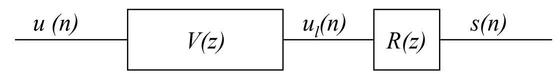
Lecture 5

Linear Prediction (LPC)

- Aims of Linear Prediction
- Derivation of Linear Prediction Equations
- Autocorrelation method of LPC
- Interpretation of LPC filter as a spectral whitener

Deller: 266++

Linear Prediction: Analysis & Coding (LPC)



$u(n) =$ glottal waveform, noise or mixture of the two

$u_l(n) =$ volume flow at the lips

$s(n) =$ pressure at the microphone

$$V(z) = \frac{Gz^{-\frac{1}{2}p}}{1-\sum_{j=1}^{p} a_j z^{-j}} = \frac{Gz^{-\frac{1}{2}p}}{A(z)}$$ a time-varying all-pole filter

$$R(z) = 1 - z^{-1}$$

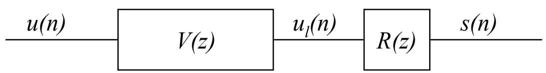
The aim of Linear Predication Analysis (LPC) is to estimate $V(z)$ from the speech signal $s(n)$.

Notes:

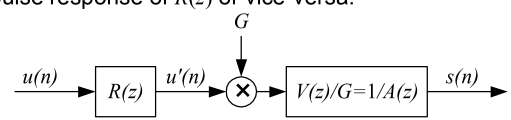
- We will neglect the pure delay term $z^{-\frac{1}{2}p}$ in the numerator of $V(z)$.
- 50% of the world puts a + sign in the denominator of $V(z)$ (this is almost essential when using MATLAB).

Deller: 266++

Prediction Error



We can reverse the order of $V(z)$ and $R(z)$ since both are linear and $V(z)$ doesn't change substantially during the impulse response of $R(z)$ or vice-versa:



$$s(n) = Gu'(n) + \sum_{j=1}^{p} a_j s(n-j)$$

If the vocal tract resonances have high gain, the second term will dominate:

$$s(n) \approx \sum_{j=1}^{p} a_j s(n-j)$$

The right hand side of this expression is a *prediction* of s(n) as a *linear sum* of past speech samples. Define the *prediction error* at sample n as

$$e(n) = s(n) - \sum_{j=1}^{p} a_j s(n-j) = s(n) - a_1 s(n-1) - a_2 s(n-2) - \ldots - a_p s(n-p)$$

or in terms of z – transforms: $E(z) = S(z)A(z)$

Given a frame of speech $\{F\}$, we would like to find the values a_i that minimize:

$$Q_E = \sum_{n \in \{F\}} e^2(n)$$

To do so, we differentiate w.r.t each a_i:

$$\frac{\partial Q_E}{\partial a_i} = \sum_{n \in \{F\}} \frac{\partial\left(e^2(n)\right)}{\partial a_i} = \sum_{n \in \{F\}} 2e(n)\frac{\partial e(n)}{\partial a_i} = -\sum_{n \in \{F\}} 2e(n)s(n-i)$$

The optimum values of a_i must satisfy p equations:

$$\sum_{n \in \{F\}} e(n)s(n-i) = 0 \quad \text{for} \quad i = 1, \ldots, p$$

$$\Rightarrow \sum_{n \in \{F\}} \left(s(n)s(n-i) - \sum_{j=1}^{p} a_j s(n-j)s(n-i) \right) = 0 \quad \text{for} \quad i = 1, \ldots, p$$

$$\Rightarrow \sum_{j=1}^{p} a_j \sum_{n \in \{F\}} s(n-j)s(n-i) = \sum_{n \in \{F\}} s(n)s(n-i)$$

$$\Rightarrow \sum_{j=1}^{p} \phi_{ij} a_j = \phi_{i0} \quad where \quad \phi_{ij} = \sum_{n \in \{F\}} s(n-i)s(n-j)$$

or in matrix form:

$$\mathbf{\Phi a} = \mathbf{c} \quad \Rightarrow \quad \mathbf{a} = \mathbf{\Phi}^{-1}\mathbf{c} \quad \text{providing } \mathbf{\Phi}^{-1} \text{ exists}$$

the matrix $\mathbf{\Phi}$ is symmetric and positive semi-definite.

