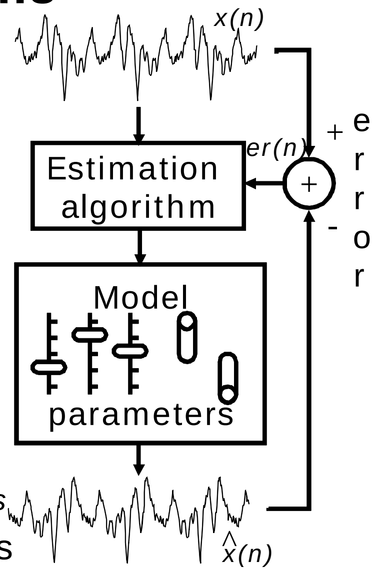
PART II (short-term) Modeling of the speech signal

Families of speech models

- ***Articulatory*** models (parameters = position of tongue, glottis, lip opening, etc.)
- ***Production*** models (electical analogy of vocal tract: combination of electrical generators and filters; parameters = switches and coefs. of filters)
- ***Phenomenological*** models (pure signal processing techniques for modeling the speech signal or its spectrum: FFT, wavelets, time-domain processing, etc.)

Definitions

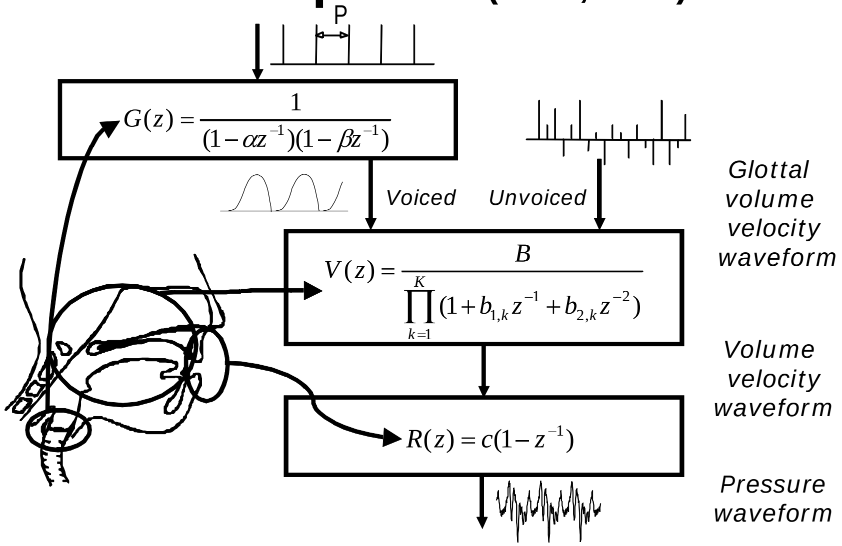
- Speech processing is based on ***speech models***
- Models have ***parameters***, like black boxes have switches and sliders
- Parameters are ***estimated*** via ***algorithms***
- Errors : output of model ≠ input signal
  - ***Modeling*** *(intrinsic) errors*
  - ***Estimation*** *(extrinsic) errors*
- Algorithms ***minimize*** errors



Contents

- Production models
  - **Autoregressive modeling of speech**
  - Autoregressive modeling of stationary random signals
    - AR model - Estimation algorithms
  - Autoregressive estimation of speech
  - Extensions of the AR model
- Transform-based models (*phenomenological*)
  - Short-term Fourier transform
    - Definition - Interpretation
    - Relationship with AR modeling
    - Hybrid harmonic+noise model - Estimation
  - Homomorphic (or *cepstral*) transform
    - Definition - Properties
    - Relationship with AR modeling
- F0 estimation

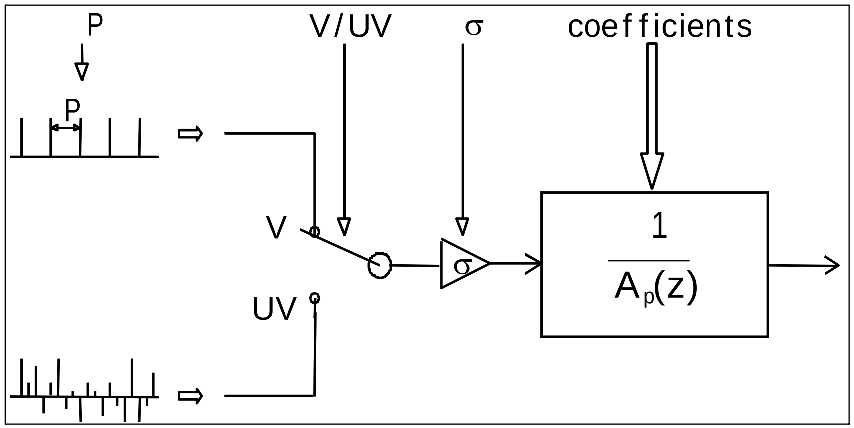
Autoregressive modeling of speech (Fant, 50's)



Contents

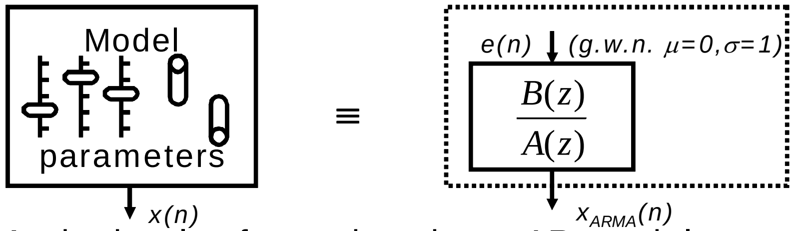
Autoregressive modeling of speech (Fant, 50's)

G(z)V(z)R(z) ≈σ /A$_p$(z) : « All pole » model

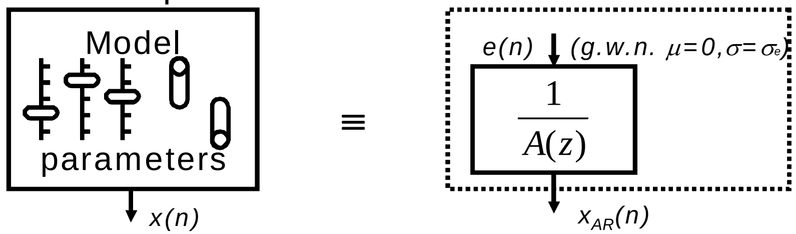


Autoregressive modeling of stationary random signals

- ARMA (autoregressive-moving average)



- Much simpler for estimation : AR model



Autoregressive modeling of stationary random signals

$$e(n) \longrightarrow \boxed{\frac{1}{A(z)}} \longrightarrow x(n)$$

- Each sample x(n) is the sum of a completely (linearly) predictable component, and an « innovation » component e(n)
- ⇒ ***Autoregressive*** or ***linear prediction*** (LP) model

$$X(z) = \frac{E(z)}{\sum_{i=1}^{p} a_i z^{-i}}$$

$$x(n) = e(n) + \sum_{i=1}^{p} - a_i x(n-i)$$

$$e(n) = \sum_{i=0}^{p} a_i x(n-i) \qquad (a_0 = 1)$$

- e(n) can be computed from a_i (i=1...p) and x(n)
- e(n) is the output of the ***inverse filter***

$$x(n) \longrightarrow \boxed{A(z)} \longrightarrow e(n)$$

YULE-WALKER equations

$$\sum_{j=1}^{p} \phi_x(i-j) a_j = -\phi_x(i) \quad (i = 1 \ldots p)$$

$$\begin{bmatrix} \phi_x(0) & \phi_x(1) & \ldots & \phi_x(p-1) \\ \phi_x(1) & \phi_x(0) & \ldots & \phi_x(p-2) \\ \ldots & \ldots & \ldots & \ldots \\ \phi_x(p-1) & \phi_x(p-2) & \ldots & \phi_x(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \ldots \\ a_p \end{bmatrix} = - \begin{bmatrix} \phi_x(1) \\ \phi_x(2) \\ \ldots \\ \phi_x(p) \end{bmatrix}$$

$$\mathbf{\Phi_x^{p-1} a} = -\boldsymbol{\varphi}_\mathbf{x}^\mathbf{p}$$

- *p* linear equations with *p* unknowns ⇒ O(p^3)?
- Toeplitz matrix ⇒ O(p^2) : recursively on the prediction order

