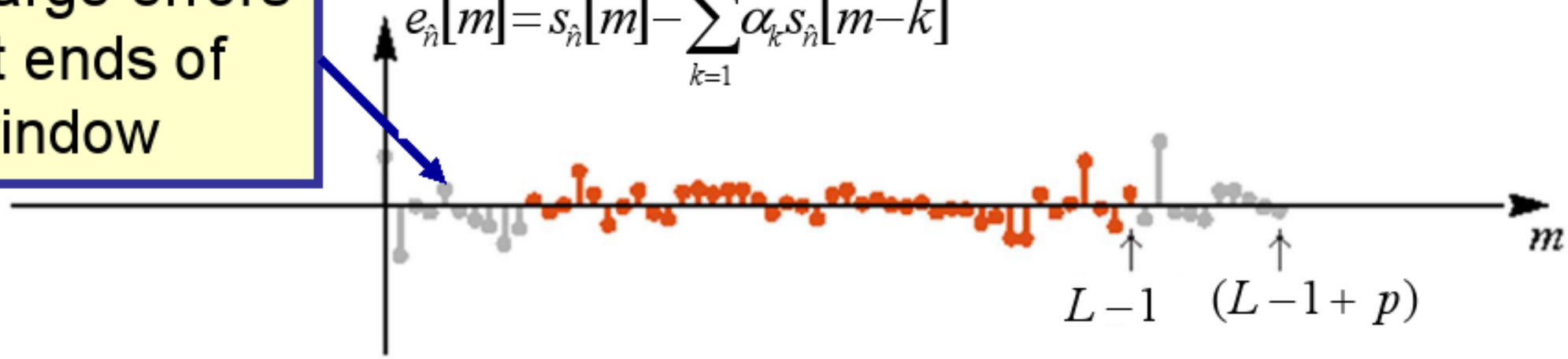


$$s_{\hat{n}}[m] = s[m+\hat{n}]w[m]$$



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

Large errors at ends of window



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

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

- there are $L-|i-k|$ non-zero terms in the computation of $\phi_{\hat{n}}(i,k)$ for each value of i and k; 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 short-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 $p \times p$ 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**):

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

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

Covariance Method

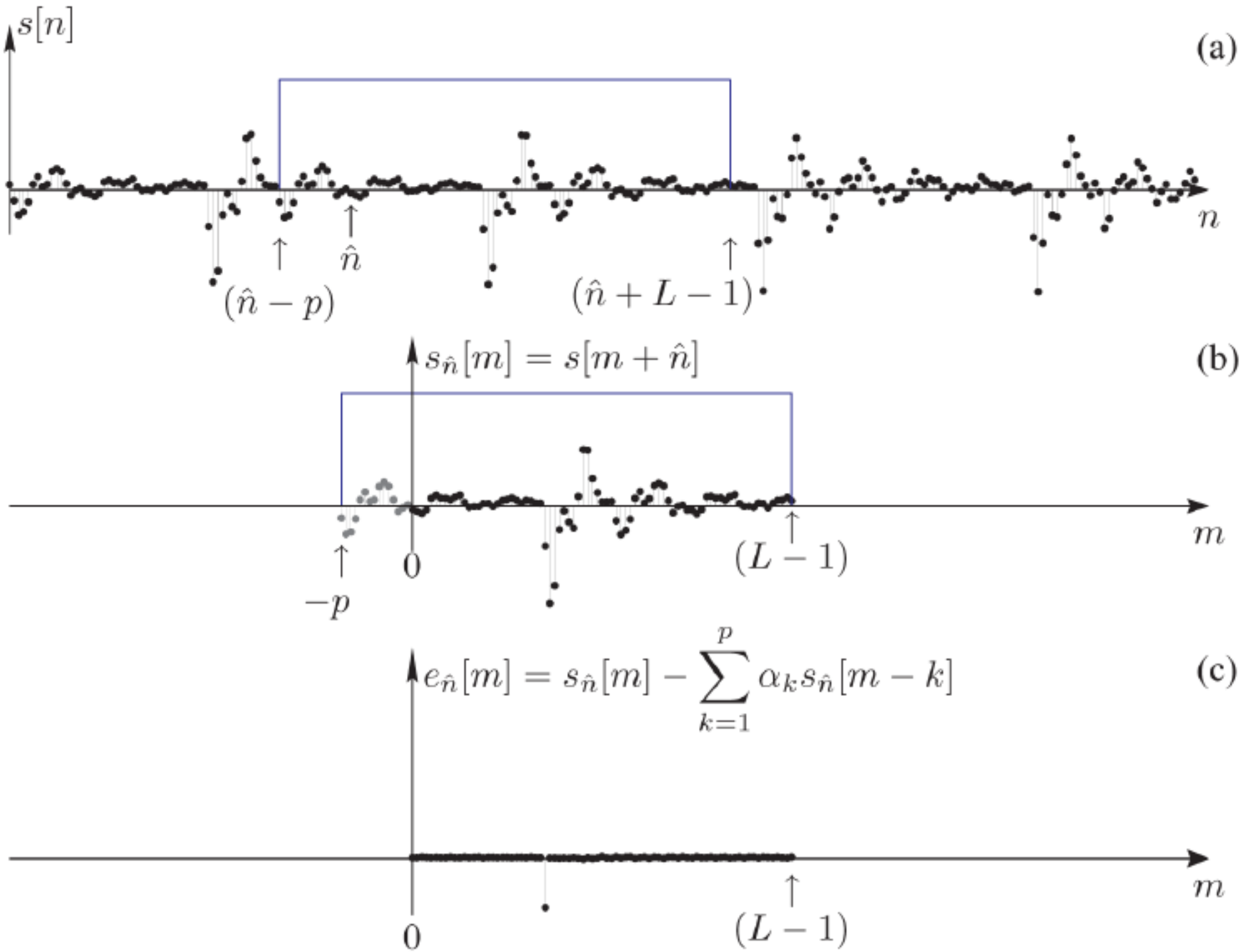
- changing the summation index gives

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

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

- 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

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

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

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

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

Covariance Method

- we have $\phi_{\hat{n}}(i,k) = \phi_{\hat{n}}(k,i)$ => symmetric but not Toeplitz matrix whose diagonal elements are related as

$$\phi_{\hat{n}}(i+1,k+1) = \phi_{\hat{n}}(i,k) + s_{\hat{n}}(-i-1)s_{\hat{n}}(-k-1) - s_{\hat{n}}(L-1-i)s_{\hat{n}}(L-1-k)$$

$$\phi_{\hat{n}}(2,2) = \phi_{\hat{n}}(1,1) + s_{\hat{n}}(-2)s_{\hat{n}}(-2) - s_{\hat{n}}(L-2)s_{\hat{n}}(L-2)$$

- 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

Summary of LP

- use p^{th} order linear predictor to predict $s(\hat{n})$ from p previous samples
- minimize mean-squared error, $E_{\hat{n}}$, over analysis window of duration L-samples
- solution for optimum predictor coefficients, $\{\alpha_k\}$, is based on solving a matrix equation

  => two solutions have evolved

  - ***autocorrelation method*** => signal is windowed by a tapering window in order to minimize discontinuities at beginning (predicting speech from zero-valued samples) and end (predicting zero-valued samples from speech samples) of the interval; the matrix $\phi_{\hat{n}}(i,k)$ is shown to be an autocorrelation function; the resulting autocorrelation matrix is Toeplitz and can be readily solved using standard matrix solutions
  - ***covariance method*** => the signal is extended by p samples outside the normal range of $0 \le m \le L-1$ to include p samples occurring prior to $m = 0$; this eliminates large errors in computing the signal from values prior to $m = 0$ (they are available) and eliminates the need for a tapering window; resulting matrix of correlations is symmetric but not Toeplitz => different method of solution with somewhat different set of optimal prediction coefficients, $\{\alpha_k\}$

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 $L \le m \le L+p-1$

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

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

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

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

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 Levinson or 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 $s(\hat{n}+L-1) \Rightarrow L+p$ 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

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

Computation of Model Gain

- it is reasonable to expect the model gain, G, to be determined by matching the signal energy with the energy of the linearly predicted samples
- from the basic model equations we have

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

- whereas for the prediction error we have

$$e(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) \Rightarrow \text{best fit to model}$$

- when $\alpha_k = a_k$ (i.e., perfect match to model), then

$$e(n) = Gu(n)$$

- since it is virtually impossible to guarantee that $\alpha_k = a_k$, cannot use this simple matching property for determining the gain; instead use energy matching criterion (energy in error signal=energy in excitation)

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

Gain Assumptions

- assumptions about excitation to solve for G
  - ***voiced speech***--$u(n) = \delta(n) \Rightarrow$ L order of a single pitch period; predictor order, p, large enough to model glottal pulse shape, vocal tract IR, and radiation
  - ***unvoiced speech***--$u(n)$-zero mean, unity variance, stationary white noise process

