Mel-Wiener Filter for Mel-LPC Based Speech Recognition

IETCE TRANSACTIONS on Information and System VOL.E90-D, No.6
JUNE 2007

Author : Md. Babul ISLAM, student Member, Kazumasa YAMAMOTO, Member, and Hiroshi MATSIMOTO, Fellow

Professor:陳嘉平
Reporter:葉佳璋

Outline

- Introduction
- Overview of Mel-LPC analysis
- Mel-Wiener Filter
- Approximation of Crosscorrelation Function
- System overview
- Experiments

Introduction

- As the front-end of a speech recognition system, spectral analysis with auditory-like frequency resolution has been shown to be more effective for speech recognition.

- In filter-bank based system, MFCC is widely used. On the other hand, as an LPC based method, we previously proposed a simple and efficient time domain technique to estimate an all-pole model on the mel-frequecy scale, which is referred to as Mel-LPC .

Introduction

- Since SS is a frequency domain filtering method, it is not appropriate for time domain front-end, such as in Mel-LPC analysis. On the contrary, Wiener filter is possible to design and/or implement in both frequency and time domain.
- For Mel-LPC based analysis, the traditional formulation of Mel-Wiener filter requires the frequency warped noisy speech signal.

- This paper proposes a novel approach to estimating the Mel-Wiener filter from input speech signal on the linear frequency scale.

Overview of Mel-LPC analysis

- The frequency warped signal $\tilde{x}[n]$(n=0, ...,∞) obtained by the bilinear transformation of a finite length windowed signal x[n](n=0,…,N-1) is defined by

$$\tilde{X}(\tilde{z}) = \sum_{n=0}^{\infty} \tilde{x}[n]\tilde{z}^{-n} = X(z) = \sum_{n=0}^{N-1} x[n]z^{-n}$$

  where $\tilde{z}^{-1}$ is the first-order all-pass filter, $\tilde{z}^{-1} = \dfrac{z^{-1} - \alpha}{1 - \alpha \cdot z^{-1}}$

- The phase response of $\tilde{z}^{-1}$ is given by

$$\tilde{\lambda} = \lambda + 2\tan^{-1}\left\{\frac{\alpha \sin\lambda}{1 - \alpha\cos\lambda}\right\}$$

This phase function determines a frequency mapping.

- Letting f(n) represent original sequence, and g(k) represent the transformed sequence, we consider linear transformations between f(n) and g(k) corresponding to expanding f(n) in terms of a set of linearly independent sequence so t $\psi_k(n)$ that

$$f(n) = \sum_{k=-\infty}^{\infty} g(k)\psi_k(n)$$

$$G(e^{j\omega}) = \sum_{k=-\infty}^{\infty} g(k)e^{-j\omega k}$$

$$F(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} f(n)e^{-j\Omega n}$$

$$\omega = \theta(\Omega)$$

$$G(e^{j\theta(\Omega)}) = F(e^{j\Omega})$$

$$\Psi_k(e^{j\Omega}) = e^{-jk\theta(\Omega)}$$

Where

$$\Psi_k(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} \psi_k(n) e^{-j\Omega n}$$

- Therefore, the function $\psi_k(n)$ must have an all-pass characteristic, that is their z transform on the unit circle must have unity magnitude independent of frequency.

$$\Psi_k(z) = \left(\frac{z^{-1} - \alpha}{1 - \alpha \cdot z^{-1}} \right)^k$$

$$f(n) = \sum_{k=-\infty}^{\infty} g(k)\psi_k(n)$$

$$\sum_{k=-\infty}^{\infty} f(n) \cdot e^{-j\Omega n} = \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} g(k)\psi_k(n) \right) \cdot e^{-j\Omega n}$$

$$F(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} g(k) \sum_{n=-\infty}^{\infty} \psi_k(n) \cdot e^{-j\Omega n}$$

$$F(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} g(k)\Psi_k(e^{j\Omega})$$

$$F(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} g(k)e^{-jk\theta(\Omega)}$$

$$F(e^{j\Omega}) = G(e^{j\theta(\Omega)})$$

Overview of Mel-LPC analysis

- In Mel-LPC analysis, the spectral envelope of $\widetilde{X}(\tilde{z})\widetilde{W}(\tilde{z})$ is approximated by the following all-pole model

$$\widetilde{H}_a(\tilde{z}) = \frac{\widetilde{\sigma}_e}{1 + \sum_{k=1}^{p} \widetilde{a}_k \tilde{z}^{-k}}$$

where $\widetilde{a}_k$ is the kth mel-prediction coefficient and $\widetilde{\sigma}_e^2$ is the residual energy