Matrices with Special Properties

- Symmetric: $\phi_{ji} = \phi_{ij} \Leftrightarrow \mathbf{\Phi}^T = \mathbf{\Phi}$
- Positive Definite:

  $\sum_{i,j} x_i \phi_{ij} x_j > 0 \Leftrightarrow \mathbf{x}^T \mathbf{\Phi} \mathbf{x} > 0$ for any $\mathbf{x} \neq 0$
- Positive Semi-Definite: as above but with $\geq$.
- Toeplitz: Constant diagonals: $\phi_{i+1,j+1} = \phi_{ij} = f(i-j)$

Inverting Matrices

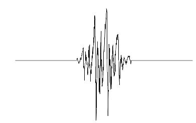
Any special properties possessed by a matrix can be used when inverting it to:

- reduce the computation time
- improve the accuracy

Matrix ($p \times p$)	**Computation**
General	$\propto p^3$
Symmetric, +ve definite	$\propto \frac{1}{2} p^3$
Toeplitz, Symmetric, +ve definite	$\propto p^2$

Autocorrelation LPC

We start with a frame of windowed speech (typ 20-30 ms):



We take $\{F\}$ to be infinite in extent $\phi_{ij} = \sum_{n=-\infty}^{+\infty} s(n-i)s(n-j)$

Because of the symmetry and the infinite sum, we have

$$\phi_{ij} = \phi_{|i-j|,0} = R_{|i-j|}$$

where the sequence R_k is the autocorrelation of the windowed speech.

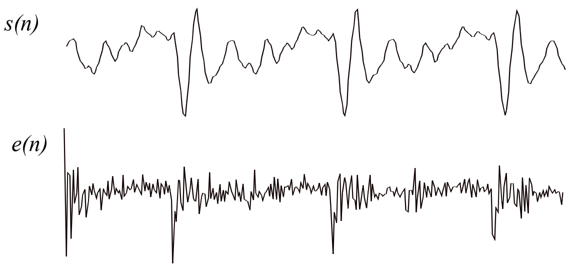
The matrix Φ is now Toeplitz (has constant diagonals) and the equations

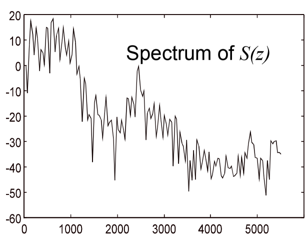
$$\mathbf{\Phi a} = \mathbf{c}$$

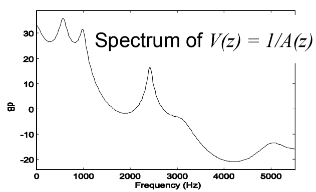
are called the *Yule-Walker* equations.

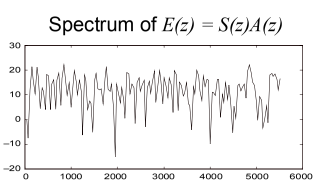
Inverting a symmetric, positive definite, Toeplitz $p \times p$ matrix takes $O(p^2)$ operations instead of the normal $O(p^3)$. Inversion procedure is known as the Levinson or Levinson-Durbin algorithm.

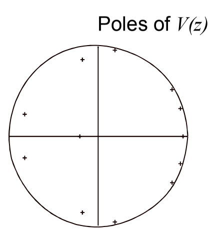
Autocorrelation LPC example: /ɑ/ from "father"











Spectral Flatness

Autocorrelation lpc finds the filter of the form

$$A(z) = 1 - a_1 z^{-1} - \ldots - a_p z^{-p}$$

that minimizes the energy of the prediction error. We will show that we can also interpret this in terms of flattening the spectrum of the error signal.

We define the normalised power spectrum of the prediction error signal $e(n)$ to be

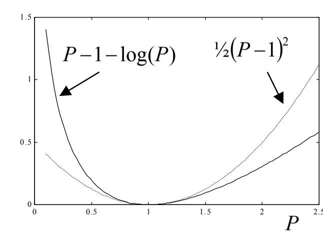
$$P_E(\omega) = \frac{\left|E\left(e^{j\omega}\right)\right|^2}{Q_E} \qquad Q_E = \sum e^2(n) = \frac{1}{2\pi}\int_{\omega=0}^{2\pi}\left|E\left(e^{j\omega}\right)\right|^2 d\omega$$

where $E(z)$ is the z-transform of the signal and Q_E is the signal energy. The average value of P_E is equal to 1.

