Linear Predictive Coding of Speech

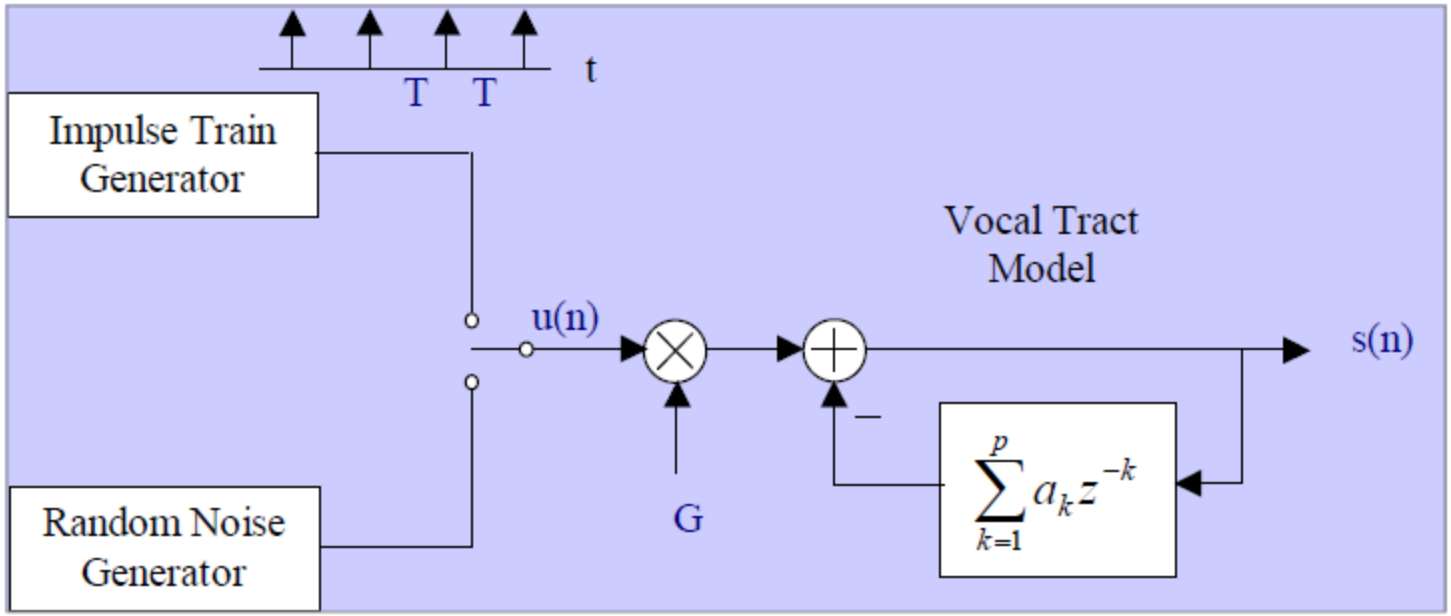
- One of the most powerful speech analysis techniques is the method of linear predictive analysis.
- This method has become the predominant technique for estimating the basic speech parameters, e.g. pitch, formants, spectra, vocal tract area functions, and for representing speech for low bit rate transmission or storage

LPC .....

- The basic idea behind linear predictive analysis is that a speech sample can be approximated as a linear combination of past speech samples.
- By minimising the sum of the squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficients can be determined.

A basic discrete-time model for speech production was developed previously

$$\frac{S(z)}{U(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$



LPC….

- The speech samples s(n) are related to the excitation u(n) by the simple difference equation:

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

- Between pitch pulses Gu(n) is zero.Therefore using the above equation, s(n) can be predicted from a linearly weighted summation of past samples. However, if Gu(n) is included then we can predict s(n) only approximately

Basis for Linear Prediction

- Between pitch pulses u(n) = 0

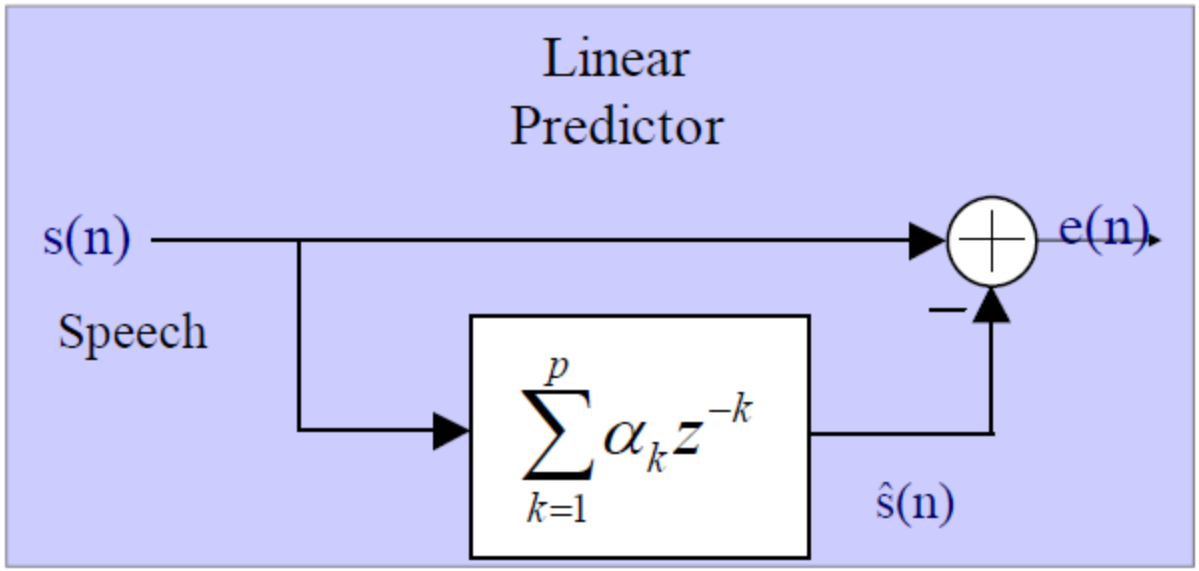
$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n)$$