- The frequency weighting function $\widetilde{W}(e^{j\tilde{\lambda}})$is defined by

$$\widetilde{W}(e^{j\tilde{\lambda}}) = \frac{\sqrt{1-\alpha^2}}{1+\alpha\tilde{z}^{-1}}$$

which is derived from

$$\frac{d\lambda}{d\tilde{\lambda}} = \widetilde{W}(e^{j\tilde{\lambda}})$$

Mel-Wiener Filter

- Formulation on the warped Frequency scale:

  briefly describes the conventional Wiener filtering on a warped frequency domain, which can be implemented by applying the traditional wiener filter to a bilinear transformed signal.
- Formulation on the Linear Frequency scale

Formulation on the warped Frequency scale

- We define a wiener filter $\widetilde{H}(\tilde{z})$ on the warped frequency scale as

$$\widetilde{H}(\tilde{z}) = \sum_{n=0}^{p-1} \tilde{h}[n]\tilde{z}^{-n} \qquad \hat{\tilde{s}}[n] = \sum_{n=0}^{p-1} \tilde{h}[k]\tilde{x}[n-k]$$

By applying $\widetilde{H}(\tilde{z})$ to the bilinear transformed signal $\tilde{x}[n]$ the estimated clean speech $\hat{\tilde{s}}[n]$

- Since the error signal $\tilde{e}[n] = \tilde{s}[n] - \hat{\tilde{s}}[n]$ is an infinite sequence, the sum of the square error is evaluated by

$$\xi\{\tilde{h}\} = \sum_{n=0}^{\infty} \tilde{e}[n]^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\widetilde{S}(e^{j\tilde{\lambda}}) - \widetilde{H}(e^{j\tilde{\lambda}})\widetilde{X}(e^{j\tilde{\lambda}})\right|^2 d\tilde{\lambda}$$

Formulation on the Linear Frequency scale

- We define the transfer function of a frequency warped Winer filter on z domain by

$$\widetilde{H}_w(\tilde{z}(z)) = \sum_{n=0}^{p-1} \widetilde{h_w}[n]\tilde{z}^{-n}$$

the estimated speech $\hat{s}_w[n]$ based on filter $\widetilde{H_w}(\tilde{z}(z))$ is given by

$$\widehat{s_w}[n] = \sum_{n=0}^{p-1} \tilde{h}_w[k]x_k[n]$$

where $\tilde{x}_k[n]$ is the output signal of k cascaded all pass filter

- In the spectral Domain can be rewritten as

$$\widehat{S}_w(e^{j\tilde{\lambda}}) = \widetilde{H}_w(e^{j\tilde{\lambda}})X(e^{j\tilde{\lambda}})$$

Let $\tilde{\hat{S}}_w(e^{j\tilde{\lambda}})$ be the spectrum of the bilinear transformed signal of $\hat{s}_w[n]$

Formulation on the Linear Frequency scale

- Since $\hat{S}_w(e^{j\tilde{\lambda}}) = \tilde{\hat{S}}_w(e^{j\tilde{\lambda}})$ from the definition of the frequency warped signal as

$$\widetilde{X}(\tilde{z}) = \sum_{n=0}^{\infty} \tilde{x}[n]\tilde{z}^{-n} = X(z) = \sum_{n=0}^{N-1} \tilde{x}[n]z^{-n}$$