LEVINSON **SCHUR(-LEROUX-GUEGEN)**

YULE-WALKER equations

$$\sigma_e^2 = E\left[e^2(n)\right]$$

$$= E\left[\sum_{i=0}^{p} a_i x(n-i) \sum_{j=0}^{p} a_j x(n-j)\right]$$

$$= \sum_{i,j=0}^{p} a_i a_j E\left[x(n-i)x(n-j)\right]$$

$$= \sum_{i,j=0}^{p} a_i a_j \phi_x(i-j)$$

$$\frac{\partial \sigma_e^2}{\partial a_i} = 0 \quad (i = 1 \ldots p)$$

$$2\sum_{j=0}^{p} \phi_x(i-j) a_j = 0$$

$$\boxed{\sum_{j=1}^{p} \phi_x(i-j) a_j = -\phi_x(i) \quad (i = 1 \ldots p)}$$

- **Problem** : find a_i (i=1...p) from x(n), not knowing e(n)
- Minimize the variance σ_e^2 of e(n)
- YULE-WALKER equations

LEVINSON algorithm

$$\begin{bmatrix} \phi_x(0) & \phi_x(1) & \ldots & \phi_x(m-1) \\ \phi_x(1) & \phi_x(0) & \ldots & \phi_x(m-2) \\ \ldots & \ldots & \ldots & \ldots \\ \phi_x(m-1) & \phi_x(m-2) & \ldots & \phi_x(0) \end{bmatrix} \begin{bmatrix} a_1^m \\ a_2^m \\ \ldots \\ a_m^m \end{bmatrix} = - \begin{bmatrix} \phi_x(1) \\ \phi_x(2) \\ \ldots \\ \phi_x(m) \end{bmatrix}$$

Find solution for order *m*+1 from sol. for *m?*

?

$$\begin{bmatrix} \phi_x(0) & \phi_x(1) & \ldots & \phi_x(m-1) & \phi_x(m) \\ \phi_x(1) & \phi_x(0) & \ldots & \phi_x(m-2) & \phi_x(m-1) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \phi_x(m-1) & \phi_x(m-2) & \ldots & \phi_x(0) & \phi_x(1) \\ \phi_x(m) & \phi_x(m-1) & \ldots & \phi_x(1) & \phi_x(0) \end{bmatrix} \begin{bmatrix} a_1^{m+1} \\ a_2^{m+1} \\ \ldots \\ a_m^{m+1} \\ a_{m+1}^{m+1} \end{bmatrix} = - \begin{bmatrix} \phi_x(1) \\ \phi_x(2) \\ \ldots \\ \phi_x(m) \\ \phi_x(m+1) \end{bmatrix}$$

(1)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_1^m\\ a_2^m\\ \dots\\ a_m^m\\ 0\end{bmatrix}=-\begin{bmatrix}\phi_x(1)\\ \phi_x(2)\\ \dots\\ \phi_x(m)\\ \phi_x(m+1)+\mu\end{bmatrix}$$

$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}1\\ a_1^m\\ \dots\\ a_{m-1}^m\\ a_m^m\end{bmatrix}=-\begin{bmatrix}\alpha\\ 0\\ \dots\\ 0\\ 0\end{bmatrix}$$

Toeplitz

(2)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_m^m\\ a_{m-1}^m\\ \dots\\ a_1^m\\ 1\end{bmatrix}=-\begin{bmatrix}0\\ 0\\ \dots\\ 0\\ \alpha\end{bmatrix}$$

(1)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_1^m\\ a_2^m\\ \dots\\ a_m^m\\ 0\end{bmatrix}=-\begin{bmatrix}\phi_x(1)\\ \phi_x(2)\\ \dots\\ \phi_x(m)\\ \phi_x(m+1)+\mu\end{bmatrix}$$

(2)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_m^m\\ a_{m-1}^m\\ \dots\\ a_1^m\\ 1\end{bmatrix}=-\begin{bmatrix}0\\ 0\\ \dots\\ 0\\ \alpha\end{bmatrix}$$

(3)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_1^{m+1}\\ a_2^{m+1}\\ \dots\\ a_m^{m+1}\\ a_{m+1}^{m+1}\end{bmatrix}=-\begin{bmatrix}\phi_x(1)\\ \phi_x(2)\\ \dots\\ \phi_x(m)\\ \phi_x(m+1)\end{bmatrix}$$

(3) = (1) + k_{m+1}(2) ?

$$\boxed{a_i^{m+1}=a_i^m+k_{m+1}a_{m+1-i}^m \qquad a_{m+1}^{m+1}=k_{m+1}}$$

(1)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_1^m\\ a_2^m\\ \dots\\ a_m^m\\ 0\end{bmatrix}=-\begin{bmatrix}\phi_x(1)\\ \phi_x(2)\\ \dots\\ \phi_x(m)\\ \phi_x(m+1)+\mu\end{bmatrix}$$

(2)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_m^m\\ a_{m-1}^m\\ \dots\\ a_1^m\\ 1\end{bmatrix}=-\begin{bmatrix}0\\ 0\\ \dots\\ 0\\ \alpha\end{bmatrix}$$

(3)
$$\begin{bmatrix}\phi_x(0)&\phi_x(1)&\dots&\phi_x(m-1)&\phi_x(m)\\ \phi_x(1)&\phi_x(0)&\dots&\phi_x(m-2)&\phi_x(m-1)\\ \dots&\dots&\dots&\dots&\dots\\ \phi_x(m-1)&\phi_x(m-2)&\dots&\phi_x(0)&\phi_x(1)\\ \phi_x(m)&\phi_x(m-1)&\dots&\phi_x(1)&\phi_x(0)\end{bmatrix}\begin{bmatrix}a_1^{m+1}\\ a_2^{m+1}\\ \dots\\ a_m^{m+1}\\ a_{m+1}^{m+1}\end{bmatrix}=-\begin{bmatrix}\phi_x(1)\\ \phi_x(2)\\ \dots\\ \phi_x(m)\\ \phi_x(m+1)\end{bmatrix}$$

(3) = (1) + k_{m+1}(2) ?

$$\boxed{\phi_x(m+1)+\mu_m+k_{m+1}\alpha_m=\phi_x(m+1) \qquad k_{m+1}=-{}^{\mu_m}\!/_{\alpha_m}}$$

LEVINSON algorithm

- Initialization

$$a_m(0) = 1 \;,\; m = 1, 2, \dots, p \qquad \alpha_0 = \phi_x(0) = \sigma_x^{\,2}$$

