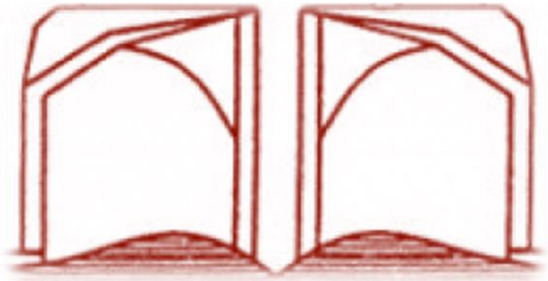


پردازش گفتار

نمایش‌های سیگنال گفتار و استخراج ویژگی

هادی ویسی

h.veisi@ut.ac.ir

دانشگاه تهران - دانشکده علوم و فنون نوین

فهرست

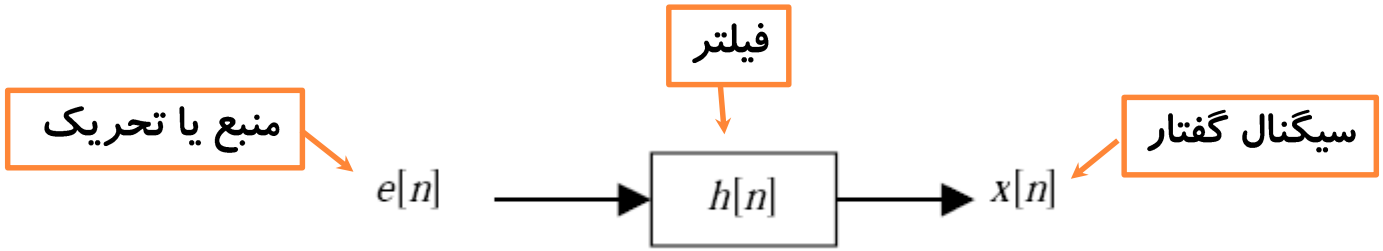
مدل منبع-فیلتر . . .

هدف

- نمایش و مدل‌سازی سیگنال گفتار
- کاربرد در کلیه سیستم‌های پردازش‌های گفتار
  - رمزگذاری، شبیه‌سازی، سنتز گفتار، تشخیص گفتار و ...

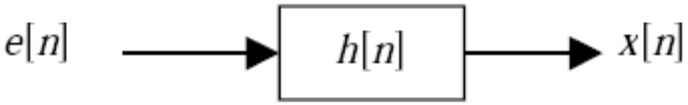
تجزیه سیگنال گفتار

- منبعی که از فیلتر خطی متغیر با زمان عبور می‌کند
- منبع = جریان هوا در تارهای صوتی
- فیلتر = نشانگر رزنانس‌های مجرای گفتار



مدل منبع-فیلتر

- تخمین فیلتر h[n]
  - روش‌های مبتنی بر مدل‌های تولید گفتار
    - رمزگذاری پیش‌بینی خطی (linear predictive coding)
    - تحلیل کپسترال (cepstral analysis)
  - روش‌های مبتنی بر مدل‌های دریافت گفتار
    - کپستروم مل-بسامد (mel-frequency cepstrum)



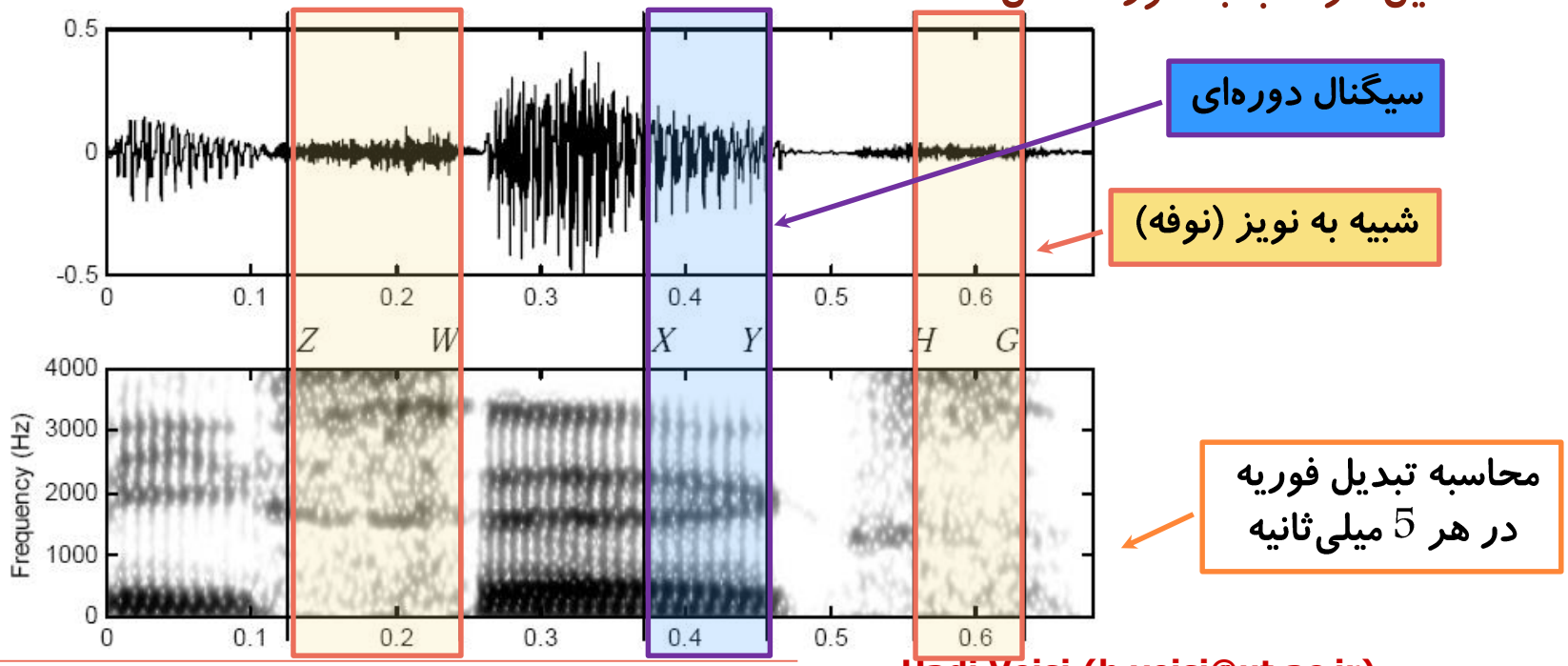
- تخمین منبع e[n]
  - بعد از تخمین فیلتر
  - محاسبه منبع با عبور دادن سیگنال گفتار از فیلتر معکوس

تحلیل فوریه کوتاه‌مدت . . .

- عدم ایستا (stationary) بودن سیگنال گفتار
  - مشخصات آن با زمان تغییر می‌کند
- سیگنال در طول ادای یک واج (۲۵ تا ۲۵۰ میلی ثانیه) تقریباً ایستاست
- با تقریب، فرض می‌شود سیگنال گفتار در زمان‌های کوتاه ایستا است
  - یک بخش کوتاه‌مدت سیگنال گفتار = فریم (قاب)
  - طول هر فریم باید به گونه‌ای باشد که شامل فقط یک واج و یا واج‌گونه باشد
- طول هر قاب در کاربردهای واقعی: بین ۱۰ تا ۵۰ میلی ثانیه

تحلیل فوریه کوتاه‌مدت . . .

- تجزیه سیگنال گفتار به مجموعه‌ای از بخش‌های کوتاه (فریم=قاب)
  - طول فریم‌ها باید به اندازه کافی کوچک = سیگنال ایستا (stationary) باشد
    - مشخصات آماری سیگنال ثابت باشد
    - ثابت ماندن رفتار سیگنال (دوره‌ای بودن یا ظاهر شبیه به نوفه داشتن)
  - تحلیل هر قاب به طور مستقل



تحلیل فوریه کوتاه‌مدت . . .

سیگنال کوتاه‌مدت

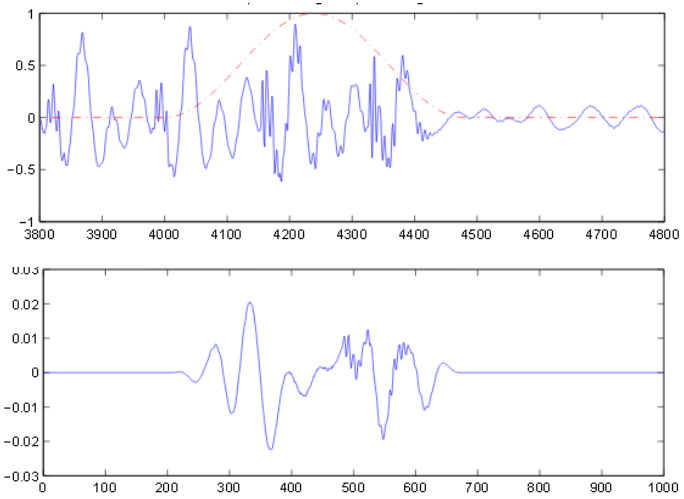
- فریم (قاب) m

قاب = سیگنال کوتاه‌مدت

$$x_m[n] = x[n]\,w_m[n]$$

پنجره (window)

- پنجره به جز در یک منطقه کوچک (طول مشخص) در همه جا صفر است
- هرچند تابع پنجره می‌تواند مقادیر مختلفی برای قالب‌های مختلف m داشته باشد اما معمولاً پنجره برای تمامی قاب‌ها یکسان است



$$w_m[n] = w[m-n]$$

- که در آن $w[n] = 0 \quad \text{for} \quad |n| > N/2$ (N=طول قاب)

- نمایش فوریه کوتاه مدت برای قاب m

$$X_m(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_m[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} w[m-n]x[n]e^{-j\omega n}$$

تحلیل فوریه کوتاه‌مدت . . .

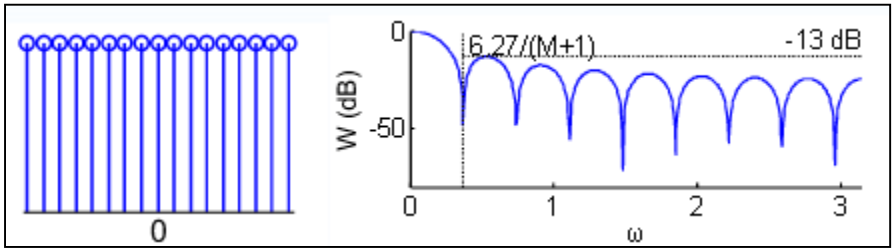
پنجره‌گذاری . . .

- مستطیلی (Rectangular)

$$h_\pi[n] = u[n] - u[n-N]$$

$$H_\pi(e^{j\omega}) = \frac{1-e^{-j\omega N}}{1-e^{-j\omega}} = \frac{\left(e^{j\omega N/2} - e^{-j\omega N/2}\right)e^{-j\omega N/2}}{\left(e^{j\omega/2} - e^{-j\omega/2}\right)e^{-j\omega/2}}$$

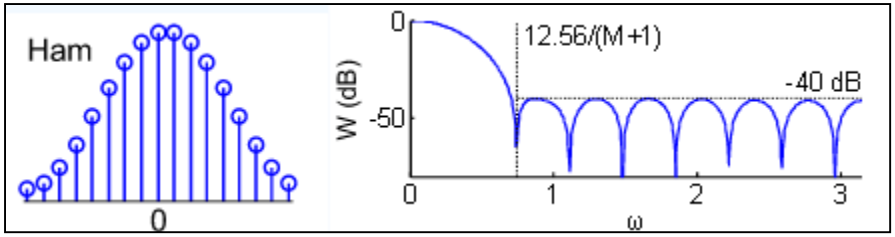
$$= \frac{\sin \omega N/2}{\sin \omega/2} e^{-j\omega(N-1)/2} = A(\omega) e^{-j\omega(N-1)/2}$$



- همینگ (Hamming)
  - معمولاً α=0.46

$$h_h[n] = \begin{cases} (1-\alpha) - \alpha\cos(2\pi n/N) & 0 \le n < N \\ 0 & otherwise \end{cases}$$

$$H_h(e^{j\omega}) = (1-\alpha)H_\pi(e^{j\omega}) - (\alpha/2)H_\pi(e^{j(\omega-2\pi/N)}) - (\alpha/2)H_\pi(e^{j(\omega+2\pi/N)})$$



تحلیل فوریه کوتاه‌مدت . . .

- پنجره‌گذاری

Rectangular: $w[n] \equiv 1$

Hanning: $0.5 + 0.5c_1$

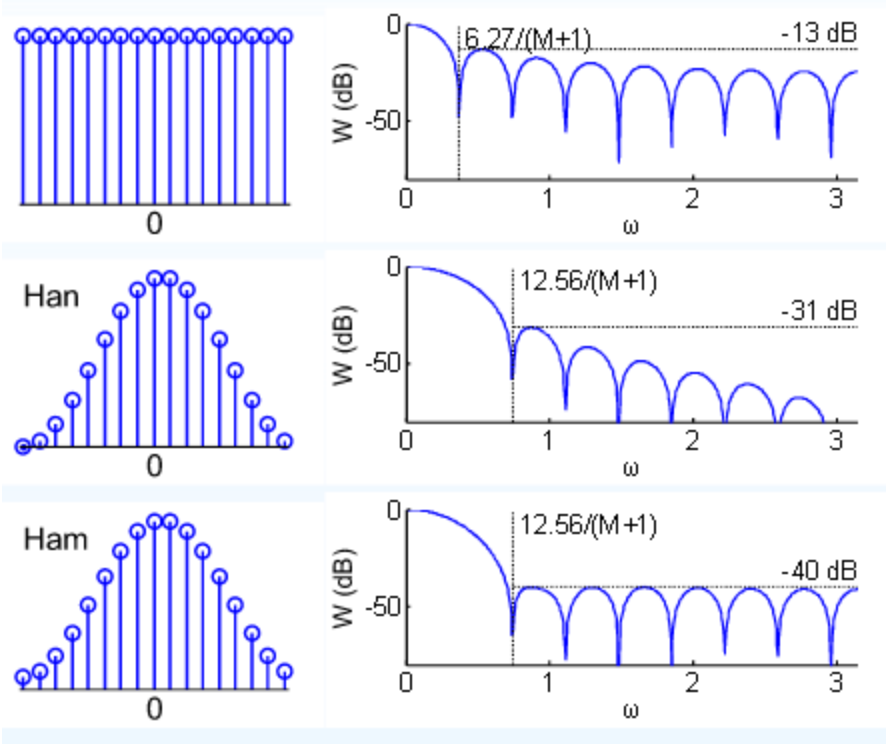
$c_k = \cos \frac{2\pi kn}{M+1}$

rapid sidelobe decay

Hamming: $0.54 + 0.46c_1$

best peak sidelobe

انتخاب بهتر!



تحلیل فوریه کوتاه‌مدت . . .