- We have following equation $\tilde{\hat{S}}_w(e^{j\tilde{\lambda}}) = \widetilde{H}_w(e^{j\tilde{\lambda}})\widetilde{X}(e^{j\tilde{\lambda}})$
- This equation shows that $\widetilde{H_w}(e^{j\tilde{\lambda}})$ is a linear filter to estimate the spectrum $\tilde{\hat{S}}_w(e^{j\tilde{\lambda}})$ from the input spectrum $\widetilde{X}_w(e^{j\tilde{\lambda}})$ on the warped frequency domain.
- Now the sum of square error is given by

$$\xi\{\tilde{h}\} = \sum_{n=0}^{\infty} \tilde{e}[n]^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\widetilde{S}(e^{j\tilde{\lambda}}) - \widetilde{H}_w(e^{j\tilde{\lambda}})\widetilde{X}(e^{j\tilde{\lambda}})\right| \cdot \left|\widetilde{W}(e^{j\tilde{\lambda}})\right|^2 d\tilde{\lambda}$$

Formulation on the Linear Frequency scale

- The minimization of equation $\xi\{\tilde{h}\}$ with respect to $\{\tilde{h}_w[k]\}$ give the following normal equations

$$\sum_{n=0}^{p-1} \tilde{\phi}_{xx}(m,k)\,\tilde{h}_w[k] = \tilde{\phi}_{xx}(0,m) \qquad (m = 0,\ldots p-1)$$

Where $\tilde{\phi}_{xx}(m,k) = \sum_{n=0}^{\infty} x_m[n]\,x_k[n]$

and $\tilde{\phi}_{sx}(m,k) = \sum_{n=0}^{\infty} s_m[n]\,x_k[n]$

- In the warped frequency domain can be rewritten as

$$\tilde{\phi}_{xx}(m,k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\widetilde{X}(e^{j\tilde{\lambda}})\widetilde{W}(e^{j\tilde{\lambda}})\right|^2 \cdot e^{j(m-k)\tilde{\lambda}}\,d\tilde{\lambda}$$

$$\tilde{\phi}_{sx}(m,k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \widetilde{S}(e^{j\tilde{\lambda}})\,\widetilde{X}^{*}(e^{j\tilde{\lambda}})\left|\widetilde{W}(e^{j\tilde{\lambda}})\right|^2 \cdot e^{j(m-k)\tilde{\lambda}}\,d\tilde{\lambda}$$

Formulation on the Linear Frequency scale

- Therefore, $\tilde{\phi}_{xx}(m,k)$ is the autocorrelation function of signal $\tilde{x}_w[n]$ whose Fourier transform is equal to the frequency warped and weighted spectrum $\widetilde{X}(e^{j\tilde{\lambda}})\widetilde{W}(e^{j\tilde{\lambda}})$
- Similarly, $\tilde{\phi}_{sx}(m,k)$ is the crosscorrelation function between $\tilde{x}_{w,m}[n]$ and $\tilde{s}_{w,k}[n]$ whose whose Fourier transform is $\widetilde{S}(e^{j\tilde{\lambda}})\widetilde{W}(e^{j\tilde{\lambda}})$
- We call $\tilde{\phi}_{xx}(m,k)$ and $\tilde{\phi}_{sx}(m,k)$ as the generalized autocorrelation and crosscorrelation function

Formulation on the Linear Frequency scale

- It should be noted that each of the $\tilde{\phi}_{xx}(m,k)$ and $\tilde{\phi}_{sx}(m,k)$ is a function of difference (k-m). Thus, both function are calculated from the sum of finite term as

$$\tilde{\phi}_{xx}(m,k) = \tilde{r}_{xx}[k-m] = \sum_{n=0}^{N-1} x_m[n]\, x_{|k-m|}[n]$$

$$\tilde{\phi}_{sx}(0,k) = \tilde{r}_{sx}[k] = \sum_{n=0}^{N-1} s[n]\, x_{|k-m|}[n]$$