$$\therefore s(n) = a_1 s(n-1) + a_2 s(n-2) + \ldots + a_q s(n-p)$$

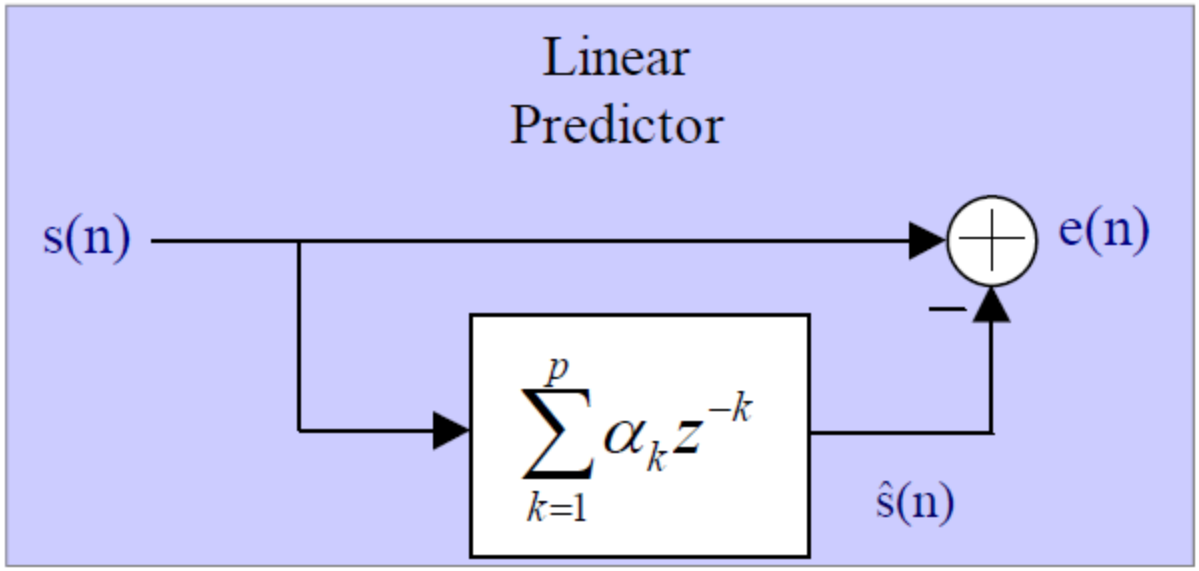
- The n^{th} speech sample can be viewed as a linear combination of p past samples

LPC……

- LPC is often referred to as "inverse filtering", as its aim is to determine the "all zero" filter which is the INVERSE of the vocal tract model



All Zero Filter



- The input to the model will be the previous speech samples, s(n-k) and output is estimated current sample
- The difference between the ACTUAL speech sample & the ESTIMATED speech sample is the error signal, e(n)

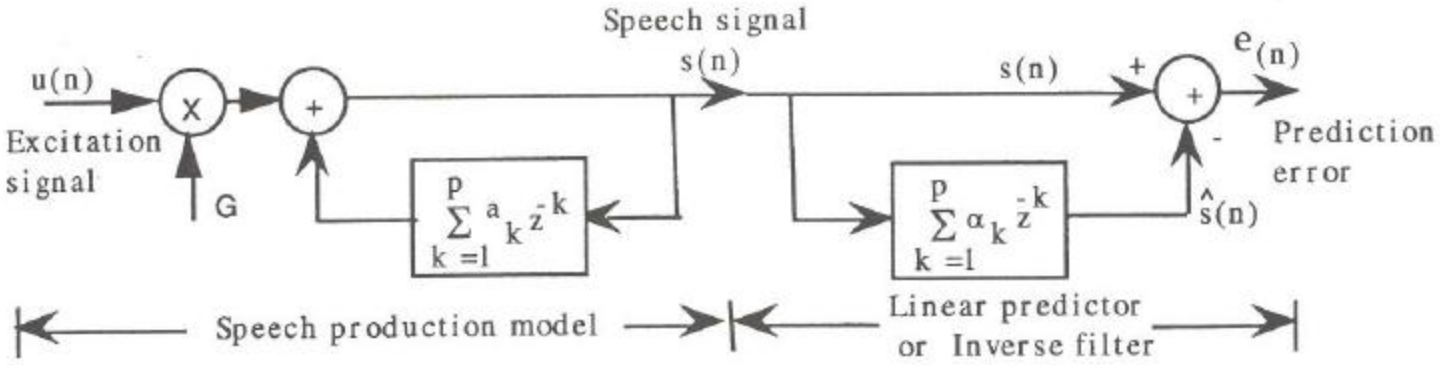
Linear Prediction Model

Suppose that we process the speech signals with a linear predictor and the predictor coefficients are α_k and the predictor output is:

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

The error between the actual signal s(n) and the predicted value $\hat{s}(n)$ is given by

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



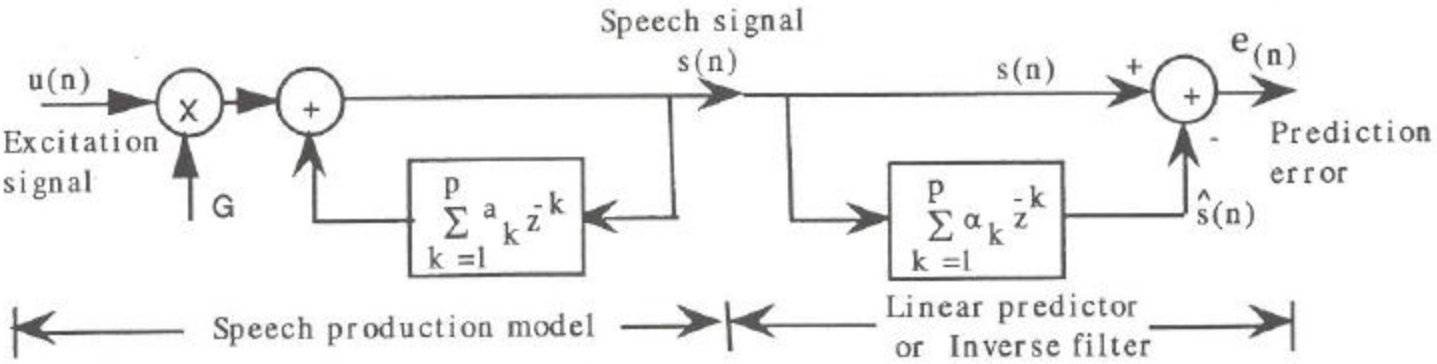
The prediction error signal e(n) is shown above.

$$\frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}}$$

<- Vocal tract(IIR filter)

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

<- Linear Predictor(FIR filter)



$$\frac{E(z)}{U(z)} = G\frac{1-\sum_{k=1}^{P}\alpha_k z^{-k}}{1-\sum_{k=1}^{P}a_k z^{-k}}$$

If $\alpha_k = a_k \Rightarrow \frac{E(z)}{U(z)} = G \Rightarrow e(n) = Gu(n)$

- Ideally, we need a technique to produce the coefficients of the model (α_k) such that they are equal to the coefficients of the speech production model (a_k)

LPC…..

- If we determine the correct coefficients, then the error signal e(n) = Gu(n) and the linear predictor is called an "inverse filter". The transfer functiion of the inverse filter is given by

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

- A by-product of the LPC analysis is the generation of the error signal e(n) and it is a good approximation to the excitation source.
- It is expected that the prediction error e(n) will be large (for voiced speech) at the beginning of each pitch period.
- Thus the pitch period can be determined by detecting the positions of the samples of e(n) which are large, and defining the period as the difference between pairs of samples of e(n) which exceed a reasonable threshold.

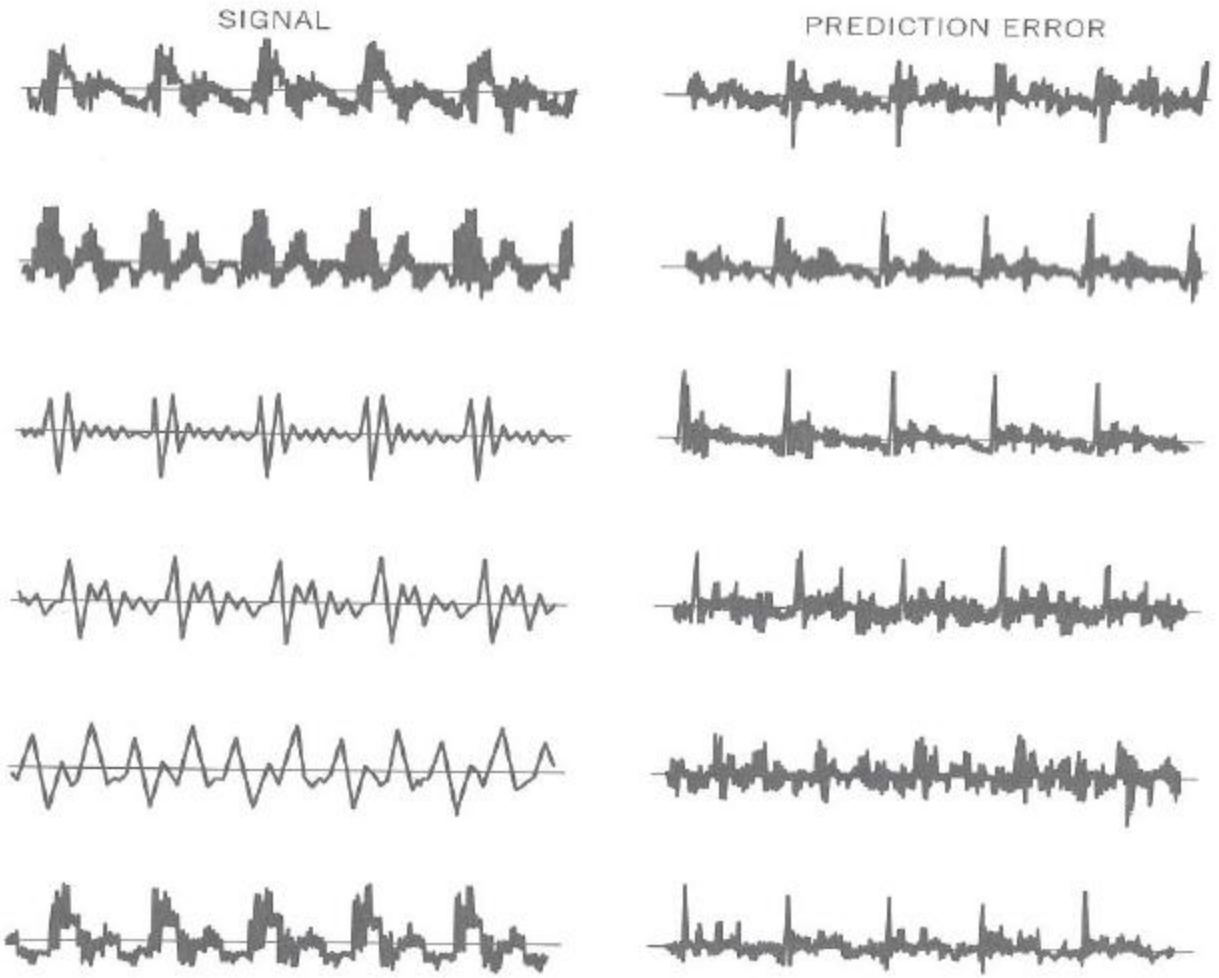
- Alternatively the pitch period can be determined by performing an autocorrelation analysis on e(n) and detecting the largest peak in the appropriate range.
- Another way of interpreting why the error signal is valuable for pitch detection is the observation that the spectrum of the error signal is approximately flat; thus the effects of the formants have been eliminated in the error signal.
- In conclusion we can say that except for a sample at the beginning of each pitch period, every sample of the voiced speech waveform can be predicted from the past p samples.