The Resulting LPC Model

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

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

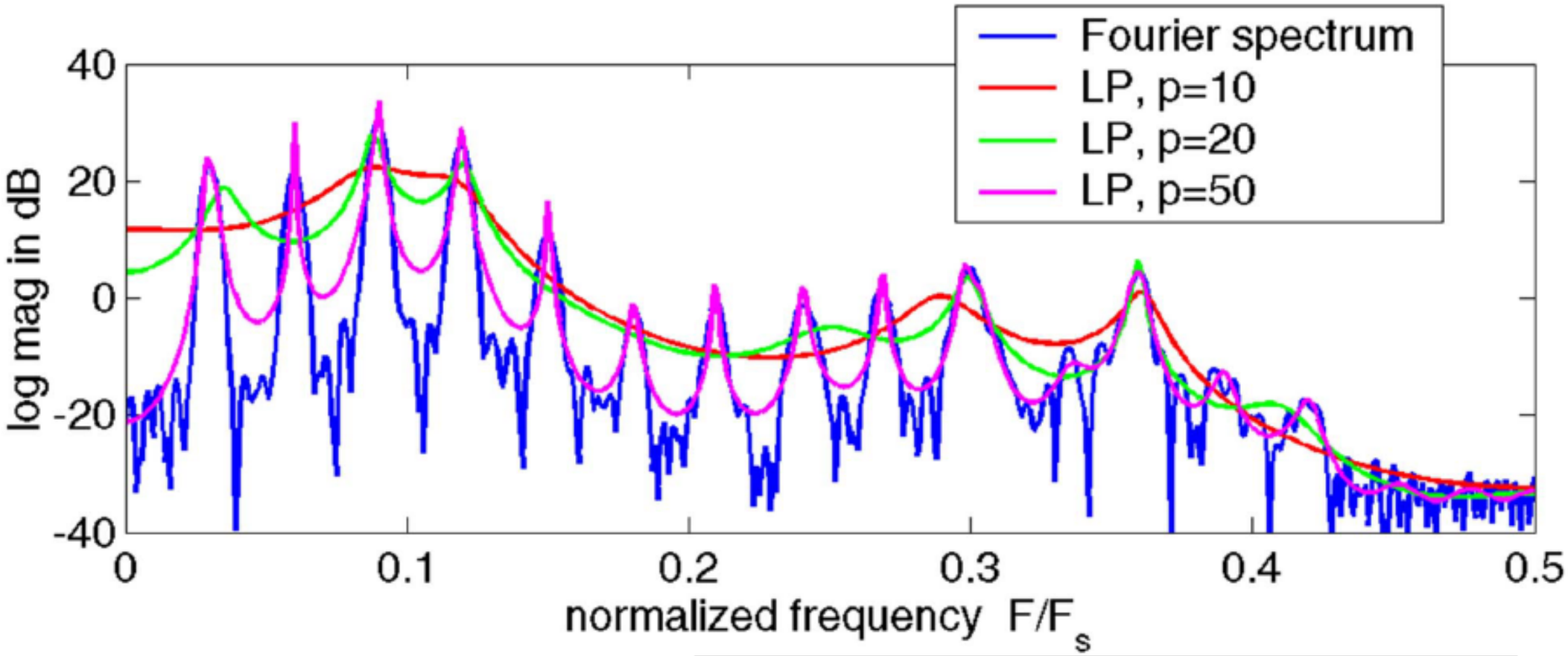
- with frequency response

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

- 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



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

```
x = s .* hamming(301);
X = fft( x , 1000 )
[ A , G , r ] = autolpc( x , 10 )
H = G ./ fft(A,1000);
```

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)

LP Short-Time Spectrum Analysis

- Defined speech segment as:

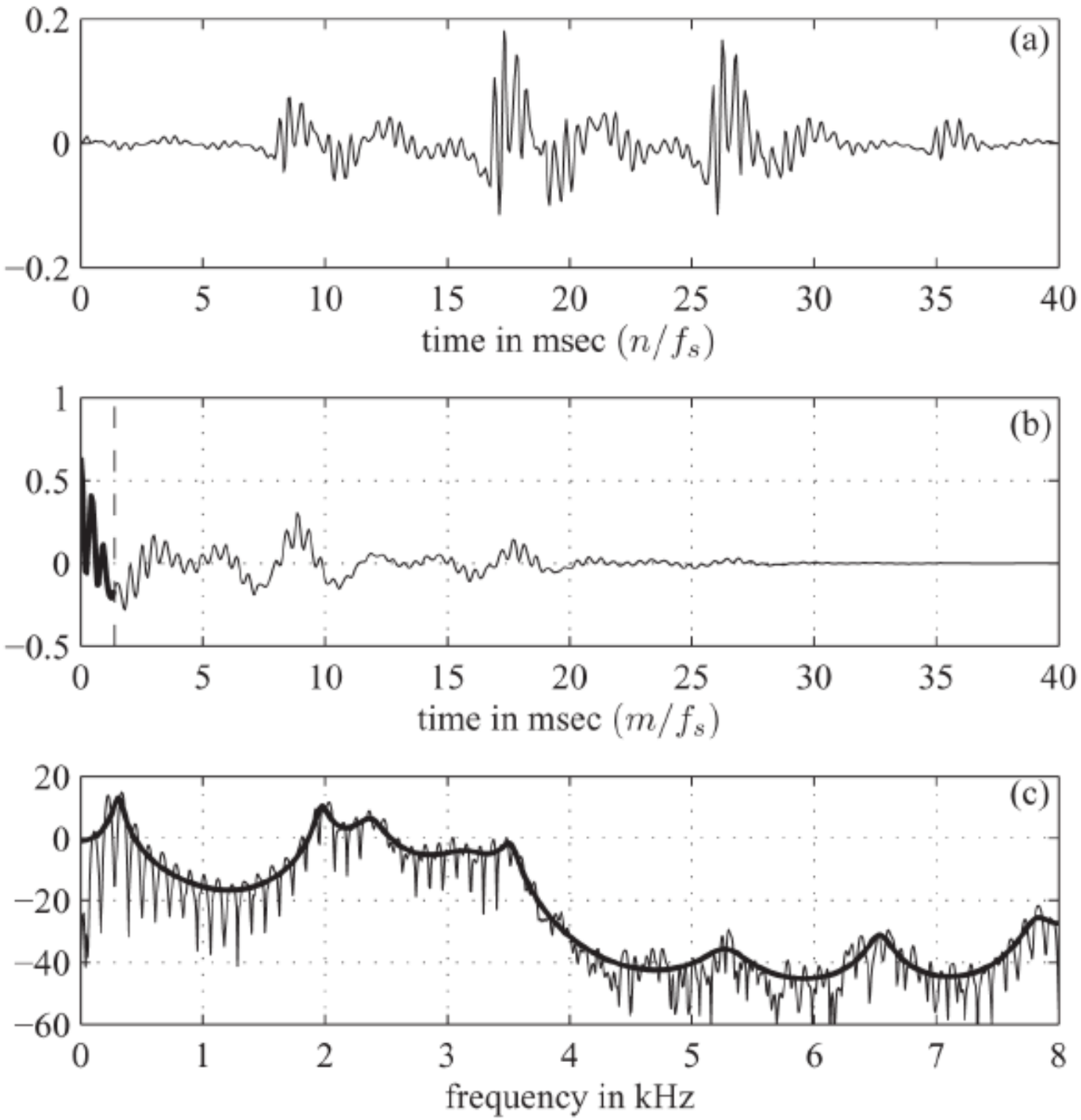
$$s_{\hat{n}}[m] = s[m + \hat{n}]w[m]$$

- The discrete-time Fourier transform of this windowed segment is:

$$S_{\hat{n}}\left(e^{j\omega}\right) = \sum_{m=-\infty}^{\infty} s[m + \hat{n}]w[m]e^{-j\omega m}$$