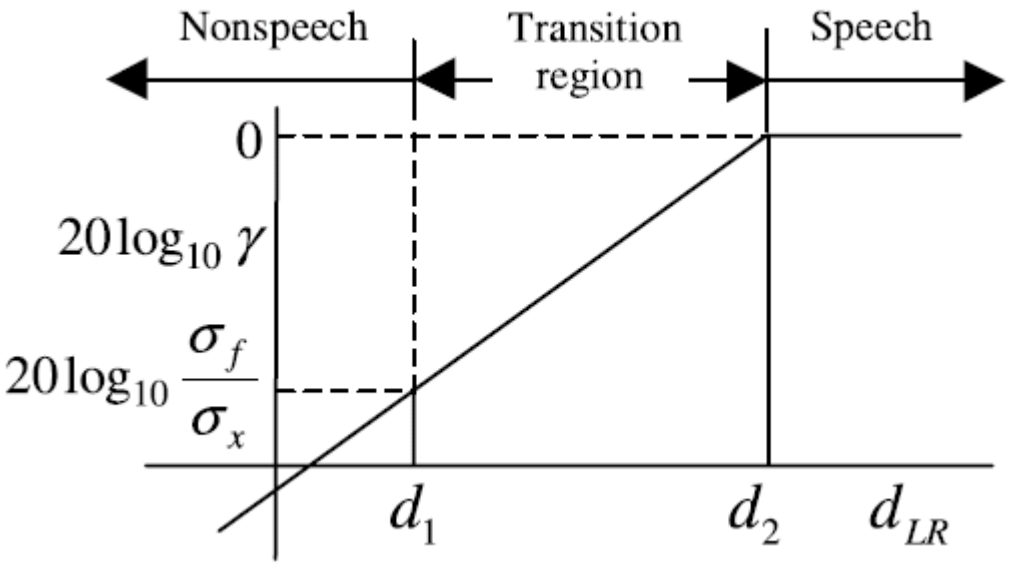
- In practical situation, since both the speech and noise are unobservable, the exact estimation of crosscorrelation function $\tilde{r}_{sx}[m]$ between clean and noisy speech is not possible, rather the approximated value is estimated.

Approximation of Crosscorrelation Function

- For the approximation of $\tilde{r}_{sx}[m]$, input speech frames are classified as non-speech, transition and speech segments depending on the value of likelihood ratio d_{LR} of each frames.
- The generalize crosscorrelation function between the clean and noisy speech is given by

$$\tilde{r}_{sx}[m] = \tilde{r}_{xx}[m] - \tilde{r}_{nn}[m] - \tilde{r}_{ns}[m]$$

since $\tilde{r}_{ns}[m]$ is negligible for the noise uncorrelatged with the speech signal and $\tilde{r}_{nn}[m]$ is approximated by $\hat{\tilde{r}}_{nn}[m]$


Nonspeech
Transition region
Speech
0
$20\log_{10}\gamma$
$20\log_{10}\frac{\sigma_f}{\sigma_x}$
d_1
d_2
d_{LR}

Approximation of Crosscorrelation Function

- $\tilde{r}_{sx}[m]$ for the segment with $d_{LR} \geq d_2$ is approximated as

$$\tilde{r}_{sx}[m] \approx \tilde{r}_{xx}[m] - v \cdot \hat{\tilde{r}}_{nn}[m]$$

where v is a scaling factor, given by

$$v = \begin{cases} 1; & if \quad \tilde{r}_{xx}[0] > \hat{\tilde{r}}_{nn}[0] \\ 0.9\tilde{r}_{xx}[m] / \hat{\tilde{r}}_{nn}[0]; & if \quad \tilde{r}_{xx}[0] \leq \hat{\tilde{r}}_{nn}[0] \end{cases}$$

The rower part is used as flooring to prevent $\tilde{r}_{sx}[0]$, i.e., the power to be negative.

Approximation of Crosscorrelation Function

- The noise segment with the lower value of likelihood ratio d_{LR} is means the noise is dominating and speech is absent.
- Hence a random sequence is introduced as floored signal f[n] instead of s[n] for the noise segment with $d_{LR} < d_2$
- The rms value of floored signal σ_f is set to -30dB from the maximum rms value of the input signal.
- Then the generalized crosscorrelation function $\tilde{r}_{fx}[m]$ between f[n] and x[n] is calculated.

