远场语音交互技术解析

—— 远场语音信号处理

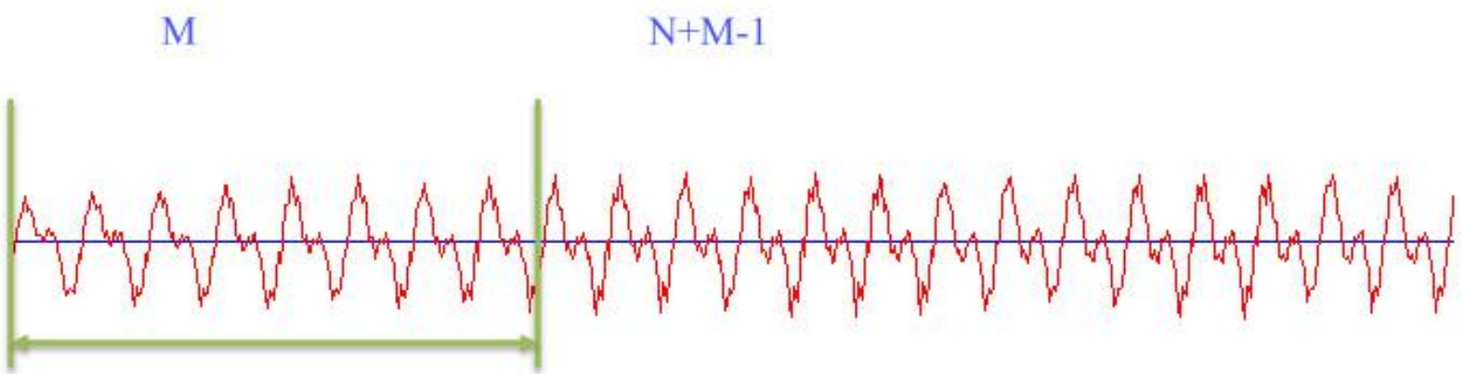
声智科技

讲师：冯大航

慕课学院
www.mooc.ai

语音信号处理基本概念

1. 短时能量分析



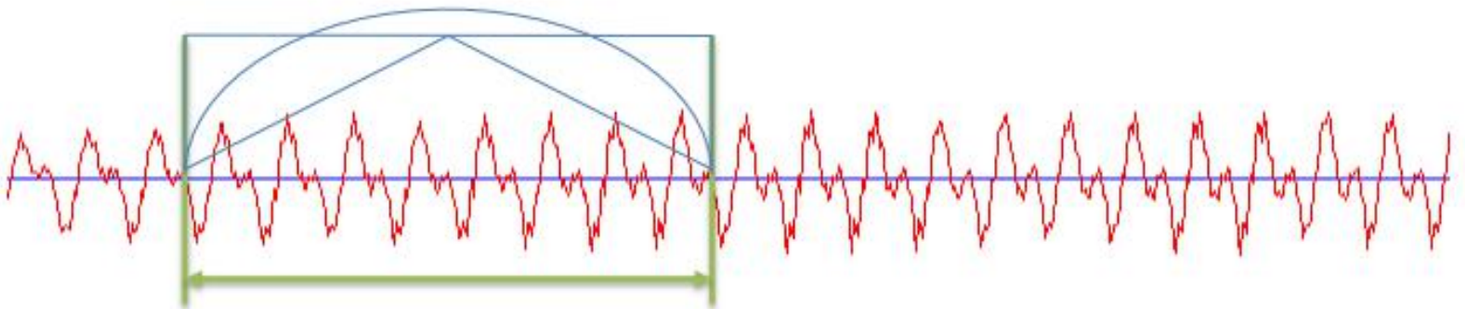
$$E_M = \sum_{m=M}^{N+M-1} [s(m)]^2$$

N称为语音短时分析的帧长 Frame Length

M称为语音短时分析的步长 Step

语音信号处理基本概念

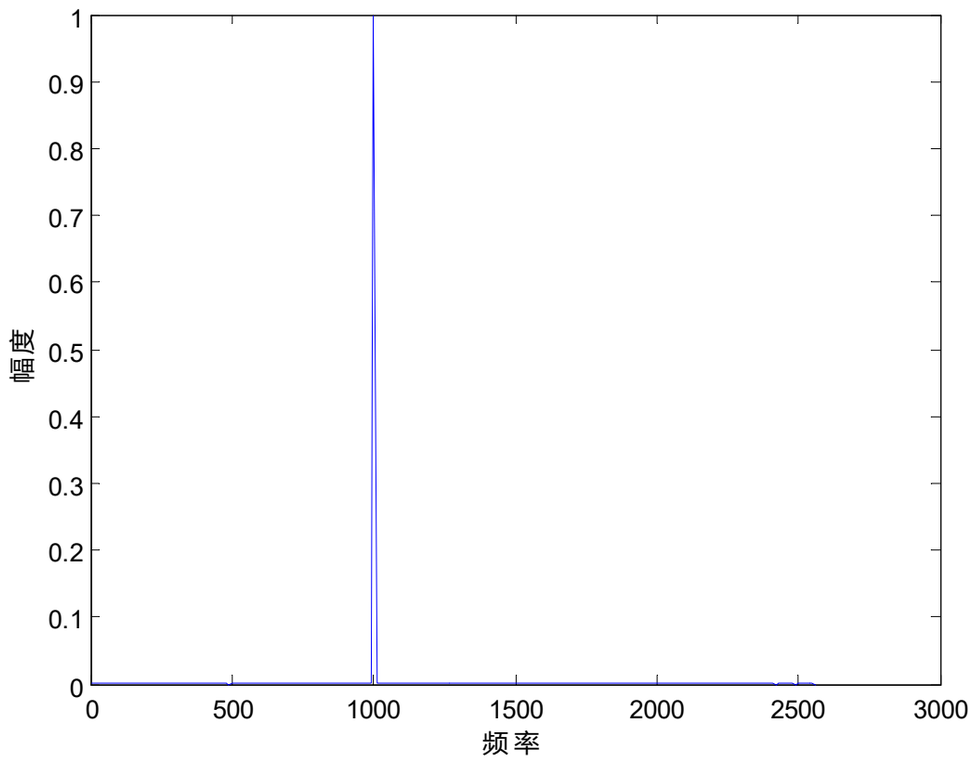
决定短时能量特性有两个条件：不同的窗口的形状和长度。窗长越长，频率分辨率越高，而时间分辨率越低