- Short-time FT and the LP spectrum are linked via short-time autocorrelation

LP Short-Time Spectrum Analysis

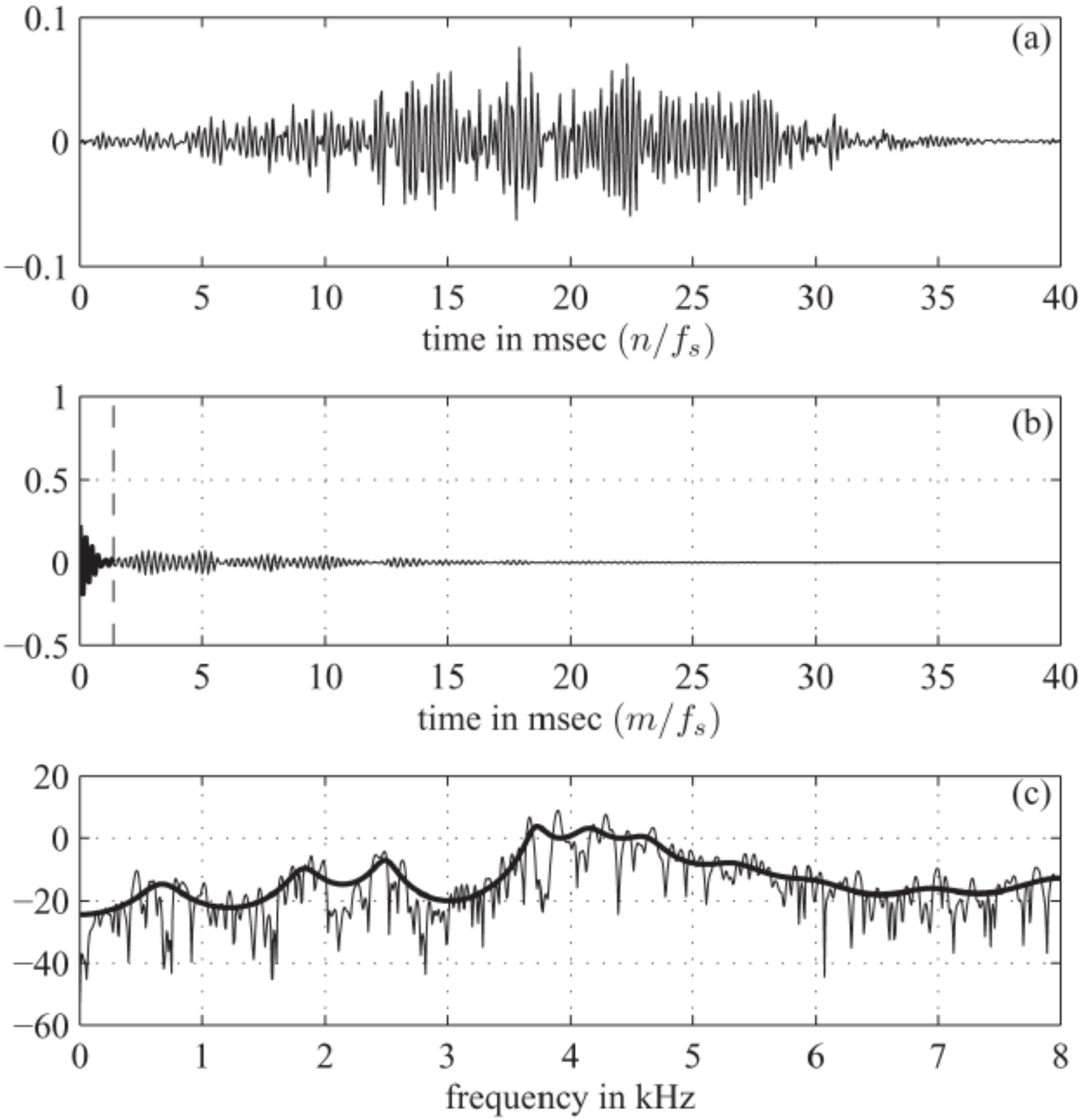


(a) Voiced speech segment obtained using a Hamming window

(b) Corresponding short-time autocorrelation function used in LP analysis (heavy line shows values used in LP analysis)

(c) Corresponding short-time log magnitude Fourier transform and short-time log magnitude LPC spectrum (F_S=16 kHz)

LP Short-Time Spectrum Analysis

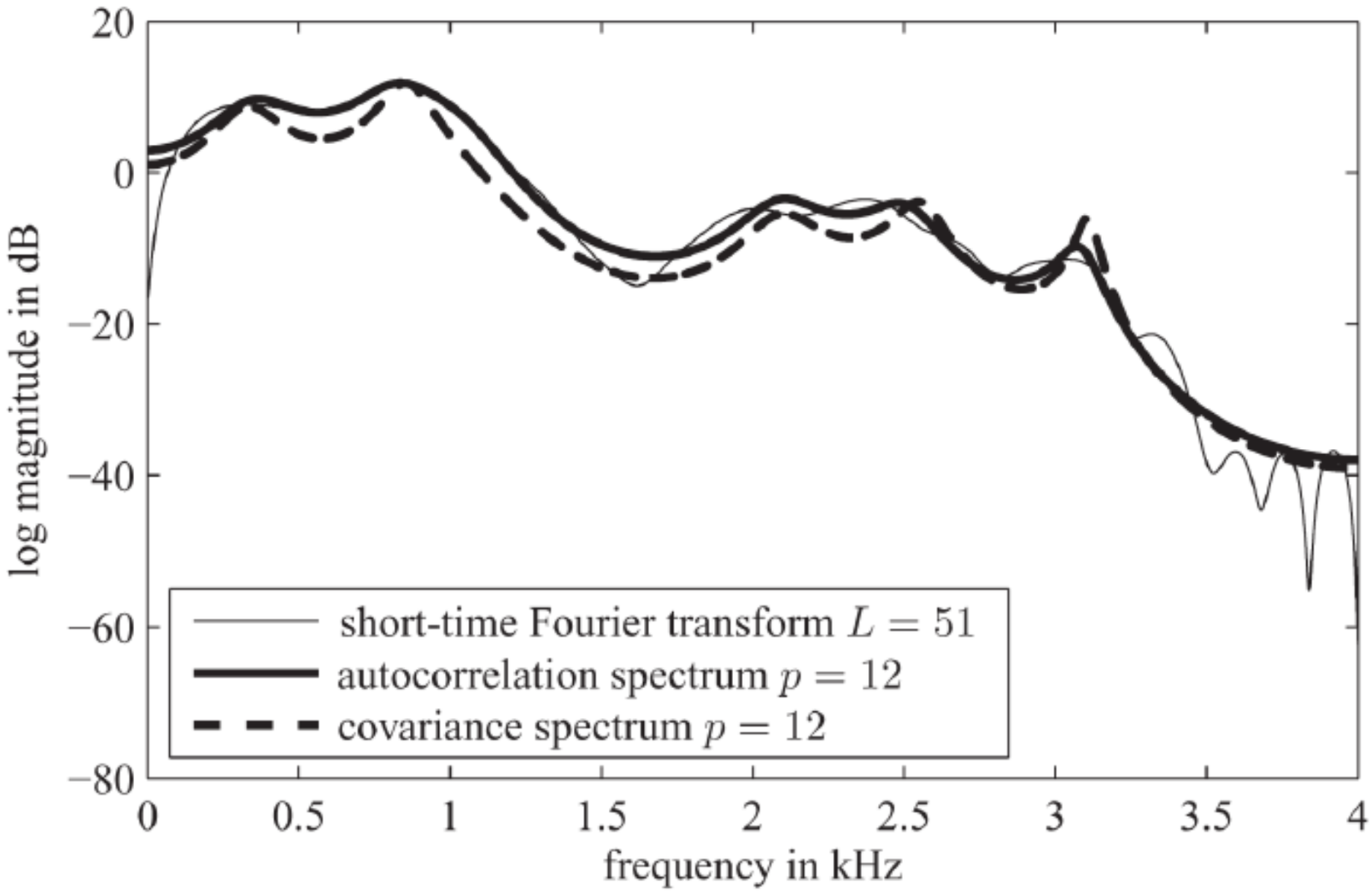


(a) Unvoiced speech segment obtained using a Hamming window

(b) Corresponding short-time autocorrelation function used in LP analysis (heavy line shows values used in LP analysis)

(c) Corresponding short-time log magnitude Fourier transform and short-time log magnitude LPC spectrum (F_S=16 kHz)

LP Interpretation Example2



Note small differences in spectral shape between STFT, autocorrelation spectrum and covariance spectrum when using short window duration (L=51 samples).

Solutions of LPC Equations

Covariance Method (Cholesky Decomposition Method)

LPC Solutions-Covariance Method

- for the covariance method we need to solve the matrix equation

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

$\phi\alpha = \psi$ (in matrix notation)

- ϕ is a positive definite, symmetric matrix with (i, j) element $\phi_{\hat{n}}(i, j)$, and α and ψ are column vectors with elements α_i and $\phi_{\hat{n}}(i,0)$
- the solution of the matrix equation is called the Cholesky decomposition, or square root method

ϕ=VDV$^{\mathbf{t}}$; V = lower triangular matrix with 1's on the main diagonal

D=diagonal matrix

LPC Solutions-Covariance Method

- can readily determine elements of V and D by solving for (i, j) elements of the matrix equation, as follows

$$\phi_{\hat{n}}(i,j) = \sum_{k=1}^{j} V_{ik} d_k V_{jk}, \qquad 1 \le j \le i-1$$

- giving

$$V_{ij} d_j = \phi_{\hat{n}}(i,j) - \sum_{k=1}^{j-1} V_{ik} d_k V_{jk}, \qquad 1 \le j \le i-1$$

- and for the diagonal elements

$$\phi_{\hat{n}}(i,i) = \sum_{k=1}^{i} V_{ik} d_k V_{ik}$$

- giving

$$d_i = \phi_{\hat{n}}(i,i) - \sum_{k=1}^{i-1} V_{ik}^2 d_k, \qquad i \ge 2$$

- with

$$d_1 = \phi_{\hat{n}}(1,1)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ V_{21} & 1 & 0 & 0 \\ V_{31} & V_{32} & 1 & 0 \\ V_{41} & V_{42} & V_{43} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_3 & 0 \\ 0 & 0 & 0 & d_4 \end{bmatrix} \begin{bmatrix} 1 & V_{21} & V_{31} & V_{41} \\ 0 & 1 & V_{32} & V_{42} \\ 0 & 0 & 1 & V_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cholesky Decomposition Example

- consider example with $p = 4$, and matrix elements $\phi_{\hat{n}}(i, j) = \phi_{ij}$

$$\begin{bmatrix} \phi_{11} & \phi_{21} & \phi_{31} & \phi_{41} \\ \phi_{21} & \phi_{22} & \phi_{32} & \phi_{42} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{43} \\ \phi_{41} & \phi_{42} & \phi_{34} & \phi_{44} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ V_{21} & 1 & 0 & 0 \\ V_{31} & V_{32} & 1 & 0 \\ V_{41} & V_{42} & V_{43} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_3 & 0 \\ 0 & 0 & 0 & d_4 \end{bmatrix} \begin{bmatrix} 1 & V_{21} & V_{31} & V_{41} \\ 0 & 1 & V_{32} & V_{42} \\ 0 & 0 & 1 & V_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cholesky Decomposition Example