Approximation of Crosscorrelation Function

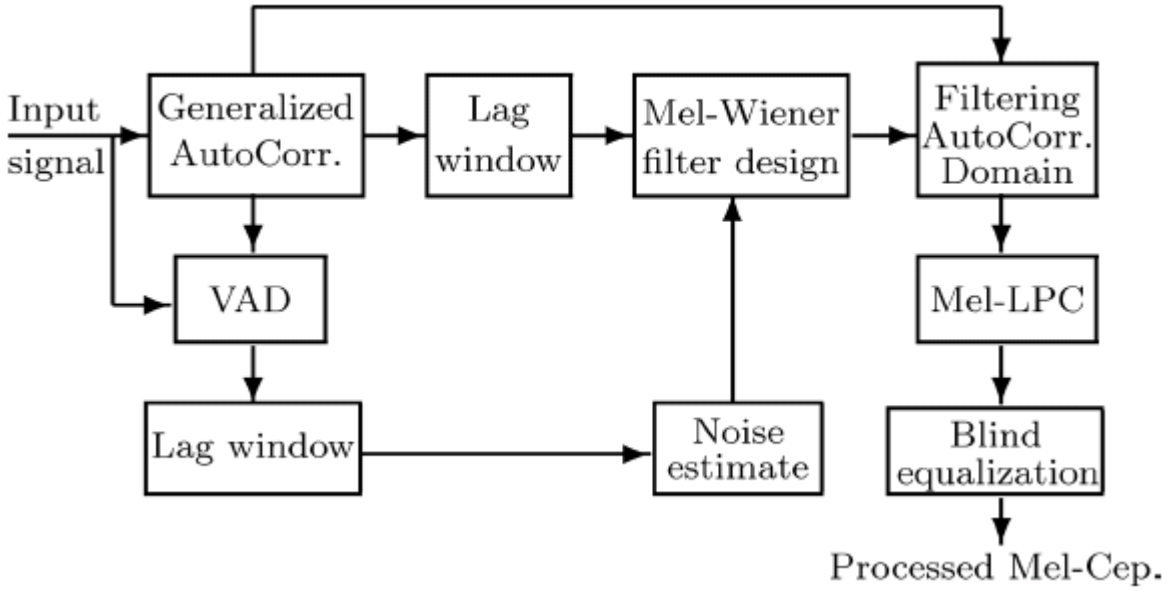
- The crosscorrelation function $\tilde{r}_{sx}[m]$ is compromised with the floored crosscorrelation function $\tilde{r}_{fx}[m]$ as

$$\tilde{r}_{sx}[m] \approx \gamma(\tilde{r}_{xx}[m] - v \cdot \tilde{r}_{nn}[m]) - \tilde{r}_{fx}[m]$$

Where $\gamma = (\sigma_f / \sigma_x)^{(d_{LR} - d_2)/(d_1 - d_2)}$

- d_1 and d_2 are experimentally tunable constants with values d_1=0.6 and d_2=0.15 for clean training and d_2=0.13 for multi-condition training.
- The parameter γ prevents the abrupt transition between non-speech and speech segments, consequently, it will prevent the sharp clipping of the parameters, which might degrades the performance of the systems.

System overview



Voice Activity Detection

- The voice activity detector (VAD) is based on the likelihood ratio measure between autoregressive(AR) of noise and input speech signal.
- From initial 20 frames a noise model is created, i.e., the model is assumed to be Mth order autoregressive with coefficients $\tilde{b}^t = [\tilde{b}_0 \tilde{b}_1 ... \tilde{b}_M]$ where $\tilde{b}_0$ =1. For each input frame, $\tilde{r}_{xx}[m]$ is calculated to estimate likelihood ratio as follows

$$d_{LR} = R_{\tilde{b}}[0]\tilde{r}_{xx}[0] + 2\sum_{i=1}^{M} R_{\tilde{b}}\tilde{r}_{xx}[i] - 1$$

where $R_{\tilde{b}}[i]$ is the autocorrelation function of AR coefficients.

- d_{LR} is compared with a threshold value η is empirical value of 0.11 for clean training and 0.1 for multicondition training

Noise model Estimate

- In the proposed filtering technique, a lag window is applied to the noise autocorrelation function to smooth the fine spectra of high order autocorrelation coefficients
- When a frame t is detected as noise, the estimated generalized autocorrelation function of noise $\hat{\hat{r}}_{nn}[m,t]$ is updated by accumulating the lag windowed $\hat{\hat{r}}_{xx}[m,t]$ as follows

$$\hat{\hat{r}}_{nn}[m,t]=\begin{cases}\beta\hat{\hat{r}}_{nn}[m,t_p]+(1-\beta)\hat{\hat{r}}_{xx}[m,t]; & if \quad \text{frame t is silence}\\ \hat{\hat{r}}_{nn}[m,t_p]; & if \quad \text{frame t is speech}\end{cases}$$

where t_p is the previous noise frame and β is the forgetting factor with value of 0.96.