LPC….

- For voiced speech, e(n) would consist of a train of impulses
- e(n) would be small most of the time except at the beginning of the pitch period
- The prediction is not valid at instants of time where the input pulses occur

Speech signal s(n)

Error signal
$\varepsilon(n) = G\,u(n)$



Residual error e(n) waveforms for several vowels (Rabiner et el. , 1993) 16

LPC….

- The prediction error e(n) = Gu(n) would be
  - Scaled pulse train for voiced speech frames
  - Scaled random noise for unvoiced speech frames
- Because of the time-varying nature of the speech, the predictor coefficients should be estimated from short segments of speech signal (10-20 ms)

Estimation of Predictor Coefficients

- The basic approach is to find a set of predictor coefficients α_k that will minimise the mean squared error , $e(n)^2$, over a short segment of speech waveform.

$$E = \sum_{n=1}^{N} e(n)^2 = \sum_{n=1}^{N} (s(n) - \hat{s}(n))^2$$

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

Estimation of Predictor Coefficients

- We are attempting to MINIMISE this error and hence we must determine the condition such that the derivative of E with respect to α_k is zero

$$\frac{\partial E}{\partial \alpha_i} = 0, \text{ for i} = 1, 2, 3, \ldots\ldots \text{p (typically 10 to 14)}$$

Minimisation of Error

$$\frac{\partial E}{\partial \alpha_i} = \sum_{n=1}^{N} 2\left[s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) \right]\left[-s(n-i) \right] = 0$$

$$\therefore \sum_{n=1}^{N} s(n)s(n-i) - \sum_{n=1}^{N}\sum_{k=1}^{p} \alpha_k s(n-k)s(n-i) = 0$$

$$\sum_{n=1}^{N} s(n)s(n-i) - \sum_{k=1}^{p} \alpha_k \sum_{N=1}^{N} s(n-k)s(n-i) = 0 \quad \text{for i} = 1,2,3 \ldots\ldots \text{p}$$

$$\sum_{k=1}^{p} \alpha_k \sum_{N=1}^{N} s(n-k)s(n-i) = \sum_{n=1}^{N} s(n)s(n-i) \quad \text{for i} = 1,2,3 \ldots\ldots \text{p}$$

Autocorrelation Method for LPC Analysis

- This yields a set of p simultaneous equations
- Closer examination of the "coefficients" of these simultaneous equations show that they are actually the autocorrelation function for difference delay values
- We have already introduced the autocorrelation function

$$\phi(k) = R(k) = \sum_{n=0}^{N-1} s(n)s(n+k)$$

$$R(k) = R(-k)$$

Autocorrelation Method for LPC Analysis

- Consider the "coefficient" of a_k in the set of simultaneous equations, namely

$$\sum_{n=1}^{N} s(n-k)s(n-i)$$

If n - k is replaced by l then this becomes

$$\sum_{l=1}^{N} s(l)s(l+k-i) = R(k-i)$$

The limits of the summation remain the same as the frame of speech is assumed to be windowed (i.e. s(l) = 0 if $l < 1$ or $l > N$)

It is assumed that the waveform segment s(n) is identically zero outside the interval $0 \le l \le N-1$ (i.e multiply the signal s(n) by a window function)

Autocorrelation Method for LPC Analysis

- Similarly, the "constant" terms of the simultaneous equation

$$\sum_{n=1}^{N} s(n)s(n-i) = R(i)$$

- Thus the set of simultaneous equations can be re-written using these autocorrelation terms

$$\sum_{k=1}^{p} \alpha_k \sum_{N=1}^{N} s(n-k)s(n-i) = \sum_{n=1}^{N} s(n)s(n-i) \quad \text{for i} = 1,2,3 \ldots\ldots \text{p}$$

$$\sum_{k=1}^{p} \alpha_k R(k-i) = R(\text{i}) \quad \text{for i} = 1,2,3 \ldots\ldots \text{p}$$