- solve matrix for $d_1, V_{21}, V_{31}, V_{41}, d_2, V_{32}, V_{42}, d_3, V_{43}, d_4$

$$d_1 = \phi_{11}$$ step 1

$$V_{21} d_1 = \phi_{21}; \quad V_{31} d_1 = \phi_{31}; \quad V_{41} d_1 = \phi_{41}$$

$$V_{21} = \phi_{21} / d_1; \; V_{31} = \phi_{31} / d_1; \; V_{41} = \phi_{41} / d_1$$ step 2

$$d_2 = \phi_{22} - V_{21}^2 d_1$$ step 3

$$V_{32} d_2 = \phi_{32} - V_{31} d_1 V_{21} \Rightarrow V_{32} = \left(\phi_{32} - V_{31} d_1 V_{21}\right) / d_2$$

$$V_{42} d_2 = \phi_{42} - V_{41} d_1 V_{21} \Rightarrow V_{42} = \left(\phi_{42} - V_{41} d_1 V_{21}\right) / d_2$$ step 4

- iterate procedure to solve for d_3, V_{43}, d_4

LPC Solutions-Covariance Method

- now need to solve for α using a 2-step procedure

$$VDV^t\alpha = \psi$$

- writing this as

$$VY = \psi \quad \text{with}$$

$$DV^t\alpha = Y \quad \text{or}$$

$$V^t\alpha = D^{-1}Y$$

- from V (which is now known) solve for column vector Y using a simple recursion of the form

$$Y_i = \psi_i - \sum_{j=1}^{i-1} V_{ij} Y_j, \qquad p \geq i \geq 2$$

- with initial condition

$$Y_1 = \psi_1$$

LPC Solutions-Covariance Method

- now can solve for α using the recursion

$$\alpha_i = Y_i / d_i - \sum_{j=i+1}^{p} V_{ji}\alpha_j, \qquad 1 \le i \le p-1$$

- with initial condition

$$\alpha_p = Y_p / d_p$$

- calculation proceeds backwards from $i = p-1$ down to $i = 1$

Cholesky Decomposition Example

- continuing the example we solve for Y

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ V_{21} & 1 & 0 & 0 \\ V_{31} & V_{32} & 1 & 0 \\ V_{41} & V_{42} & V_{43} & 1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix}$$

- first solving for $Y_1 - Y_4$ we get

$$Y_1 = \psi_1$$
$$Y_2 = \psi_2 - V_{21}Y_1$$
$$Y_3 = \psi_3 - V_{31}Y_1 - V_{32}Y_2$$
$$Y_4 = \psi_4 - V_{41}Y_1 - V_{42}Y_2 - V_{43}Y_3$$

Cholesky Decomposition Example

- next solve for α from equation

$$\begin{bmatrix} 1 & V_{21} & V_{31} & V_{41} \\ 0 & 1 & V_{32} & V_{42} \\ 0 & 0 & 1 & V_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 1/d_1 & 0 & 0 & 0 \\ 0 & 1/d_2 & 0 & 0 \\ 0 & 0 & 1/d_3 & 0 \\ 0 & 0 & 0 & 1/d_4 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} Y_1/d_1 \\ Y_2/d_2 \\ Y_3/d_3 \\ Y_4/d_4 \end{bmatrix}$$

- giving the results

$$\alpha_4 = Y_4 / d_4$$
$$\alpha_3 = Y_3 / d_3 - V_{43}\alpha_4$$
$$\alpha_2 = Y_2 / d_2 - V_{32}\alpha_3 - V_{42}\alpha_4$$
$$\alpha_1 = Y_1 / d_1 - V_{21}\alpha_2 - V_{31}\alpha_3 - V_{41}\alpha_4$$

- completing the solution

Covariance Method Minimum Error

- the minimum mean squared error can be written in the form

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

$$= \phi_{\hat{n}}(0,0) - \alpha^t \psi$$

- since $\alpha^t = Y^t D^{-1} V^{-1}$ can write this as

$$E_{\hat{n}} = \phi_{\hat{n}}(0,0) - Y^t D^{-1} Y$$

$$= \phi_{\hat{n}}(0,0) - \sum_{k=1}^{p} Y_k^2 \,/\, d_k$$

- this computation for $E_{\hat{n}}$ can be used for all values of LP order from 1 to $p \Rightarrow$ can understand how LP order reduces mean-squared error

Cholesky Matrix Inversion Algorithm

%% Find first column of $\boldsymbol{V}$

$$d_1 = \phi_{11} \tag{9.66}$$

for $i = 2, 3, \ldots, p$

$$V_{i1} = \phi_{i1}/d_1 \tag{9.67}$$

end

%% Find $\boldsymbol{D}$ and remaining columns of $\boldsymbol{V}$

for $j = 2, 3, \ldots, p-1$

$$d_j = \phi_{jj} - \sum_{k=1}^{j-1} V_{jk}^2 d_k \tag{9.68}$$

for $i = j+1, \ldots, p$

$$V_{ij} = (\phi_{ij} - \sum_{k=1}^{j-1} V_{ik} d_k V_{jk})/d_j \tag{9.69}$$

end

end

$$d_p = \phi_{pp} - \sum_{k=1}^{p-1} V_{pk}^2 d_k \tag{9.68}$$

%% Find $\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{\alpha}$

$$Y_1 = \psi_1 \tag{9.72}$$

for $i = 2, 3, \ldots, p$

$$Y_i = \psi_i - \sum_{j=1}^{i-1} V_{ij} Y_j \tag{9.73}$$

end

%% Find $\boldsymbol{\alpha}$ from $\boldsymbol{Y}$

$$\alpha_p = Y_p/d_p \tag{9.76}$$

for $i = p-1, p-2, \ldots, 1$

$$\alpha_i = Y_i/d_i - \sum_{j=i+1}^{p} V_{ij}\alpha_j \tag{9.77}$$

end

Solutions of LPC Equations

Autocorrelation Method via Levinson-Durbin Algorithm

Levinson-Durbin Algorithm 1

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

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

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

- $\mathbf{R}$ 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:

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

- 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 next 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 $\gamma^{(i-1)}$ 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 coefficients.
- With this choice of $\gamma^{(i-1)}$, the vector of ith 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 7

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

Levinson-Durbin Algorithm

Levinson-Durbin Algorithm

$$\mathcal{E}^{(0)} = R[0] \tag{9.98}$$

for $i = 1, 2, \ldots, p$

$$k_i = \left(R[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R[i-j] \right) / \mathcal{E}^{(i-1)} \tag{9.93}$$

$$\alpha_i^{(i)} = k_i \tag{9.96b}$$

if $i > 1$ then for $j = 1, 2, \ldots, i-1$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \tag{9.96a}$$

end

$$\mathcal{E}^{(i)} = (1 - k_i^2) \mathcal{E}^{(i-1)} \tag{9.94}$$

end

$$\alpha_j = \alpha_j^{(p)} \quad j = 1, 2, \ldots, p \tag{9.97}$$

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

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

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

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

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

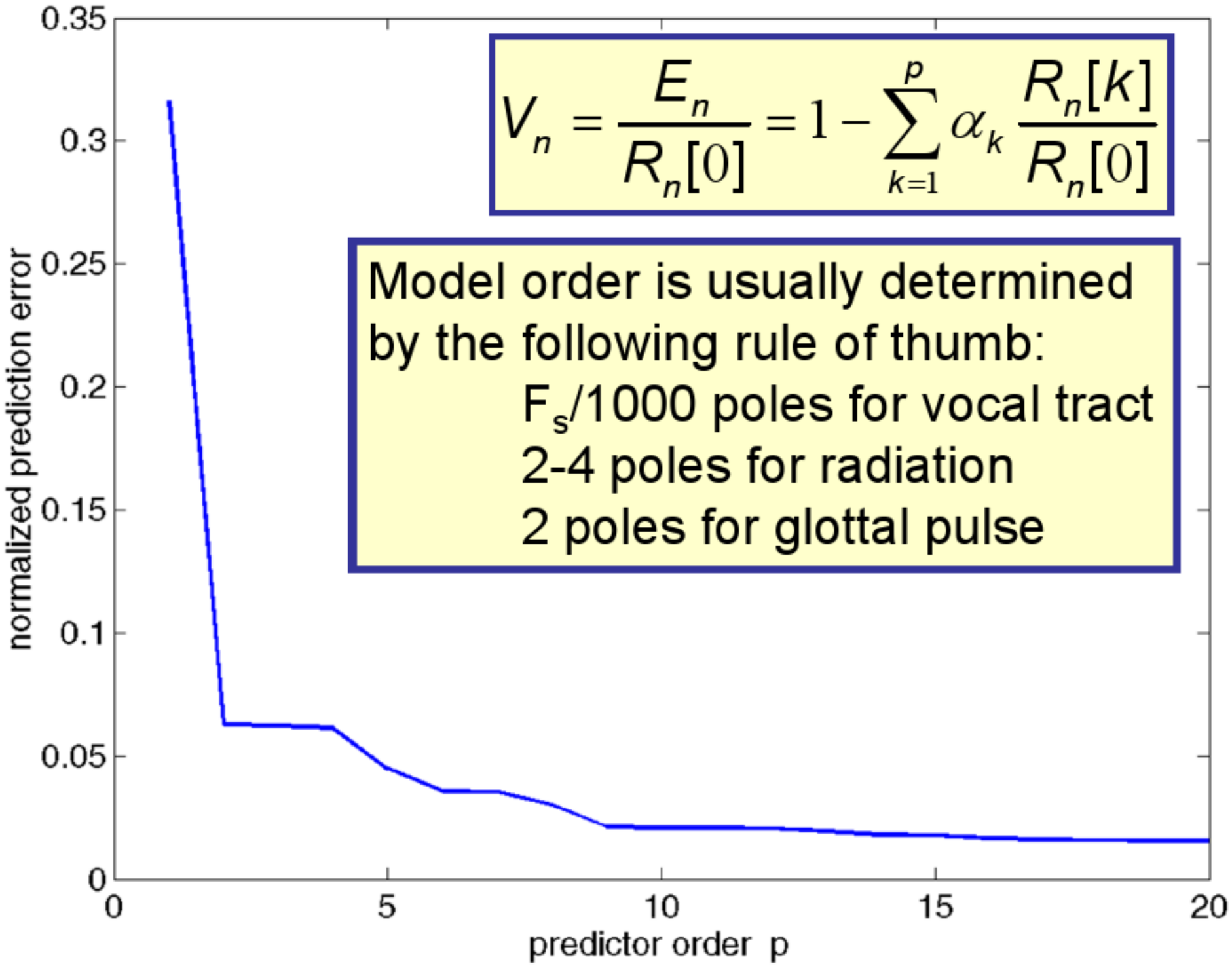
- with final coefficients

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

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

$E^{(i)}$ = 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

4.7.Application of LPC Parameters

- 4.7.1. Pitch detection using LPC parameters
- 4.7.2. Formant analysis
- 4.7.3. VELP
- 4.7.4. CELP