- برای سیگنال دوره‌ای $X_m[n]$ با دوره M داریم

$$X_m(e^{j\omega}) = \sum_{k=-\infty}^{\infty} X_m[k]\delta(\omega - 2\pi k/M)$$

- تبدیل فوریه پنجره $w[n]$ =

$$W(e^{j\omega}) = \sum_{n=-\infty}^{\infty} w[n]e^{-j\omega n}$$

  - تبدیل فوریه $w[m-n]$ =

$$W(e^{-j\omega})e^{-j\omega m}$$

- بنابراین (تبدیل فوریه یک قاب)
  - ضرب در حوزه زمان ($x[n]w[m-n]$) معادل کانولوشن در حوزه فرکانس است

$$X_m(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_m[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} w[m-n]x[n]e^{-j\omega n} = \sum_{k=-\infty}^{\infty} X_m[k]W(e^{j(\omega-2\pi k/N)})e^{j(\omega-2\pi k/N)m}$$

    - جمع وزن‌دار $W(e^{j\omega})$ها

تحلیل فوریه کوتاه‌مدت . . .

- برای بازیابی $x_m[n]$ از $X_m(e^{j\omega})$ باید $W(e^{j\omega}) \approx 0 \text{ for } |\omega - \omega_k| > \lambda$
  - پاسخ فرکانسی پنجره در خارج از lobe اصلی صفر باشد
  - برای پنجره مستطیلی داریم $\lambda = 2\pi/N$ و برای پنجره همینگ $\lambda = 4\pi/N$
  - پس برای پنجره مستطیلی باید $N \geq M$ (M= دوره تناوب سیگنال)
    - طول پنجره حداقل یک دوره تناوب زیروبمی (pitch period) باشد
  - و برای پنجره همینگ $N \geq 2M$
    - طول پنجره حداقل دو دوره تناوب زیروبمی (pitch period) باشد
- در عمل: مقدار زیروبمی را نداریم = درنظر گرفتن کمترین مقدار F_0
  - برای $F_0 = 50Hz$ باید $N = 20$ ms (برای مستطیلی) و $N = 40$ ms (برای همینگ)
  - اگر سیگنال با طول 40 ms غیرایستا باشد؟؟
    - پنجره مستطیلی تفکیک زمانی (Time Resolution) بهتری نسبت به پنجره همینگ فراهم می‌کند

تحلیل فوریه کوتاه‌مدت . . .

- از طرفی
  - پاسخ فرکانسی خارج از lobe اصلی صفر نیست
  - دامنه فرکانس در lobe دوم در مستطیلی 17 dB و برای همینگ حدود 44 dB کمتر از lobe اصلی است (برای هنینگ حدود 31 dB)
    - پس هارمونیک kام $X_m(e^{j2\pi k/M})$ نه تنها حاوی $X_m(k)$ بلکه حاوی جمع وزن‌دار $X_m(i)$ نیز است
      - نشت طیفی (spectral leakage)

- پنجره مستطیلی یا همینگ؟

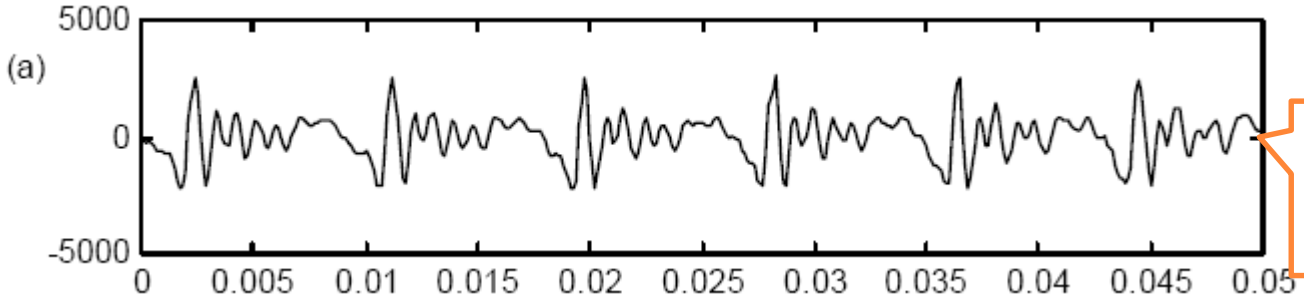
در عمل
انتخاب بهتر!

ویژگی	مستطیلی	همینگ	مقایسه
طول پنجره (Time Resolution)	یک دوره تناوب زیروبمی	دو دوره تناوب زیروبمی	مستطیلی بهتر است
نشت طیفی (Spectral Leakage)	17 dB کمتر از lobe اصلی	44 dB کمتر از lobe اصلی	همینگ بهتر است

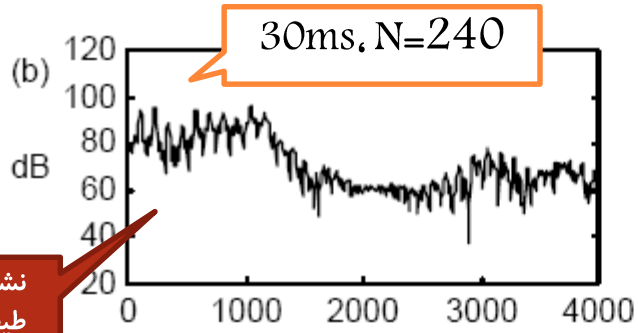
در عمل: طول پنجره حدود 20 تا 30 میلی‌ثانیه

تحلیل فوریه کوتاه‌مدت . . .

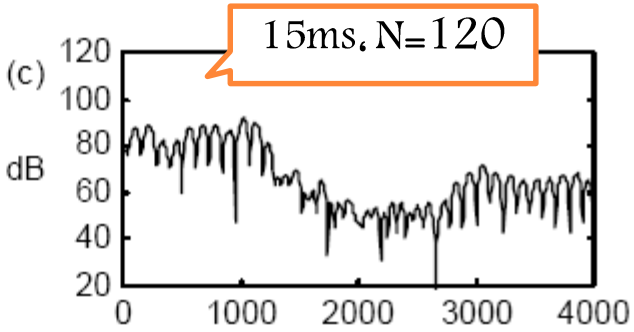
- مثال: /ah/



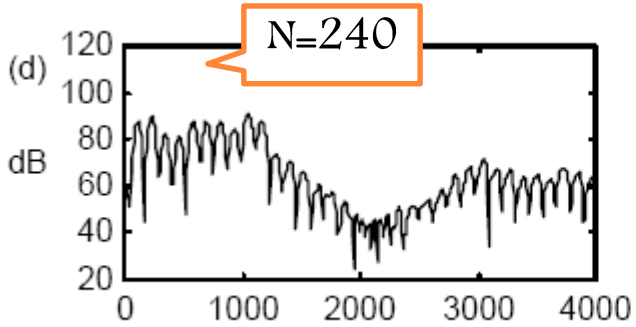
نمونه برداری = 8kHz
دوره تناوب زیروبمی: 110 Hz
دوره تناوب زیروبمی: M=72

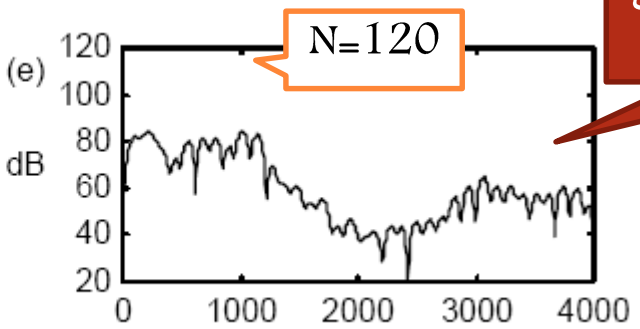


نشت طیفی



- مستطیلی



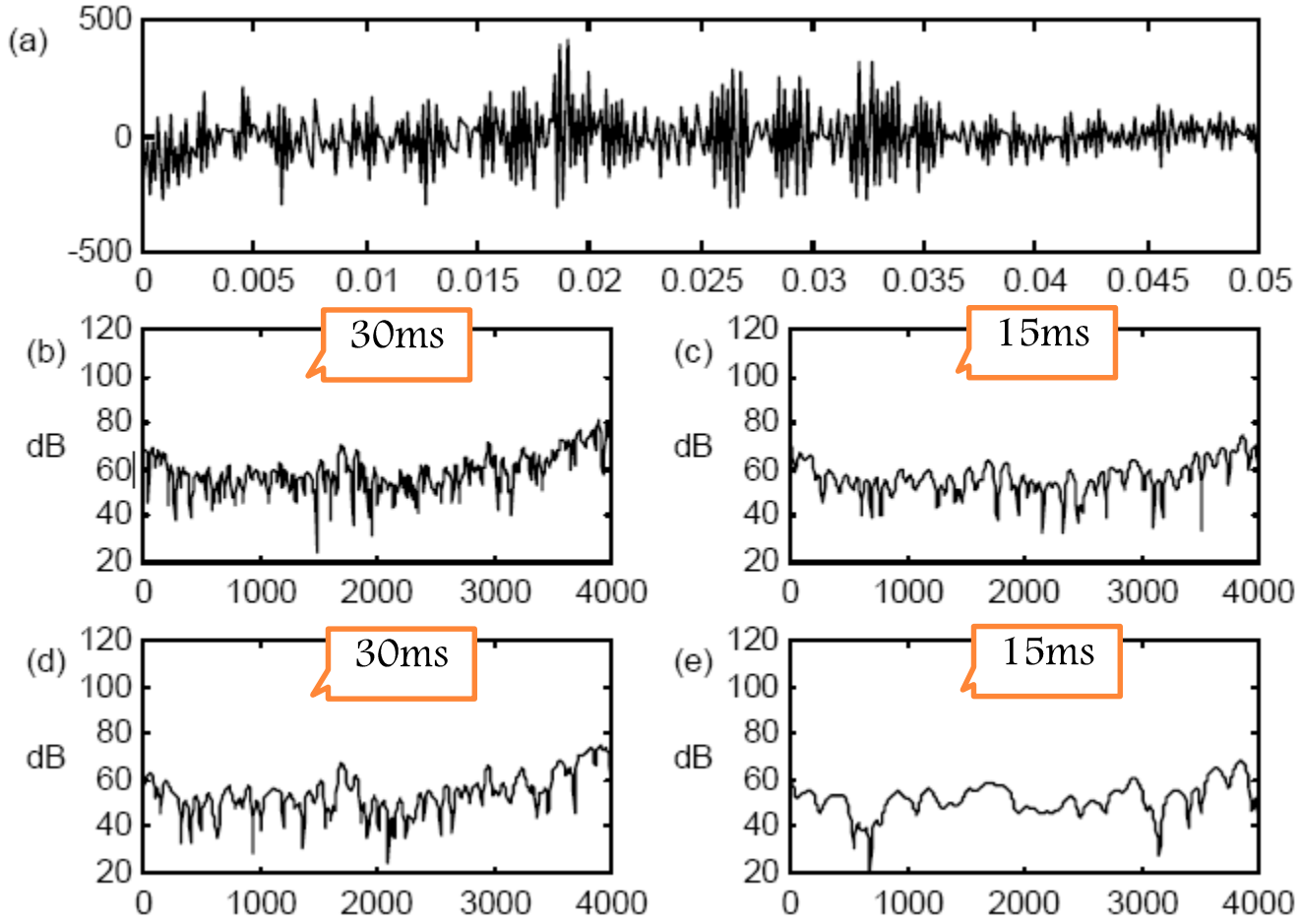


طول پنجره کمتر از دو برابر تناوب
عدم نمایش درست lobeها

- همینگ

تحلیل فوریه کوتاه‌مدت . . .