We define the *spectral roughness* of the signal as:

$$R_E = \frac{1}{2\pi}\int_{\omega=0}^{2\pi} P_E(\omega) - 1 - \log\left(P_E(\omega)\right) d\omega$$

R_E is similar to the variance of P_E since the integrand is similar to $\tfrac{1}{2}(P_E - 1)^2$ where $\text{mean}(P_E) = 1$.



LPC.PPT(4/15/2002) 5.9

Spectral Flatness (continued)

We can find an alternative expression for R_E

$$R_E = \frac{1}{2\pi}\int_{\omega=0}^{2\pi} P_E(\omega) - 1 - \log(P_E(\omega))\, d\omega$$

$$= \frac{1}{2\pi}\int_{\omega=0}^{2\pi} -\log(P_E(\omega))\, d\omega \quad \text{since} \quad \int P_E(\omega)d\omega = 1$$

$$= \log(Q_E) - \frac{1}{2\pi}\int_{\omega=0}^{2\pi} \log\left(\left|E(e^{j\omega})\right|^2\right) d\omega$$

Thus the spectral roughness of a signal equals the difference between its log energy and the average of its log energy spectrum.

LPC.PPT(4/15/2002) 5.10

Spectral Flatness (continued)

We know that $E(z) = S(z) \times A(z)$, hence

$$\log\left(\left|E(e^{j\omega})\right|^2\right) = \log\left(\left|S(e^{j\omega})\right|^2\right) + \log\left(\left|A(e^{j\omega})\right|^2\right)$$

Substituting this in the expression for R_E gives

$$R_E = \log(Q_E) - \frac{1}{2\pi}\int_{\omega=0}^{2\pi} \log\left(\left|E(e^{j\omega})\right|^2\right) d\omega$$

$$= \log(Q_E) - \frac{1}{2\pi}\int_{\omega=0}^{2\pi} \log\left(\left|S(e^{j\omega})\right|^2\right) d\omega - \frac{1}{2\pi}\int_{\omega=0}^{2\pi} \log\left(\left|A(e^{j\omega})\right|^2\right) d\omega$$

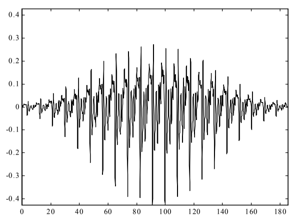
We saw in the section on filter properties that the term involving A is zero since $a_0=1$ and all roots of A lie in the unit circle. Hence

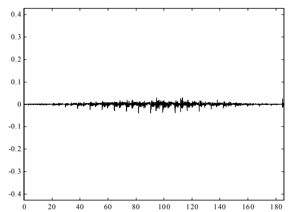
$$R_E = \log(Q_E) - \frac{1}{2\pi}\int_{\omega=0}^{2\pi} \log\left(\left|S(e^{j\omega})\right|^2\right) d\omega$$

The term involving S is independent of A. It follows that if A is chosen to minimize Q_E, it will also minimize R_E, the spectral roughness of $e(n)$. The filter $A(z)$ is called a whitening filter because it makes the spectrum flatter.

Spectral Flatness example

These two graphs show a windowed speech signal, /ɑ/, and the error signal after filtering by $A(z)$

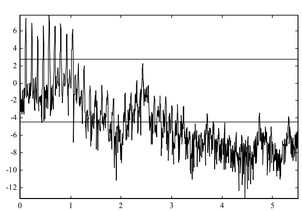


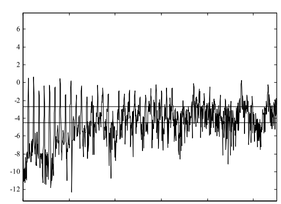


The two lower graphs show the log energy spectrum of each signal.

The two horizontal lines on each graph are the mean value (same for both graphs) and the log of the total energy.

The spectral roughness is the difference between the two.