Experimental Setup

- The proposed system was evaluated on Aurora 2 database, which is a subset of TI digits database contaminated by additive noises and channel effect
- The speech signal was windowed using Hamming window of length 20ms with 10ms frame period.
- The frequency warping factor was set to 0.4.
- As front-end, 14 cepstral coefficients and their delta confidents including 0^{th} terms were used.
- For VAD, the order of AR was set to 10 and the window length was 40 ms with same frame period as in Mel-LPC

Table 2 Word accuracy w/o Wiener filter for clean training.

Set A								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	99.02	95.52	86.98	68.22	43.48	23.00	11.58	63.44
Babb.	99.09	84.70	62.58	35.19	10.94	−1.54	−1.45	38.38
Car	98.84	94.66	79.45	51.27	25.59	13.06	9.60	52.81
Exhi.	99.11	95.87	86.79	66.95	34.93	16.11	9.60	60.13
Aver.	99.02	92.69	78.95	55.41	28.74	12.66	7.34	53.69
Set B								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Rest.	99.02	87.90	69.30	43.23	14.28	−5.13	−7.58	41.92
Stre.	99.09	94.11	84.28	59.58	31.05	12.70	6.14	56.35
Airp.	98.84	89.35	70.56	45.33	17.57	1.64	−0.51	44.89
T-St.	99.11	90.16	73.65	46.34	17.68	4.91	5.68	46.55
Aver.	99.02	90.38	74.45	48.62	20.15	3.53	0.94	47.43
Set C (MIRS)								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	98.93	89.41	78.78	62.88	42.49	21.80	11.36	59.08
Stre.	98.73	92.44	83.98	67.32	44.20	23.55	12.48	62.30
Aver.	98.83	90.93	81.38	65.10	43.35	22.68	11.92	60.69

Table 3 Word accuracy with proposed filter for clean training.

Set A								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	99.17	97.82	96.96	93.18	86.03	64.54	28.46	87.71
Babb.	98.64	97.67	96.28	92.41	82.59	52.15	17.29	84.22
Car	98.90	98.42	97.97	95.74	90.10	69.16	27.50	90.28
Exhi.	99.04	97.66	96.17	92.50	80.22	57.64	29.07	84.84
Aver.	98.94	97.90	96.85	93.46	84.74	60.88	25.58	86.77
Set B								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Rest.	99.17	97.88	95.21	90.91	77.62	49.19	15.87	82.17
Stre.	98.64	97.70	96.83	93.74	84.43	59.76	29.69	86.50
Airp.	98.90	98.15	97.17	94.75	86.01	63.47	26.78	87.91
T-St.	99.04	97.99	96.82	93.98	84.33	64.24	28.05	87.48
Aver.	98.94	97.93	96.51	93.35	83.10	59.17	25.10	86.02
Set C (MIRS)								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	99.02	97.97	96.62	92.88	83.33	57.63	23.61	85.69
Stre.	98.73	97.07	96.70	93.26	80.74	55.99	26.21	84.76
Aver.	98.88	97.52	96.66	93.07	82.04	56.81	24.91	85.22

Table 4 Recognition accuracy w/o Wiener filter for multi-condition training.