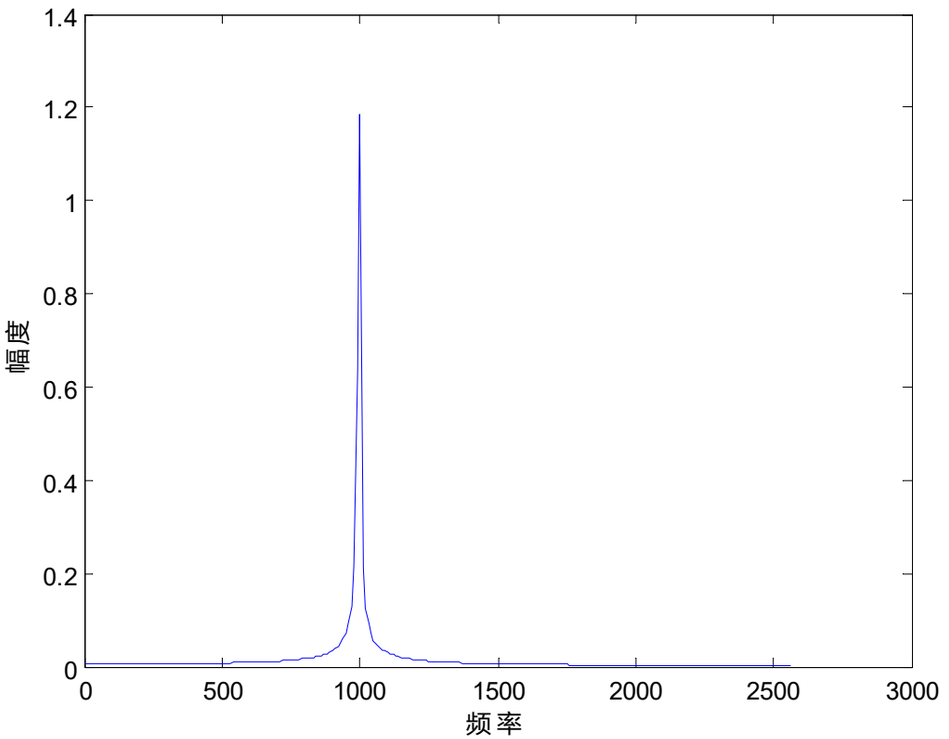


$$E_M = \sum_{m=M}^{N+M-1} [s(m)w(m)]^2$$

时域上相乘等价于频域上做卷积运算，相当于在每个频点上加了一个滤波器

语音信号处理基本概念





$$x = \sin\left(2\pi f_1 t\right)$$

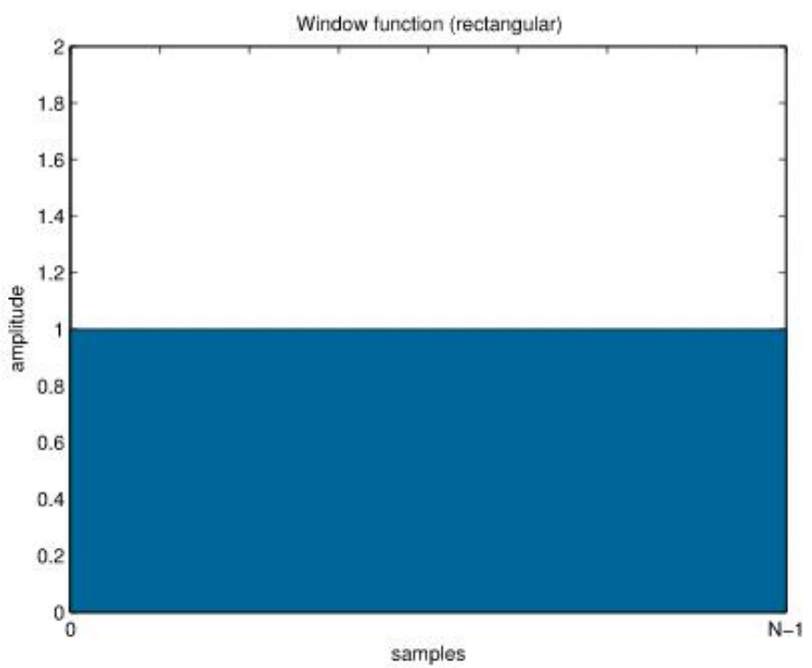
$$x = \sin\left(2\pi f_1 t\right) + \sin\left(2\pi f_2 t\right)$$

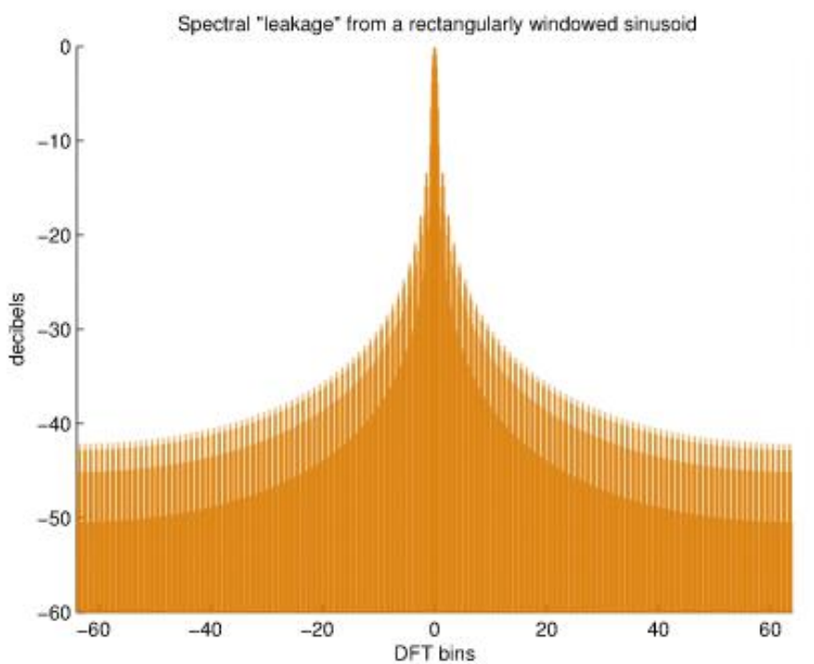
语音信号处理基本概念

典型窗函数

矩形窗：

$$w(n) = \begin{cases} 1 & 0 \le n \le M-1 \\ 0 & \text{其它} \end{cases}$$



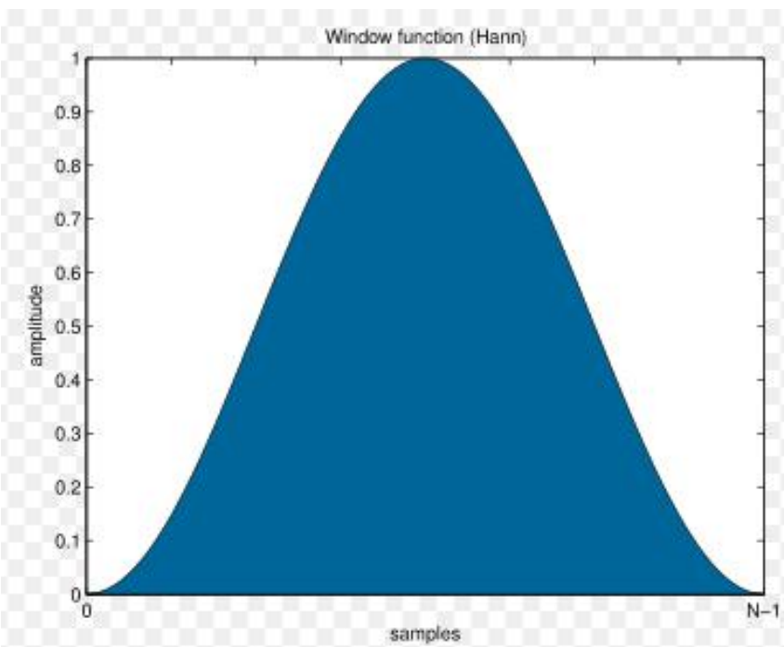


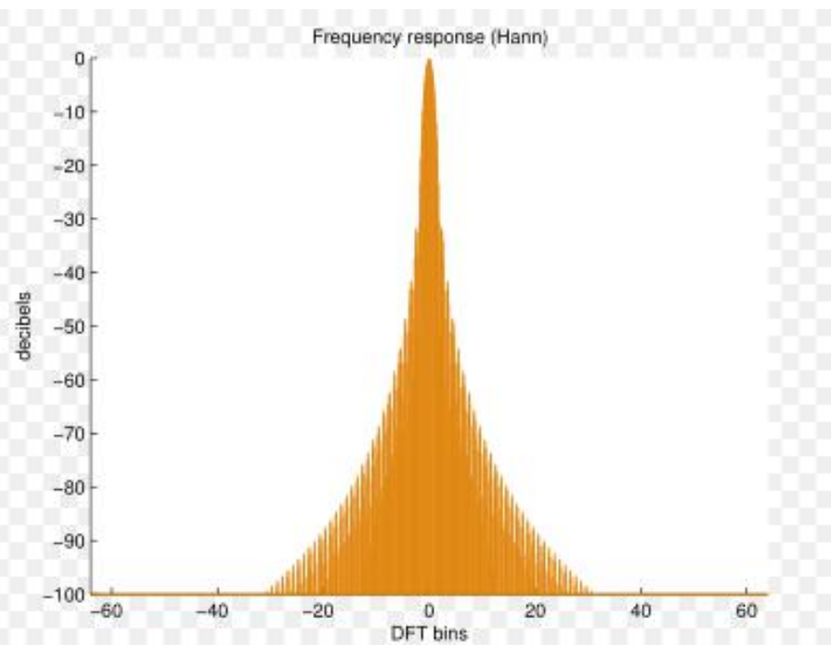
语音信号处理基本概念

典型窗函数

汉宁窗：

$$w(n) = \begin{cases} 0.5 - 0.5\cos(2\pi n / (M-1)) \\ 0 \quad \text{其它} \end{cases}$$