Lecture 6

Linear Prediction (part 2)

- Covariance method of LPC
- Preemphasis
- Closed Phase Covariance LPC
- Alternative LPC parameter sets:
  - Pole positions
  - Reflection Coefficients
  - Log Area Ratios

Deller:309+

We consider two variants of LPC analysis which differ only in their choice of speech frame, $\{F\}$:

– Autocorrelation LPC Analysis

- Requires a windowed signal $\Rightarrow$ tradeoff between spectral resolution and time resolution
- Requires >20 ms of data
- Has a fast algorithm because Φ is toeplitz
- Guarantees a stable filter $V(z)$

– Covariance LPC Analysis (Prony's method)

- No windowing required
- Gives infinite spectral resolution
- Requires >2 ms of data
- Slower algorithm because Φ is not Toeplitz
- Sometimes gives an unstable filter $V(z)$

Deller: 292+ & 309+

Covariance LPC

From slide 5.4:

$$\sum_{j=1}^{p}\phi_{ij}a_j=\phi_{i0} \quad \text{where} \quad \phi_{ij}=\sum_{n\in\{F\}}s(n-i)s(n-j)$$

We chose $\{F\}$ to be a finite segment of speech: $\{F\}=s(n)$ for $0\le n\le (N\text{-}1)$ then we have:

$$\phi_{ij}=\sum_{n=0}^{N-1}s(n-i)s(n-j)$$

The matrix Φ is still symmetric but is no longer Toeplitz:

$$\phi_{ij}=\sum_{n=-1}^{N-2}s(n-i+1)s(n-j+1)$$
$$=s(-i)s(-j)-s(N-i)s(N-j)+\sum_{n=0}^{N-1}s(n-i+1)s(n-j+1)$$
$$=s(-i)s(-j)-s(N-i)s(N-j)+\phi_{i-1,j-1}$$

This allows the entire matrix Φ to be calculated recursively from its first row or column.

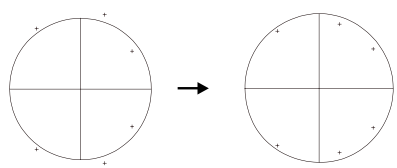
Since the matrix is not Toeplitz, the computation involved in inverting Φ is $\propto p^3$ rather than $\propto p^2$ and so takes longer.

Covariance LPC generally gives better results than Autocorrelation LPC but is more sensitive to the precise position of the frame in relation to the vocal fold closures.

Unstable Poles

Covariance LPC does not necessarily give a stable filter $V(z)$ (though it usually does).

We can force stability by replacing an unstable pole at $z = p$ by a stable one at $z = 1/p^*$.



As we have seen in the section on filter properties, reflecting a pole in the unit circle leaves the magnitude response unchanged except for multiplying by a constant (equal to the magnitude of the pole).

Thus the spectral flattening property of LPC is unaltered by this pole reflection.

Discovering which poles lie outside the unit circle is quite expensive: this is a further computational disadvantage of covariance LPC.

Haykin: Adaptive Filter Theoryp139, Deller:329

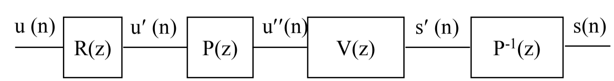
Preemphasis

The matrix Φ is always non-singular, but not necessarily by very much. A measure of how close a matrix is to being singular is given by its *condition number*: for a symmetric +ve definite matrix, this is the ratio of its largest to its smallest eigenvalue.

For large p, the condition number of Φ tends to the ratio $S_{max}(\omega)/S_{min}(\omega)$. We can thus improve the numerical properties of the LPC analysis procedure by flattening the speech spectrum before calculating the autocorrelation matrix Φ.

For *voiced* speech, the input to $V(z)$ is $u_g'(n)$ whose spectrum falls off at high frequencies at around –6dB/octave. This can be compensated with a 1st-order high-pass filter with a zero near z=1: $P(z) = 1 - \alpha z^{-1}$

$P(z)$ is approximately a differentiator. The normalised corner frequency of $P(z)$ is approximately $(1-\alpha)/2\pi$: This is typically placed in the range 0 to 150 Hz. From a spectral flatness point of view, the optimum value of α is ϕ_{10}/ϕ_{00} (obtained from autocorrelation LPC with $p = 1$).



Deller: 339

Closed-Phase Covariance LPC

From slide 5.3,

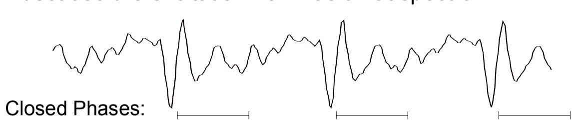
$$s(n) = Gu'(n) + \sum_{j=1}^{p} a_j s(n-j)$$

we have neglected the term $Gu'(n)$ because we don't know what it is and it is assumed to be much smaller than the second term.