Set A								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	98.86	98.25	97.36	95.86	90.67	69.45	30.76	90.32
Babb.	98.76	97.82	96.86	94.38	85.73	59.16	23.67	86.79
Car	98.66	97.97	97.49	95.79	89.56	61.41	20.91	88.45
Exhi.	98.92	97.96	96.91	95.06	89.29	67.63	28.05	89.37
Aver.	98.80	98.00	97.16	95.28	88.82	64.42	25.85	88.74
Set B								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Rest.	98.86	97.64	96.50	92.08	82.25	57.51	23.64	85.20
Stre.	98.76	97.58	96.49	94.23	85.82	64.42	27.30	87.71
Airp.	98.66	98.12	97.61	94.72	88.16	67.88	31.26	89.30
T-St.	98.92	97.81	96.24	92.97	84.23	59.89	21.01	86.23
Aver.	98.80	97.79	96.71	93.50	85.12	62.43	25.81	87.11
Set C (MIRS)								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	98.99	97.76	96.38	93.15	83.79	52.01	18.70	84.62
Stre.	98.40	96.46	96.49	93.41	85.22	60.07	27.36	86.33
Aver.	98.70	97.11	96.44	93.28	84.51	56.04	23.03	85.48

Table 5 Recognition accuracy with Wiener filter for multi-condition training.

Set A								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	98.89	98.00	97.30	95.49	91.10	76.11	40.71	91.60
Babb.	98.55	97.79	97.07	95.86	90.36	68.11	28.08	89.84
Car	98.63	97.73	97.58	96.60	92.57	76.68	36.24	92.24
Exhi.	98.77	97.87	97.28	95.06	88.34	73.22	42.27	90.36
Aver.	98.71	97.85	97.31	95.76	90.60	73.53	36.83	91.01
Set B								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Rest.	98.89	98.00	97.18	94.87	87.04	65.31	25.58	88.48
Stre.	98.55	97.97	97.07	95.28	88.97	70.86	38.21	90.03
Airp.	98.63	98.06	96.96	95.71	90.64	74.02	35.49	91.08
T-St.	98.77	98.03	97.25	95.53	89.69	72.01	35.98	90.51
Aver.	98.71	98.02	97.12	95.35	89.09	70.55	33.82	90.03
Set C (MIRS)								
Noise	cln	20 dB	15 dB	10 dB	5 dB	0 dB	−5 dB	Aver.
Subw.	98.65	97.88	97.27	95.21	89.38	69.33	34.60	89.82
Stre.	98.31	96.89	96.80	94.95	88.24	66.32	34.19	88.64
Aver.	98.48	97.39	97.04	95.08	88.81	67.83	34.40	89.23

Table 6 Comparative result for proposed system and ETSI AFE.

	Mode	Set A	Set B	Set C	Overall
Proposed	**Multi**	**91.01**	**90.03**	**89.23**	**90.26**
	Clean	**86.77**	**86.02**	**85.22**	**86.16**
	Aver.	**88.89**	**88.03**	**87.23**	**88.21**
ETSI AFE	Multi	92.20	91.54	89.21	91.34
	Clean	87.18	86.29	83.25	86.04
	Aver.	89.69	88.92	86.23	88.69

Table 7 Aurora 2 relative improvement.

Training Mode	Set A	Set B	Set C	Overall
Multi	12.72%	20.30%	18.38%	16.89%
Clean	66.50%	72.67%	61.59%	67.98%
Average	39.61%	46.49%	39.98%	42.43%