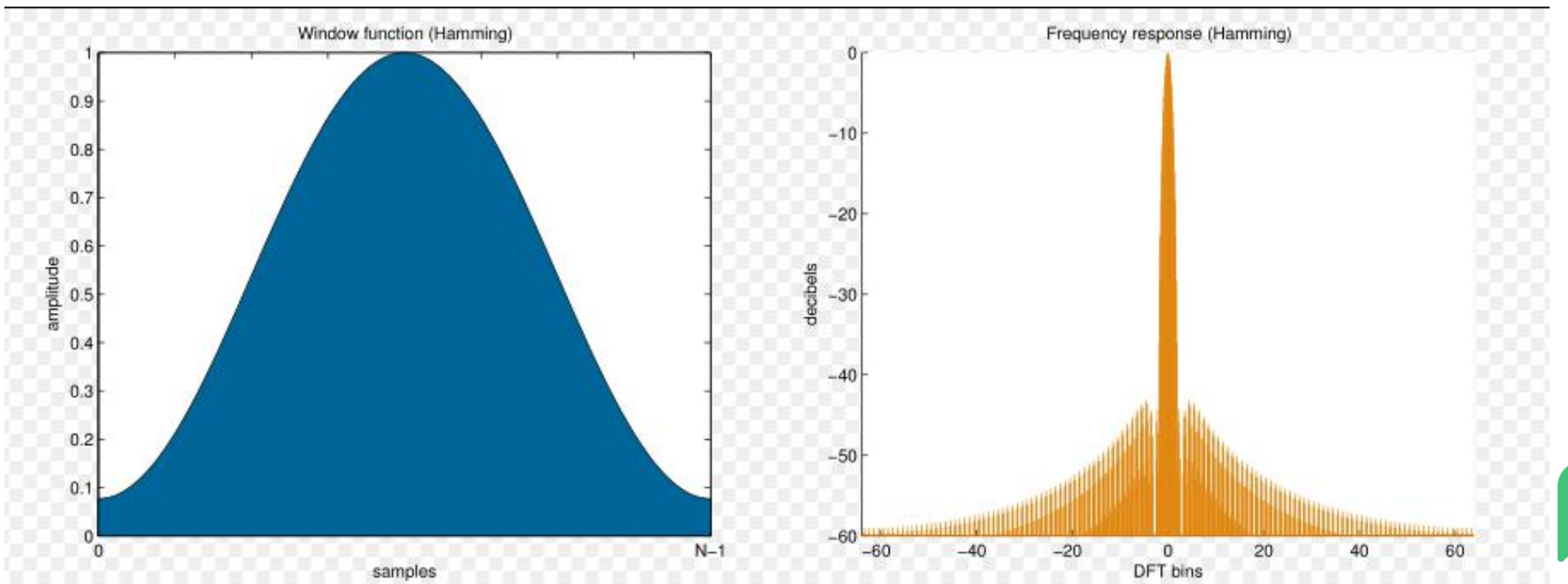


语音信号处理基本概念

典型窗函数

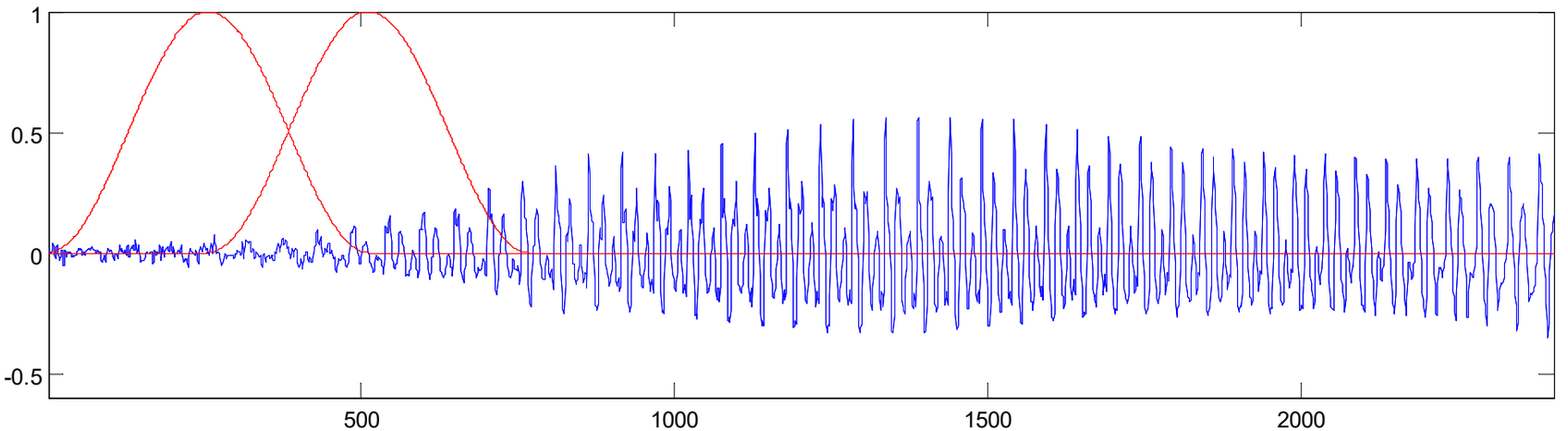
汉明窗

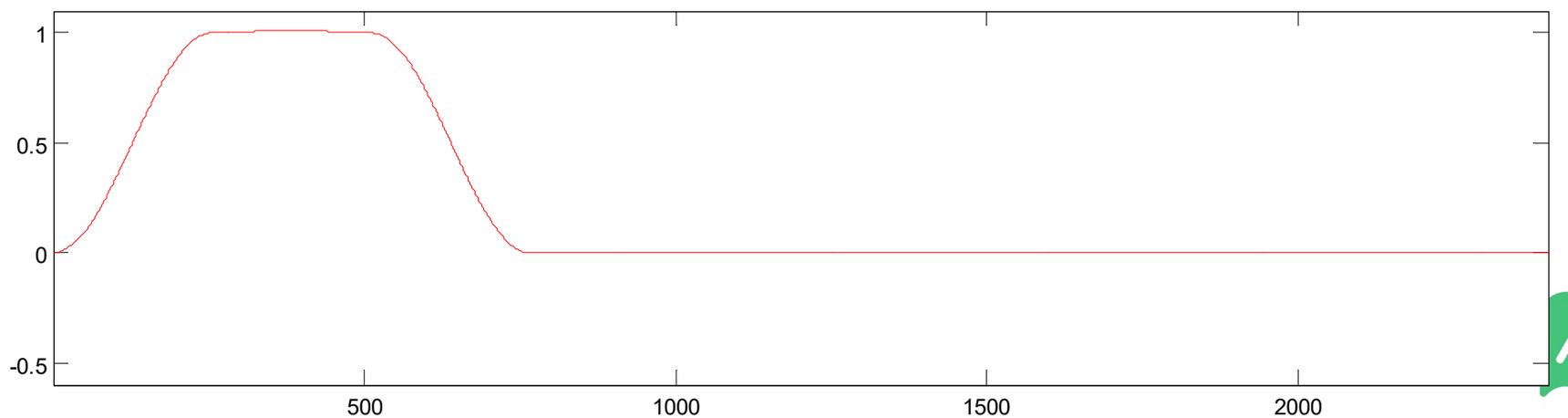
$$w(n) = \begin{cases} 0.53836 - 0.46164 \cos\left(\frac{2\pi n}{N-1}\right) \\ 0 \quad \text{其它} \end{cases}$$