- مثال: واج بی‌واک

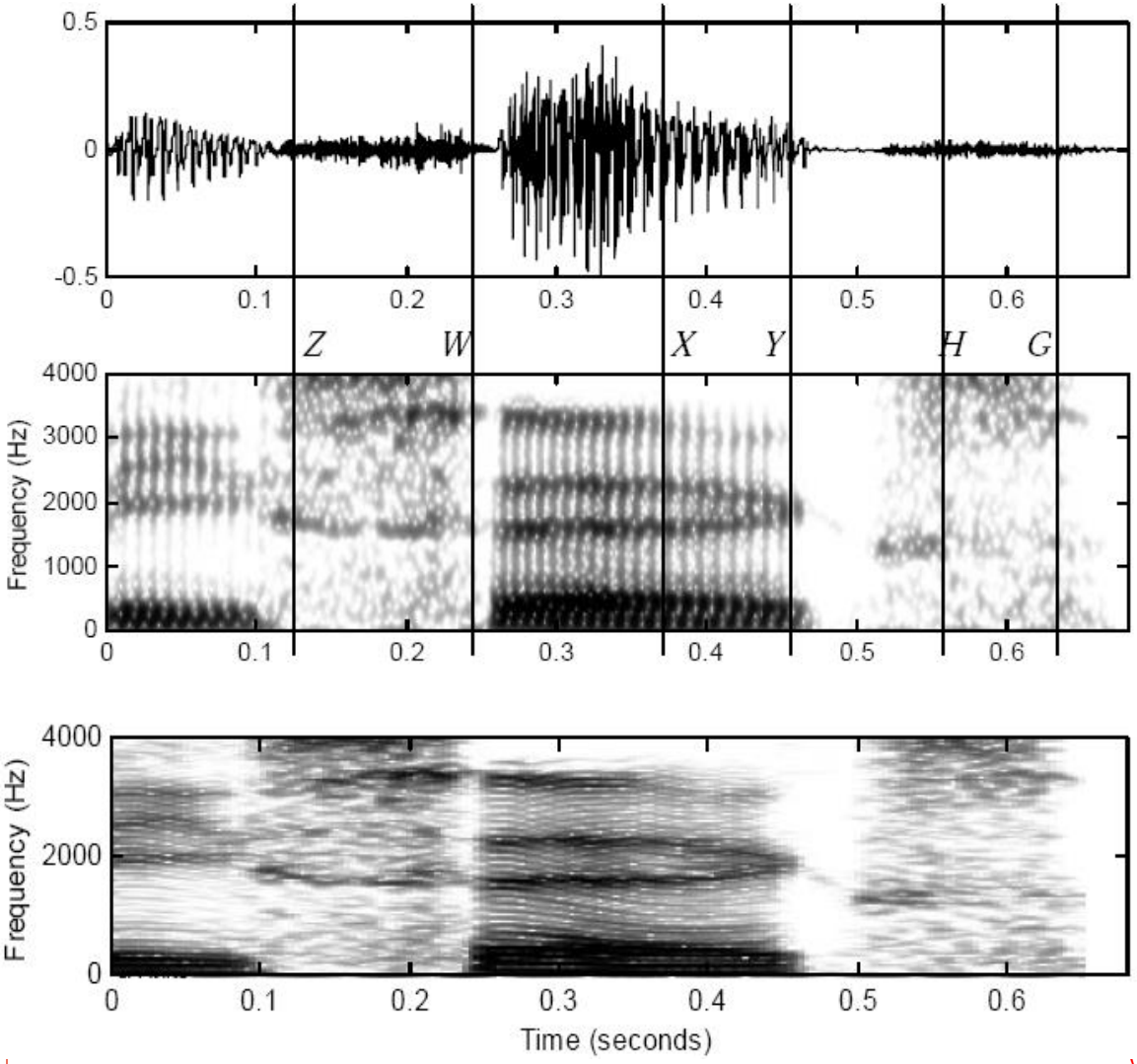


- مستطیلی

سیگنال نامنظم

- همینگ

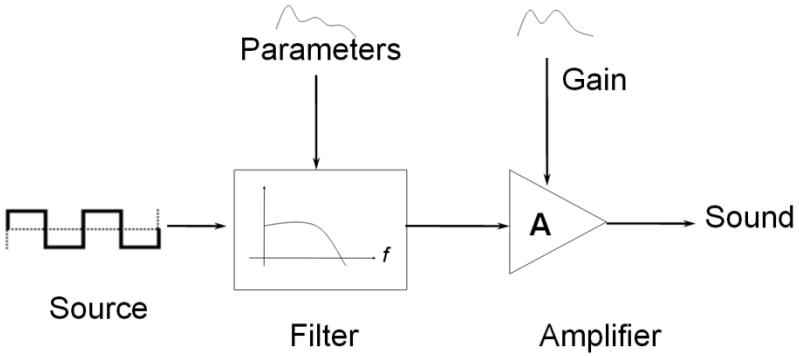
تحلیل فوریه کوتاه‌مدت



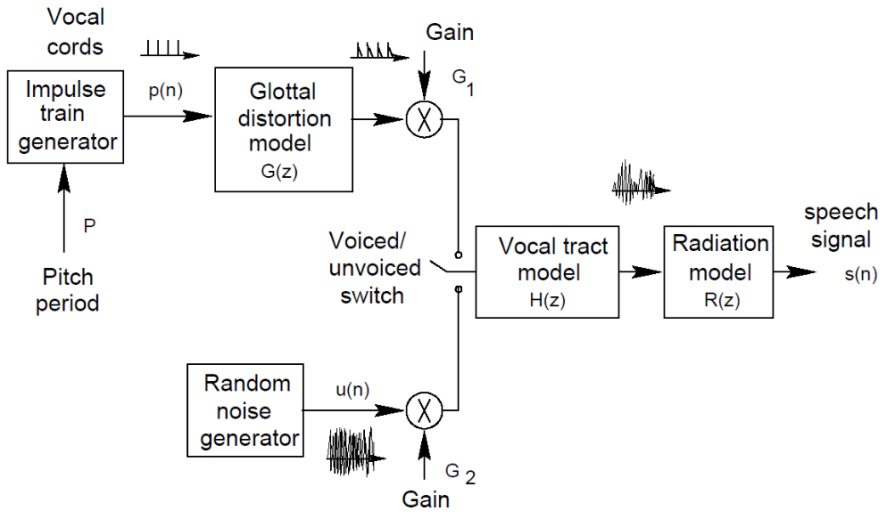
- طیف‌نگار

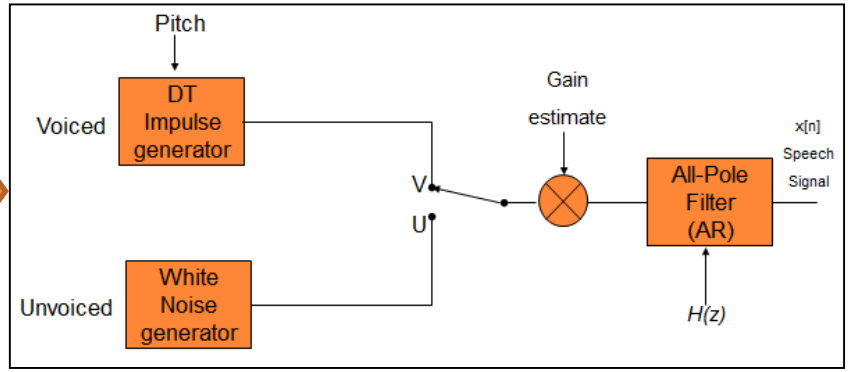
- **باند پهن**
  - پنجره زمانی کوتاه
    - کمتر از ۱۰ میلی ثانیه
  - تفکیک زمانی خوب
  - تفکیک بسامد پایین‌تر
    - فیلترهای پهن (>200Hz)
- **باند باریک**
  - پنجره زمانی بلند
    - بزرگتر از ۲۰ میلی ثانیه
  - تفکیک زمانی کمتر
  - تفکیک بسامد بهتر
    - فیلترهای باریک (<100Hz)

مدل منبع-فیلتر



- مدل منبع-فیلتر (Source-Filter)
  - تولید گفتار واک‌دار و بی‌واک

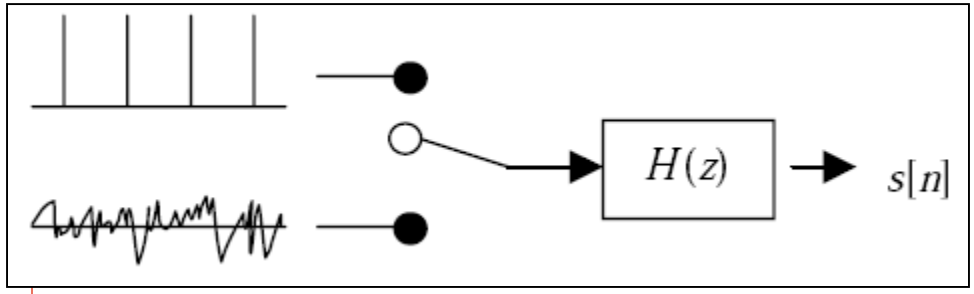




- نحوه تخمین فیلتر؟ منبع؟

تحلیل LPC ...

- رمزگذاری پیش‌بینی کننده خطی (LPC: Linear Predictive Coding)
  - مبانی ریاضی از سال ۱۹۲۷ (Yule) و ۱۹۳۱ (Walker)
  - الگوریتم بدست آوردن ضرایب در ۱۹۴۷ (Levinson) و ۱۹۶۰ (Durbin)
- تحلیل LPC یا مدل‌سازی خود بازگشتی (AR: Auto-Regressive)
- روش پرکاربرد در نمایش سیگنال گفتار و تخمین پارامترهای اصلی آن (سریع و ساده)
- تخمین فیلتر (و منبع) در مدل منبع-فیلتر
  - امکان مدل کردن $H(z)$ با یک فیلتر تمام-قطب (با تعداد قطب‌های کافی)



تعداد P قطب

$$H(z)=\frac{X(z)}{E(z)}=\frac{1}{1-\sum_{k=1}^{p}a_k z^{-k}}=\frac{1}{A(z)}$$

تحلیل LPC ...

- فیلتر معکوس (تولید صدا)

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$$

- در حوزه زمان (معکوس تبدیل z)

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

تخمین نمونه nام سیگنال از روی P نمونه قبلی (و سیگنال تحریک)= LP

P = مرتبه تحلیل LPC

a_k= ضرایب تحلیل LPC

سیگنال تحریک (Excitation)
باقی‌مانده (Residual)

- تخمین

$$\tilde{x}[n] = \sum_{k=1}^{p} a_k x[n-k]$$

- خطای تخمین

$$e[n] = x[n] - \tilde{x}[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k]$$

- مساله: نحوه پیدا کردن ضرایب a_k؟

تحلیل LPC: محاسبه ضرایب ...

- هدف: کمینه کردن خطای تخمین
  - مربعات خطا برای قاب سیگنال گفتار $x_m[n]$

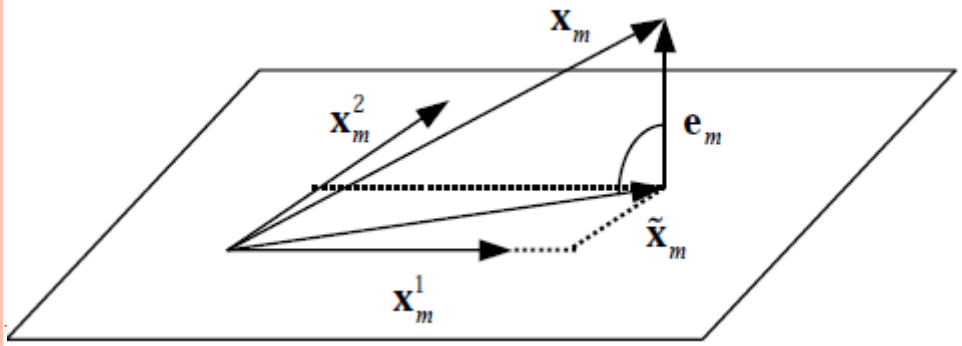
$$E_m = \sum_n e_m^2[n] = \sum_n \left(x_m[n] - \tilde{x}_m[n]\right)^2 = \sum_n \left(x_m[n] - \sum_{j=1}^{p} a_j x_m[n-j] \right)^2$$

  - مشتق‌گیری بر حسب a_j و برابر صفر قرار دادن

ضرب داخلی خطا و سیگنال

$$< \mathbf{e}_m , \mathbf{x}_m^i >= \sum_n e_m[n] x_m[n-i] = 0 \qquad 1 \le i \le p$$

اصل تعامد (Orthogonality Principle): بردار خطای تخمین‌گر بهینه (از نظر مربعات خطا) بر نمونه‌های قبلی سیگنال متعامد است



تحلیل LPC: محاسبه ضرایب ...

معادلات

$$< \mathbf{e}_m , \mathbf{x}_m^i >= \sum_n e_m[n] x_m[n-i] = 0 \qquad 1 \le i \le p$$

$$e[n] = x[n] - \tilde{x}[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k]$$

- تعداد P معادله خطی

$$\sum_n x_m[n-i] x_m[n] = \sum_{j=1}^{p} a_j \sum_n x_m[n-i] x_m[n-j] \qquad i = 1, 2, \ldots, p$$

- تعریف ضرایب هم‌بستگی

$$\phi_m[i,j] = \sum_n x_m[n-i] x_m[n-j]$$

- آنگاه داریم (معادله Yule-Walker):

$$\sum_{j=1}^{p} a_j \phi_m[i,j] = \phi_m[i,0] \qquad i = 1, 2, \ldots, p$$

- معمولاً خطای تخمین به گونه‌ای نرمال می‌شود که انرژی آن واحد باشد

$$\sum_n u_m^2[n] = 1$$

- رابطه بین خطای اصلی و خطای نرمال شده

$$e_m[n] = G u_m[n]$$

- آنگاه مقدار خطای تخمین =

$$E_m = \sum_n e_m^2[n] = G^2 \sum_n u_m^2[n] = G^2$$

تحلیل LPC: محاسبه ضرایب . . .

- روش‌های حل معادلات
  - روش کوواریانس (Covariance)
  - روش خودهمبستگی (Autocorrelation)
  - روش لاتیس: لوینسون–دوربین (Levinson-Durbin)

تحلیل LPC: محاسبه ضرایب . . .

- روش کواریانس . . .

$$\phi_m[i,j]=\sum_{n=0}^{N-1}x_m[n-i]x_m[n-j]=\sum_{n=-i}^{N-1-j}x_m[n]x_m[n+i-j]=\phi_m[j,i]$$

$$\sum_{j=1}^{p}a_j\phi_m[i,j]=\phi_m[i,0] \qquad i=1,2,\ldots,p$$

$$\begin{pmatrix}\phi_m[1,1] & \phi_m[1,2] & \phi_m[1,3] & \cdots & \phi_m[1,p]\\ \phi_m[2,1] & \phi_m[2,2] & \phi_m[2,3] & \cdots & \phi_m[1,p]\\ \phi_m[3,1] & \phi_m[3,2] & \phi_m[3,3] & \cdots & \phi_m[3,p]\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \phi_m[p,1] & \phi_m[p,2] & \phi_m[p,3] & \cdots & \phi_m[p,p]\end{pmatrix}\begin{pmatrix}a_1\\ a_2\\ a_3\\ \cdots\\ a_p\end{pmatrix}=\begin{pmatrix}\phi_m[1,0]\\ \phi_m[2,0]\\ \phi_m[3,0]\\ \cdots\\ \phi_m[p,0]\end{pmatrix} \Rightarrow \Phi\mathbf{a}=\psi$$

- تجزیه ماتریس Φ: $\Phi=\mathbf{VDV}^t$
  - ماتریس V پایین مثلثی با قطر اصلی ۱ است
  - ماتریس D قطری است

$$\phi[i,j]=\sum_{k=1}^{j}V_{ik}d_kV_{jk} \qquad 1\le j<i \Rightarrow V_{ij}d_j=\phi[i,j]-\sum_{k=1}^{j-1}V_{ik}d_kV_{jk} \qquad 1\le j<i$$

$$\phi[i,i]=\sum_{k=1}^{i}V_{ik}d_kV_{ik} \Rightarrow d_i=\phi[i,i]-\sum_{k=1}^{i-1}V_{ik}^2d_k, \quad i\ge 2 \Rightarrow d_1=\phi[1,1]$$

تحلیل LPC: محاسبه ضرایب . . .

- روش کواریانس

$$\begin{cases} \Phi \mathbf{a} = \psi \\ \Phi = \mathbf{V}\mathbf{D}\mathbf{V}^t \end{cases} \Rightarrow \begin{cases} \mathbf{V}\mathbf{Y} = \psi \\ \mathbf{Y} = \mathbf{D}\mathbf{V}^t\mathbf{a} \end{cases} \Rightarrow \mathbf{V}^t\mathbf{a} = \mathbf{D}^{-1}\mathbf{Y}$$

- با داشتن V، مقدار Y (به صورت بازگشتی) قابل محاسبه است

$$Y_i = \psi_i - \sum_{j=1}^{i-1} V_{ij} Y_j \ , \qquad 2 \le i \le p \quad \Rightarrow \quad Y_1 = \psi_1$$

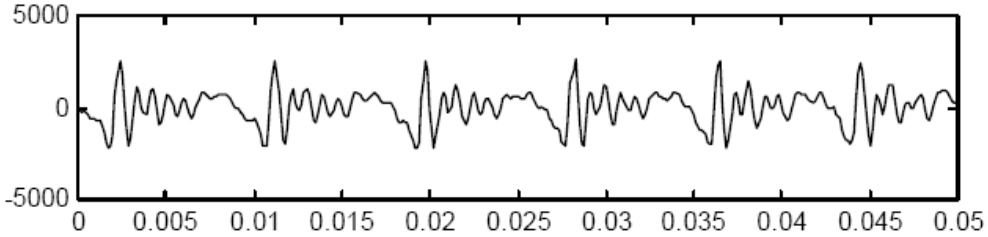
- با داشتن Y، ضرایب a بدست می‌آید

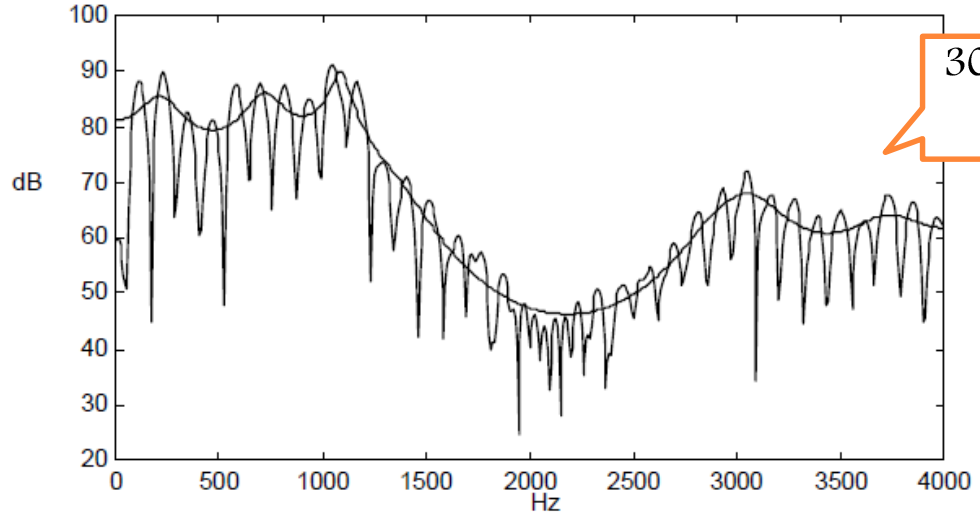
$$\begin{cases} a_i = Y_i / d_i - \sum_{j=i+1}^{p} V_{ji} a_j \ , & 1 \le i < p \\ a_p = Y_p / d_p \end{cases}$$

تحلیل LPC: تحلیل طیفی . . .