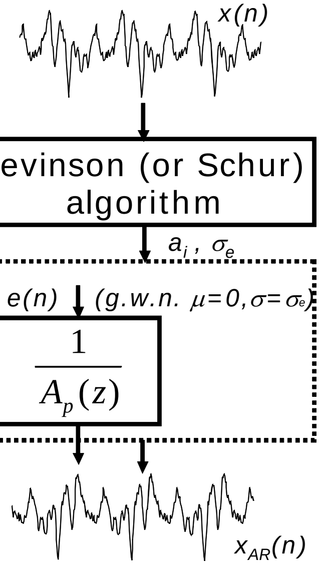
- Recursion on m=0...p-1

$$\textit{for } m = 0, 1, \dots, p-1$$
$$k_{m+1} = -(1/\alpha_m)\,.\sum_{i=0}^{m-1} a_{m-1}(i)\,.\,\phi_x(m-i)$$
$$\textit{for } i = 1, 2, \dots, m-1$$
$$a_m(i) = a_{m-1}(i) + k_m\,.\,a_{m-1}(m-i)$$
$$a_{m+1}(m) = k_{m+1}$$
$$\alpha_{m+1} = \alpha_m(1-k^2{}_{m+1})$$

k_i : PARCOR coefficients

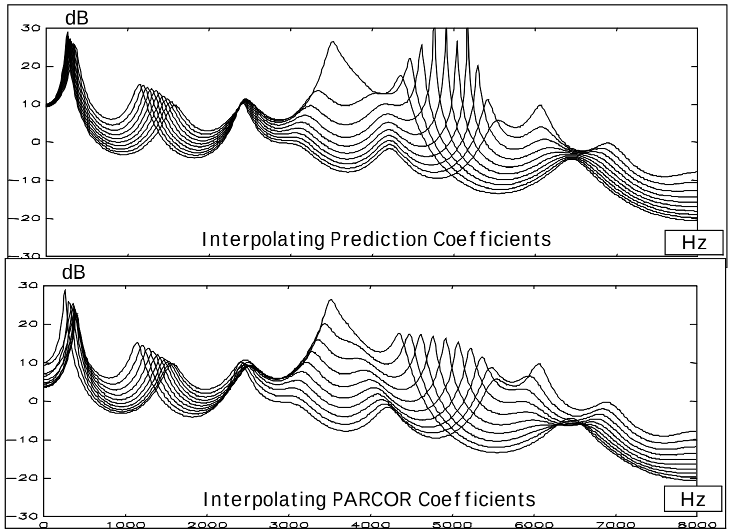
- Naturally appear in the LEVINSON recursion
- Can be **physically** interpreted as area ratios of acoustic tubes in series
- Equivalent to prediction coefficients (there is a recursion formula for $\{a_i\}\rightarrow\{k_i\}\rightarrow\{a_i\}$
- Transfer function of 1/A(z) much less sensitive to Δk_i then to Δa_i
- Good **interpolation** properties
- Correspond to the ***lattice filter structure*** for 1/A(z)
- 1/A(z) stable if $-1<k_i<+1$

How good are the a_i 's or k_i 's ?



$$x \overset{?}{\equiv} x_{AR}$$

Interpolating PARCORs



How good are the a_i 's or k_i 's ?

$$\phi_x(k) \equiv \phi_{AR}(k)$$
$$\text{for} \quad k = 0 \ldots p$$

Contents

- Production models
  - Autoregressive modeling of speech
  - Autoregressive modeling of stationary random signals
    - AR model - Estimation algorithms
  - **Autoregressive estimation of speech**
  - Extensions of the AR model
- Transform-based models (*phenomenological*)
  - Short-term Fourier transform
    - Definition - Interpretation
    - Relationship with AR modeling
    - Hybrid harmonic+noise model - Estimation
  - Homomorphic (or *cepstral*) transform
    - Definition - Properties
    - Relationship with AR modeling
- F0 estimation

Yule-Walker equations(again)

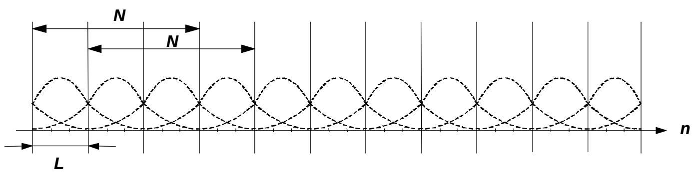
$$E_p = \sum_{n=-\infty}^{+\infty} e^2(n)$$
$$= \sum_{n=-\infty}^{+\infty}\left[\sum_{i=0}^{p} a_i x(n-i) \sum_{j=0}^{p} a_j x(n-j)\right]$$
$$= \sum_{i,j=0}^{p} a_i a_j \sum_{n=-\infty}^{+\infty}\left[x(n-i)x(n-j)\right]$$
$$= \sum_{i,j=0}^{p} a_i a_j r_x(i-j)$$
$$\frac{\partial E_p}{\partial a_i} = 0 \qquad (i=1 \ldots p)$$
$$2\sum_{j=0}^{p} r_x(i-j) a_i = 0$$
$$\boxed{\sum_{j=1}^{p} r_x(i-j) a_i = -r_x(0) \qquad (i=1 \ldots p)}$$

- Minimize the energy E_p of the prediction error *e(n)*
- Same formalism as in the stationnary case, applied here to a signal which takes non-zero values only for *n*=0..*N*-1
- Same Yule-Walker equations, except ϕ_x is estimated as r_x

$$r_x(k) = \sum_{n=-\infty}^{+\infty}\left[x(n)x(n-k)\right]$$

Short-term analysis of speech

- Speech is not stationnary (if it were, information ≡ 0)
- *« Pseudo-stationnary »* on 30 ms
- Frame-by-frame (typ.: N=30ms/L=10ms)



- ex: Fs=10kHz, L=100 samples, N=300 samples

Yule-Walker equations(again)

$$\sum_{j=1}^{p} r_x(i-j) a_i = -r_x(0) \qquad (i=1 \ldots p)$$

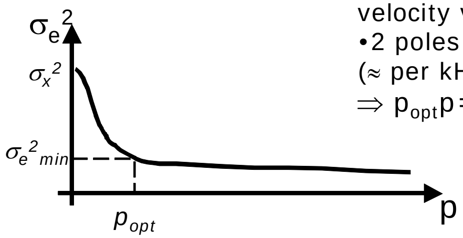
$$\begin{bmatrix} r_x(0) & r_x(1) & \ldots & r_x(p-1) \\ r_x(1) & r_x(0) & \ldots & r_x(p-2) \\ \ldots & \ldots & \ldots & \ldots \\ r_x(p-1) & r_x(p-2) & \ldots & r_x(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \ldots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_x(1) \\ r_x(2) \\ \ldots \\ r_x(p) \end{bmatrix}$$

$$\mathbf{R}_\mathbf{x}^{\mathbf{p-1}}\mathbf{a} = -\mathbf{r}_\mathbf{x}^{\mathbf{p}}$$