Autocorrelation Method for LPC Analysis

$$\sum_{k=1}^{p} \alpha_k R(k-i) = R(\mathrm{i}) \quad \text{for i} = 1,2,3 \ldots\ldots \mathrm{p}$$

$$\alpha_1 R(0) + \alpha_2 R(1) + \alpha_3 R(2) + \ldots\ldots\ldots \alpha_{\mathrm{p}} R(p-1) = R(1)$$

$$\alpha_1 R(1) + \alpha_2 R(0) + \alpha_3 R(1) + \ldots\ldots\ldots \alpha_{\mathrm{p}} R(p-2) = R(2)$$

$$\alpha_1 R(2) + \alpha_2 R(1) + \alpha_3 R(0) + \ldots\ldots\ldots \alpha_{\mathrm{p}} R(p-3) = R(3)$$

....

$$\alpha_1 R(p) + \alpha_2 R(p-1) + \alpha_3 R(p-2) + \ldots\ldots\ldots \alpha_{\mathrm{p}} R(0) = R(p)$$

Matrix Form of Simultaneous Equations

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

Solving the Simultaneous Equations

- It is necessary to invert the "autocorrelation matrix" (R) in order to determine the LPC coefficients

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

- This can be quite cumbersome, given that it is a pxp matrix, with p typically between 10 and 14 in most applications

Solving the Simultaneous Equations

- However, since R is what is known as a Toeplitz Matrix
  - Symmetric
  - Elements along diagonals are equal

  there are a number of ITERATIVE methods which can be used to solve the system (e.g. Durbin's algorithm)

Durbin's Algorithm

$\alpha_k^{(i)}$ is the k^{th} LPC coefficient value after the i^{th} iteration

$E^{(i)}$ is the residual error after the ith iteration

Steps

1. Initially set $E^{(0)} = R(0)$

2. Calculate, $k_i = \dfrac{\left[R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j) \right]}{E^{(i-1)}}$

3. Set $\alpha_i^{(i)} = k_i$ and $\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$ for $1 \le j \le i$

4. Calculate $E^{(i)} = \left(1 - k_i^2\right) E^{(i-1)}$

5. Repeat 2,3 and 4 until $i = p$ (the order of the LPC model)

Consider an example of obtaining the predictor coefficients for a predictor of order 2. The original matrix equation is of the form:

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

Using the Durbin method, we get the following steps:

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

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

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

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

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

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

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

$$\rightarrow \quad \alpha_1 = \alpha_1^2$$

$$\alpha_2 = \alpha_2^2$$

Durbin's Algorithm

The recursion allows the prediction of the i^{th} order filter coefficients from the $(i-1)^{th}$ order filter coefficients in such a way as to minimise the short-time average prediction error E.

$\alpha_k^{(i)}$ is the j^{th} predictor coefficient for a predictor order of i where k_i is the reflection coefficient for a predictor order i.

$E^{(i)}$ is the prediction error for a predictor of order i. Thus at each stage of the computation the prediction error for a predictor of order i can be monitored

Durbin's Algorithm

If the autocorrelation coefficients R(i) are replaced by a set of normalised autocorrelation coefficients R(i)/R(0), then the solution to the matrix equation remains unchanged.

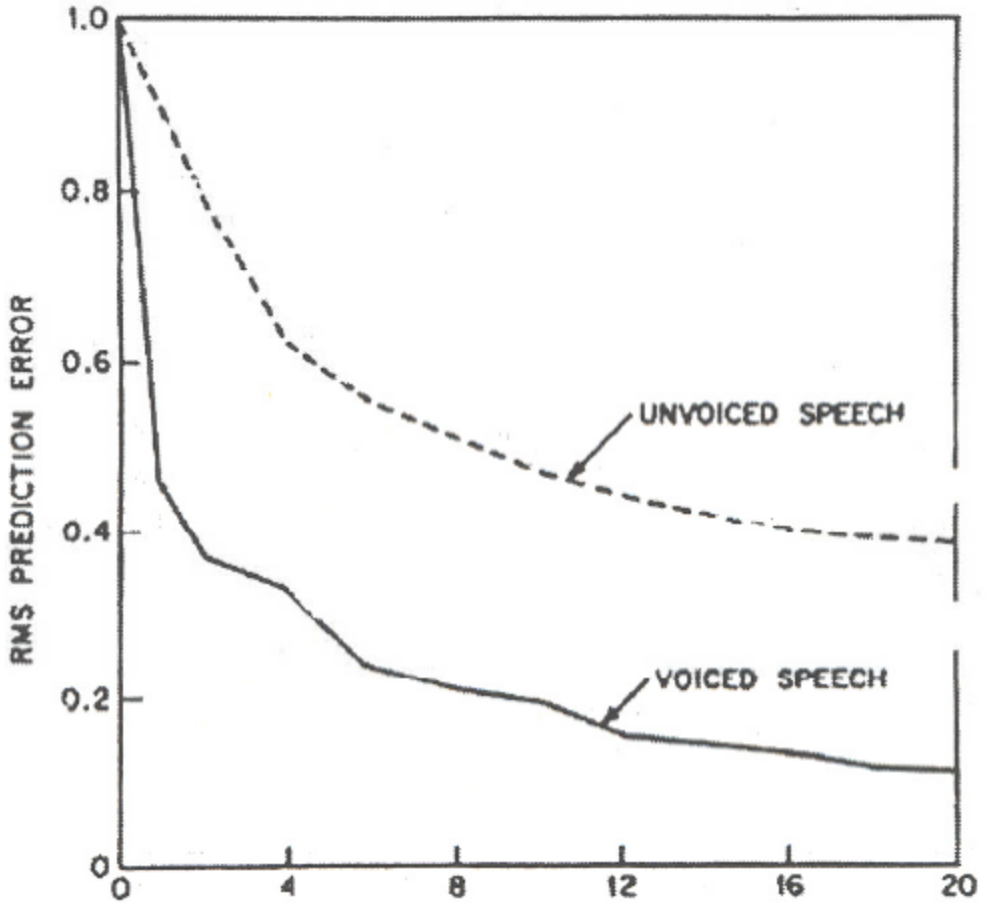
However, the error $E^{(i)}$ is now interpreted as a normalised error. If we now call this normalised error $V^{(i)}$, then