语音信号处理基本概念

Overlap





语音信号处理基本概念

功率谱估计

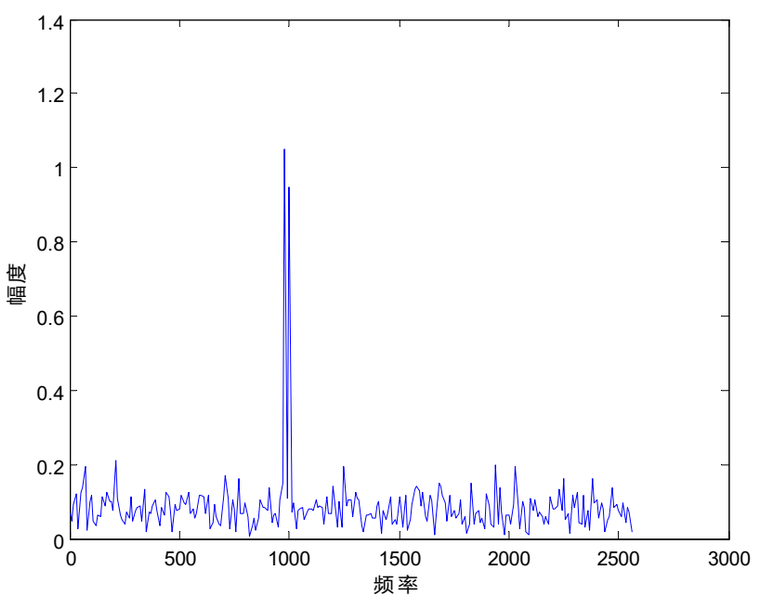
1. 经典功率谱估计

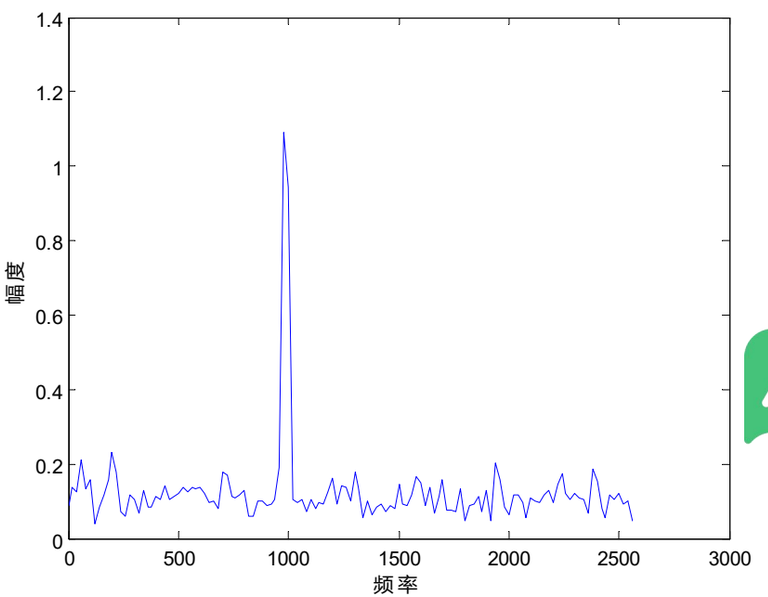
- 周期图法

$$P(k)=\frac{1}{N}\left|X_N(k)\right|^2$$

- 平均周期图法

$$P(k)=\frac{1}{M}\sum_{m=0}^{M}\left|X_N^{(m)}(k)\right|^2$$



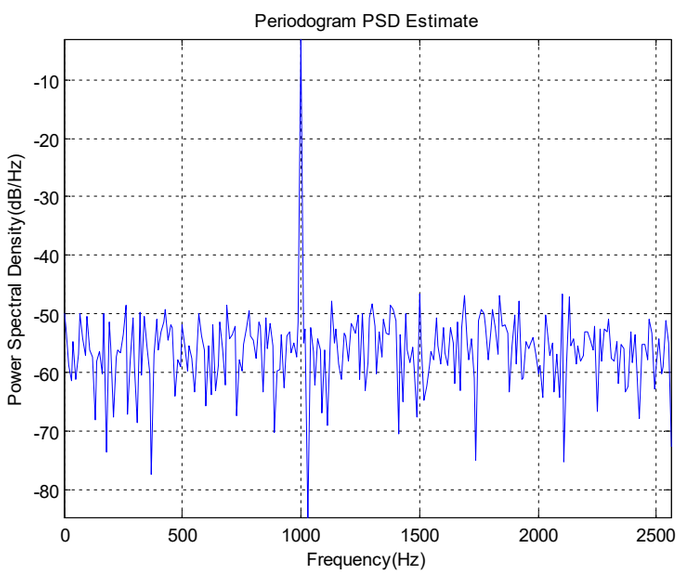


语音信号处理基本概念

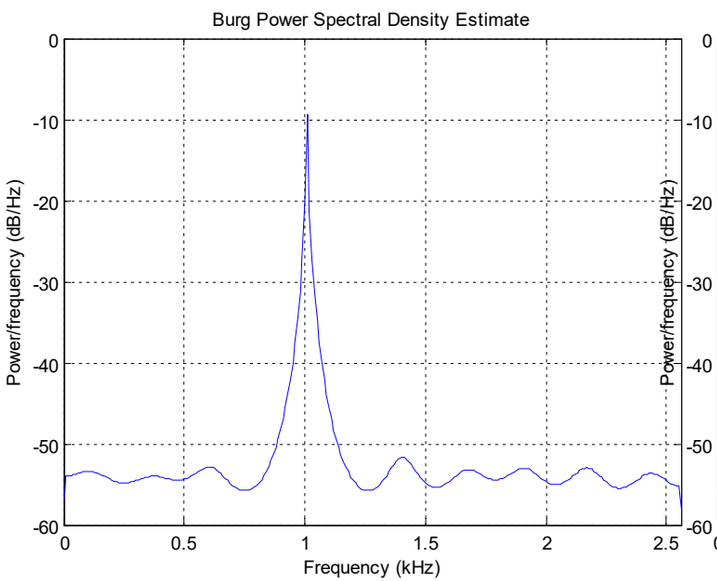
功率谱估计

2. 参数法功率谱估计

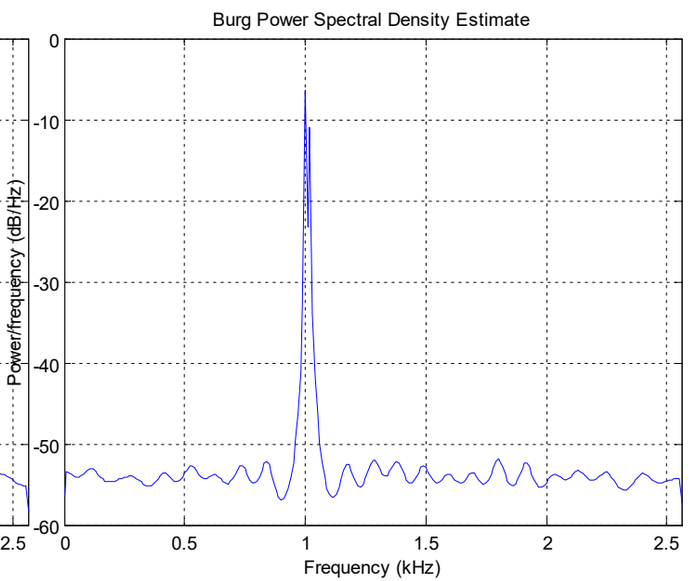
- AR
- ARMA
- 子空间方法



周期图法



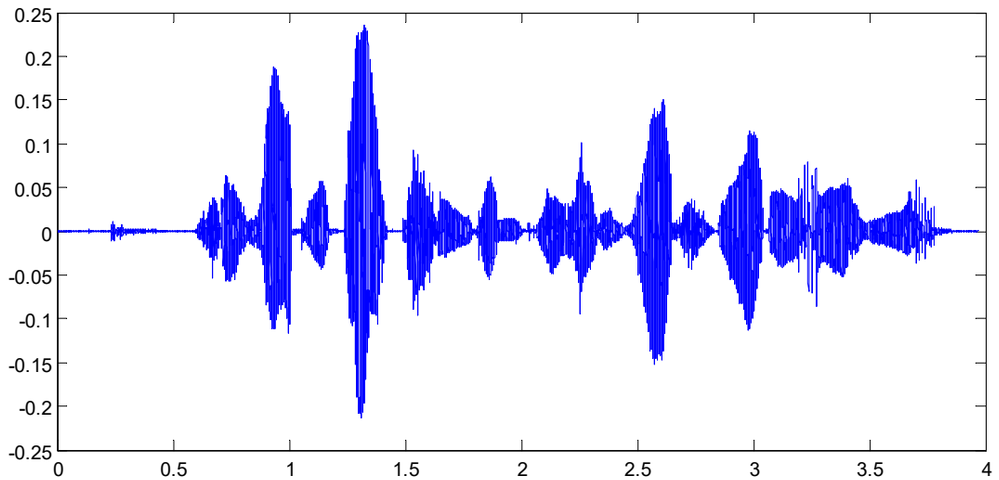
AR谱估计，20阶