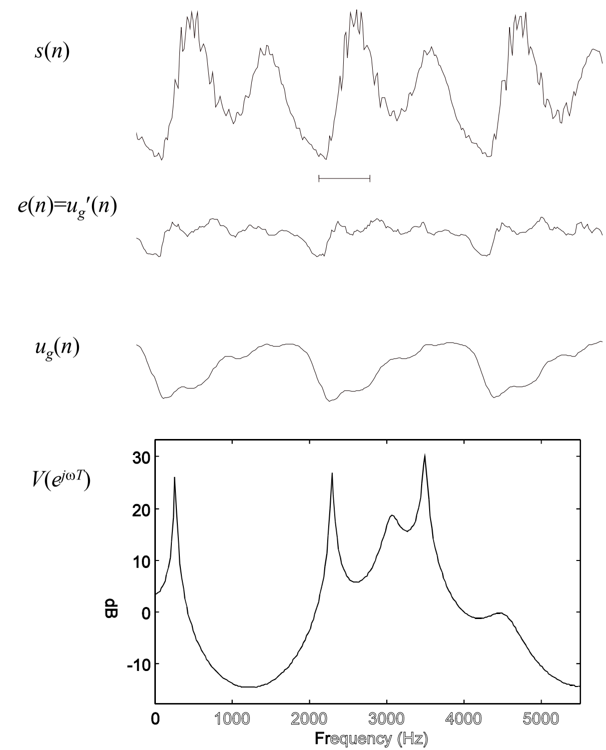
If we knew when the vocal folds were closed, we could restrict $\{F\}$ to those particular intervals. We can estimate the times of vocal fold closure in two ways:

- Looking for spikes in the $e(n)$ signal
- Using a Laryngograph (or Electroglottograph or EGG): this instrument measures the radio-frequency conductance across the larynx.
  - Conductance $\propto$ Vocal fold contact area.
  - Accurate but inconvenient.

In Closed-Phase LPC, we choose our analysis interval $\{F\}$ to consist of one or more closed phase intervals (not necessarily contiguous). No preemphasis is necessary because the excitation now has a flat spectrum.



Closed Phase Covariance LPC: /i/ from "bee"



Alternative Parameter Sets

The vocal tract filter is defined by $p+1$ parameters:

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

The LPC (or AR) coefficients a_k have some bad properties:

- The frequency response is very sensitive to small changes in a_k (such as quantizing errors in coding)
- There is no easy way to verify that the filter is stable
- Interpolating between the parameters that correspond to two different filters will not vary the frequency response smoothly from one to the other: stability is not even guaranteed.

There are several alternative parameter sets that are equivalent to the a_k (most require G to be specified as well):

<u>Pole Positions</u>

We can factorize the denominator of $V(z)$ to give its poles:

$$1 - \sum_{k=1}^{p} a_k z^{-k} = \prod_{k=1}^{p} \left(1 - x_k z^{-1}\right)$$

The polynomial roots x_k are either real or occur in complex conjugate pairs. $|x_k|$ must be <1 for stability. Factorizing polynomials is computationally expensive. The frequency response is sensitive to pole position errors near $|z|=1$.

<u>Reflection Coefficients of equivalent tube</u>

Any all-pole filter is equivalent to a tube with p sections: this is characterised by p reflection coefficients (assuming $r_0=1$). We can convert between the reflection coefficients and the polynomial coefficients by using the formulae given on slide 2.9.

Properties:

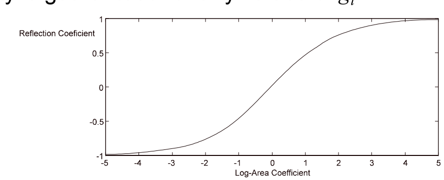
- An all-pole filter is stable iff the corresponding reflection coefficients all lie between -1 and +1.
- Interpolating between two of reflection coefficient sets will give a smoothly changing frequency response.
- High coefficient sensitivity near ±1.

The negative reflection coefficients are sometimes called the *PARCOR* coefficients (PARCOR = partial correlation).

<u>Log Area Ratios of equivalent tube</u>

$$g_i = \log\left(\frac{A_{i+1}}{A_i}\right) = \log\left(\frac{1+r_i}{1-r_i}\right) \quad \Leftrightarrow \quad r_i = \frac{e^{g_i} - 1}{e^{g_i} + 1} = \tanh(\tfrac{1}{2} g_i)$$

Stability is guaranteed for any values of g_i.