- Reference (Manju)

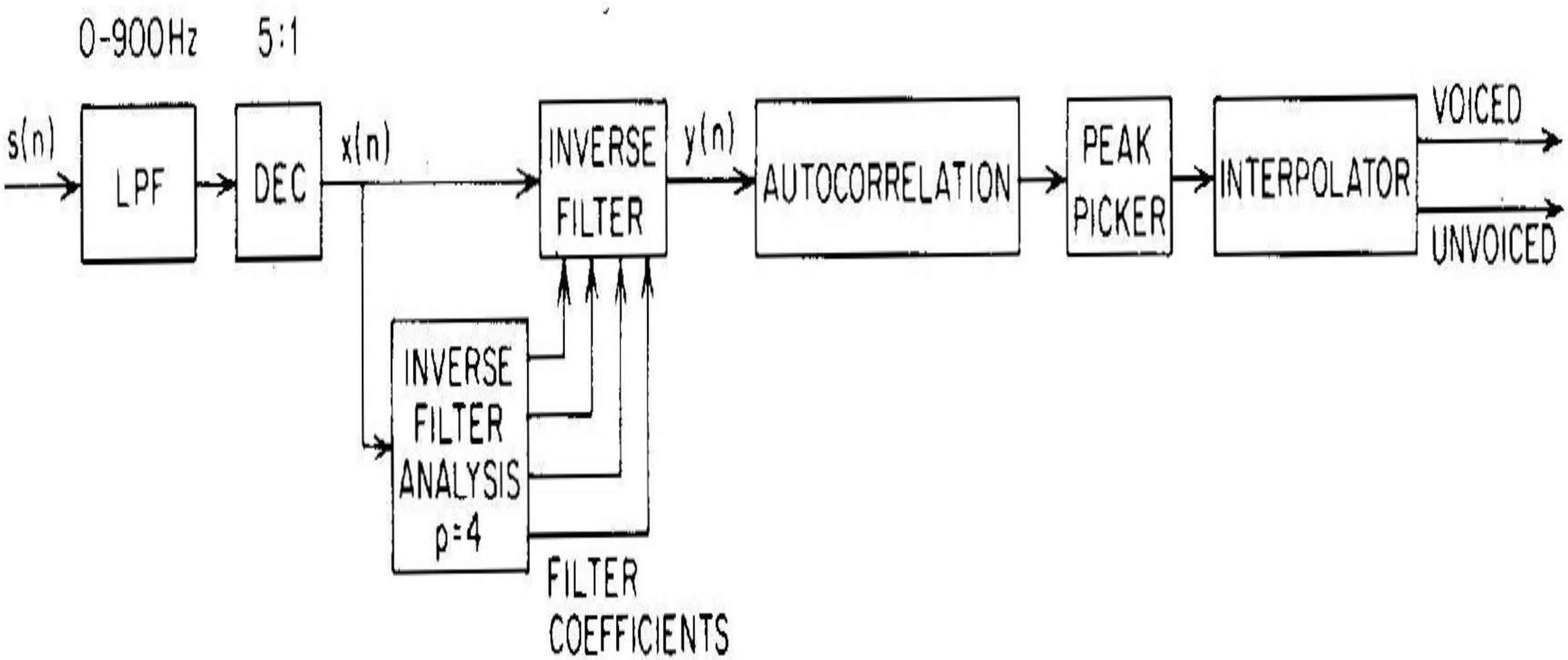
4.7.Application of LPC Parameters

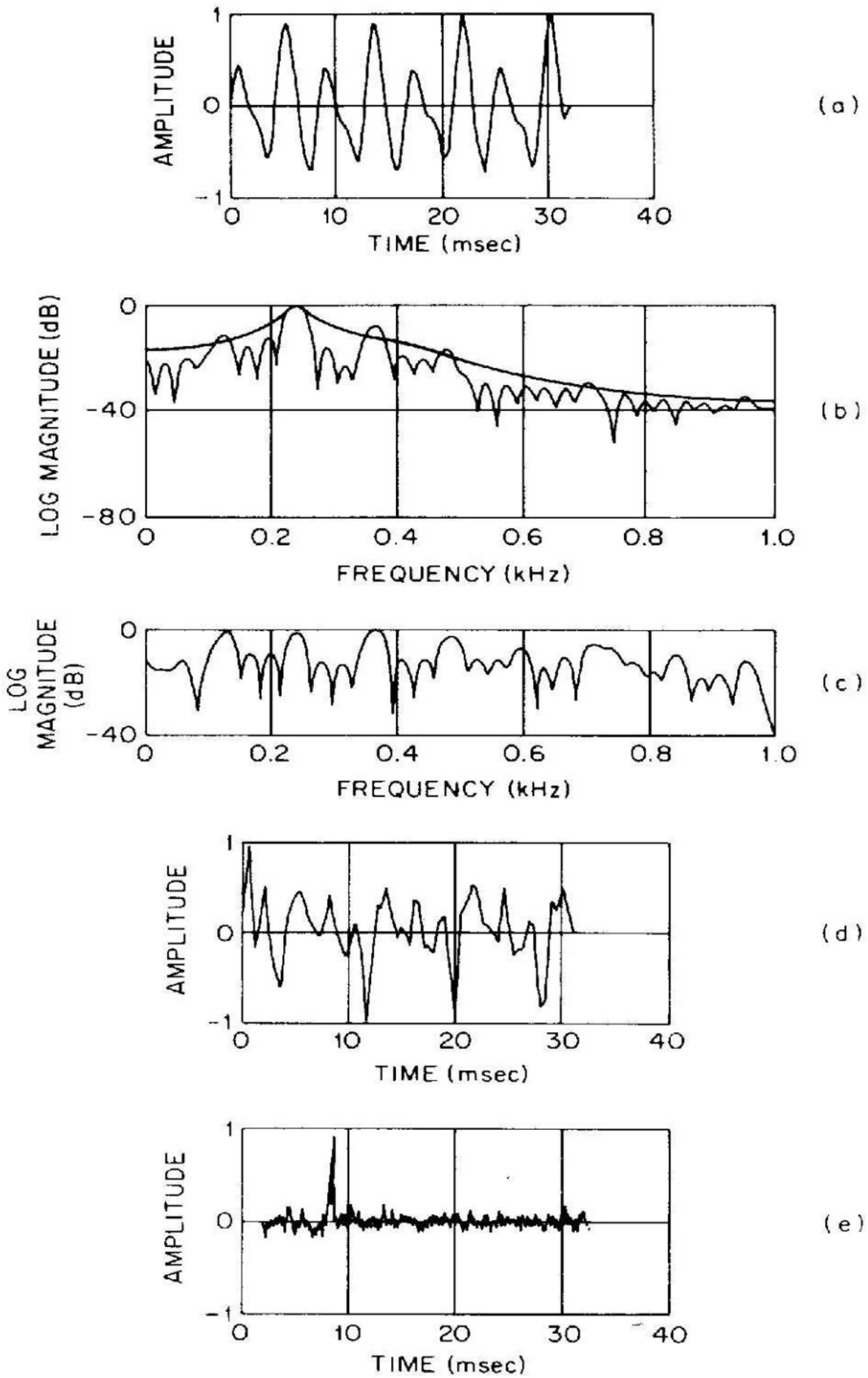
- LPC methods are the most widely used in speech coding, speech synthesis, speech recognition, speaker recognition and verification and for speech storage
- LPC methods provide extremely accurate estimates of speech parameters, and does it extremely efficiently.

Pitch detection using LPC parameter

- Error signal e(n) from LPC analysis can be used to estimate the pitch period directly.
- To overcome the limitation of above method Simple Inverse Filtering Tracking(SIFT) method is introduced.

Block diagram of SIFT algorithm for pitch detection





Input waveform

Input spectrum and LPC parameter

Output spectrum after inverse-filtering with LPC

Output waveform
normalized

Autocorrelation of
output waveform

Problems with SIFT

- 1. For some sounds (e.g. nasals, vowel-to-silence transitions) the signal is dominated by a single harmonic (close to sine wave).
  LPC analysis and inverse filtering can remove the formant information *and* the glottal-source information, leaving only white noise.