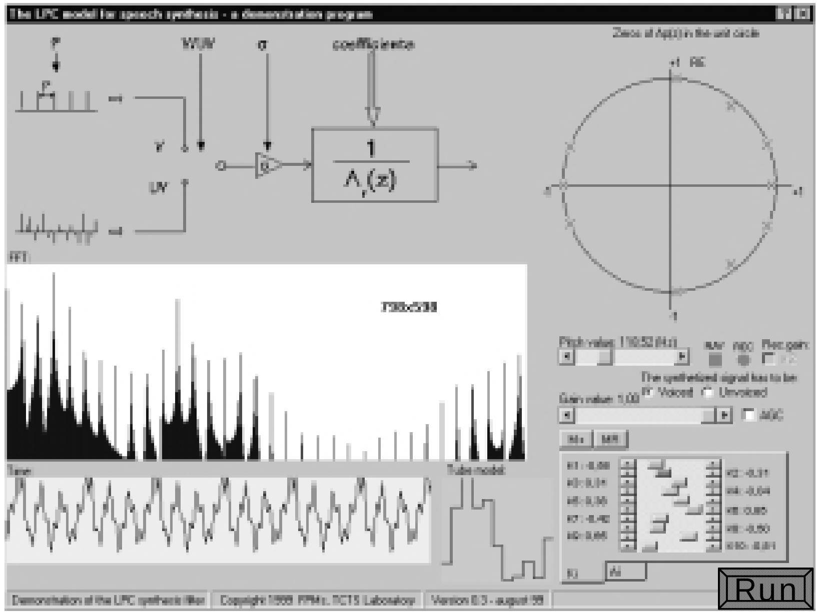
- Toeplitz matrix ⇒ O(p^2) using Levinson or **Schur** (still slightly faster than Levinson)
- NB : this is actually called the « autocorrelation approach »; in contrast, the so-called « covariance approach » is based on a different expression of the energy of e(n) and leads to a non-Toeplitz matrix…

In practice

- Sampling frequency
  - telephone speech : 8 kHz
  - speech recognition: 10 kHz
  - speech synthesis : 16 kHz
  - multimedia applications 11.25 kHz, 22.5 kHz et 44.1 kHz
- Prediction order p



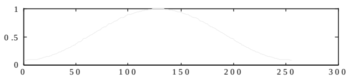
•2 poles for shaping the glottal velocity volume waveform
•2 poles (1 resonator) per formant (≈ per kHz of bandpass)
$\Rightarrow p_{opt} p = 2 + F_s$



In practice

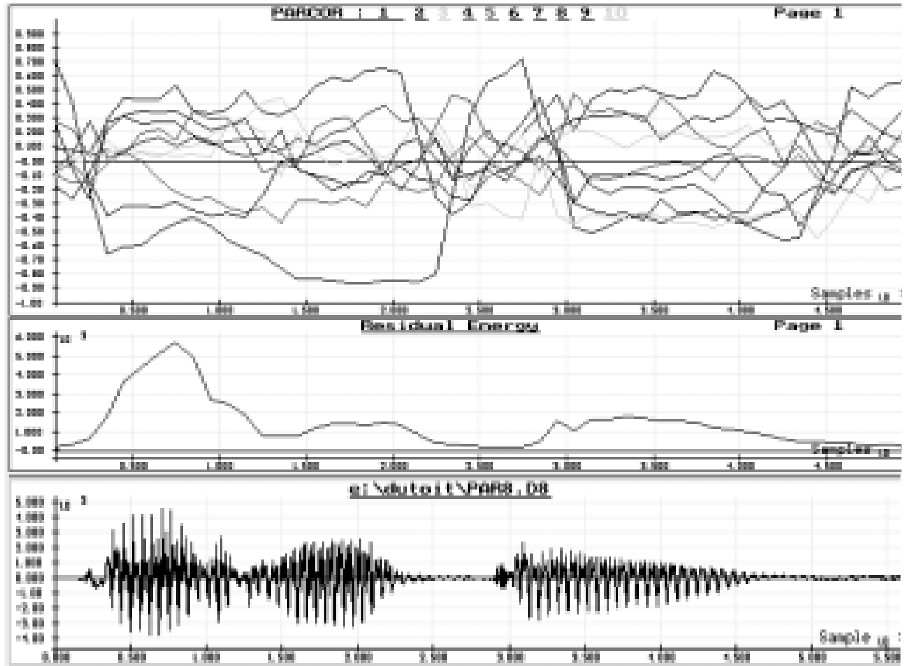
- (Pre-accentuation : filter $P(z) = 1 - \mu z^{-1}$)
- (Hamming) Weighting window
  - prevent e(n) from naturally taking high values at the beginning and at the end of the frame
  - let each set of prediction coefficients be more representataive of the 10 central ms

$$w(n) = 0.54 + 0.46\cos(2\pi\frac{n}{N})$$



- Computational load

		N=300,p=10
Weighting	N	300
Autocorrelation	(N – p / 2) (p + 1)	3245
Schur	p (p + 1)	110



Contents

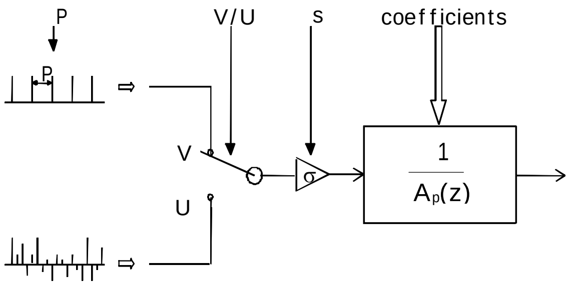
- Production models
  - Autoregressive modeling of speech
  - Autoregressive modeling of stationary random signals
    - AR model - Estimation algorithms
  - Autoregressive estimation of speech
  - **Extensions of the AR model**
- Transform-based models (*phenomenological*)
  - Short-term Fourier transform
    - Definition - Interpretation
    - Relationship with AR modeling
    - Hybrid harmonic+noise model - Estimation
  - Homomorphic (or *cepstral*) transform
    - Definition - Properties
    - Relationship with AR modeling
- F0 estimation

Problem with pseudo-stationarity

- r_x means nothing for plosives $\Rightarrow$ meaning of $\{a_i\}$?
- Explosions of plosives last 1-2 ms; they are synthesized on 10 ms at best
- In practice, the ear is much less sensitive to the spectrum of highly transient sounds

LP Speech synthesis

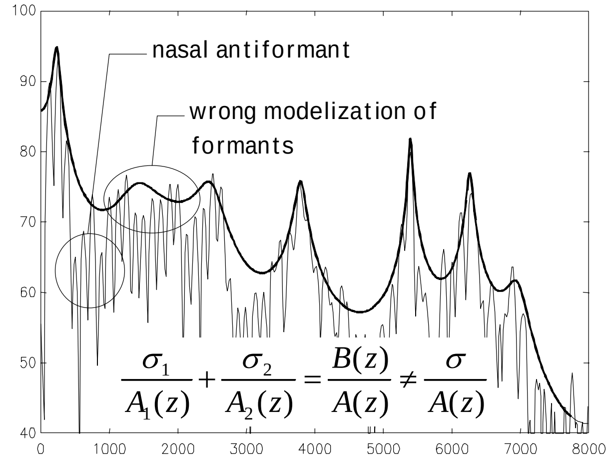
- Parameters are changed every 10 ms



- e(n) is replaced by a pulse sequence or by white noise (μ=0,σ=1), and amplified with $\sigma = \sigma_e$