Lecture 7

Alternative LPC Parameter Sets

- Cepstral Coefficients
  - Relation to pole positions
  - Relation to LPC filter coefficients
- Line Spectrum Frequencies
  - Relation to pole positions and to formant frequencies
- Summary of LPC parameter sets

Most *speech recognisers* describe the spectrum of speech sounds using *cepstral coefficients*. This is because they are good at discriminating between different phonemes, are fairly independent of each other and have approximately Gaussian distributions for a particular phoneme.

Most *speech coders* describe the spectrum of speech sounds using *line spectrum frequencies*. This is because they can be quantised to low precision without distorting the spectrum too much.

Cepstral Coefficients: Calculating from x_k

Cepstrum :**inverse fourier transform of log spectrum (periodic spectrum ⇒ discrete cepstrum)**:

$$c_n = \frac{1}{2\pi} \int_{\omega=-\pi}^{+\pi} \log\left(V(e^{j\omega})\right) e^{j\omega n} d\omega$$

The coefficients c_n can be obtained directly from the x_k :

Define $C(z) = \sum_{n=-\infty}^{+\infty} c_n z^{-n} \quad \Rightarrow \quad c_n = \frac{1}{2\pi} \int_{\omega=-\pi}^{+\pi} C(e^{j\omega}) e^{j\omega n} d\omega$

This is the standard inverse z-transform derived by taking the inverse fourier transform of both sides of the first equation.

By equating the fourier transforms of the two expressions for c_n, we get

$$C(z) = \log\left(V(z)\right)$$
$$= \log\left(\frac{G}{A(z)}\right) = \log(G) - \log\left(A(z)\right)$$

where $A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} = \prod_{k=1}^{p}\left(1 - x_k z^{-1}\right)$

By using the Taylor series

$$\log(1-y) = -\sum_{n=1}^{\infty} \frac{y^n}{n} \quad \text{for } |y| < 1$$

$$C(z) = \log(G) - \log(A(z))$$
$$= \log(G) - \sum_{k=1}^{p} \log\left(1 - x_k z^{-1}\right)$$
$$= \log(G) + \sum_{k=1}^{p} \sum_{n=1}^{\infty} \frac{x_k^n}{n} z^{-n}$$

By collecting all the terms in z^{-n}, we can get c_n in terms of x_k:

$$c_n = \begin{cases} 0 & \text{for } n < 0 \\ \log(G) & \text{for } n = 0 \\ \sum_{k=1}^{p} \frac{x_k^n}{n} & \text{for } n > 0 \end{cases}$$

Because $|x_k| < 1$ the c_n decrease exponentially with n.

Cepstral Coefficients: Calculating from a_k

Differentiating $C(z) = \log(G) - \log(A(z))$ with respect to z:

$$C'(z) = \frac{-A'(z)}{A(z)} \quad \Rightarrow \quad A(z)C'(z) = -A'(z)$$
$$\Rightarrow \quad A(z)zC'(z) = -zA'(z)$$

This gives:

$$\left(1 - \sum_{k=1}^{p} a_k z^{-k}\right)\left(z \sum_{m=0}^{\infty} -mc_m z^{-(m+1)}\right) = -z \sum_{n=1}^{p} + na_n z^{-(n+1)}$$

$$\Rightarrow \left(1 - \sum_{k=1}^{p} a_k z^{-k}\right)\left(\sum_{m=1}^{\infty} mc_m z^{-m}\right) = \sum_{n=1}^{p} na_n z^{-n}$$

$$\Rightarrow \sum_{n=1}^{\infty} nc_n z^{-n} - \sum_{k=1}^{p} \sum_{m=1}^{\infty} mc_m a_k z^{-(m+k)} = \sum_{n=1}^{p} na_n z^{-n}$$

replacing m by n-k (to make the z exponent uniform) gives:

$$\Rightarrow \sum_{n=1}^{\infty} nc_n z^{-n} = \sum_{n=1}^{p} na_n z^{-n} + \sum_{k=1}^{p} \sum_{n=k+1}^{\infty} (n-k) c_{(n-k)} a_k z^{-n}$$

now take the coefficient of z^{-n} in the above equation noting that $n \ge k+1 \Rightarrow k \le n-1$:

$$nc_n = na_n + \sum_{k=1}^{\min(p,n-1)} (n-k) c_{(n-k)} a_k$$

$$\Rightarrow \quad c_n = a_n + \frac{1}{n} \sum_{k=1}^{\min(p,n-1)} (n-k) c_{(n-k)} a_k$$