- 2. Still have the problem that a high frequency signal this method is not applicable because of the presence of more than one formants. This may cause “pitch halving” error.

- Rather than use autocorrelation, we can measure F0 from information in the spectrum, by identifying harmonics

Formant analysis

- Formants are the resonances of the vocal tract.
- Formants position and bandwidth estimation can be done in following ways:
- Factoring the predictive polynomial roots
- Peak picking method

Factoring the predictive polynomial roots

- Formants can be estimated using the parameter obtained from linear prediction.
- Factoring the predictor polynomial and based on the root obtained, try to decide which are formants and which corresponds to spectral shaping poles.

Advantages of using LPC method for formant analysis

- Formant center frequency and bandwidth accurately determined by factoring predictor polynomial
- Pole correspond to appropriate formant decision is easier then other methods since number of poles reduced.
- Extra poles are easily isolated since their bandwidth are very large compared to that of the speech formant under consideration.

Disadvantages of using LPC for formant analysis

- For nasal sound or nasalized vowels physical significance of the predictor polynomial is unclear i.e. If the poles related to nasal pole or nasal zero or if they are all related to vocal tract resonance, since all pole model is used to model speech spectrum
- It is not clear that how bandwidth of the root is related to the actual formant bandwidth

Voice-excited linear prediction

- The main idea behind the voice-excitation is to avoid the imprecise detection of the pitch and the use of an impulse train while synthesizing the speech.
- It avoids direct estimation of the excitation parameter.

Block diagram of VELP

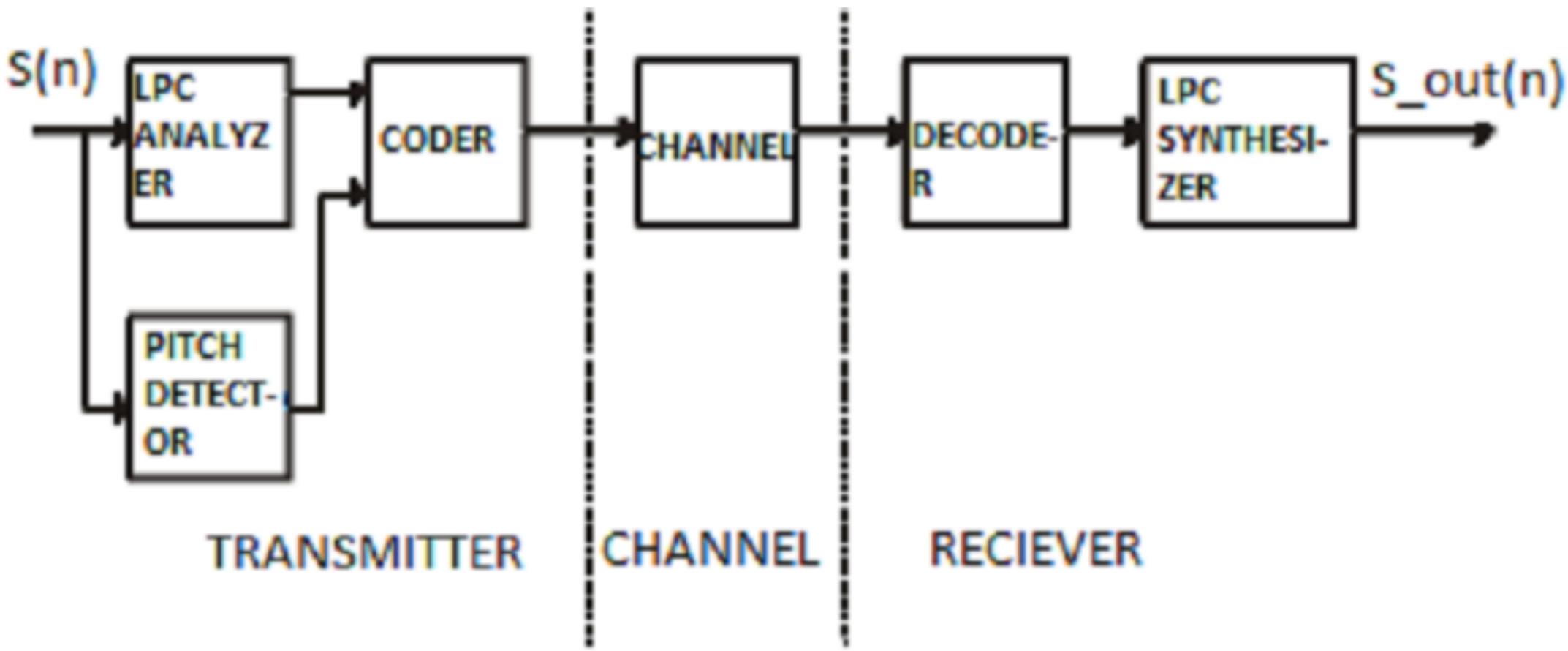


FIGURE 3.BLOCK DIAGRAM OF AN LPC VOCODER

Quantization of LPC coefficient

- Direct quantization is avoided
- Quantization of predictor polynomial coefficient
- Bit rate for transmission or storage is given as
- 72.fs
  - Where fs is number of frame per second whose value are 100, 67,33 gives bit rate of 7200, 4800 and 2400 bits per second respectively

Original Vs LPC Vs VELP

A. Original Speech Signal

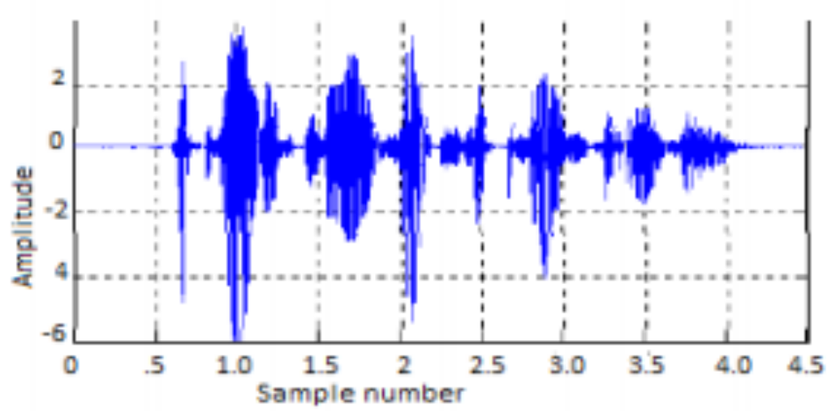


FIGURE 5.WAVEFORM OF ORIGINAL SPEECH SIGNAL

B. LPC reconstructed Speech Signal

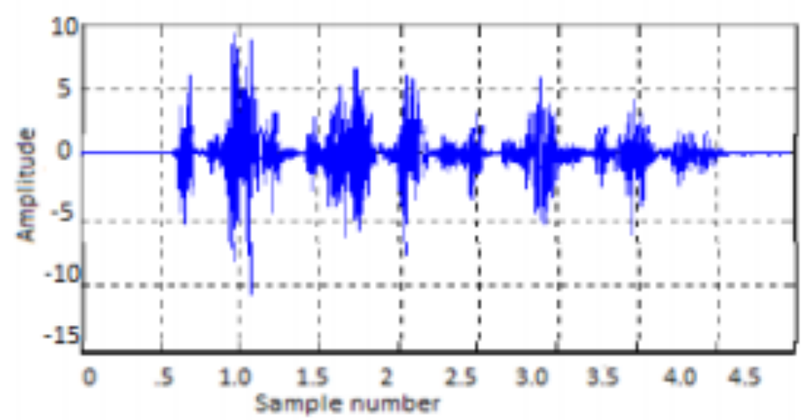


FIGURE 6.WAVEFORM OF LPC RECONSTRUCTED SPEECH SIGNAL

- The reconstructed speech has a lower Quality
- The LPC reconstructed speech has a lower pitch than the original sound.
- The voice-excited LPC reconstructed file sounds more spoken and less whispered