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

- تحلیل طیفی با LPC . . .
  - یک فیلتر تمام-قطب (IIR)
  - با رسم $H(e^{j\omega})$ قله‌هایی را در ریشه‌های مخرج داریم





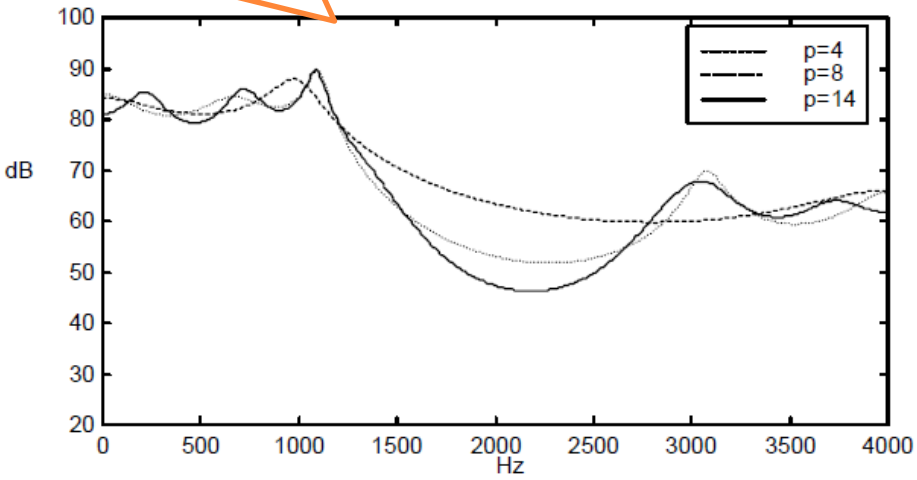
پنجره همینگ 30 ms
مرتبه LPC = 14

تحلیل LPC: تحلیل طیفی . . .

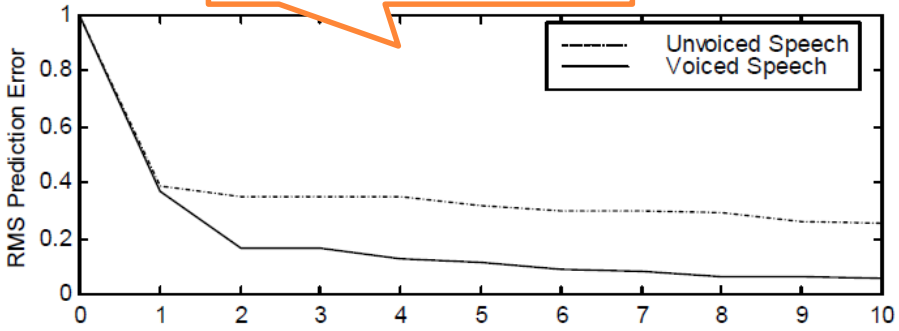
انتخاب مرتبه P

- مدل تمام-قطب مادامی‌که از تعداد قطب‌های زیاد استفاده می‌کنند، تقریب خوبی دارد
  - حتی برای واج‌های خیشومی که صفر هم دارند
- به طور میانگین طیف گفتار شامل یک قطب در هر کیلو هرتز
- در عمل $P = F_s + (2\ or\ 4)$ تقریب خوبی است
  - بسامد نمونه‌گیری برحسب کیلوهرتز F_s

افزایش مرتبه= جزئیات بیشتر در طیف



افزایش مرتبه= کاهش خطا

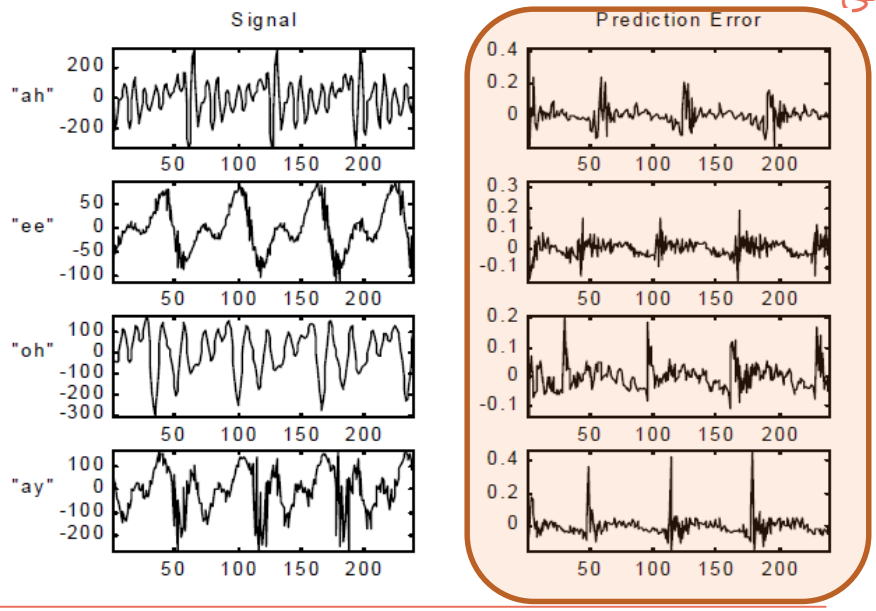


تحلیل LPC: خطای پیش‌بینی . . .

- خطای پیش‌بینی (سیگنال تحریک یا باقی‌مانده)

$$e[n] = x[n] - \tilde{x}[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k]$$

- گفتار بی‌واک: به‌طور تقریبی باید نوفه سفید باشد (در عمل، این تقریب کاملا خوب است)
- گفتار واک‌دار: که به‌طور تقریبی باید زنجیره پالسی باشد (در عمل اینگونه نیست)
  - گفتار واقعی کاملاً دوره‌ای نیست (یک مؤلفه تصادفی نیز دارد) و فرض تمام-قطب در مجموع معتبر نیست (صفرها با فیلتر LPC مدل‌سازی نشده)
  - تولید گفتار (سنتز) با این روش = گفتار رباتی



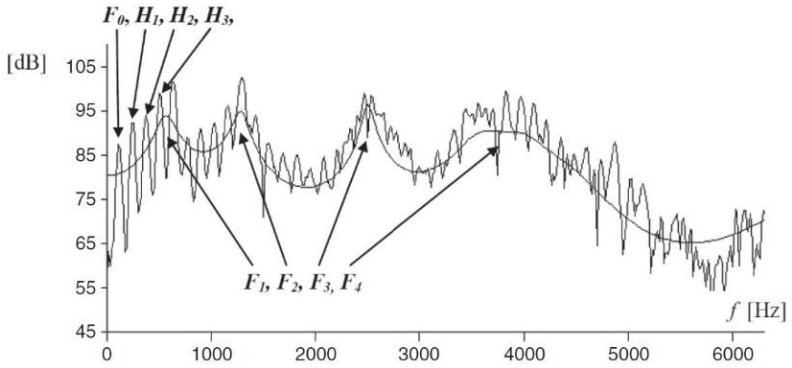
می‌بایست زنجیره پالس باشند

تحلیل LPC: کاربردها ...

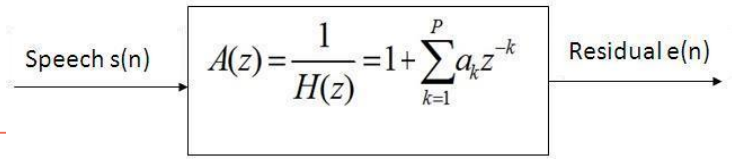
- استفاده در کدینگ و سنتز گفتار
- استفاده از ضرایب LP به عنوان ویژگی (در پردازش گفتار)
- تخمین طیف (پوش طیف) از روی ضرایب LP

$$X(e^{j\omega}) = \frac{G}{FT([1, a_1, a_2, \ldots, a_P])}$$

- تخمین فرکانس فرمنت‌ها
  - قله‌های پوش طیف گفتار

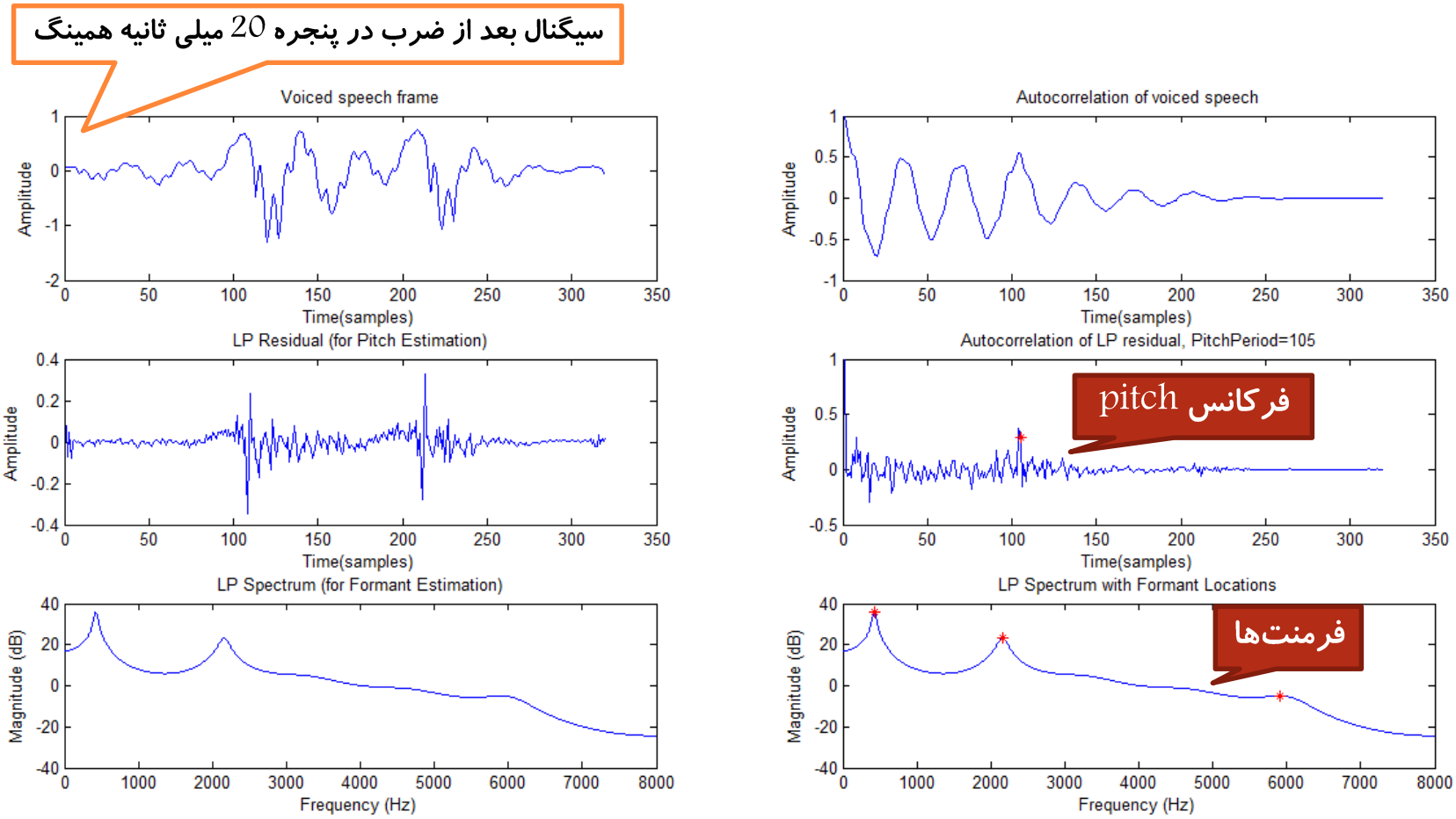


- تخمین فرکانس زیروبمی (Pitch)
  - محاسبه سیگنال تحریک (باقی‌مانده) با فیلتر کردن معکوس
  - محاسبه هم‌بستگی و یافته نقطه بیشینه جهت تخمین دوره تناوب آن



تحلیل LPC: مثال . . .

- یک فریم از واج /e/ فارسی



تحلیل LPC: مثال

- نمونه کد
  - از سایت بردارید

```
FrameLen = 20; % in ms
LPCOrder = 10;
[SpeechSig,Fs,Bits]=wavread('voiced-e.wav');
FrameLen = FrameLen*Fs/1000;
SpeechSig=SpeechSig./(1.01*abs(max(SpeechSig))); % Normalize to [-1,1]
SpeechSig=SpeechSig(FrameLen+1:2*FrameLen);
Win=hamming(FrameLen);
SpeechSig=SpeechSig.*Win; % windowing
SpeechCorr=xcorr(SpeechSig,SpeechSig);
SpeechCorr=SpeechCorr./(abs(max(SpeechCorr)));
SpeechCorr = SpeechCorr(end/2:end);

% Compute LP coeffs
A=SpeechCorr(1:LPCOrder); % P order autocorr
r=SpeechCorr(2:(LPCOrder+1));
A=toeplitz(A); % Toeplitz autocorr matrix
L=-inv(A)*r;
LPCoeffs = [1;L]; % this is eq. to LPCoeffs=lpc(SpeechSig,LPCOrder)

% Compute residuals
Residual =conv(SpeechSig,LPCoeffs); % convolution of speech signal and the filter (inve
Residual=Residual(round(LPCOrder/2):length(Residual)-round(LPCOrder/2)-1);

% Pitch Estimation from Residuals
ResidualCorr=xcorr(Residual,Residual); %auto-correlation
ResidualCorr = ResidualCorr(end/2:end);
ResidualCorr=ResidualCorr./(abs(max(ResidualCorr)));
MinPitch=20; %min pitch period
MaxPitch=160; % max pitch
ResidualCorrInterval=ResidualCorr(MinPitch:MaxPitch);
[PitchVal,PicthLoc]=max(ResidualCorrInterval); % find the (second) peak
PitchPeriod=MinPitch+PicthLoc;
PitchFreq=(1./PitchPeriod)*Fs;

% Formant Estimation from LP Spectrum
LPSpec=abs(fft(LPCoeffs,Fs)); %Calculate LP Spectrum (takig FFT from LP coeffs)
LPSpec=LPSpec.^(-1);
LPSpec=20*log10(LPSpec);
```

تحلیل LPC ...