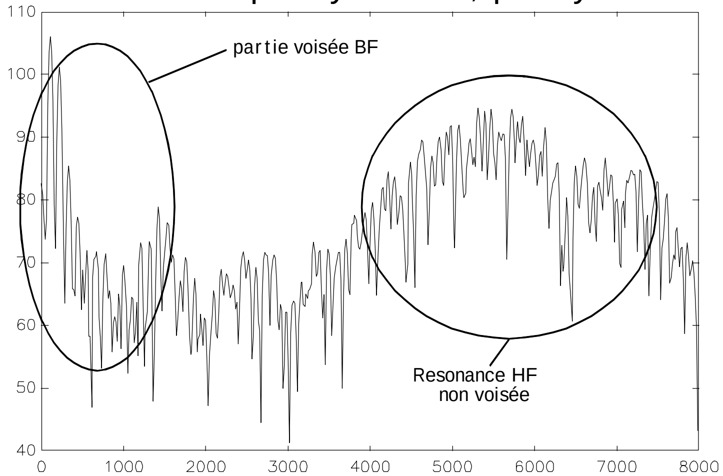
Problem with anti-formants

ARMA would be better for nasal sounds

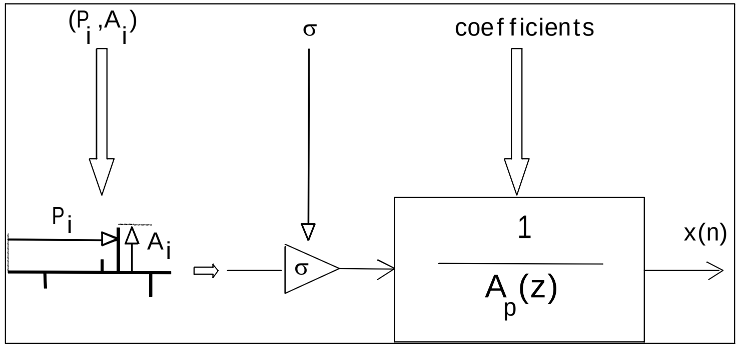


Problem with mixed voicing

Voiced fricatives:partly voiced, partly unvoiced



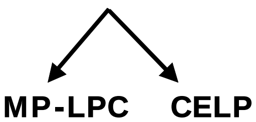
Multipulse Linear Prediction

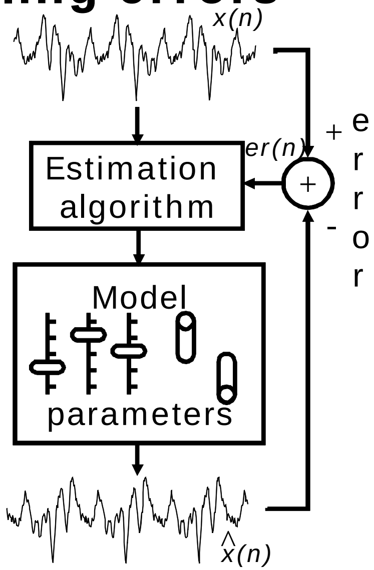


MP-LPC=LP model excited with small number of pulses, whose positions and amplitudes have to be adjusted

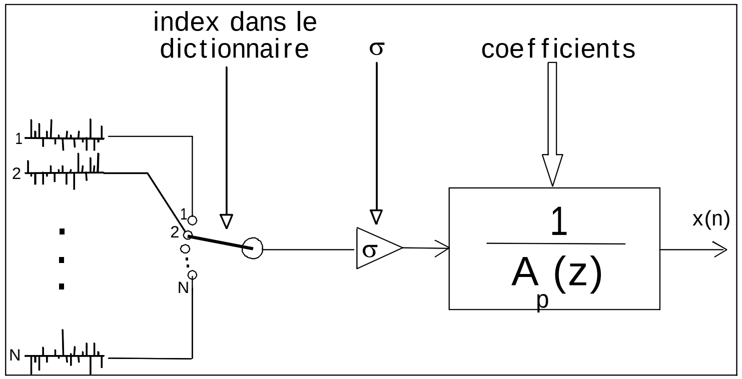
Decreasing modeling errors

- If the prediction error e(n) was used as input for 1/A(z): **er(n) ≡0**
- In practice, the excitation waveforms used are a **very rough approximation** of e(n)
- Find more realistic excitation waveforms?





Code-Excited Linear Prediction



CELP=LP model excited with a real excitation signal, taken from a list (=*codebook*) of typ. 1024 or 2018

Contents

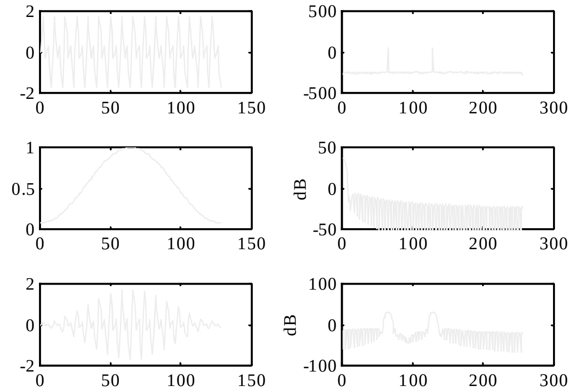
- Production models
  - Autoregressive modeling of speech
  - Autoregressive modeling of stationary random signals
    - AR model - Estimation algorithms
  - Autoregressive estimation of speech
  - Extensions of the AR model
- **Transform-based models (*phenomenological*)**
  - **Short-term Fourier transform**
    - Definition - Interpretation
    - Relationship with AR modeling
    - Hybrid harmonic+noise model - Estimation
  - Homomorphic (or *cepstral*) transform
    - Definition - Properties
    - Relationship with AR modeling
- F0 estimation

Short-term Fourier Transform

Convolutive effect

$$x(t) \overset{\Im}{\leftrightarrow} X(\omega) \qquad w(t) \overset{\Im}{\leftrightarrow} W(\omega)$$

$$x(t)w(t-\tau) \overset{\Im}{\leftrightarrow} X(\omega) * W(\omega)$$



Short-term Fourier Transform

- Continuous: $x(t) \overset{\Im}{\leftrightarrow} X(t,\omega) = \int_{-\infty}^{\infty} x(\tau) w(t-\tau) e^{-j\omega\tau} d\tau$

- Discrete: $x(n) \overset{\Im}{\leftrightarrow} X(n,k) = \sum_{i=0}^{N-1} x(i) w(n-i) e^{-jk\omega_N i}$

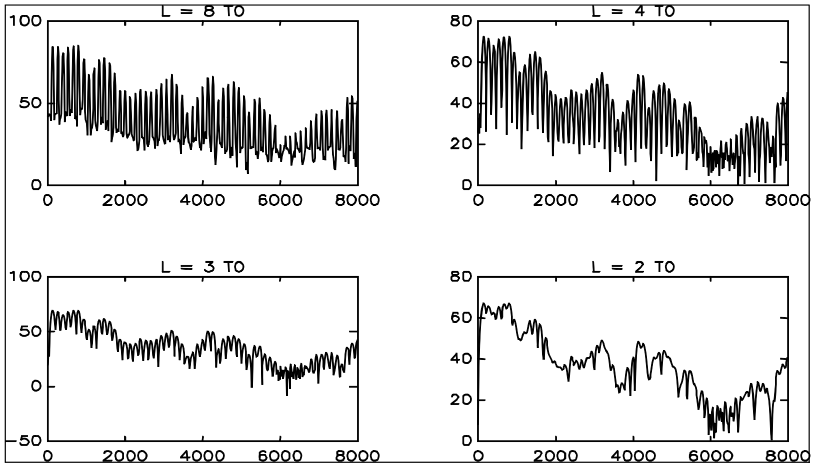
- Short-term spectral density function:

$$S_x(n,\phi) = \left| X(n,k) \right|^2 \qquad with \quad \phi = k\frac{2\pi}{N}$$