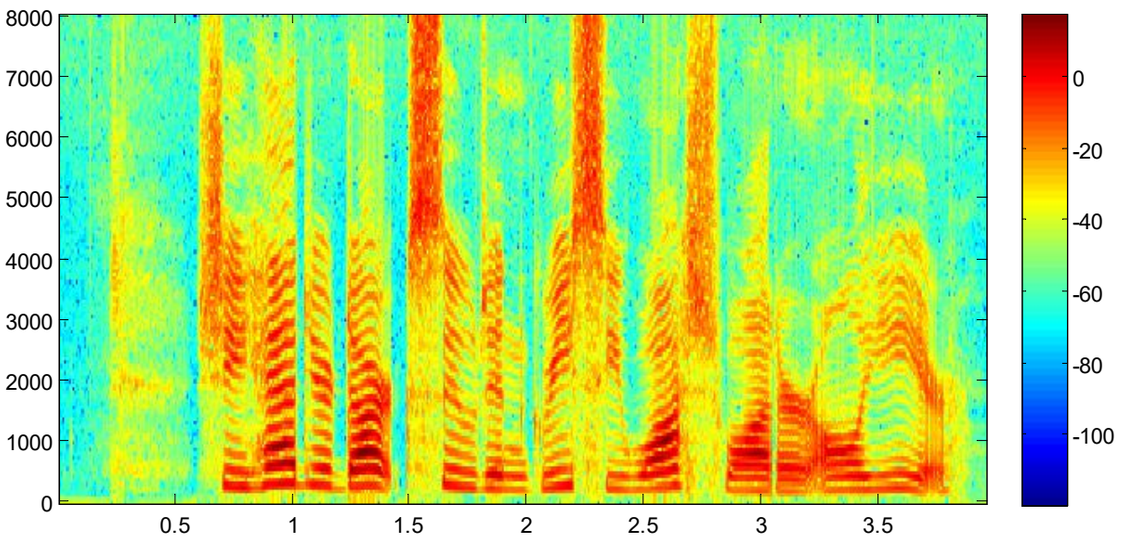


AR谱估计，50阶

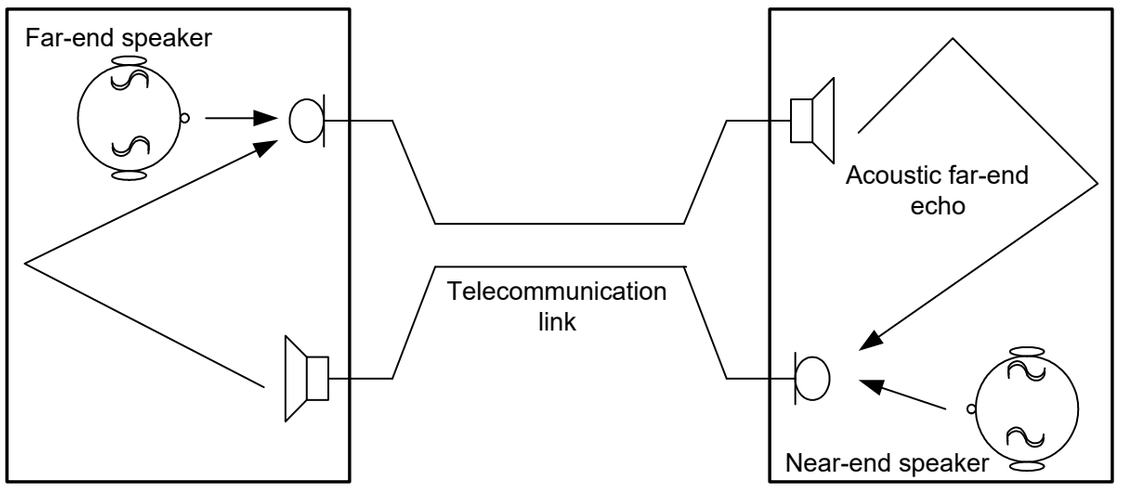
语音信号处理基本概念

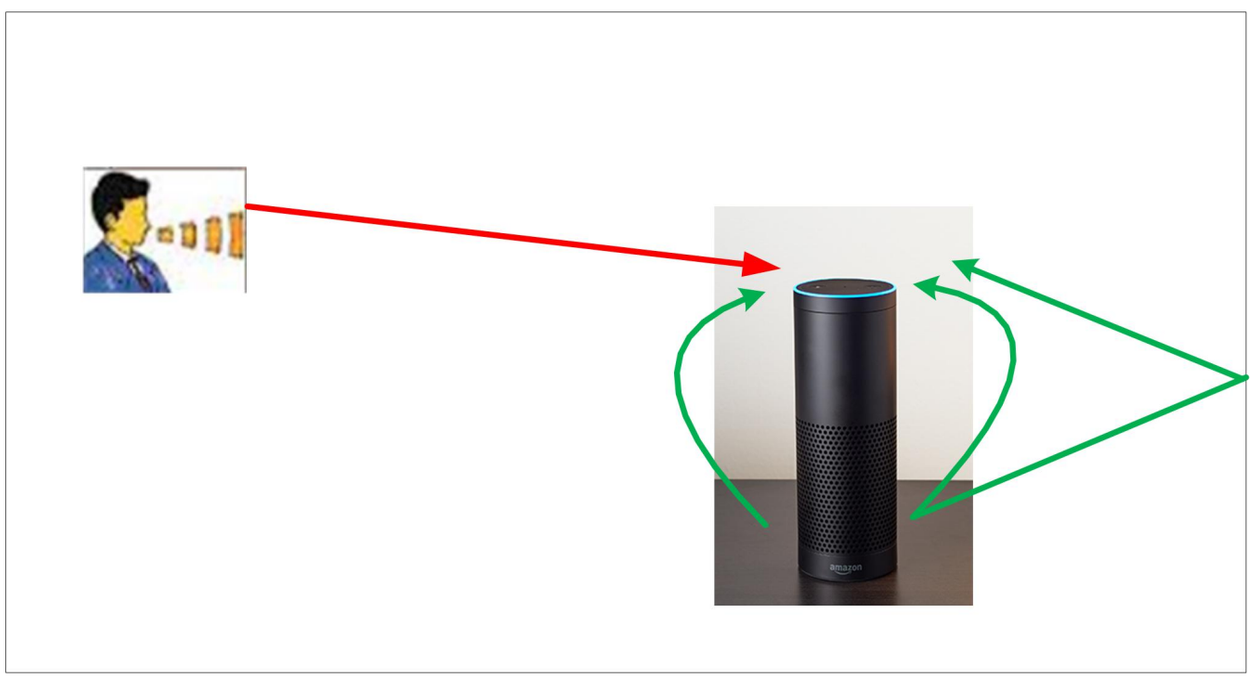
语谱图



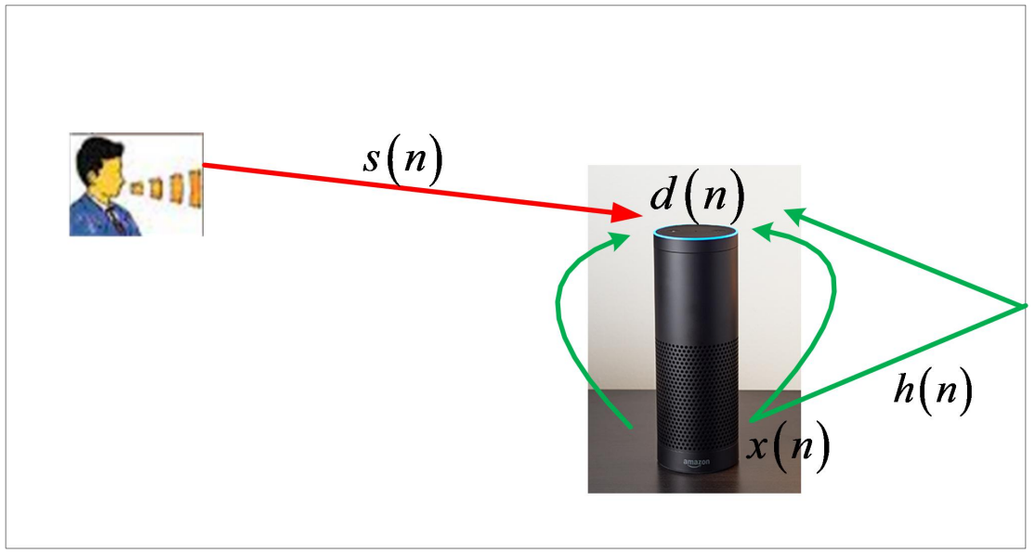


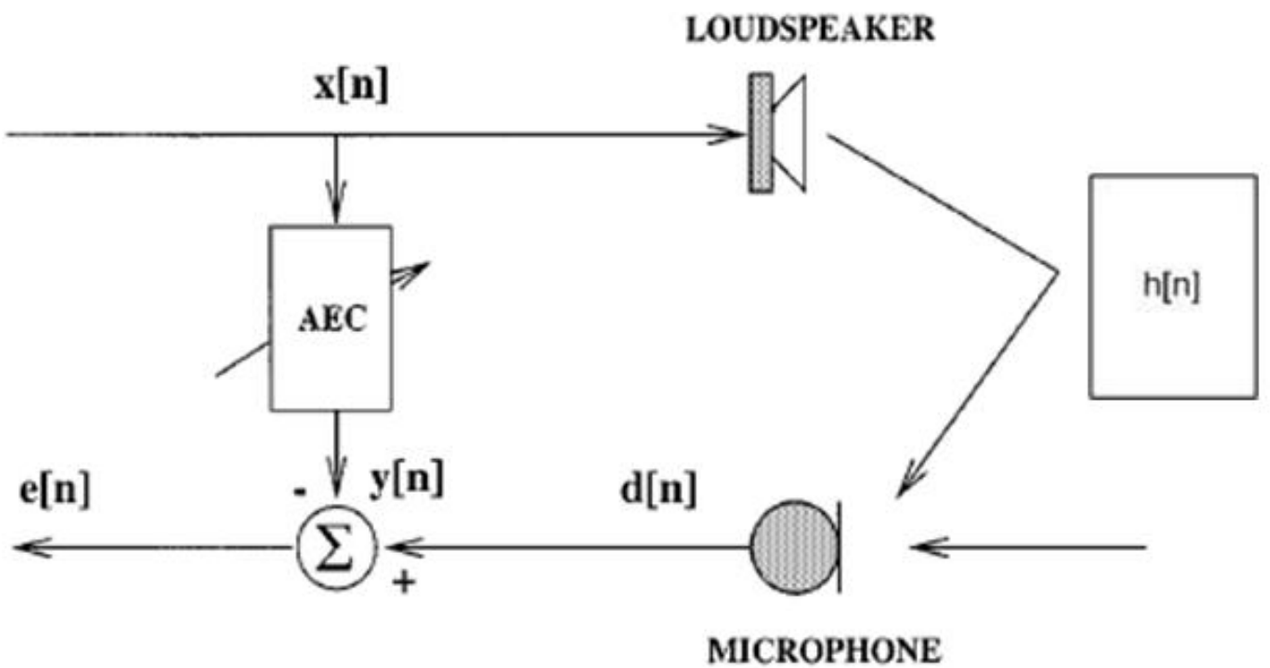
语音信号处理-回声消除





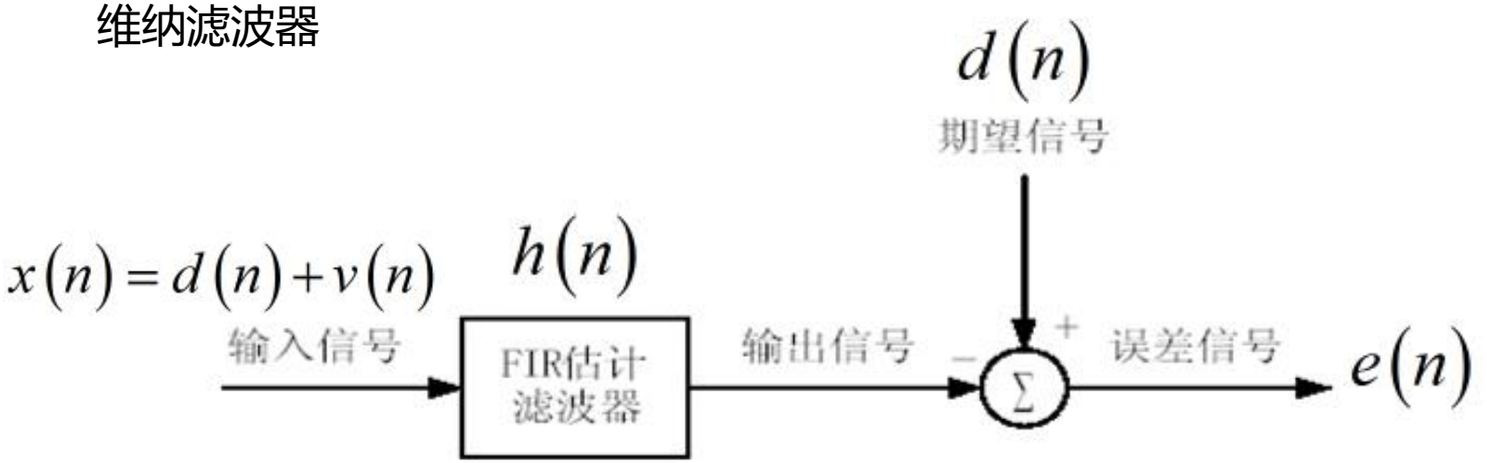
语音信号处理-回声消除

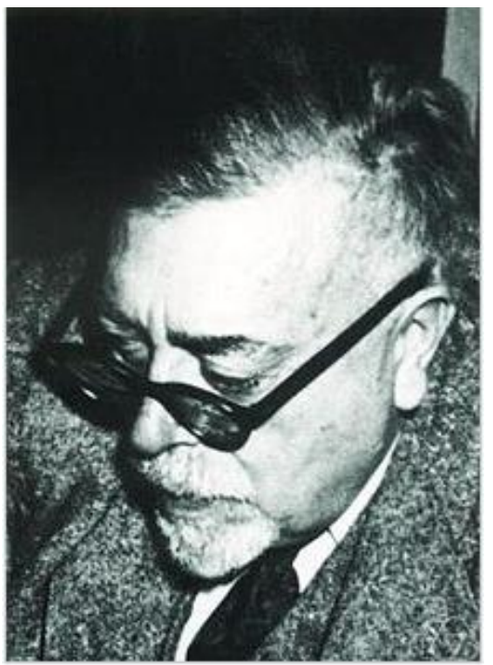




语音信号处理-回声消除

维纳滤波器



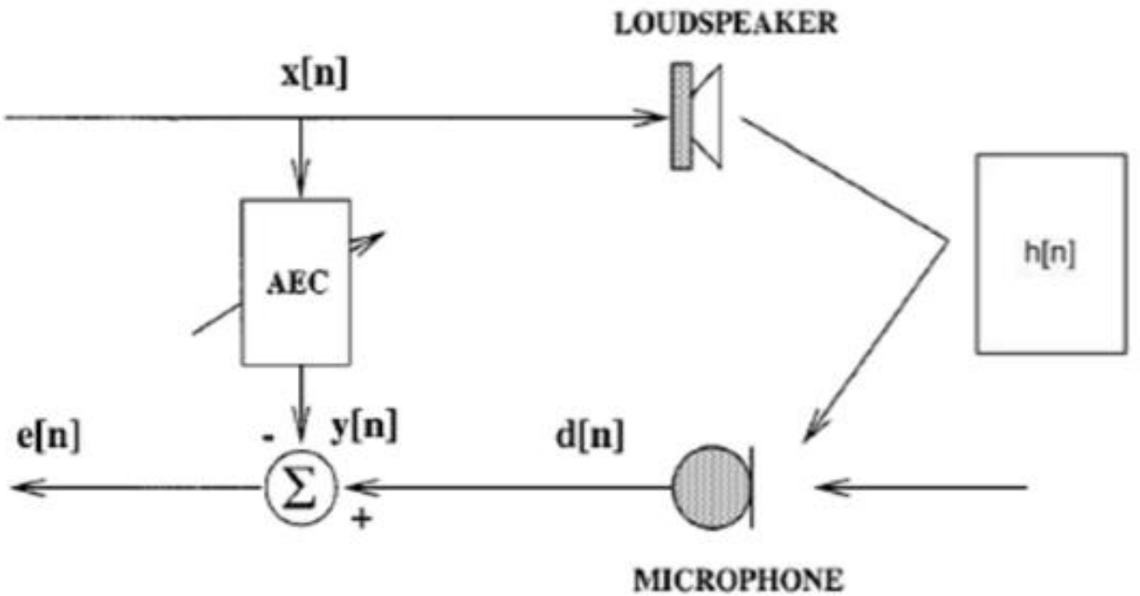


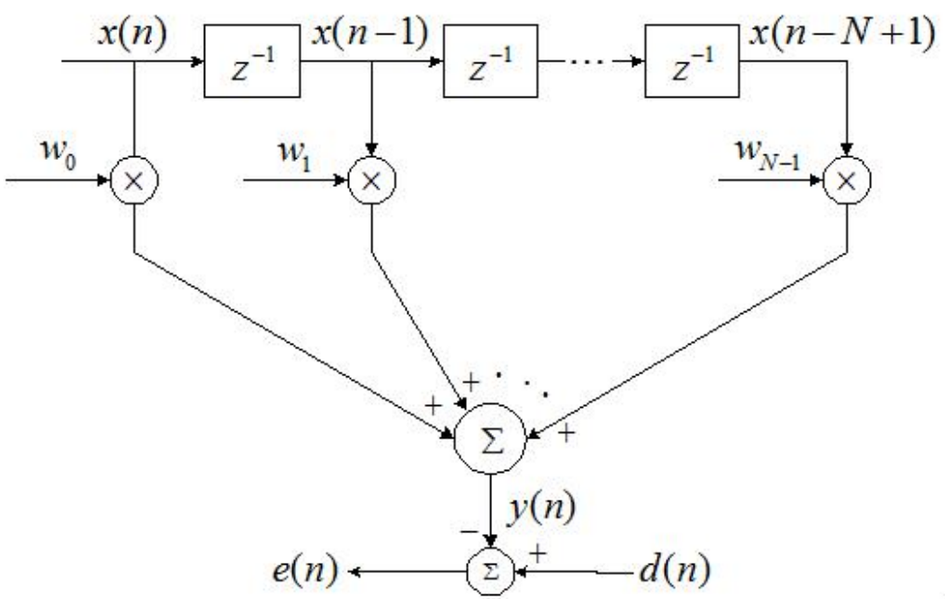
$$e(n)=x(n)*h(n)-d(n)$$