- روش‌های معادل
  - بسامدهای طیفی خط (LSF: Line Spectral Frequencies)
    - رایج در کدینگ
  - ضرایب بازتاب (Reflection Coefficients)
  - نسبت‌های لگاریتم-مساحت (Log-Area Ratios)
  - ریشه‌های چندجمله‌ای (Roots of Polynomial)

تحلیل LPC

- **مدل عمومی تخمین**

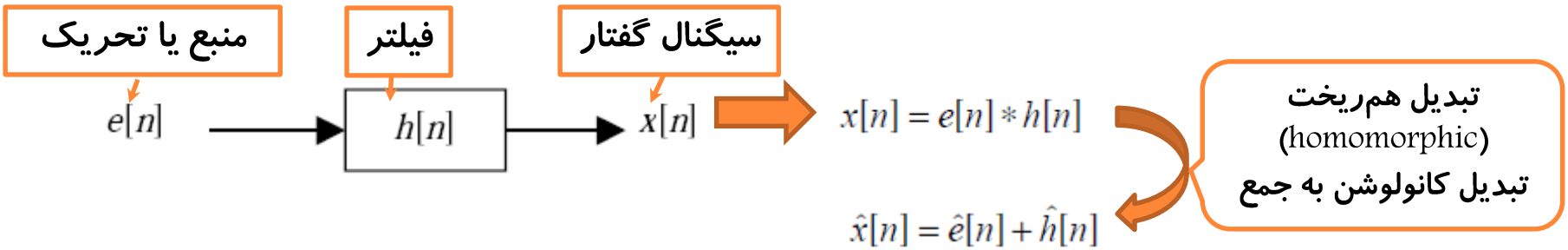
$$x[n] = \sum_{i=1}^{p} a_i x[n-i] + \sum_{j=0}^{q} a_j e[n-j]$$

- تخمین نمونه زمان n بر حسب p نمونه قبلی (x) خودش و q نمونه ورودی (e)
- اگر q=0: تخمین فقط بر حسب نمونه‌های قبلی خودش = مدل AR: Auto-Regressive
  - مدل تمام قطب
- اگر p=0: تخمین فقط بر حسب نمونه‌های ورودی = مدل MA: Moving Average
  - مدل تمام صفر
- اگر p≠0 و q≠0: تخمین بر حسب نمونه‌های قبلی و ورودی = مدل ARMA
  - مدل ترکیبی صفر-قطب

تحلیل کپستروم . . .

- ایده تحلیل کپستروم (Cepstrum)
  - در سال ۱۹۶۴ توسط Bogert (Healy , John Tukey)

- تخمین فیلتر (h) و منبع (e) در مدل منبع-فیلتر با جدا کردن آنها از هم‌دیگر



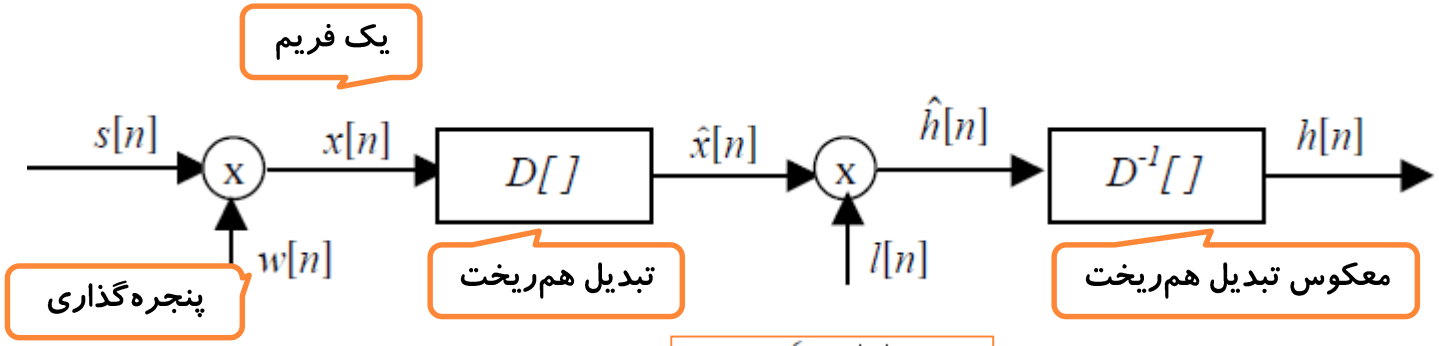
- با داشتن این رابطه، می‌توان فیلتر (h) و منبع (e) را بدست آورد

- کپستروم (Cepstrum) = یک تبدیل هم‌ریخت برای تبدیل کانولوشن به جمع

Hadi Veisi (h.veisi@ut.ac.ir)

تحلیل کپستروم . . .

- تفکیک فیلتر (h) و منبع (e) با تبدیل هم‌ریخت



$$l[n]=\begin{cases}1 & |n|<N\\ 0 & |n|\geq N\end{cases}$$

- در صورتی که بخواهیم e[n] را تخمین بزنیم، از فیلتر l[n] زیر استفاده می‌کنیم

$$l[n]=\begin{cases}1 & |n|\geq N\\ 0 & |n|<N\end{cases}$$

- بخشی از خروجی تبدیل هم‌ریخت معادل فیلتر و بخش دیگر معادل منبع است

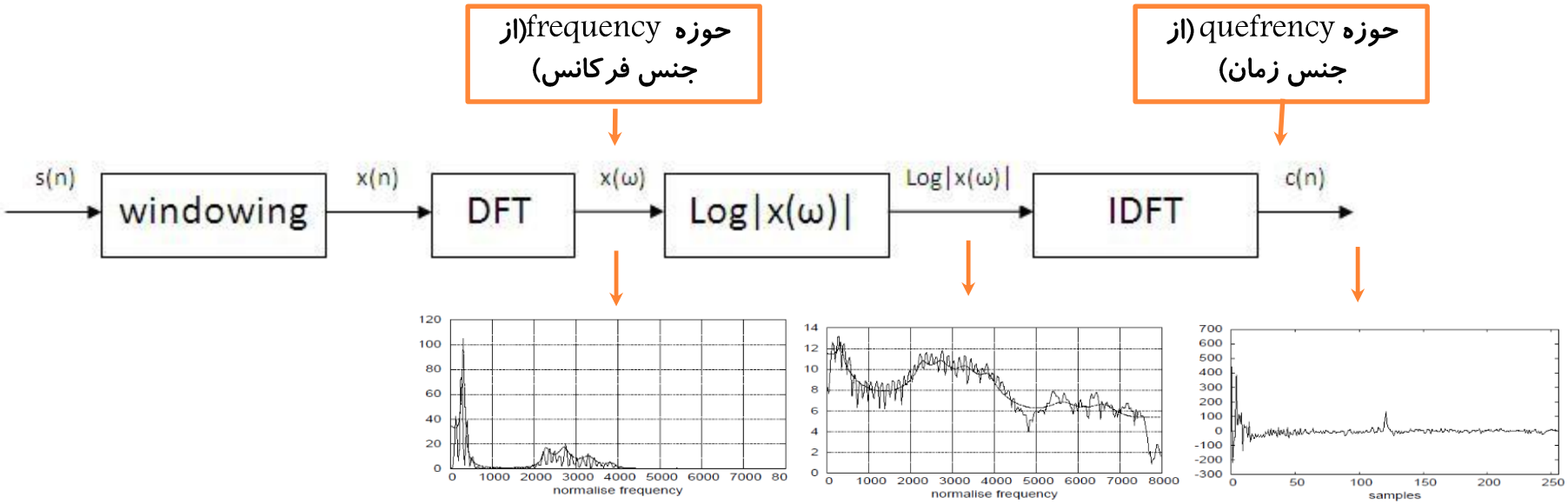
تحلیل کپستروم . . .

تعریف کپستروم حقیقی

- معکوس تبدیل فوریه لگاریتم دامنه تبدیل فوریه

$$c[n] = \mathcal{F}^{-1}\{\log|\mathcal{F}\{x[n]\}|\}$$

$$c[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln|X(e^{j\omega})|\, e^{j\omega n} d\omega$$



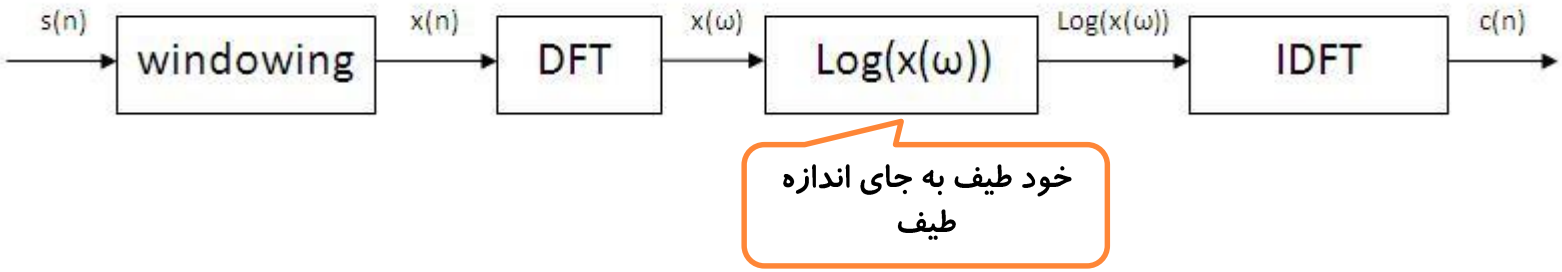
تحلیل کپستروم . . .

- تعریف کپستروم موهومی

$$\hat{x}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln X(e^{j\omega})\, e^{j\omega n} d\omega$$

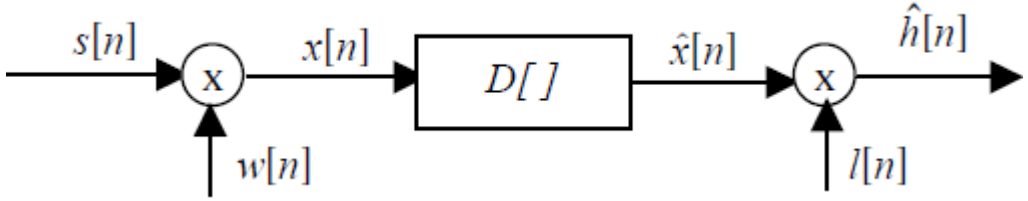
$$\hat{X}(e^{j\omega}) = \ln X(e^{j\omega}) = \ln |X(e^{j\omega})| + j\theta(\omega)$$

$$\theta(\omega) = \arg\left[X(e^{j\omega})\right]$$



تحلیل کپستروم . . .

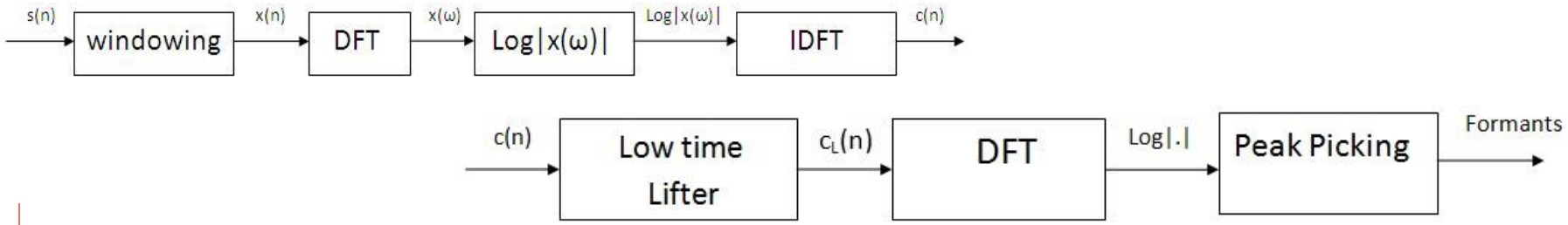
- **لیفتر کردن (Liftering) . . .**
  - **جدا کردن منبع از فیلتر با فیلتر کردن کپستروم**
  - **لیفتر کردن زمان پایین (Low-time liftering)**
    - حفظ بخش ابتدایی کپستروم



$$l[n]=\begin{cases}1 & |n|<N\\ 0 & |n|\geq N\end{cases}$$

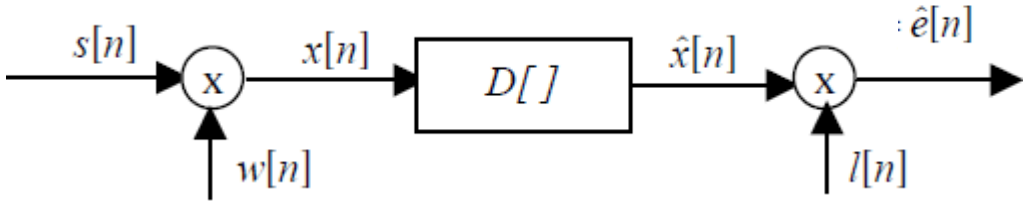
مقدار 15 یا 20

- محاسبه فرمنت‌ها: گرفتن تبدیل فوریه (و بدست آوردن لگاریتم طیف)، و یافتن نقاط بیشینه



تحلیل کپستروم . . .

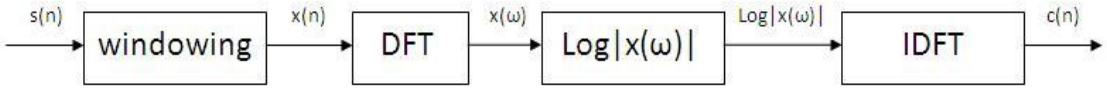
- لیفتر کردن (Liftering) . . .
  - لیفتر کردن زمان بالا (High-time liftering)
    - بخش انتهایی کپستروم = سیگنال تحریک