Thus we have a recurrence relation to calculate the c_n from the a_k coefficients:

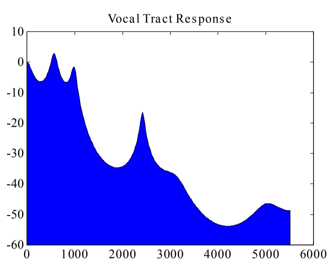
$$c_n = a_n + \frac{1}{n}\sum_{k=1}^{\min(p,n-1)} (n-k)c_{(n-k)}a_k$$

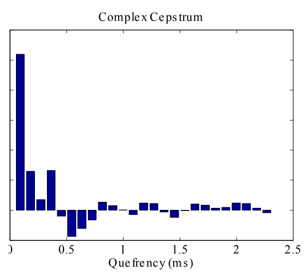
From this we get:

$$c_1 = a_1$$
$$c_2 = a_2 + \tfrac{1}{2}c_1 a_1$$
$$c_3 = a_3 + \tfrac{1}{3}\left(2c_2 a_1 + c_1 a_2\right)$$
$$c_4 = a_4 + \tfrac{1}{4}\left(3c_3 a_1 + 2c_2 a_2 + c_1 a_3\right)$$
$$c_5 = \cdots$$

These coefficients are called the *complex cepstrum* coefficients (even though they are real). The *cepstrum* coefficients use $\log|V|$ instead of $\log(V)$ and (except for c_0) are half as big.

Note the cute names: spectrum→cepstrum, frequency→quefrency, filter→lifter, etc





Deller: 331

Line Spectrum Frequencies (LSFs)

$$A(z) = G \times V^{-1}(z) = 1 - \sum_{j=1}^{p} a_j z^{-j} = 1 - a_1 z^{-1} - a_2 z^{-2} - \ldots - a_p z^{-p}$$

We can form symmetric and antisymmetric polynomials:

$$P(z) = A(z) + z^{-(p+1)}A^*(z^{*-1})$$ (see slide 4.10)

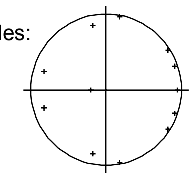
$$= 1 - (a_1 + a_p)z^{-1} - (a_2 + a_{p-1})z^{-2} - \ldots - (a_p + a_1)z^{-p} + z^{-(p+1)}$$

$$Q(z) = A(z) - z^{-(p+1)}A^*(z^{*-1})$$

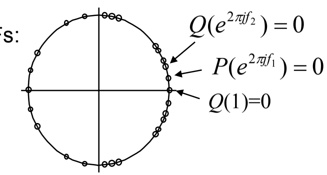
$$= 1 - (a_1 - a_p)z^{-1} - (a_2 - a_{p-1})z^{-2} - \ldots - (a_p - a_1)z^{-p} - z^{-(p+1)}$$

$V(z)$ is **stable** if and only if the roots of $P(z)$ and $Q(z)$ **all lie on the unit circle** and **they are interleaved**.

Poles:



LSFs:



If the roots of $P(z)$ are at $\exp(2\pi j f_i)$ for i=1,3,… and those of $Q(z)$ are at $\exp(2\pi j f_i)$ for i=0,2,… with $f_{i+1} > f_i \geq 0$ then the LSF frequencies are defined as $f_1, f_2, \ldots, f_p$.

Note that it is always true that f_0=+1 and f_{p+1}=–1

E.g. $A(z) = 1 - 0.7z^{-1} + 0.5z^{-2}$ $\quad P(z) = 1 - 0.2z^{-1} - 0.2z^{-2} + z^{-3}$

$z^{-3}A^*(z^{*-1}) = 0.5z^{-1} - 0.7z^{-2} + z^{-3}$ $\quad Q(z) = 1 - 1.2z^{-1} + 1.2z^{-2} - z^{-3}$

Proof that roots of P(z) and Q(z) lie on the unit circle

$$P(z)=0 \quad \Leftrightarrow \quad A(z)=-z^{-(p+1)}A^*(z^{*-1}) \quad \Leftrightarrow \quad H(z)=-1$$

$$Q(z)=0 \quad \Leftrightarrow \quad A(z)=+z^{-(p+1)}A^*(z^{*-1}) \quad \Leftrightarrow \quad H(z)=+1$$

$$\text{where } H(z)=\frac{A(z)}{z^{-(p+1)}A^*(z^{*-1})}=z\prod_{i=1}^{p}\frac{(1-x_i z^{-1})}{z^{-1}(1-x_i^* z)}=z\prod_{i=1}^{p}\frac{(z-x_i)}{(1-x_i^* z)}$$

here the x_i are the roots of $A(z)=V^{-1}(z)$.