$$E\left[e^2(n)\right]=E\left[\left(x(n)*h(n)-d(n)\right)\text{^}2\right]$$

$$h(n)=\mathbf{R}_{xx}^{-1}\mathbf{r}_{xd}$$

语音信号处理-回声消除





维纳滤波器准则

$$\xi = \mathrm{E}\left[\left|e(n)\right|^2\right]$$

$$w(n) = \mathbf{R}_{\mathbf{xx}}^{-1}\mathbf{r}_{\mathbf{x}d}$$

$$\mathbf{R}_{\mathbf{xx}} = E\left[\mathbf{x}(n)\mathbf{x}(n)^T\right]$$

$$\mathbf{r}_{\mathbf{x}d} = E\left[\mathbf{x}(n)d(n)\right]$$

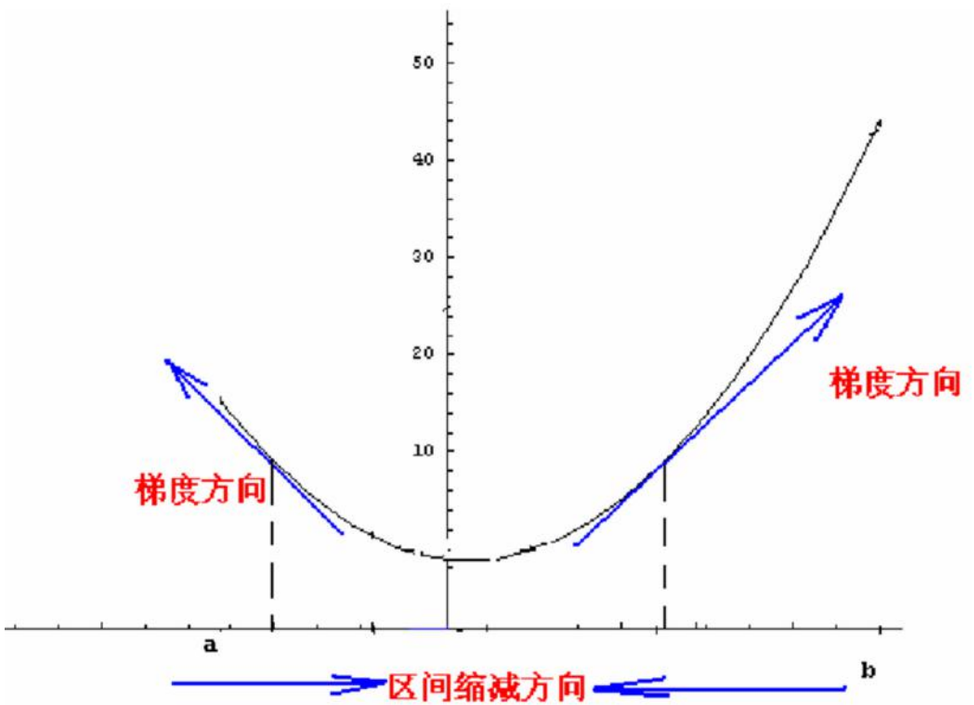
$$\mathbf{x}(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^{\mathrm{T}}$$