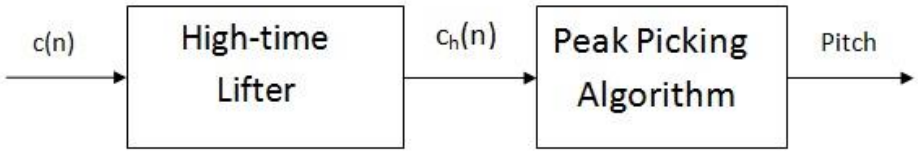


$$l[n] = \begin{cases} 1 & |n| \geq N \\ 0 & |n| < N \end{cases}$$

مقدار 15 یا 20

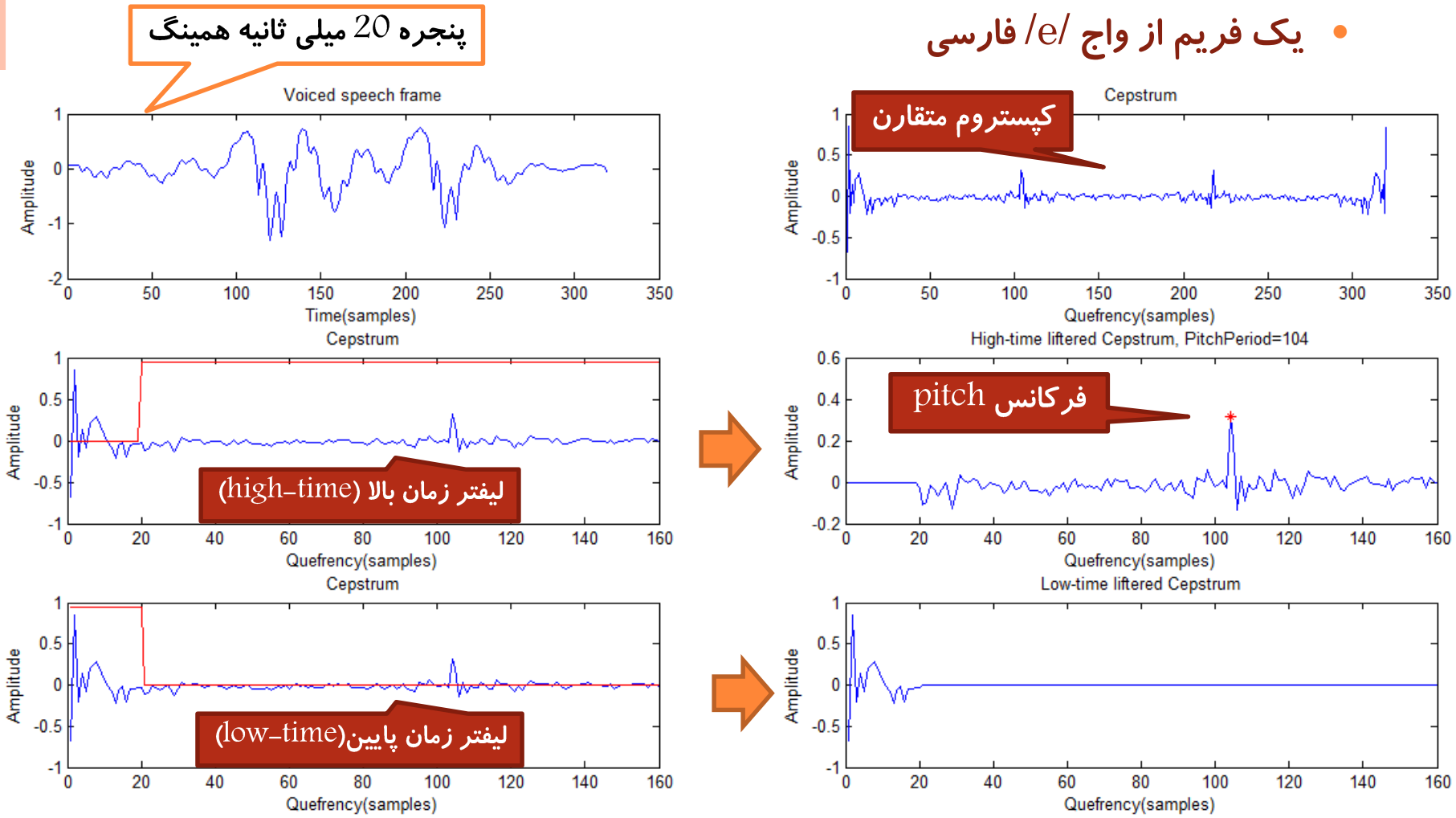
- محاسبه فرکانس زیروبمی (pitch): یافتن نقطه بیشینه روی کپستروم لیفتر شده



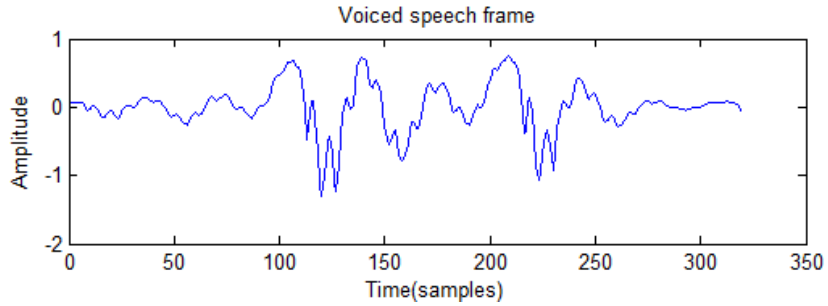


تحلیل کپستروم: مثال . . .

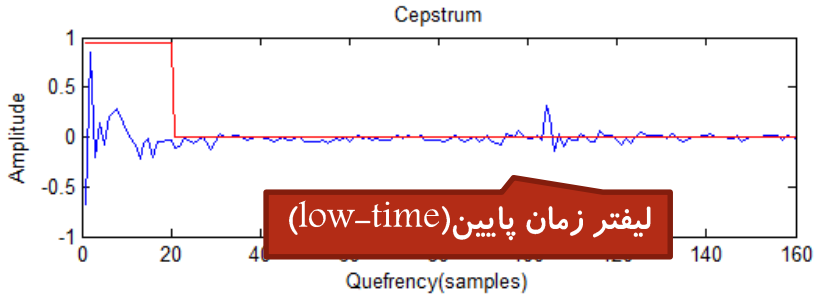
- برای سیگنال واک‌دار
  - یک فریم از واج /e/ فارسی

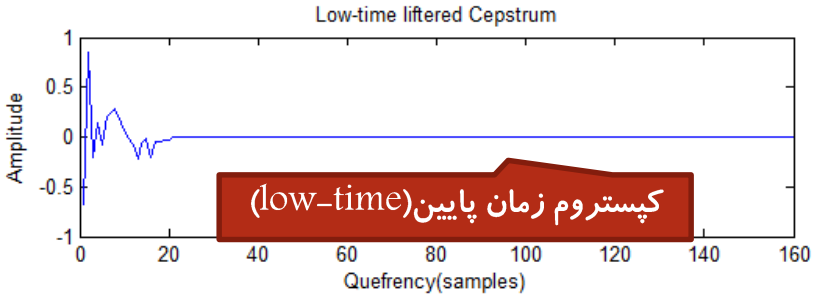


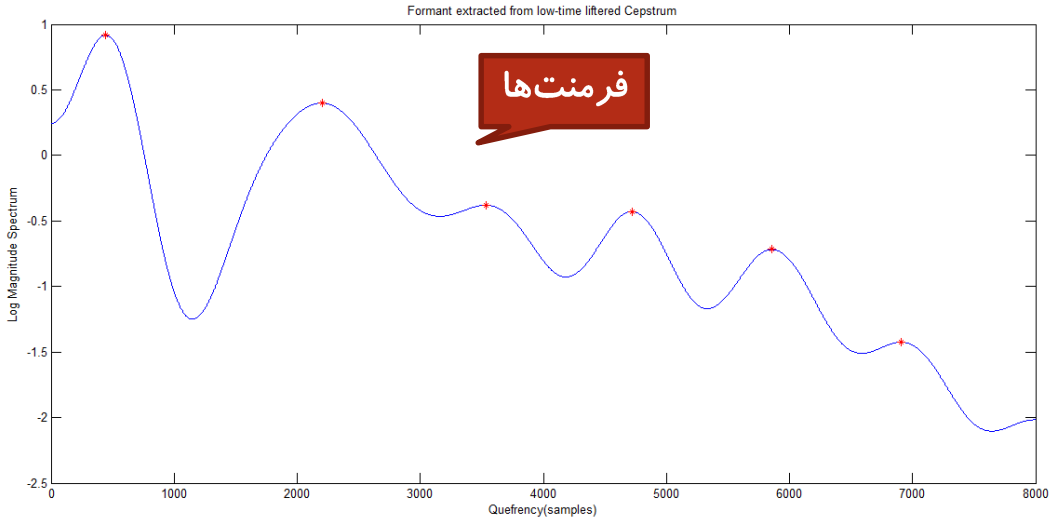
تحلیل کپستروم: مثال . .



- برای سیگنال واک‌دار
  - یک فریم از واج /e/ فارسی





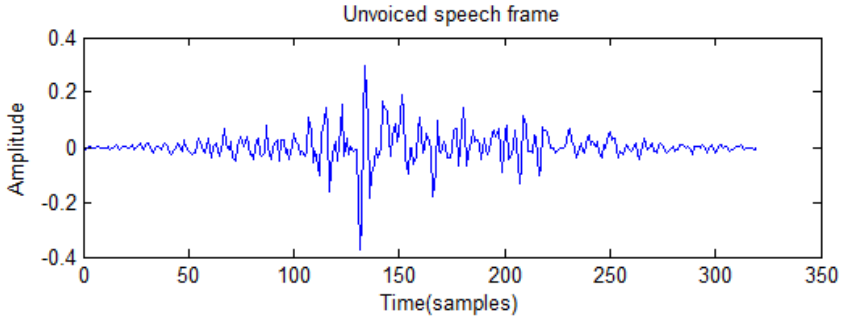


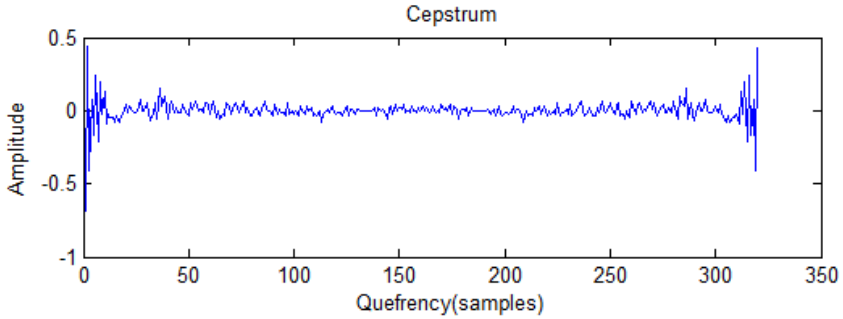
گرفتن تبدیل فوریه
(محاسبه لگاریتم طیف) و
یافتن نقاط بیشینه

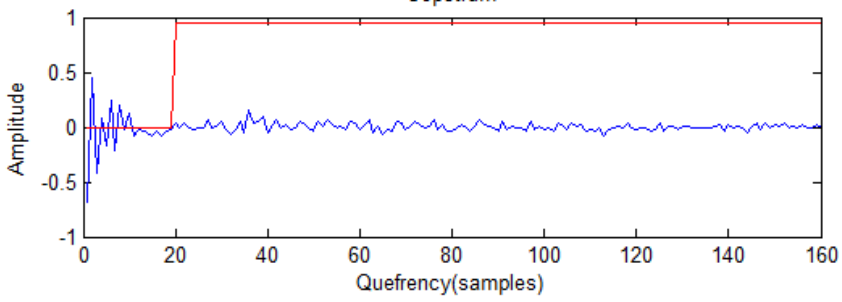
(h.veisi@ut.ac.ir)

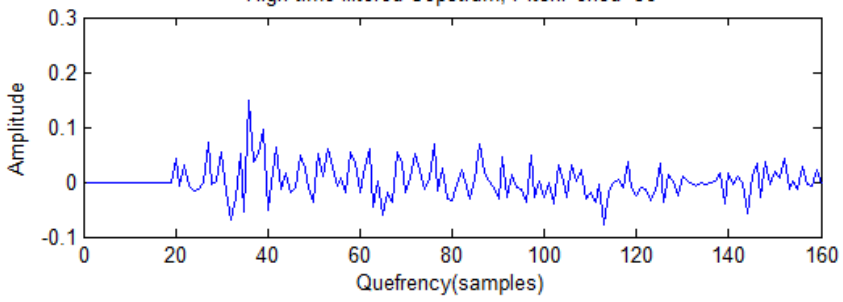
تحلیل کپستروم: مثال . . .

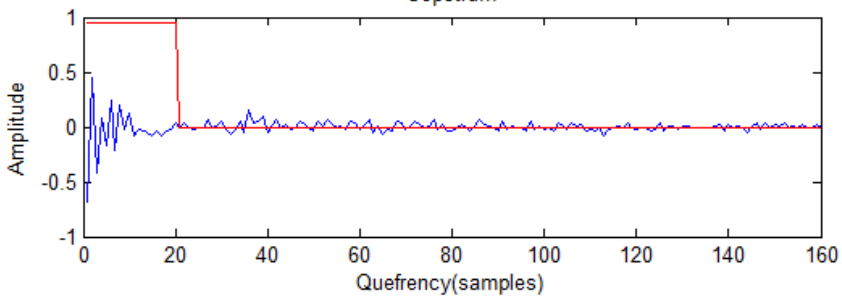
- برای سیگنال بی‌واک
  - یک فریم از واج /f/ فارسی

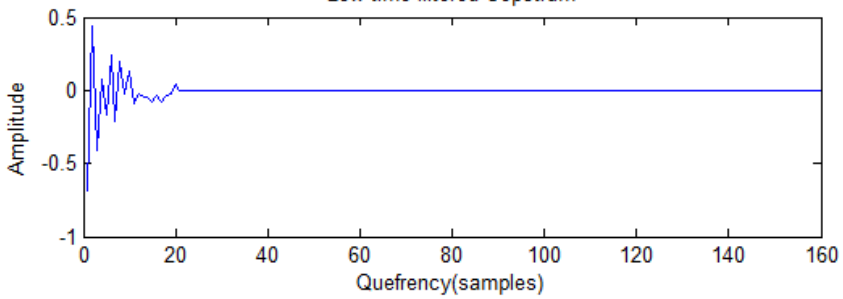












تحلیل کپستروم: مثال

- نمونه کد
  - از سایت بردارید

```
FrameLen = 20; % in ms
LifterCutOff = 20; % cut off of the liftering, 15 or 20
[SpeechSig,Fs,Bits]=wavread('voiced-e.wav');


FrameLen = FrameLen*Fs/1000;
SpeechSig=SpeechSig./(1.01*abs(max(SpeechSig))); % Normalize to [-1,1]
SpeechSig=SpeechSig(FrameLen+1:2*FrameLen);
Win=hamming(FrameLen);
SpeechSig=SpeechSig.*Win; % windowing
Cepstrum=log(abs(fft(SpeechSig)));
Cepstrum=ifft(Cepstrum);

% Liftering (High-time)
Cepstrum2=Cepstrum(1:length(Cepstrum)/2)'; % the cepstrum is symmetric
LifterHigh=zeros(1,length(Cepstrum2));
LifterHigh(LifterCutOff:length(LifterHigh))=1;
CepstrumHighTime=real(Cepstrum2.*LifterHigh);

% Liftering (Low-time)
LifterLow=zeros(1,length(Cepstrum2));
LifterLow(1:LifterCutOff)=1;
CepstrumLowTime=real(Cepstrum2.*LifterLow);

% Pitch estimation
[PitchVal,PitchLoc]=max(CepstrumHighTime);
PitchPeriod=PitchLoc;
PitchFreq=(1/PitchPeriod)*Fs;

% Formant estimation
CepstrumLowTime2=CepstrumLowTime(1:LifterCutOff);
CepstrumLowTime2Spec=fft(CepstrumLowTime2,Fs); % spectrum of low-time cepstrum
CepstrumLowTime2Spec2=CepstrumLowTime2Spec(1:Fs/2); % spectrum is symmetric
CepstrumLowTime2Spec2=real(CepstrumLowTime2Spec2);
k=1;
for i=2:length(CepstrumLowTime2Spec2)-1
  if (CepstrumLowTime2Spec2(i-1)<CepstrumLowTime2Spec2(i)) & (CepstrumLowTime2Spec2(i+1)<Ce
```