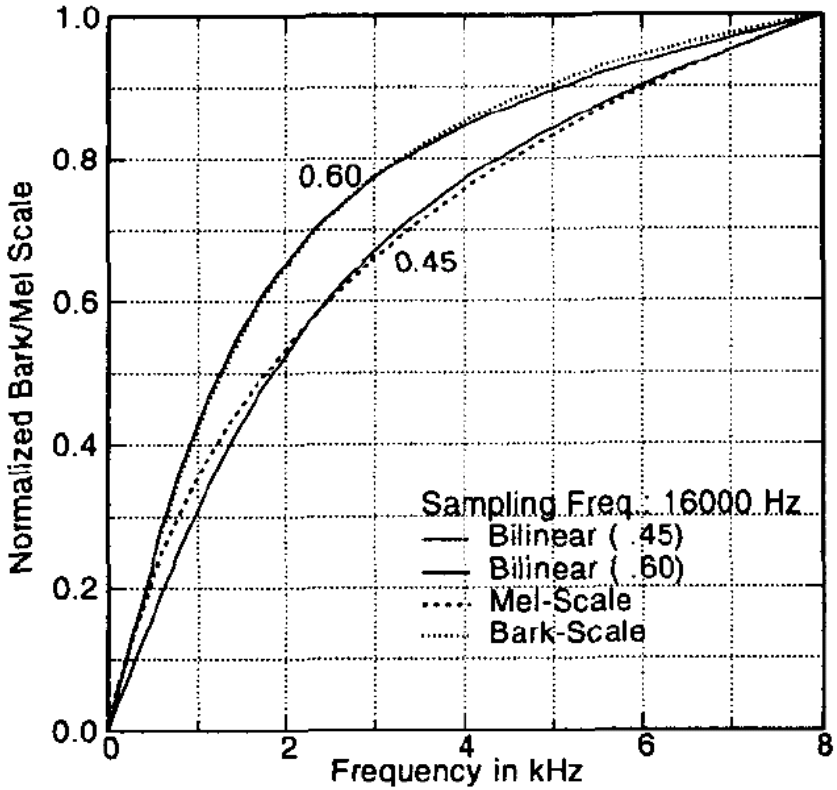


Fig. 1. The frequency mapping function in the bilinear transformation.

Autoregressive model

- In statistics and signal processing, an **autoregressive** (**AR**) **model** is a type of random process which is often used to model and predict various types of natural phenomena.

$$x(n) = \sum_{k=1}^{P} a_k x(n-k) + \Theta_0 e(n)$$

Filtering in Autocorrelation Domain

- Filtering is done in the autocorrelation Domain to estimate the generalized autocorrelation function of the filtered speech $\hat{s}_w[n]$ as follows

$$\tilde{r}_{ss}[m] = \sum_{n=0}^{\infty} \hat{s}_m[n]\hat{s}_{w,m}[n] = \sum_{n=0}^{\infty} r_{hh}[k]\tilde{r}_{xx}[m-k]$$

where $\tilde{r}_{xx}[m]$ is the generalized autocorrelation function of the noisy speech, and $\tilde{r}_{\tilde{h}\tilde{h}}[m]$ is the autocorrelation function of $\hat{h}_w[m]$.

Blind Equalization

- Blind equalization is applied on the cepstral coefficients In order to minimize the channel effects.
- This technique is based on the least mean square algorithm, which minimizes the mean square error computed as a difference between the current and reference cepstrum.

$$wt = Min(1, Max(0, lnE - 4.75))$$

$$step = 0.008\, wt$$

$$c_{eq}[i] = c[i] - bias[i], \qquad 0 \le i \le 13$$

$$bias[i] = bias[i] + step \cdot (c_{eq}[i] - c^{Ref}[i]), \qquad 0 \le i \le 13$$

Where wt is the weighting parameter, lnE indicates the log energy of the current frame, bias[i] is initialized on 0.0(0<=i<=13) and $c^{Ref}[i]$ is the reference cepstrum.

The long –term cepstrum of training clean speech is used as reference cepstrum.

Levinson-Durbin recursion

$$r_x(m)-\sum_{k=1}^{P}a_k r_x \mid m-k \mid =0 \Rightarrow \begin{pmatrix} r(0) & \cdots & r(p-1) \\ \vdots & \ddots & \vdots \\ r(p-1) & \cdots & r(0) \end{pmatrix}\begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}=\begin{pmatrix} r(0) \\ \vdots \\ r(P) \end{pmatrix}$$

$$\begin{pmatrix} r(0) \\ \vdots \\ r(P) \end{pmatrix}-\begin{pmatrix} r(0) & \cdots & r(p-1) \\ \vdots & \ddots & \vdots \\ r(p-1) & \cdots & r(0) \end{pmatrix}\begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}=\vec{0}$$

$$\xi_{\min}{}^{m}=r_x(0)-\sum_{k=1}^{P}a_k{}^{(m)}r_x(k)=[r_x(0)\cdots r_x(m)]\begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}$$

$$\begin{pmatrix} r(0) & \cdots & r(p-1) \\ \vdots & \ddots & \vdots \\ r(p-1) & \cdots & r(0) \end{pmatrix}\begin{pmatrix} 1 \\ -a_1 \\ \vdots \\ -a_p \end{pmatrix}=\begin{pmatrix} \zeta^{m}{}_{\min} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$