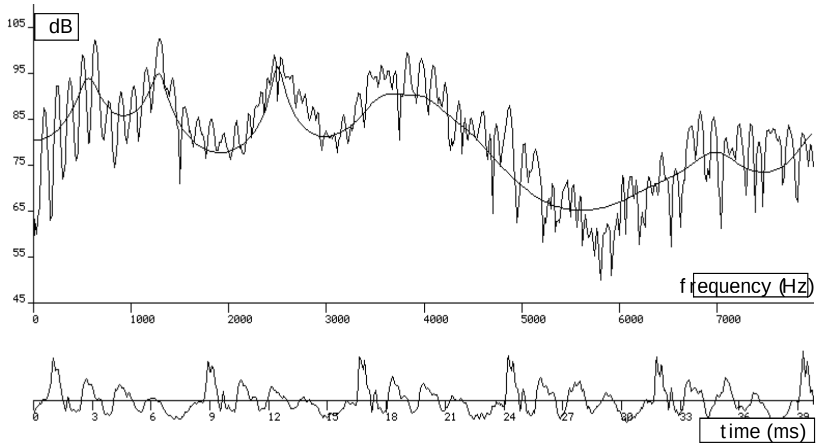
- Weighting window $w(n)$
  - if too long, averaging occurs (+stationnarity?)
  - if too short, frequency resolution falls down
  - shape?

Short-term Fourier Transform

From wide-band to narrow-band

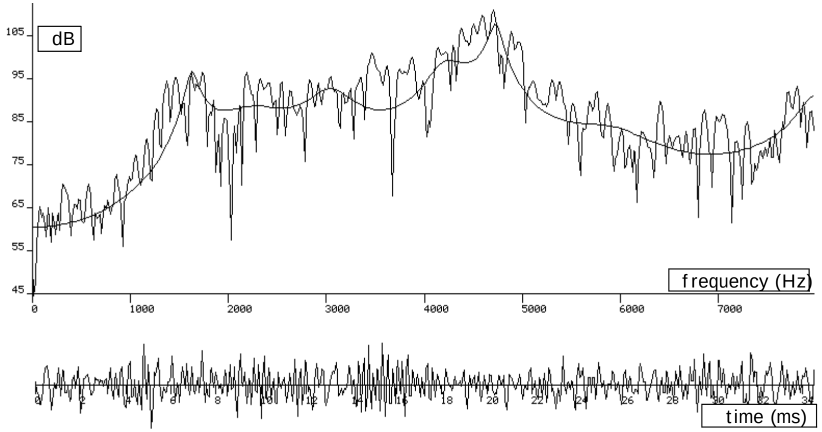


Narrow-band: voiced



[a], Hamming(30ms)

Narrow-band: unvoiced



[ʃ], Hamming(30ms)

$$NB: \quad |X(n,k)|^2 \xrightarrow{N\to\infty} S_x(\phi)$$

Contents

- Production models
  - Autoregressive modeling of speech
  - Autoregressive modeling of stationary random signals
    - AR model - Estimation algorithms
  - Autoregressive estimation of speech
  - Extensions of the AR model
- Transform-based models (*phenomenological*)
  - **Short-term Fourier transform**
    - Definition - Interpretation
    - **Relationship with AR modeling**
    - Hybrid harmonic+noise model - Estimation
  - Homomorphic (or *cepstral*) transform
    - Definition - Properties
    - Relationship with AR modeling
- F0 estimation

STFT ↔ AR model?

$$\left.\begin{array}{l}\phi_x(k) \xleftrightarrow{\Im} S_x(\phi) = \sum_k \phi_x(k) e^{jk\phi} \\ \phi_{xAR}(k) \xleftrightarrow{\Im} S_{xAR}(\phi) \\ S_{xAR}(\phi) = S_u(\phi)\left|\dfrac{\sigma}{A(e^{j\phi})}\right|^2 = \left|\dfrac{\sigma}{A(e^{j\phi})}\right|^2 \end{array}\right\} \text{by definition}$$

$$\phi_{xAR}(k) = \phi_x(k) \quad for\ k = 0,1,\ldots,p$$

$$\Downarrow$$

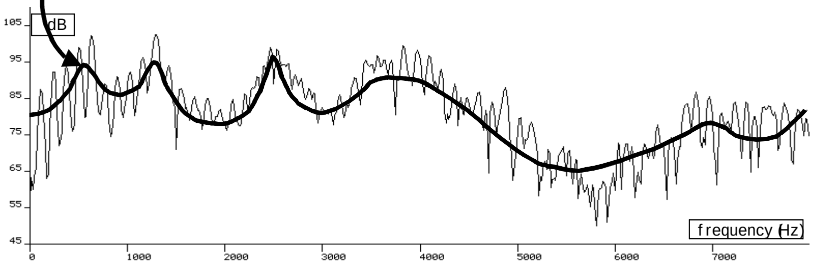
$$\boxed{\left|\frac{\sigma}{A(e^{j\phi})}\right|^2 \quad \textit{approximates the envelope of } S_x(\phi)}$$

NB: this is also appliable to *short term* estimations

STFT $\leftrightarrow$ AR model?

$\left|\dfrac{\sigma}{A(e^{j\phi})}\right|^2$ can be easily computed from the $\{a_i\}$

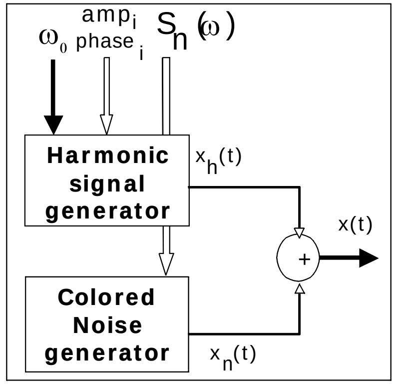
$$\underbrace{\{1, a_1, a_2, ..., a_p, 0,0,0,...,0\}}_{\text{N samples}} \xleftrightarrow{FFT} A(e^{jk\frac{2\pi}{N}})$$



Contents

- Production models
  - Autoregressive modeling of speech
  - Autoregressive modeling of stationary random signals
    - AR model - Estimation algorithms
  - Autoregressive estimation of speech
  - Extensions of the AR model
- Transform-based models (*phenomenological*)
  - **Short-term Fourier transform**
    - Definition - Interpretation
    - Relationship with AR modeling
    - **Hybrid harmonic+noise model - Estimation**
  - Homomorphic (or *cepstral*) transform
    - Definition - Properties
    - Relationship with AR modeling
- F0 estimation

Hybrid harmonic/noise model



- Harmonic component defined by the amplitudes and phases of harmonics : $\{amp_i\}$, $\{phase_i\}$
- Noise component defined by the s.d.f. of $x_n(t)$

Hybrid H/N estimation

- Rough F0 estimation (comb-filter-like)
- Estimate the parameters of the harmonic part

$$e_p(t) = \int_{-\infty}^{\infty} w^2(t-\tau)\left|x(\tau) - x_h(\tau)\right|^2 d\tau = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left|S_x(t,\omega) - S_{x_h}(t,\omega)\right|^2 d\omega$$

with $\quad x_h(t) = \sum_{i=-N}^{N} \underline{a}_i e^{ji\omega_0 t} \quad with\ \underline{a}_i = \underline{a}_{-i}^{\ *}$