C. Voice Excited LPC reconstructed Speech Signal

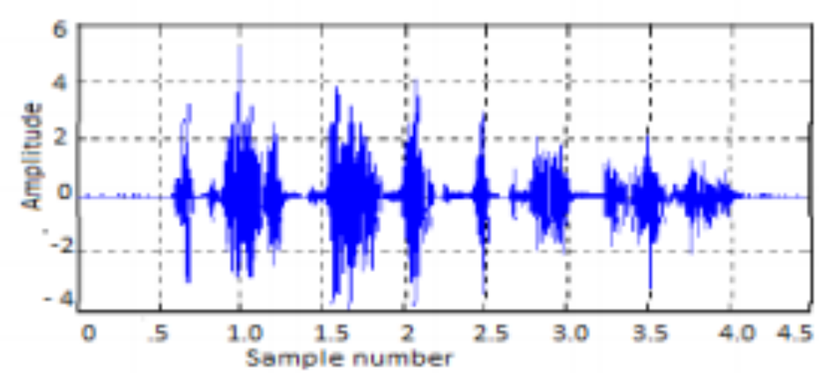


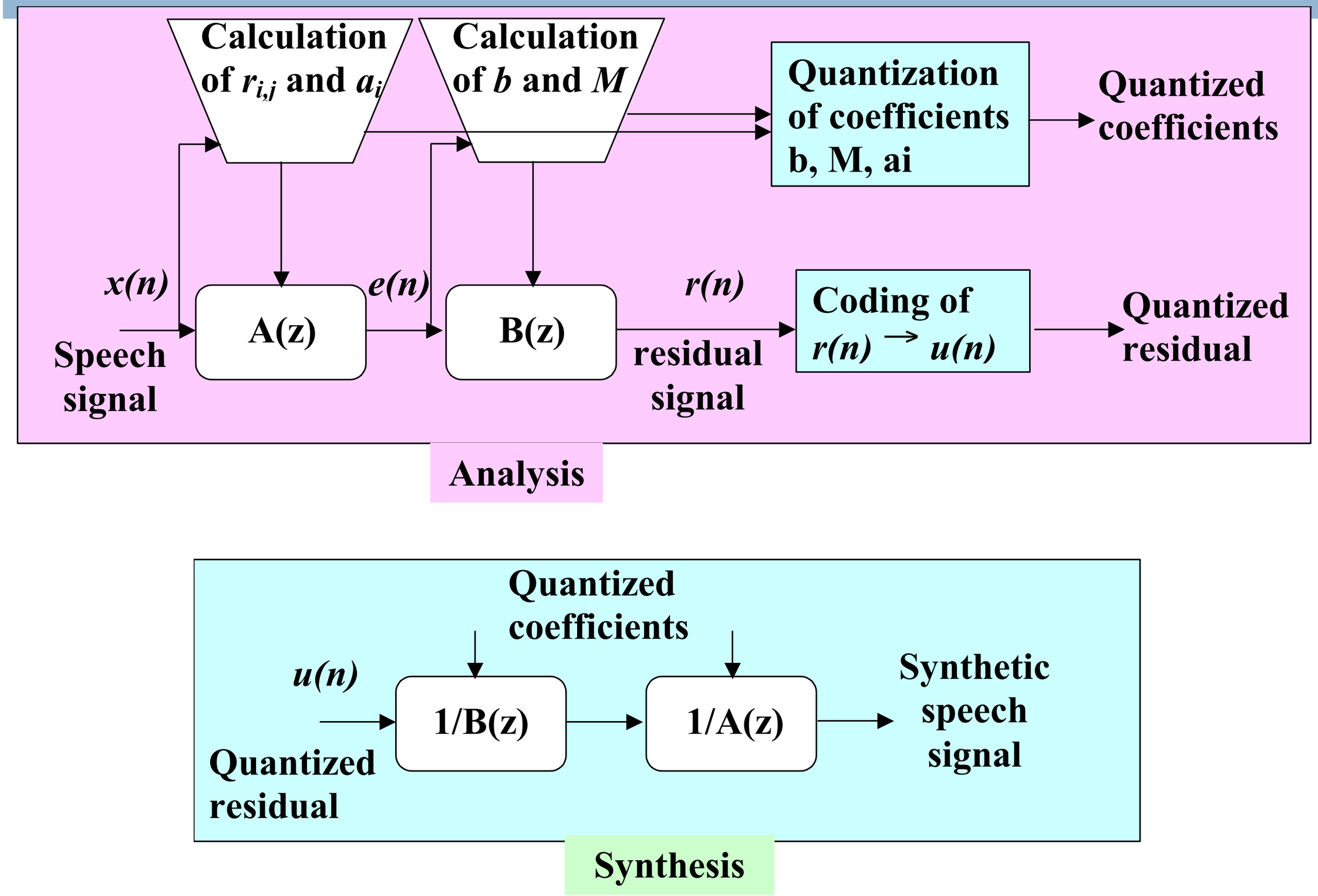
FIGURE 7.WAVEFORM OF VOICE EXCITED LPC RECONSTRUCTED SPEECH SIGNAL

Advantages and disadvantages of using VELP

- Increase in quality(pitch almost same as original signal).
- Voice identity is well preserved.
- the bits per sample increase causing an increase in Bandwidth of signal.

Code-excited linear prediction Coders using Short Term and Long Term Prediction

RELP → MPE LP→CELP



Code-excited linear prediction

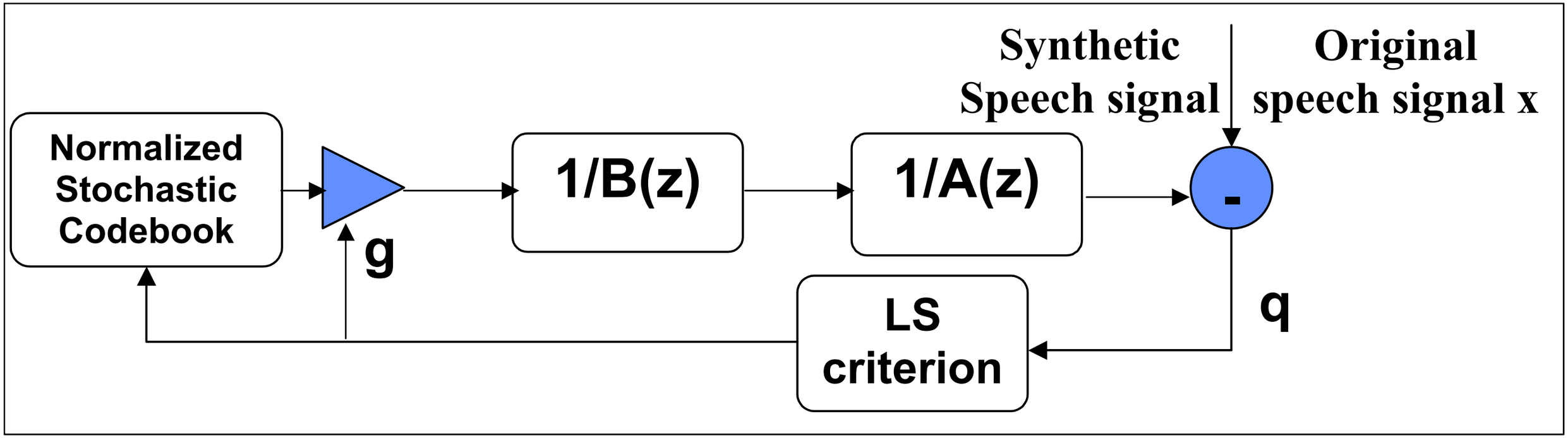
- Uses analysis by synthesis method
- CELP stream would be produced by trying all possible bit combinations and selecting the one that produces the best-sounding decoded signal.
- High complexity
- In CELP coders, vector quantization is applied on the signal.
  - Each frame of residual signal is compared to sequences of signal stored in a codebook. The codebook sequences are white and the codebook is called stochastic codebook.

CELP(continued..)

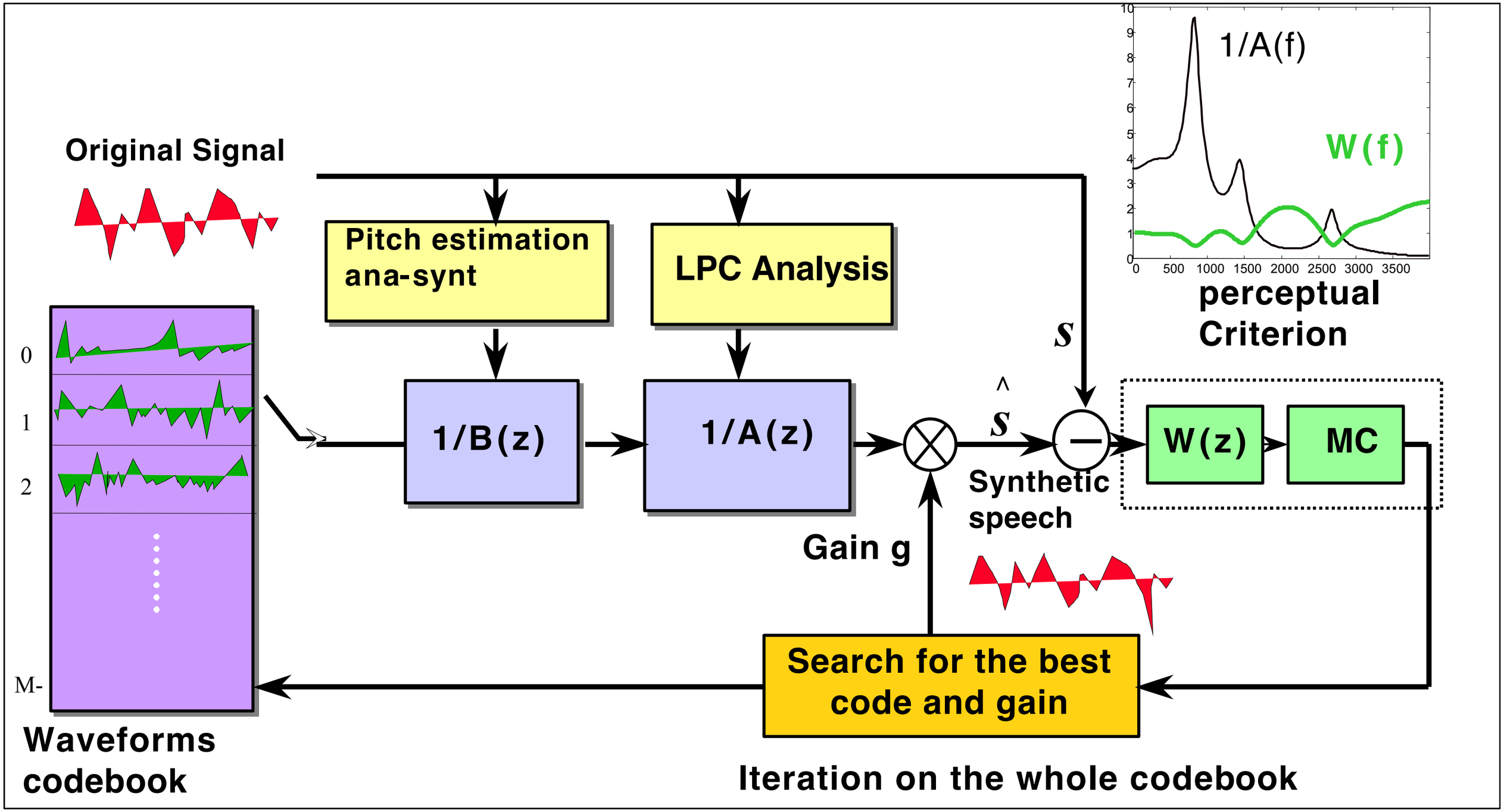
- The CELP algorithm is based on four main ideas:
- Using the source-filter model of speech production through linear prediction (LP)
- Using an adaptive and a fixed codebook as the input (excitation) of the LP model;
- Performing a search in closed-loop in a “perceptually weighted domain”.
- Applying vector quantization (VQ)