تحلیل کپستروم

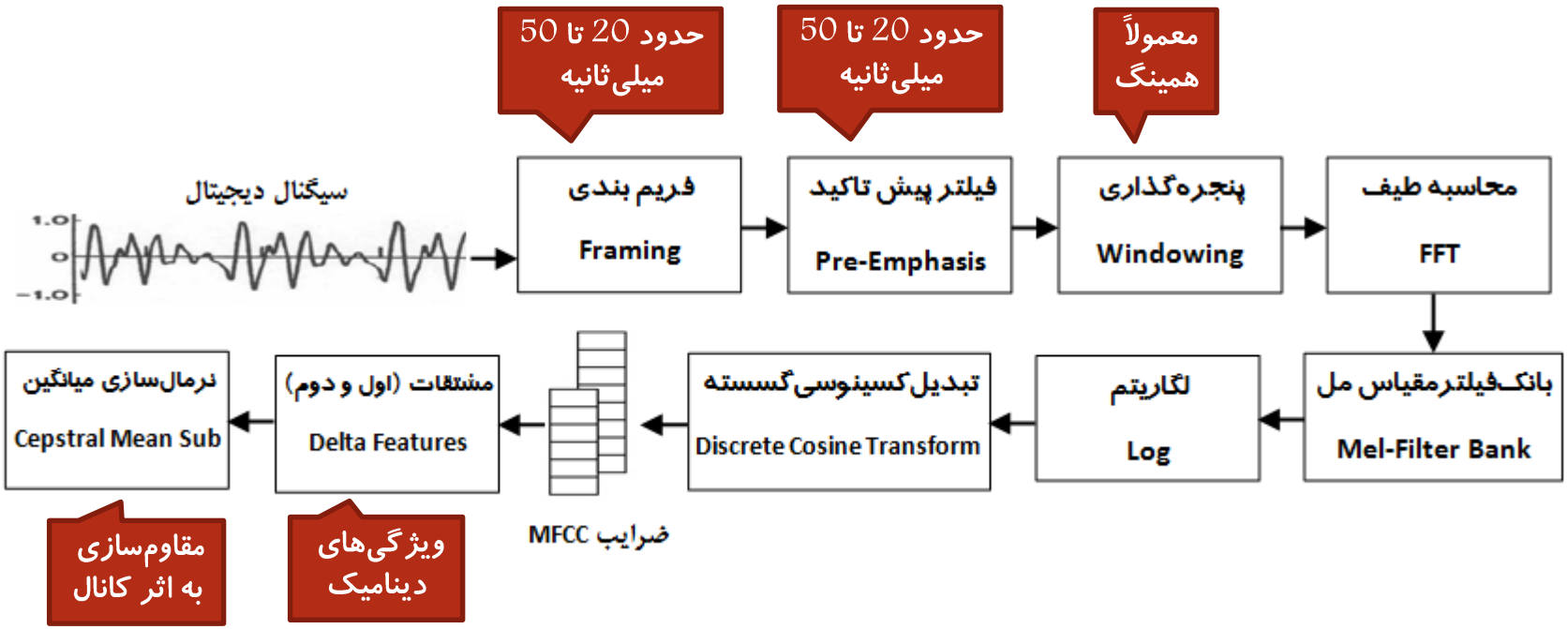
- در نظر گرفتن لگاریتم طیف به عنوان شکل موج سیگنال
- استفاده به عنوان ویژگی: نمایش فشرده پوش طیف
  - ضرایب مستقل از هم
- توانایی بالا در تشخیص واک‌داری و فرکانس Pitch
- نام‌گذاری: معکوس کردن هجای اول کلمات معادل
  - کپستروم (Cepstrum) در مقابل اسپکتروم (طیف) (Spectrum)
  - حوزه Quefrency در مقابل Frequency
  - لیفتر کردن (Liftering) در مقابل فیلتر کردن (Filtering)

روش MFCC . . .

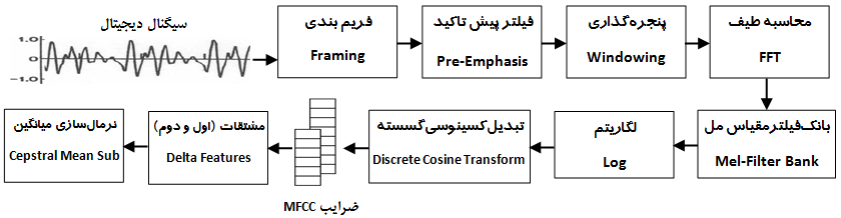
- **ضرایب کپستروم در مقیاس مل**
  - MFCC: Mel-Frequency Cepstral Coefficients
- **در سال ۱۹۸۰ توسط** Davis و Mermelstein
- **پرکاربردترین ویژگی گفتار در سیستم‌های تشخیص گفتار**
- **در واقع نوعی کپستروم حقیقی است (با تفاوت‌های زیر)**
  - تبدیل فوریه مورد استفاده در آن FFT است
  - تبدیل معکوس فوریه DCT است
  - یک مقیاس غیرخطی (مل) فرکانسی در آن استفاده می‌شود = شبیه‌سازی رفتار سیستم شنوایی

روش MFCC . . .

- مراحل



روش MFCC ...

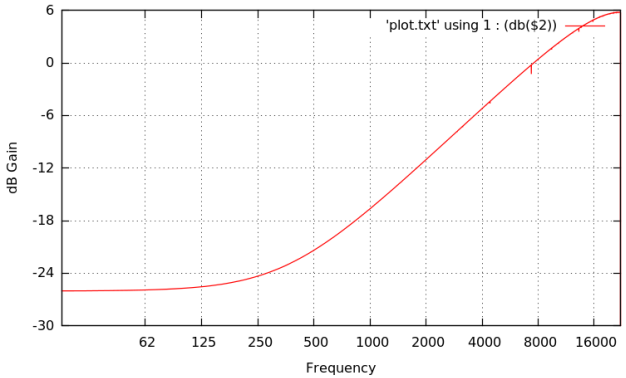


۱- فریم‌گذاری

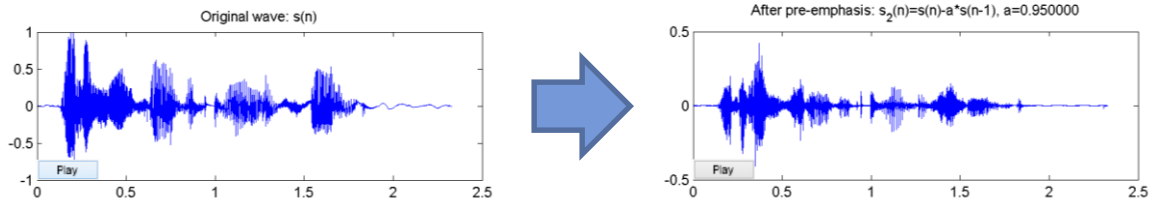
- طول هر فریم از کاربردی به کاربرد دیگر متفاوت است (معمولاً بین ۱۰ تا ۵۰ میلی ثانیه)
- فریم‌های متوالی با همدیگر همپوشانی (Overlap) دارند (۲۵٪ تا ۷۵٪ طول فریم)
- برای سیگنالی با نرخ نمونه برداری 16kHz و فریم‌های 20 ms: بردار N=320 بُعدی

۲- فیلتر پیش‌تأکید (Pre-Emphasis Filter)

$$x[n] = x[n] - \alpha x[n-1], \quad 0 \le n < N$$

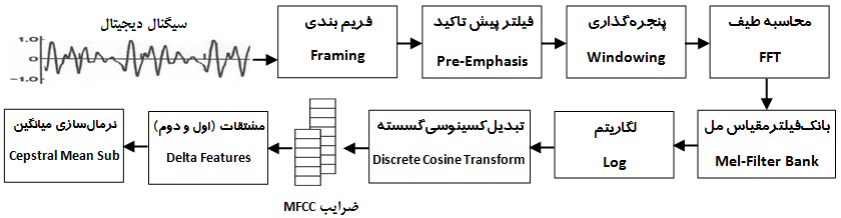


- فیلتر بالاگذر برای تقویت فرکانس‌های (فرمنت‌های) بالا
  - حذف اثرات طیفی حنجره (دو قطب) و لب‌ها (یک صفر)
- مقدار معمول برای ضریب پیش‌تاکید $\alpha=0.95$
  - همواره $0.9 \le \alpha \le 1.0$
- بردار N=320 بُعدی



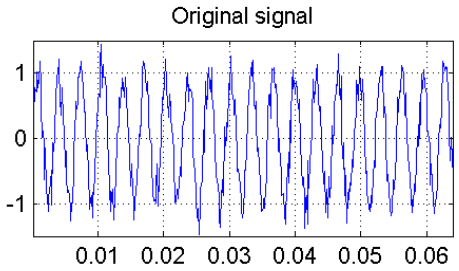
Hadi Veisi (h.veisi@ut.ac.ir)

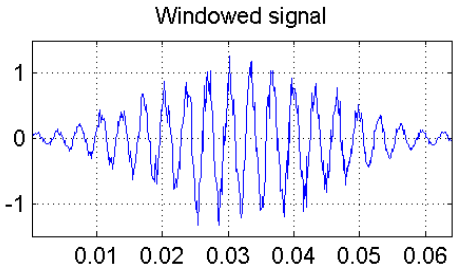
روش MFCC ...



۳- پنجره‌گذاری

- معمولاً پنجره همینگ- بردار N=320 بُعدی

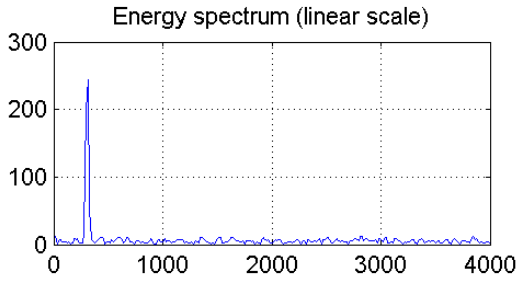


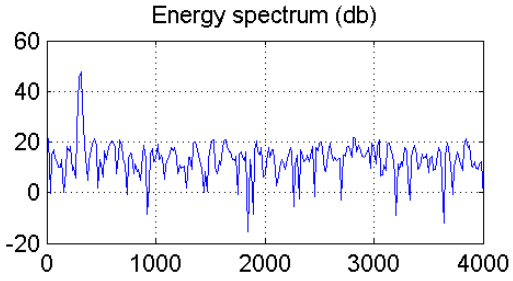


۴- محاسبه (توان) طیف

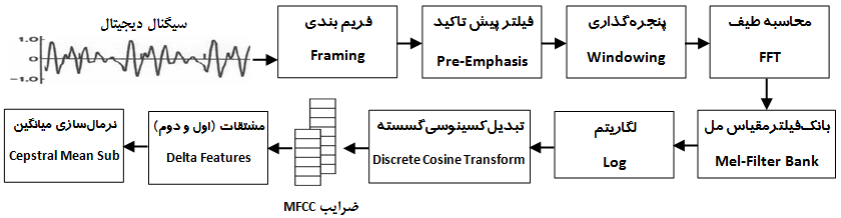
$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \qquad 0 \le k < N$$

- استفاده از FFT
- چون سیگنال حقیقی است، متقارن است، کافیست نصف آن را نگه داریم
- بردار N=160 بُعدی



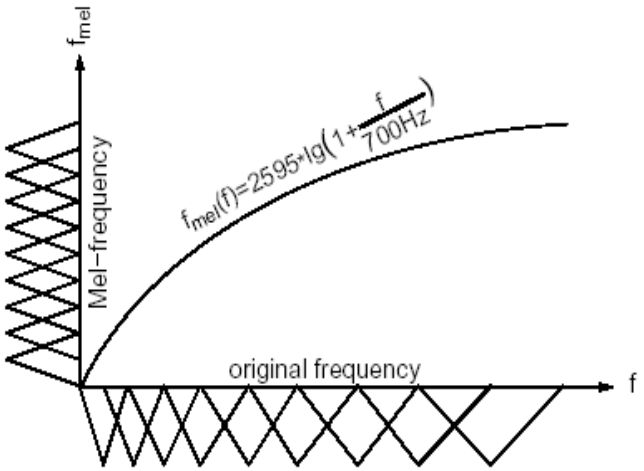


روش MFCC . . .



۵- بانک فیلتر در مقیاس مل . . .

- مدل کردن حساسیت گوش انسان نسبت به حوزه‌های مختلف فرکانس
- گوش به اطلاعات فرکانس پایین ارزش بیشتری می‌دهد
- عملکرد گوش برای فرکانس‌های کمتر از یک کیلو هرتز، خطی و برای فرکانس‌های بالاتر لگاریتمی است



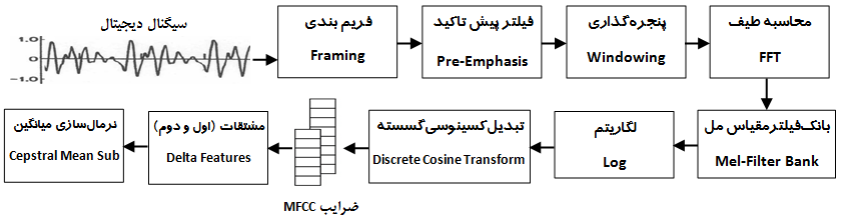
- استفاده از تعداد محدودی فیلتر
  - بین ۲۰ تا ۳۰ فیلتر (مقدار رایج = ۲٤)
- کاهش ابعاد بردار ویژگی: N=24 بُعدی
- ضرب فیلترها در طیف و محاسبه انرژی هر فیلتر

فیلترها

تعداد فیلترها

$$S[m] = \left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k] \right], \quad 0 \le m < M$$

روش MFCC ...

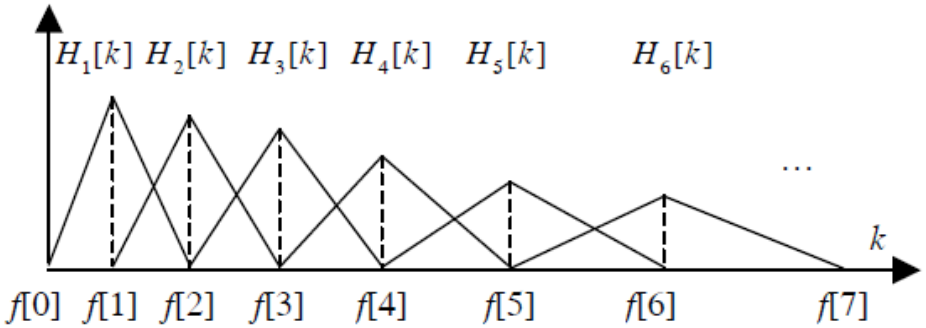


۵- بانک فیلتر در مقیاس مل

- فیلترها

Index	Mel Scale	
	Center Freq. (Hz)	BW (Hz)
1	100	100
2	200	100
3	300	100
4	400	100
5	500	100
6	600	100
7	700	100
8	800	100
9	900	100
10	1000	124
11	1149	160
12	1320	184
13	1516	211
14	1741	242
15	2000	278
16	2297	320
17	2639	367
18	3031	422
19	3482	484
20	4000	556
21	4595	639
22	5278	734
23	6063	843
24	6964	969

برای 8 kHz



$$H_m'[k]=\begin{cases}0 & k<f[m-1]\\ \dfrac{(k-f[m-1])}{(f[m]-f[m-1])} & f[m-1]\le k\le f[m]\\ \dfrac{(f[m+1]-k)}{(f[m+1]-f[m])} & f[m]\le k\le f[m+1]\\ 0 & k>f[m+1]\end{cases}$$

اندازه FFT

کمینه فرکانس

بیشینه فرکانس

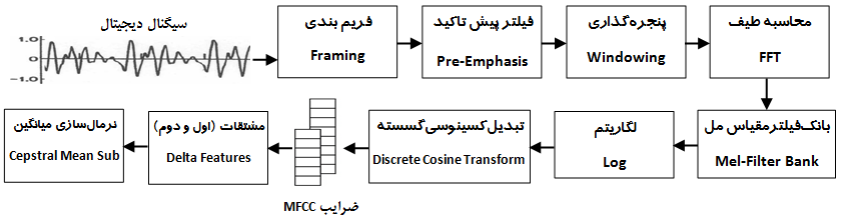
تعداد فیلترها

$$f[m]=\left(\frac{N}{F_s}\right)B^{-1}\left(B(f_l)+m\frac{B(f_h)-B(f_l)}{M+1}\right)$$

$$B(f)=1125\ln(1+f/700)$$

$$B^{-1}(b)=700\left(\exp(b/1125)-1\right)$$

روش MFCC ...