- Typical least squares minimization problem :

$$\mathbf{A}\underline{\mathbf{a}} = \mathbf{b}$$

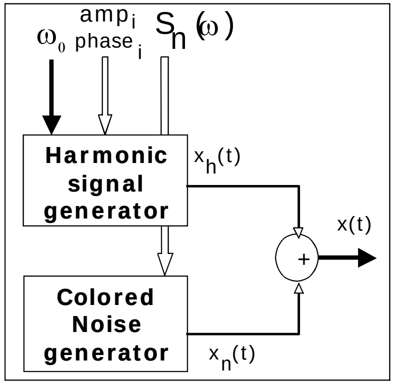
- Solution : $\mathbf{A^H A}\underline{\mathbf{a}} = \mathbf{A^H b} \quad \equiv \quad \mathbf{R}\underline{\mathbf{a}} = \mathbf{r}$
  - trivial if **R** is diagonal
  - Levinson if **R** is Toeplitz
  - Cholesky otherwise (**R** is allways symetric)

Hybrid H/N estimation

- Repeat on a grid of F_0 values around initial F_0
  - Precision on *N*th harmonic is *N* x precision on F_0!
    - F_0=100 Hz; $T_0=F_s/100$ samples
      if error = 1 sample on T_0
      = 100 - $F_s/(F_s/(100+1))$ Hz on F_0
      = $10000/(F_s+10000)$ Hz on F_0
      $N_{harm} = F_s/2/100$
      error on *N*th harm. = N_{harm} x error on $F_0 \approx$ 25 to 50Hz!!!
  - Precision required : at least 1/8th sample
- Get the s.d.f of the « noise » part with:

$$x_n(t) \approx x(t) - x_h(t)$$

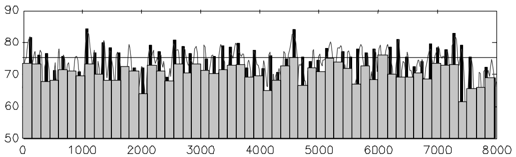
Hybrid H/N synthesis



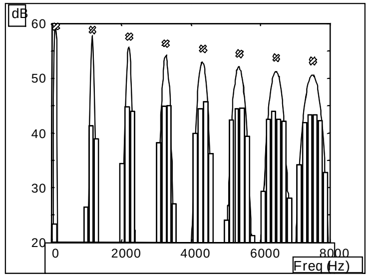
- Harmonics:
  - Σ of cosines
  - Σ of outputs of digital oscillators whose frequencies are set to that of harmonics
  - Σ of main lobes in the freq. domain + IFFT
- Noise:
  - Σ of narrow band noises, obtained by (amplitude) modulating a low frequency noise with harmonics
  - frequency-domain noise matching the required s.d.f. + IFFT

H/N modeling errors

- Harmonics are found in noise (no orthogonality)

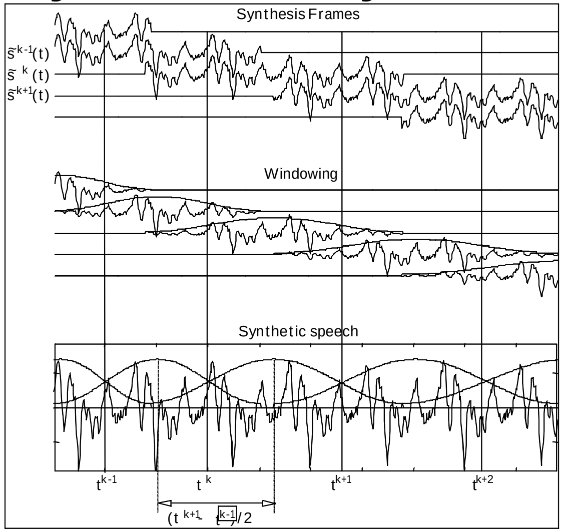


- F_0 is not constant on the analysis frame $\Rightarrow$ broadening of harmonic lobes at high frequencies



- In real speech, noise is correlated with harmonics in the time domain

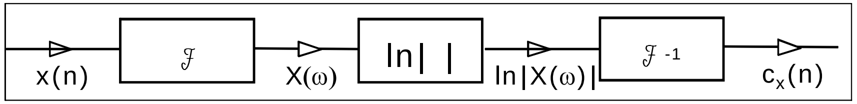
Hybrid H/N synthesis



Overlap-Add (OLA) of successive frames

Contents

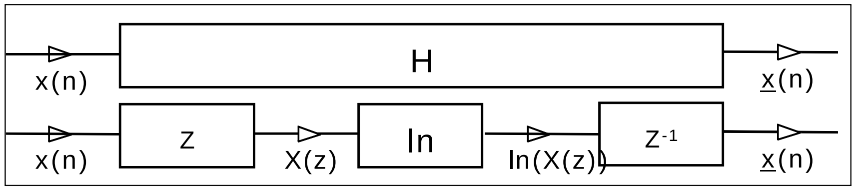
Real cepstrum



- $c_x(n)$ is the $\mathcal{F}^{-1}$ of the amplitude spectrum of $x(n)$ in neper (≈ in dB)
- $c_x(n)$ can be computed with FFT^{-1}, provided N_{FFT} is big enough (to avoid undersampling of $X(\omega)$)
- NB: *cepstrum, quefrency, liftering* :-)

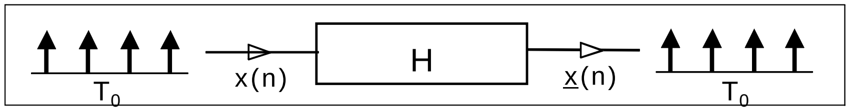
Homomorphic transform

- Complex cepstrum $\underline{x}(n)$



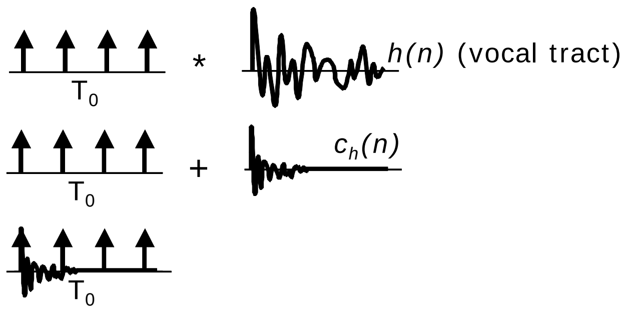
- Convolution becomes summation:

$$\boxed{x(n)=u(n)*h(n) \rightarrow X(z)=U(z)H(z) \rightarrow \underline{x}(n)=\underline{u}(n)+\underline{h}(n)}$$

- If $X(z)$ rational, then $\underline{x}(n)$ falls faster than 1/n
- 

Applications to speech

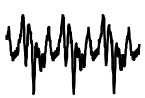
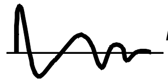
- If x(n) =



Use?
- ***Measure T_0*** :easier on $c_x(n)$ than on x(n)
- ***Measure the spectral envelope of x(n):*** *Isolate $c_h(n)$ →* ***h(n)***

Applications to speech

Cepstral Mean Substraction

If x(n) =  *  *h(n)* (tel. line)

$c_x(n) \quad = c_{speech}(n) \; + c_{line}(n)$

$E[c_x(n)] = E[c_{speech}(n)] \; + E[c_{line}(n)]$

$\qquad = E[c_{line}(n)] + k$ (estimator of $c_{line}(n)$)

$c^{CMS}{}_x(n) = c_x(n) - E[c_x(n)]$

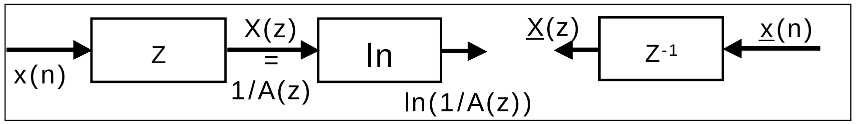
$c^{CMS}{}_x(n)$ is independent from the channel