$$V^{(i)} = \frac{E^{(i)}}{R(0)}$$

Block Diagram of the LPC processor

- Briefly explain why pre-emphasis is required in the above diagram

- Briefly explain why Hamming window is required in the above scheme

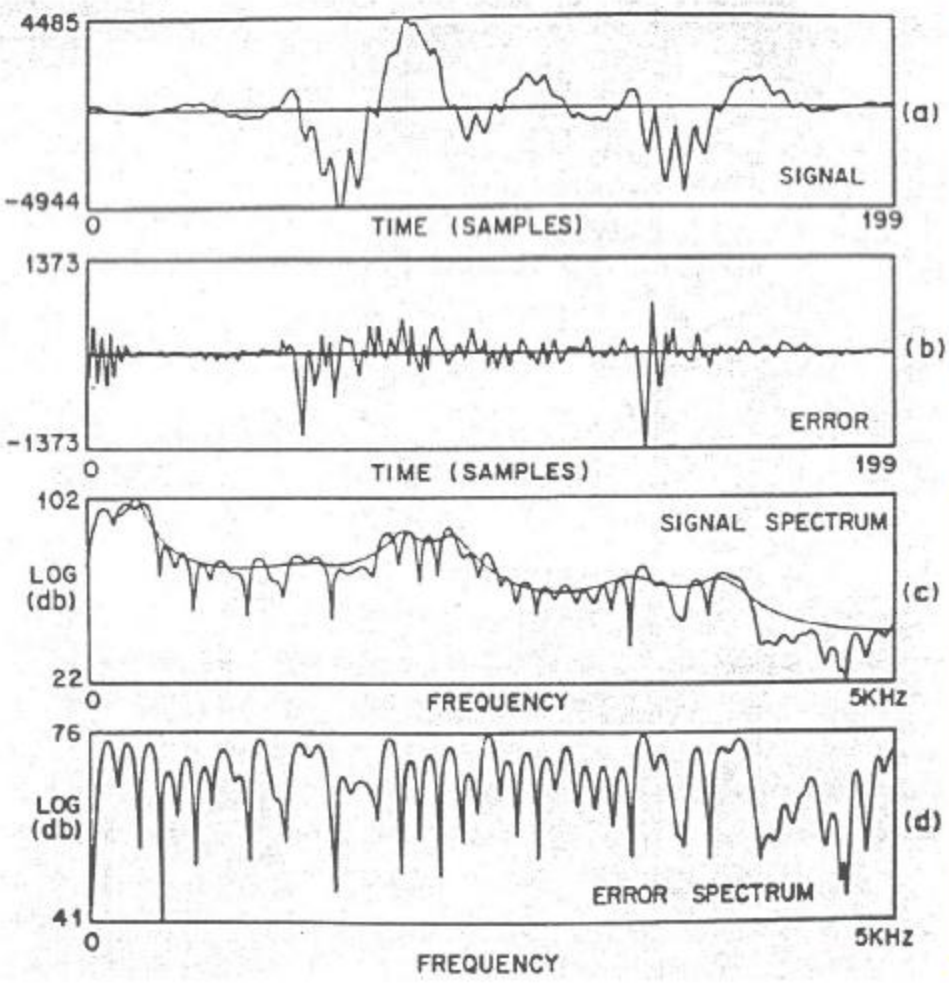


RMS prediction Error with the number of prediction coefficients p (Rabiner et al, 1993)

The normalised prediction error for unvoiced speech, for a given value of p, is significantly higher than for voiced speech.