CELP Coder Basic Scheme

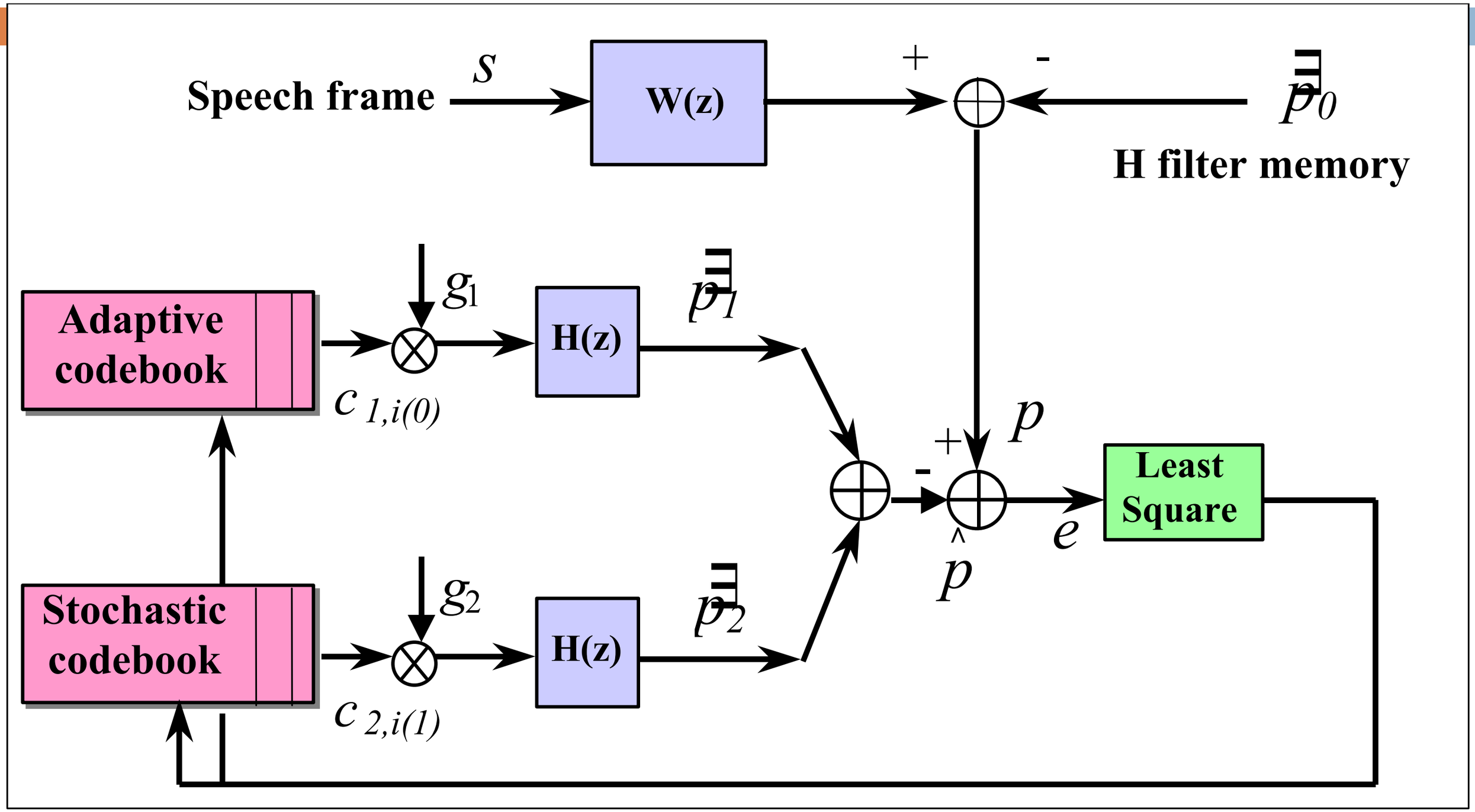
- Analysis by synthesis (closed loop) to find the best excitation sequence.



CELP Coder with Perceptual Filter

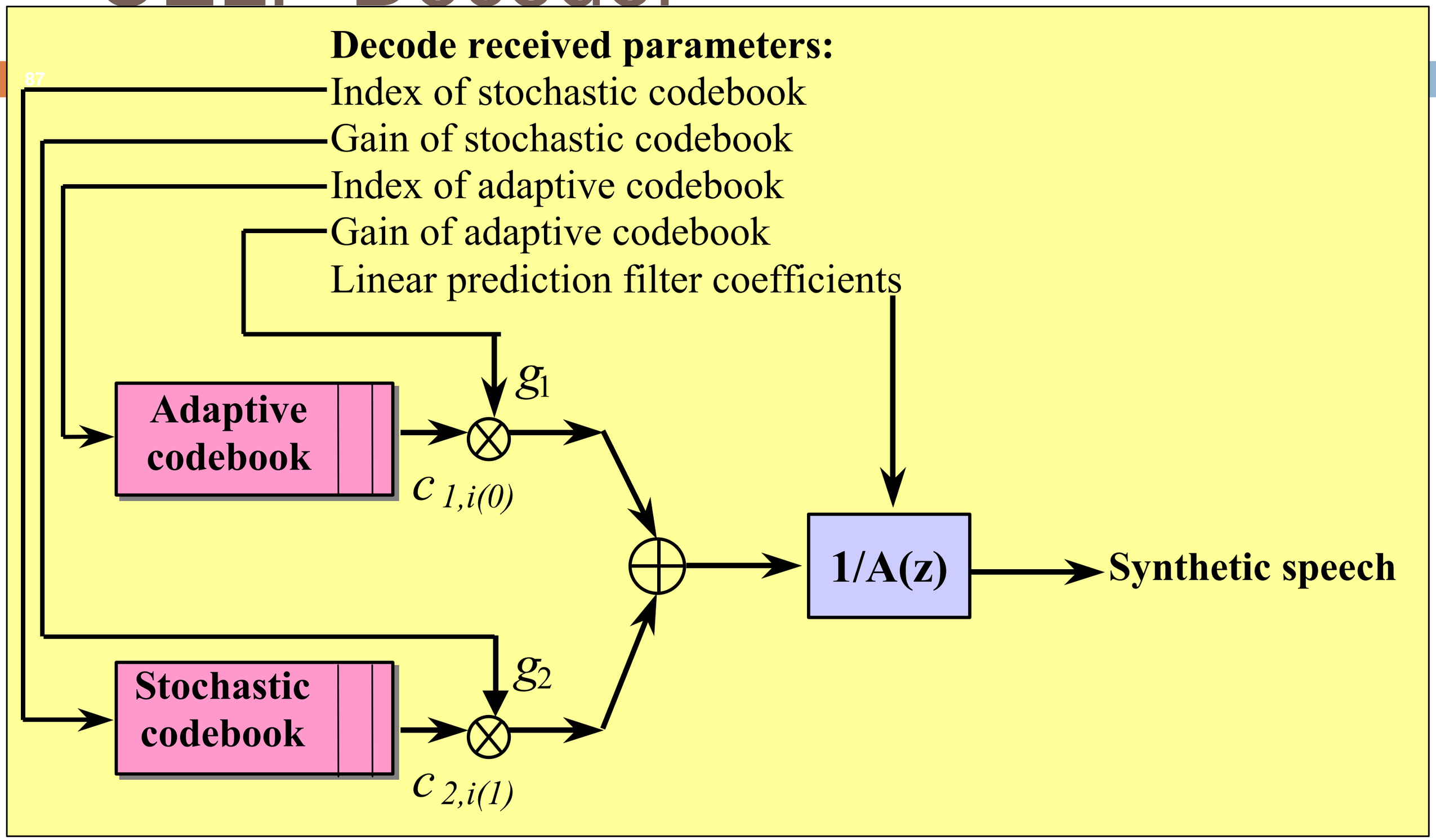


CELP with Stochastic Codebook



The adaptive codebook stores the past residual frames. It is called adaptive because its content changes with time.

CELP Decoder



CELP Coding

- Standards from 4.8 kbps to 16 kbps
- Federal standard (DOD) (4.8 kbps)
  - frame = 260 samples (30 ms)
  - LPC 8 --> (LSP coding 34 bits)
  - adaptive codebook (256 vectors (fractional pitch))
  - stochastic codebook (512 vectors (-1,0,1))