It turns out that providing all the x_i lie inside the unit circle, the absolute values of the terms making up $H(z)$ are either all > 1 or else all < 1. Taking | | of a typical term:

$$\left|\frac{(z-x_i)}{(1-x_i^* z)}\right|>1 \quad \Leftrightarrow \quad |1-x_i^* z|<|z-x_i|$$

$$\Leftrightarrow \quad (1-x_i^* z)(1-x_i^* z)^* < (z-x_i)(z-x_i)^*$$

$$\Leftrightarrow \quad (1-x_i^* z)(1-x_i z^*) < (z-x_i)(z^*-x_i^*)$$

$$\Leftrightarrow \quad 1-x_i^* z-x_i z^*+x_i x_i^* zz^* < zz^*-x_i^* z-x_i z^*+x_i x_i^*$$

$$\Leftrightarrow \quad 1-x_i x_i^*-zz^*+x_i x_i^* zz^* < 0$$

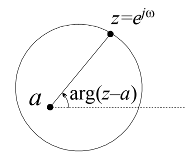
$$\Leftrightarrow \quad (1-|x_i|^2)(1-|z|^2)<0 \quad \Leftrightarrow \quad |z|>1 \quad \text{since each } |x_i|<1$$

Thus each term is greater or less than 1 according to whether $|z|>1$ or $|z|<1$. Hence $|H(z)|=1$ if and only if $|z|=1$ and so the roots of $P(z)$ and $Q(z)$ must lie on the unit circle.

Proof that the roots of P(Z) and Q(z) are interleaved

We want to find the values of $z = e^{j\omega}$ that make $H(z) = \pm 1$ or equivalently that make $\arg(H(z))$ = a multiple of π.

If $z = e^{j\omega}$ then $\arg(H(e^{j\omega}))=\arg\left(e^{j(1-p)\omega}\prod_{i=1}^{p}\frac{(e^{j\omega}-x_i)}{(e^{-j\omega}-x_i^*)}\right)$



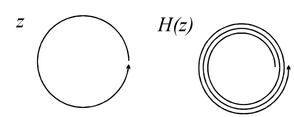
$$=(1-p)\omega+\sum_{i=1}^{p}\left(\arg(e^{j\omega}-x_i)-\arg(e^{-j\omega}-x_i^*)\right)$$

$$=(1-p)\omega+2\sum_{i=1}^{p}\arg(e^{j\omega}-x_i)$$

As ω goes from 0 to 2π, $\arg(z-a)$ changes monotonically by $+2\pi$ if $|a|<1$.

Therefore as ω goes from 0 to 2π, $\arg(H(e^{j\omega}))$ increases by

$$(1-p)\times 2\pi+2p\times 2\pi=(1+p)\times 2\pi$$



Since $H(e^{j\omega})$ goes round the unit circle (1+p) times, it must pass through each of the points +1 and –1 alternately (1+p) times.

$\arg(H(z))$ varies most rapidly when z is near one of the x_i so the LSF frequencies will cluster near the formants.

Summary of LPC parameter sets

Filter Coefficients: a_i

- – Stability check difficult; Sensitive to errors; Cannot interpolate

Pole Positions: x_i

- + Stability check easy; Can interpolate but unordered.
- – Hard to calculate; Sensitive to errors near $|x_i|=1$

Reflection Coefficients: r_i

- + Stability check easy; Can interpolate
- – Sensitive to errors near ± 1

Log Area Ratios: g_i

- + Stability guaranteed; Can interpolate

Cepstral Coefficients : c_i

- + Good for speech recognition
- – Stability check difficult

Line Spectrum Frequencies: f_i

- + Stability check easy; Can interpolate; Vary smoothly in time; Strongly correlated $\Rightarrow$ better coding; Related to spectral peaks (formants).
- – Awkward to calculate