Used in spech recogniton

Contents

Relationship with AR model

The first values of the cepstrum are characteristic of the transfer function of the vocal tract ⇒ must be related to $\{a_i\}$



$$\ln\left(\frac{1}{A(z)}\right) = \sum_{n=1}^{\infty} \underline{x}(n)\, z^{-n}$$

$$-A'(z)/A(z) = \sum_{n=1}^{\infty} n\, \underline{x}(n)\, z^{-(n+1)}$$

$$-\sum_{i=1}^{p} i\, a_i\, z^{-(i+1)} = \left[\sum_{j=0}^{p} a_i\, z^{-j}\right]\left[\sum_{n=1}^{\infty} n\, \underline{x}(n)\, z^{-(n+1)}\right] \quad \ldots$$

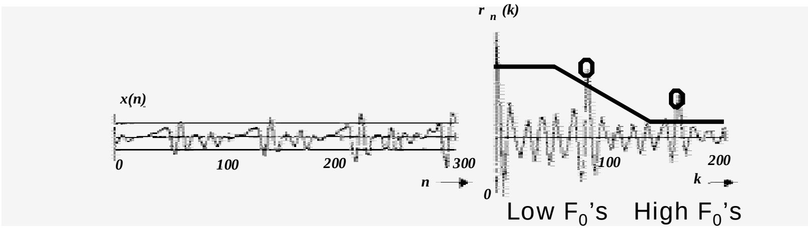
F0 estimation

- **Short-term estimation** of F0 on an analysis frame (at least 2 times the local pitch period); produces candidates for F0, with *score*
- **Long-term post-processing**, making the ultimate choice among candidates (on a 3 frames-basis, usually), as a function of their score and taking into account constraints and correcting estimation biases

Autocorrelation-based F0 estimation

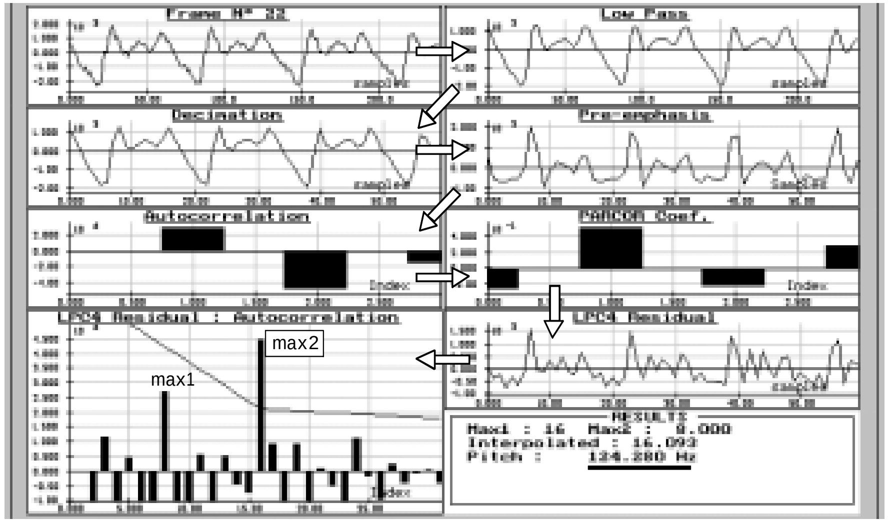
- Periodicity of *x* implies a max in $r_x(k)$ for $kT_s \approx T_0$

$$r(k) = \sum_{n=-\infty}^{\infty} x(n)x(n+k) \quad for\ k = K_{\min},\dots,K_{\max}$$

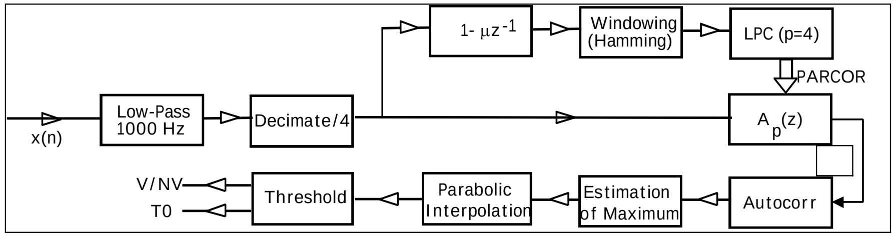


- Computational load >>>
- F0 doubling for females, because of F1≈H2 $\Rightarrow$ use **threshold(k)**

SIFT : an example



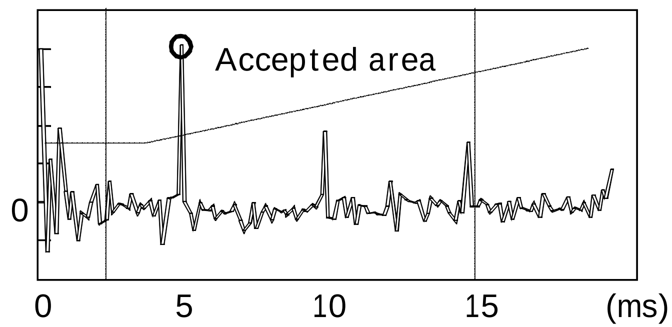
Simplified Inverse Filtering Technique (SIFT)



- Finds the max of the autocorrelation of the error signal ($\Rightarrow$ F1≠H2 : flat envelope!)
- Decimation-interpolation for lower computational cost

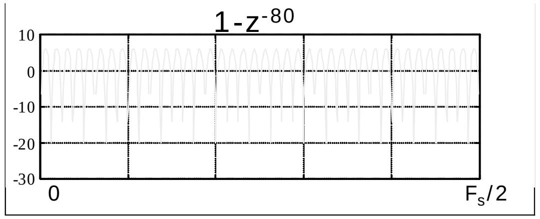
Cepstrum-based F0 estimation

- Cepstrum is FFT^{-1}(logSpectrum)
- Log spectrum is flatter than Spectrum
- Cepstrum is more pulse-like than signal



Estimating F0 with a comb filter

- Idea : filter signal with comb filter whose F0 varies from 70 to 500 Hz, and measure the energy of the output: F_0=F(max(energy))



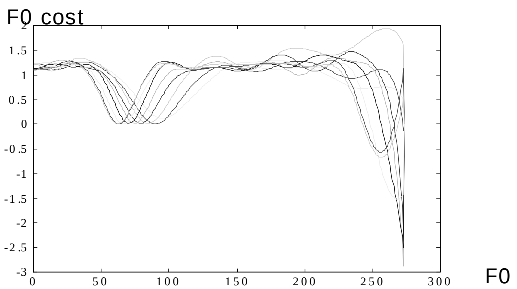
- Usually performed in the frequency domain direclty : F0=F(max(<FFT(x)²,H²(filter)>))

Conclusion

- **LP** model is good at modeling the short-term **spectral envelope** (id. for cepstrum)
- **MP-LPC, CELP, Hybrid H/N** add a better modeling of the contribution of the **excitation** component (the *fine* spectral structure)
- NB : these were models of the acoustic component of speech only...

Post-processing

- Based on **Dynamic Programming** :
  From the possible F0 values for successive frames, find the sequence which minimizes a cost function



- Logical filtering
  - ex: one V frame in an island of UV frames ⇒ UV