语音信号处理-回声消除

最陡下降法

$$\mathbf{R}_{xx}\mathbf{w} = \mathbf{r}_{xd}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu\nabla_k\xi \qquad 0 < \mu < \frac{1}{\lambda_{\max}}$$



语音信号处理-回声消除

LMS算法

最陡下降法中，性能函数为 $\xi = \mathrm{E}\left[e^2(n)\right]$

使用估计误差平方的即时估计作为性能函数 $\hat{\xi} = e^2(n)$

LMS 算法

- 滤波（Filtering）：

$$y(n) = \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)$$

- 误差估计（Error estimation）：

$$e(n) = d(n) - y(n)$$

- 权值更新（Tap-weight update）：

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n)$$

语音信号处理-回声消除

NLMS算法

NLMS 算法

- 滤波

$$y(n)=\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)$$

- 误差估计

$$e(n)=d(n)-y(n)$$

- 功率计算

$$P_{xx}(n)=\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n)$$

- 权值更新

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\frac{\mu}{P_{xx}(n)+\psi}e(n)\mathbf{x}(n)$$

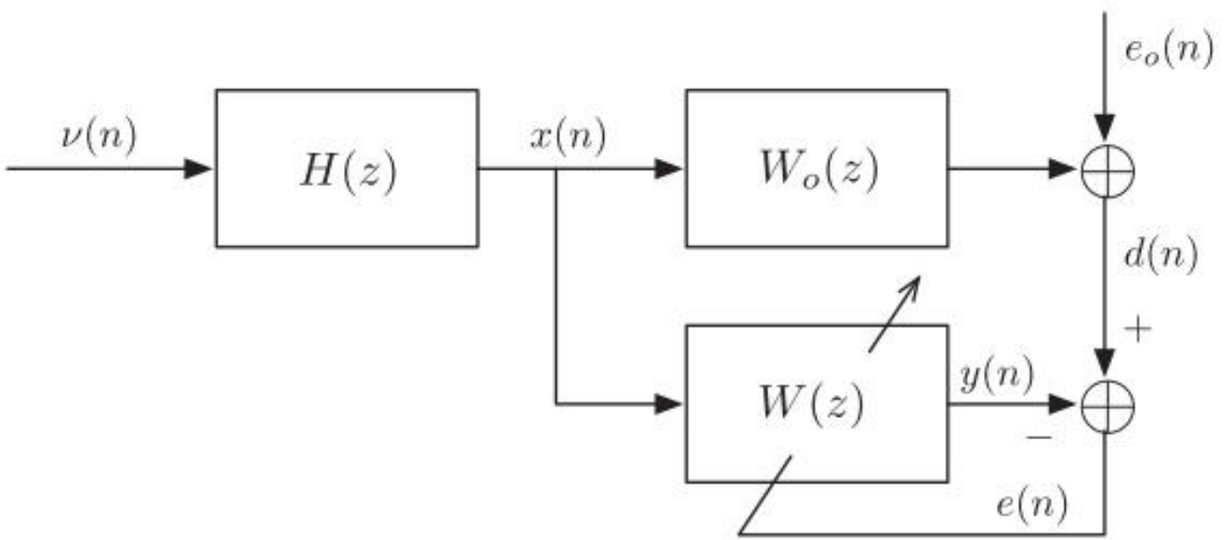
直接计算： $P_{xx}(n)=\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n)=\sum_{i=1}^{N}x_n^{\ 2}(i)$