۶- تبدیل لگاریتم

$$S[m] = \ln\left[S[m] \right]$$

- بردار ویژگی: N=24 بُعدی

۷- تبدیل کسینوسی گسسته (DCT: Discrete Cosine Transform)

- معادل معکوس تبدیل فوریه در کپستروم حقیقی

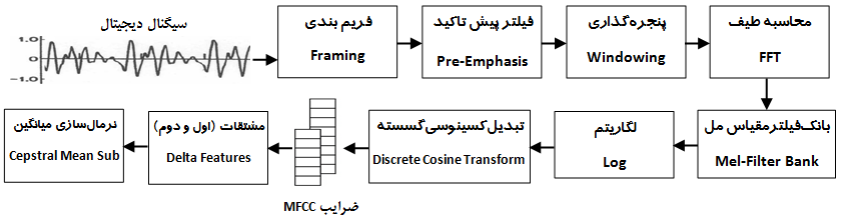
$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left(\pi n (m + 1/2) / M\right) \qquad 0 \le n < M$$

معمولاً این مقدار مقدار نصف تعداد فیلترهاست

ویژگی‌های MFCC

- کاهش ابعاد بردار ویژگی: N=12 بُعدی
- کاهش ابعاد بردار ویژگی از ۳۲۰ به ۱۲!

روش MFCC . . .



مشتقات ضرایب MFCC

- ضرایب MFCC فقط شامل اطلاعات استاتیک هر فریم است
- اطلاعات پویا و اثر فریم‌های مجاور (به دلیل غیر ایستان بودن گفتار) نیز لازم است
- محاسبه مشتقات زمانی از روی فریم‌های مجاور
  - مشتق اول(دلتا) و مشتق دوم(دلتا-دلتا)
- روش رگرسیون خطی (k فریم قبل و k فریم بعد)

مقدار معمول =2

$$\Delta C[n]=\sum_{i=-k}^{k}(i.C[n+i])/\sum_{i=-k}^{k}i^2$$

$$\Delta^2 C[n]=\frac{2\left\{\left(\sum_{i=-k}^{k}i^2\right)\left(\sum_{i=-k}^{k}C[n+i]\right)-(2k+1)\sum_{i=-k}^{k}\left(i^2C[n+i]\right)\right\}}{\left(\sum_{i=-k}^{k}i^2\right)^2-(2k+1)\left(\sum_{i=-k}^{k}i^4\right)}$$

- روش تفاضل (ساده‌تر)

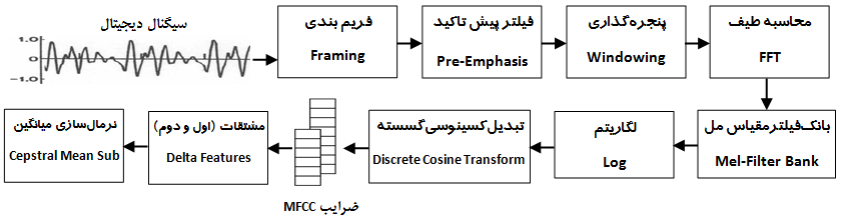
مقدار معمول 1 (یا 2)

$$\Delta C[n]=C[n+k]-C[n-k]$$

$$\Delta^2 C[n]=\Delta C[n+k]-\Delta C[n-k]=C[n+2k]-2C[n]+C[n-2k]$$

- اضافه کردن مشتق اول و دوم به بردار ویژگی‌ها: اندازه بردار ویژگی: $N=3*12$ بُعدی

روش MFCC ...



نرمال‌سازی میانگین

- تفاضل میانگین کپسترال (CMS: Cepstral Mean Subtraction)
- نرمال‌سازی میانگین کپسترال (CMN: Cepstral Mean Normalization)

- یکی از رایج‌ترین، ساده‌ترین و مؤثرترین روش‌های نرمال‌سازی
- برای حذف اثر کانال (مثل خط تلفن، تنوع میکروفون و ...) و نویزهای کانوالوشونده

- متوسط بردارهای ویژگی کپسترال (در طول چند صد فریم) محاسبه و سپس این مقدار میانگین از هر یک از بردارها کم می‌شود
  - نویز کانوالوشونده در حوزهٔ زمان = ضرب شونده در حوزه طیف = جمع‌شونده در حوزهٔ کپسترال

$$\mathbf{C}[k] = \mathbf{C}[k] - \boldsymbol{\mu} \qquad \boldsymbol{\mu} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{C}[k]$$

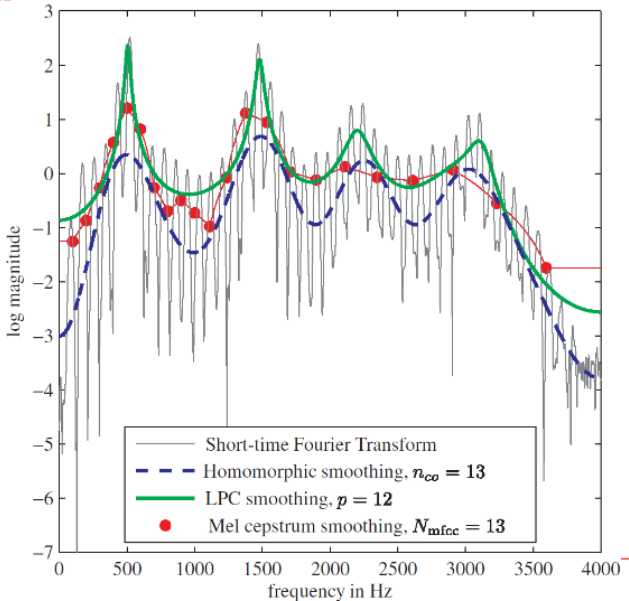
بردار ویژگی‌های MFCC

روش MFCC

- **تبدیلی هم‌ریخت نیست**
  - مگر اینکه جای لگاریتم گرفتن و محاسبه انرژی فلیتر بانک عوض شود

$$S[m]=\ln\left[\sum_{k=0}^{N-1}\left|X_a[k]\right|^2 H_m[k]\right],\quad 0\le m<M \quad\Rightarrow\quad S[m]=\sum_{k=0}^{N-1}\ln\left(\left|X_a[k]\right|^2 H_m[k]\right),\quad 0\le m<M$$

  - اما تقریبی از یک تبدیل هم‌ریخت است
  - مزیت محاسبه به ترتیب بیان شده در MFCC مقاوم بودن به نویز در تخمین طیف



- **برای تشخیص گفتار بسیار مناسب است**
  - در سایر کاربردها هم کارایی خوبی دارد!

فرکانس زیروبمی (Pitch) ...

کاربردها

- ویژگی بسیار مهمی در سنتز گفتار برای مدل‌سازی نوا
- استفاده در زبان‌های زبان‌های (Tonal) در تشخیص گفتار
  - تغییر زیروبمی سبب تغییر معنی می‌شود
- استفاده به عنوان یک ویژگی (در تشخیص گوینده)
  - در مدل منبع-فیلتر یکی از پارامترهای تولید گفتار است

روش‌های تشخیص

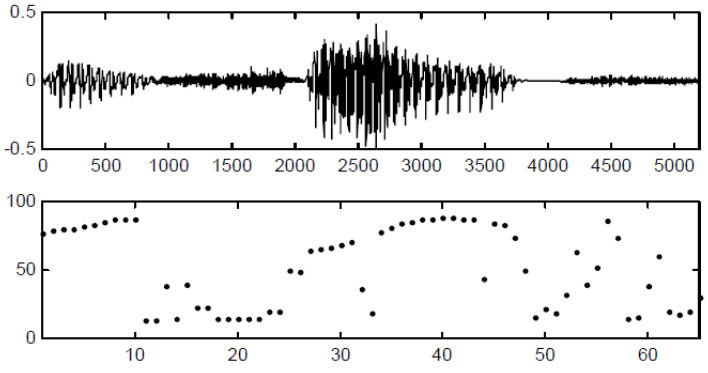
- مبتنی بر کپستروم
- خودهمبستگی (بیشینه مقدار خودهمبستگی به غیر از نقطه صفر)
- هم‌بستگی متقاطع نرمال شده (Normalized Cross-Correlation)

فرکانس زیروبمی (Pitch) . . .

- روش خودهمبستگی . . .
  - بیشترین مقدار تابع خودهمبستگی در قاب (فریم)
    - نقطه $m=0$ (شروع سیگنال) بیشینه‌ی مطلق تابع خودهمبستگی است و باید آن را نادیده گرفت.
    - N= تعداد نمونه ها

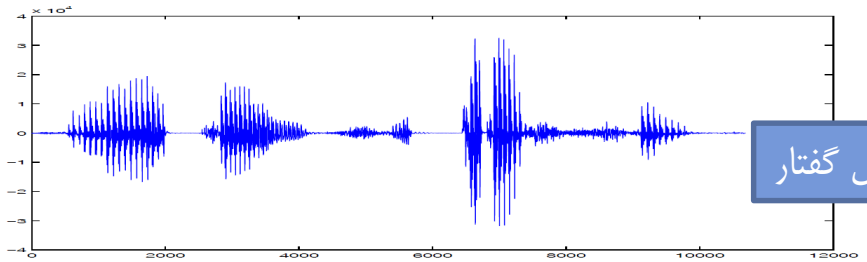
$$\hat{R}[m] = \frac{1}{N} \sum_{n=0}^{N-1-|m|} w[n]\mathbf{x}[n]w[n+|m|]\mathbf{x}[n+|m|]$$

  - دوره‌های زیروبمی می‌توانند حداقل ۴۰ هرتز (برای صدای مردانه با زیروبمی بسیار پایین) و حداکثر ۶۰۰ هرتز (برای صدای زنانه یا کودکانه با زیروبمی بسیار بالا) باشند
    - جستجو برای بیشینه درون یک بازه صورت می‌گیرد
  - مقادیر زیروبمی در مناطق بی‌واک تصادفی هستند

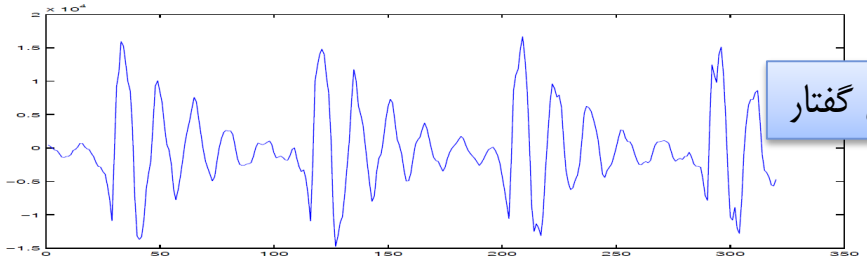


فرکانس زیروبمی (Pitch) ...

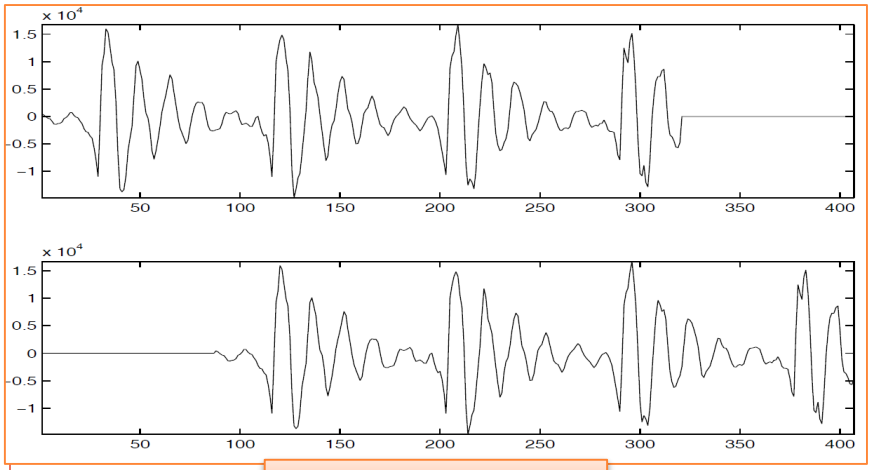
- روش خودهمبستگی



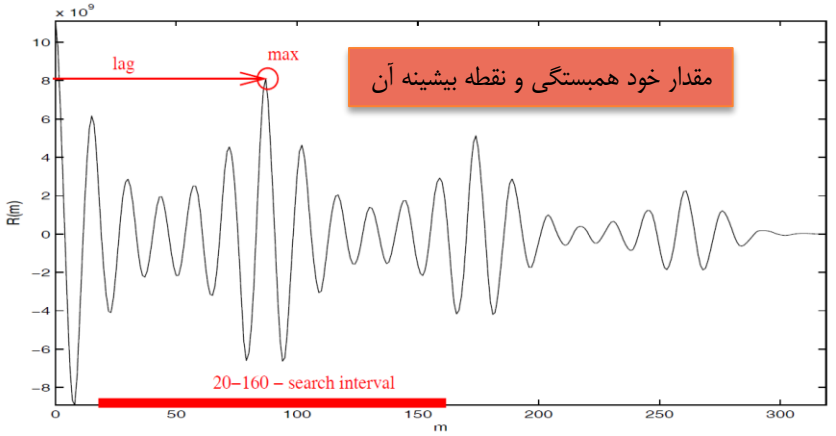
سیگنال گفتار



یک فریم از سیگنال گفتار

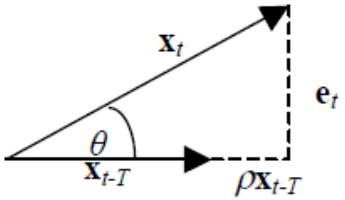


فریم اصلی و نسخه جابجا شده آن



مقدار خود همبستگی و نقطه بیشینه آن

فرکانس زیروبمی (Pitch)



- روش هم‌بستگی متقاطع نرمال شده
  - مشابه روش خودهمبستگی با مقداری بهبود

$$\alpha_t(T) = \cos(\theta) = \frac{< \mathbf{x}_t, \mathbf{x}_{t-T} >}{|\mathbf{x}_t||\mathbf{x}_{t-T}|} = \frac{\sum_{n=-N/2}^{N/2-1} x[t+n]x(t+n-T)}{\sqrt{\sum_{n=-N/2}^{N/2-1} x^2[t+n] \sum_{m=-N/2}^{N/2-1} x^2[t+m+T]}}$$

- این تخمین غیراریب است (روش خودهمبستگی اریب دارد)
- واریانس پایین‌تری از تخمین خودهمبستگی دارد
- برخلاف روش خودهمبستگی، طول پنجره می‌تواند کمتر از دوره زیروبمی باشد
  - به‌طوری‌که فرض مانا بودن صحیح‌تر است و وضوح زمانی بیشتری دارد
- ردیابی زیروبمی با این روش معمولاً بهتر از خودهمبستگی هستند اما مستلزم محاسبات بیشتری هستند

Hadi Veisi (h.veisi@ut.ac.ir)