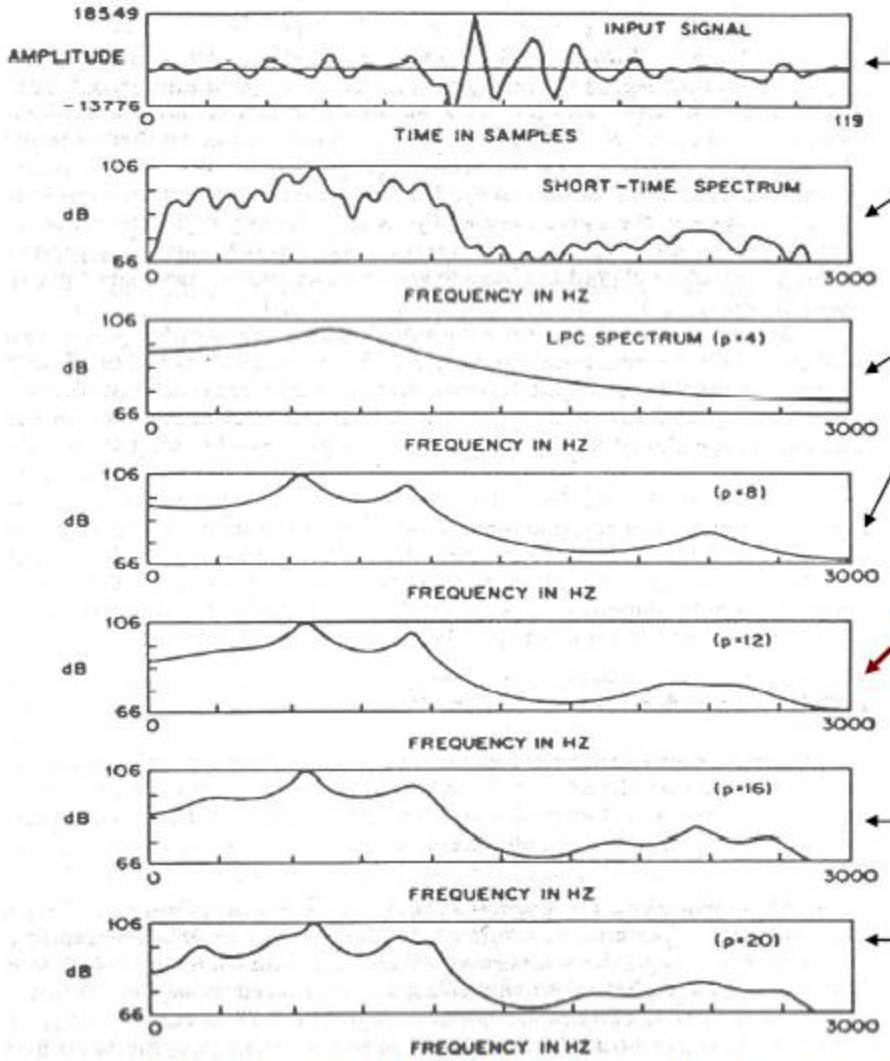
The interpretation of this results is that unvoiced speech is less linearly predictable than voiced speech

Examples of LPC analysis:



- A vowel spoken by a male speaker.
- Frame size = 20 ms. (200 samples at 10 kHz sampling rate)
- LPC analysis, p = 14

- Windowed speech (a)
- Prediction Error signal (b)
- Signal log spectrum (FFT-based) fitted by an LPC log spectrum(c)
- Log spectrum of the prediction error signal (d)



Spectra of vowel sound for several values of p

-Input speech segment

-Fourier Transform of that segment

-Liner predictive spectra of p from 4 to 20.

-It is clear that as p increases, more of the detailed properties of the signal spectrum are preserved in the LPC spectrum. When p become large, LPC spectrum often rises to fit individual pitch harmonics of the speech signal.

Reflection Coefficients

- Particularly in speech coding applications, it is common to "encode" the LPC model using "reflection coefficients (r_i)" rather than using the LPC coefficients (α_i)
- Reflection coefficients arise from the acoustic analysis of the vocal tract as a series of inter-connection cylindrical tubes with various cross sections

Reflection Coefficients

- The reflection coefficient between a tube with cross sectional area A_i and a tube with cross sectional area A_{i+1} is given by

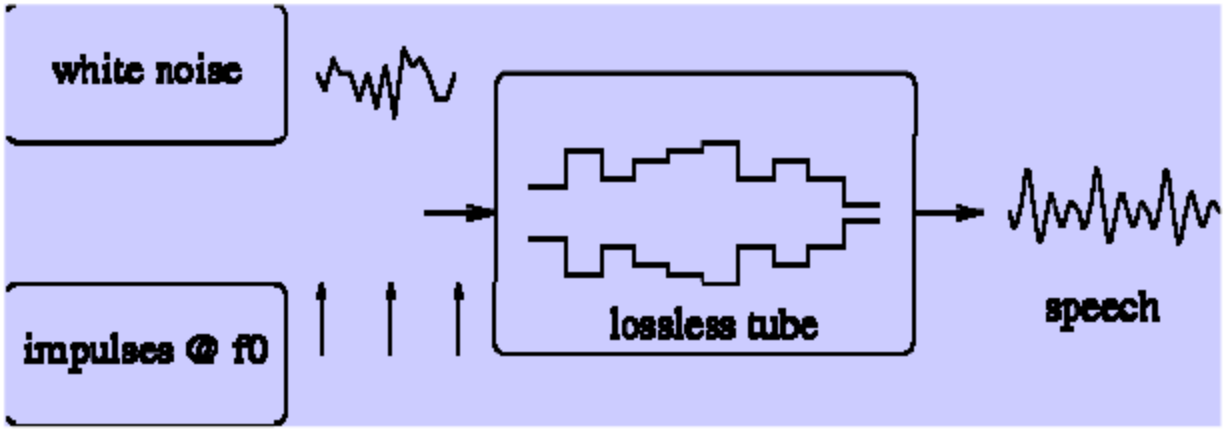
$$r_i = \frac{A_{i+1} - A_i}{A_{i+1} + A_i}$$

- r_i gives measure of how much energy is reflected back at each junction
- Note if two sections have the same area there is no relection

PARCOR Coefficients

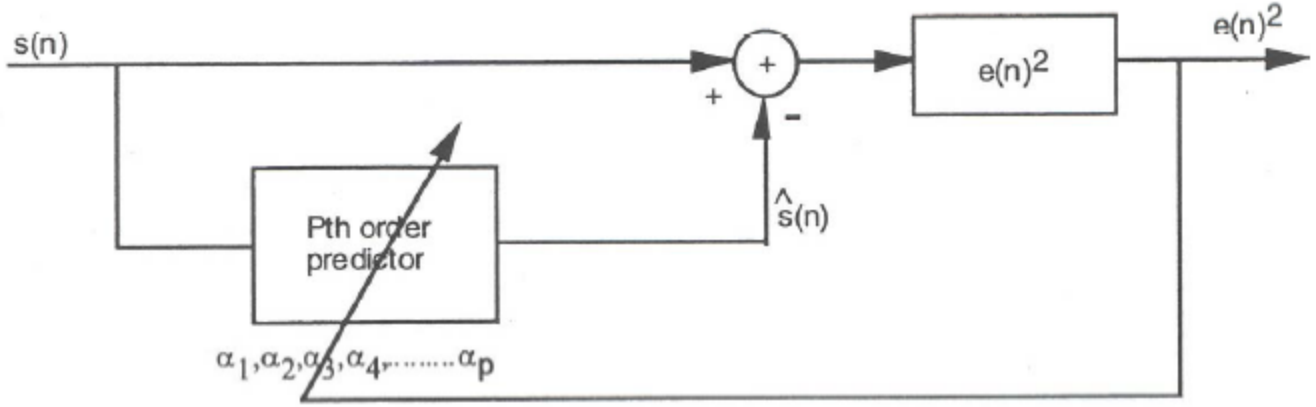
- However, it can be shown that they are related to the k_i values used in Durbin's algorithm (which are known as PARCOR coefficients) $r_i = -k_i$
- It can also be show that

$$A_{i+1} = \left[\frac{1-k_i}{1+k_i}\right] A_i$$



Steepest Descent Algorithm (for LPC calculation)

- A block diagram of a linear predictor is shown below



The predictor coefficients are adjusted continually during adaptation to reduce the squared prediction error $e(n)^2$ toward it's minimum value

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

Steepest Descent Algorithm

The updating of the predictor coefficients is carried out using the steepest descent algorithm. The predictor coefficients are updated on a sample by sample basis as follows:

$$\alpha_k(n+1) = \alpha_k(n) - c\frac{\partial[e(n)^2]}{\partial \alpha_k}$$

c = learning rate, $0 < c < 1$

$$\frac{\partial[e(n)^2]}{\partial \alpha_k} = 2e(n)\frac{\partial e(n)}{\partial \alpha_k} = -2\,e(n)s(n-k)$$

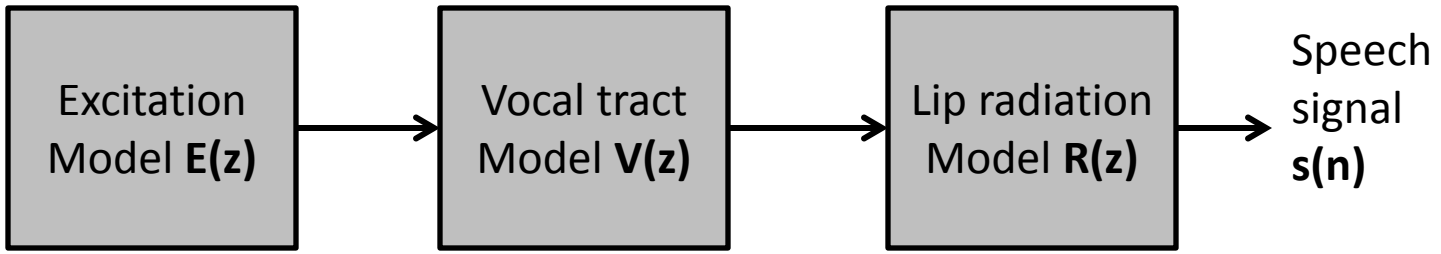
$$\boxed{\alpha_k(n+1) = \alpha_k(n) + c\,e(n)s(n-k)} \qquad 1 \le k \le p$$

Exercises on LPC

- Calculate LPC parameters of speech
  -Use LPC function and perform pre-emphasis
- Determination of formants using LPC
- Formant tracking using LPC
- Inverse filtering to determine e(n)
- Determination of pitch period from e(n)
- Pitch period Tracking

Exercise: Speech Synthesis

1. Generate an excitation train of impulses for a pitch of 100Hz. (Fs = 10000).
2. Use given coefficients for vocal tract V(z)
3. Apply this input to a model of the vocal tract including glottis model and lip radiation model and listen to the output



Exercise: Speech Synthesis – cont.

$$R(z) = 1 - z$$

$$G(z) = \frac{z^{-1}}{(1 - az^{-1})^2}$$ Where, a = 0.98

$$V(z) = \frac{1}{B(z)}$$

$$B(z) = 1 - 0.0460z^{-1} - 0.6232z^{-2} + 0.3814z^{-3}$$
$$+ 0.2443z^{-4} + 0.1973z^{-5} + 0.2873z^{-6} + 0.3655z^{-7}$$
$$- 0.4806z^{-8} - 0.1153z^{-9} + 0.7100z^{-10}$$

10th order LPC model