递推计算： $P_{xx}(n)=P_{xx}(n-1)-x_{n-1}^{\ 2}(N)+x_n^{\ 2}(1)$

遗忘计算： $P_{xx}(n)=(1-\alpha)\cdot P_{xx}(n-1)+\alpha\cdot x^2(n)$

语音信号处理-回声消除

算法仿真



$v(n)$ 是方差为1的白噪声

$H(z)$ 将白噪声变为有色噪声

$W_0(z)$ 待估计滤波器

$W(z)$ 自适应滤波器

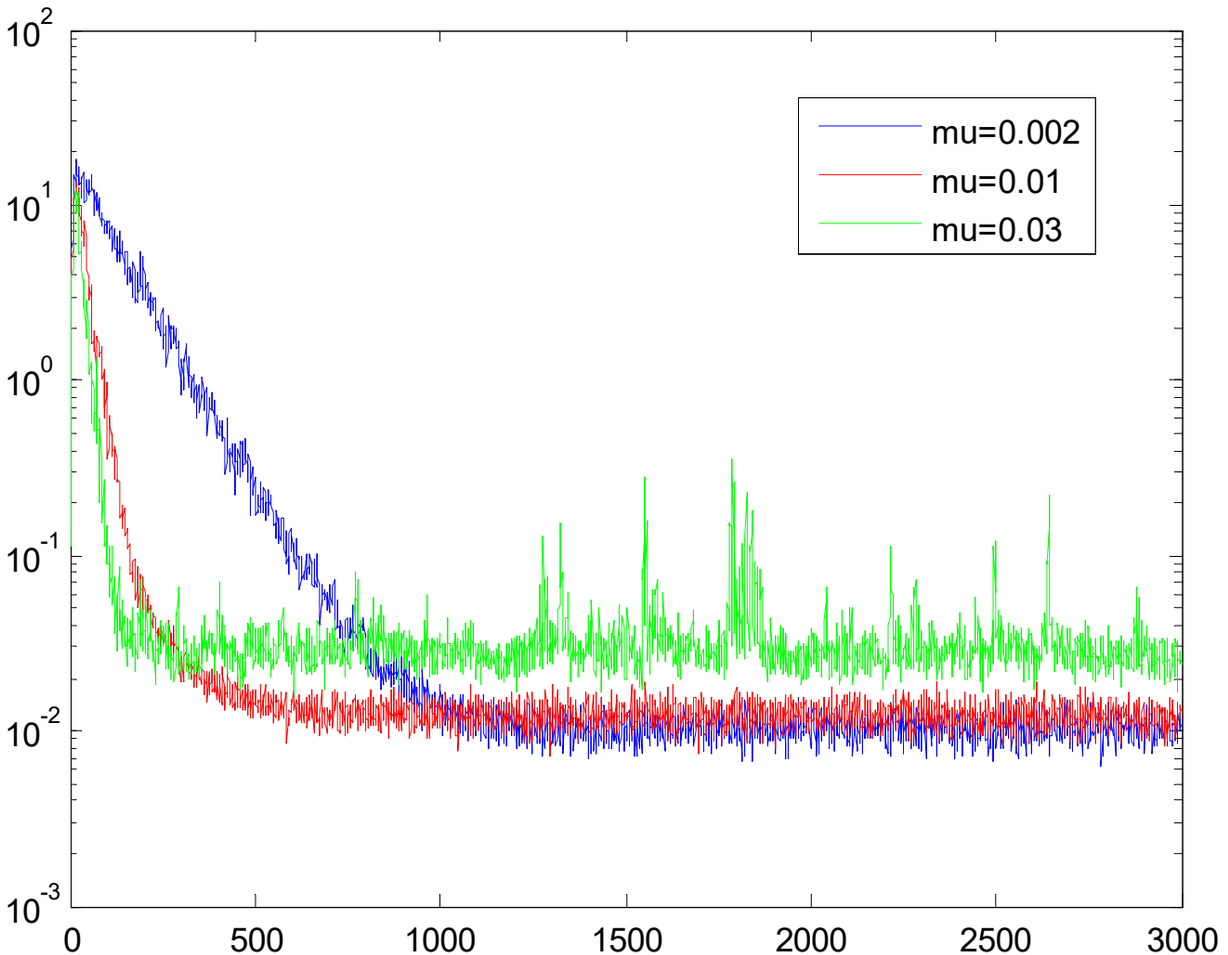
$e_0(n)$ 方差为0.01的白噪声干扰

$$H_1(z) = 0.35 + z^{-1} - 0.35z^{-2}$$

$$W_o(z) = \sum_{i=0}^{7} z^{-i} - \sum_{i=8}^{14} z^{-i}$$

语音信号处理-回声消除

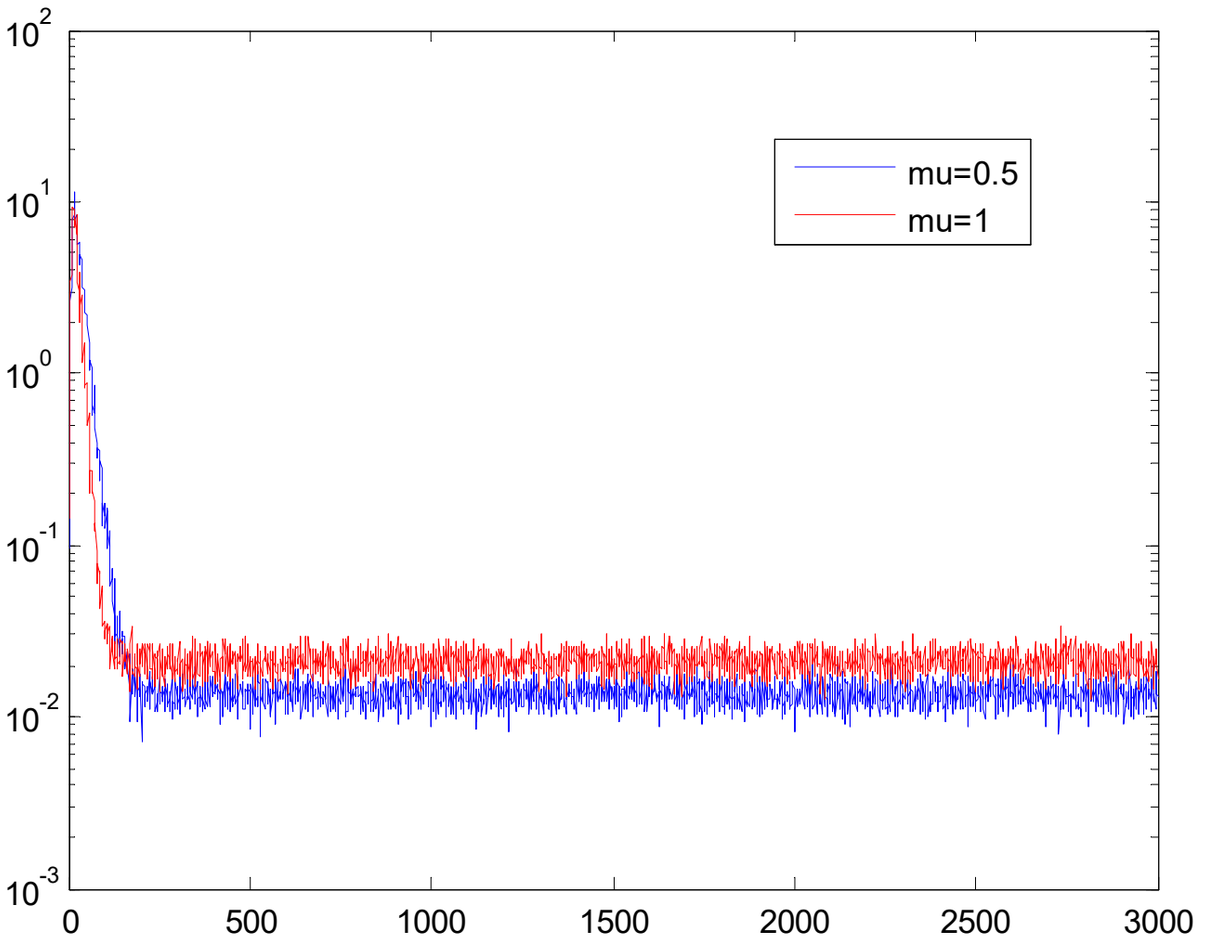
算法仿真



LMS算法

语音信号处理-回声消除

算法仿真



NLMS算法

语音信号处理-回声消除

RLS算法

$$h(n) = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xd}$$

$$(\mathbf{A} + \alpha\mathbf{a}\mathbf{a}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1} - \frac{\alpha\mathbf{A}^{-1}\mathbf{a}\mathbf{a}^{\mathrm{T}}\mathbf{A}^{-1}}{1 + \alpha\mathbf{a}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{a}}$$

Input: Tap-weight vector estimate, $\hat{\mathbf{w}}(n-1)$,
Input vector, $\mathbf{x}(n)$, desired output, $d(n)$,
and the matrix $\mathbf{R}_{xx}^{-1}(n-1)$

Output: Filter output, $\hat{y}_{n-1}(n)$,
tap-weight vector update, $\hat{\mathbf{w}}(n)$,
and the updated matrix $\mathbf{R}_{xx}^{-1}(n)$

1. Computation of the gain vector:

$$\mathbf{u}(n) = \mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)$$

$$\mathbf{k}(n) = \frac{1}{\lambda + \mathbf{x}^{\mathrm{T}}(n)\mathbf{u}(n)}\mathbf{u}(n)$$

2. Filtering:

$$\hat{y}_{n-1}(n) = \hat{\mathbf{w}}^{\mathrm{T}}(n-1)\mathbf{x}(n)$$

3. Error estimation:

$$\hat{e}_{n-1}(n) = d(n) - \hat{y}_{n-1}(n)$$

4. Tap-weight vector adaptation:

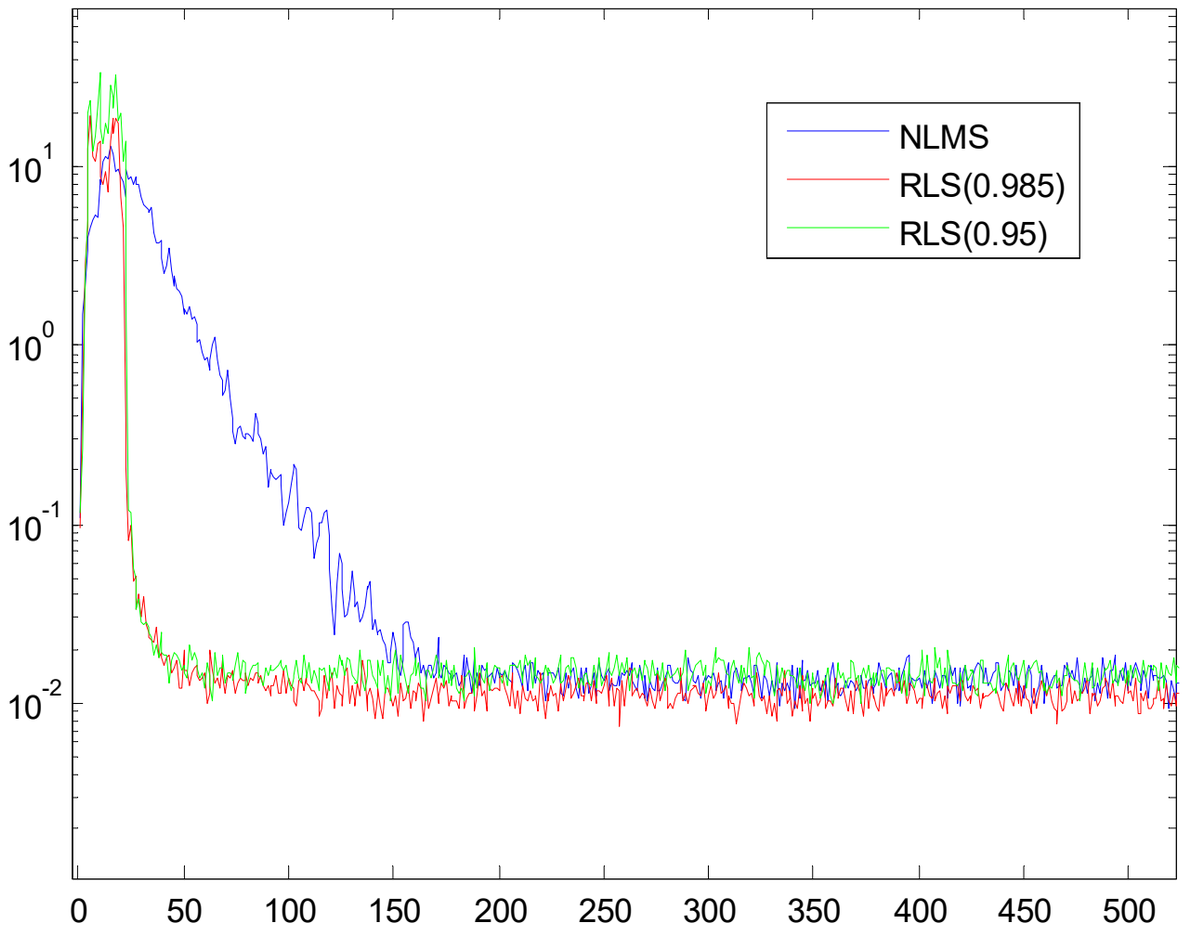
$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\hat{e}_{n-1}(n)$$

5. $\mathbf{R}_{xx}^{-1}(n)$ update:

$$\mathbf{R}_{xx}^{-1}(n) = \lambda^{-1}\left(\mathbf{R}_{xx}^{-1}(n-1) - \mathbf{k}(n)\left[\mathbf{x}^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)\right]\right)$$

语音信号处理-回声消除

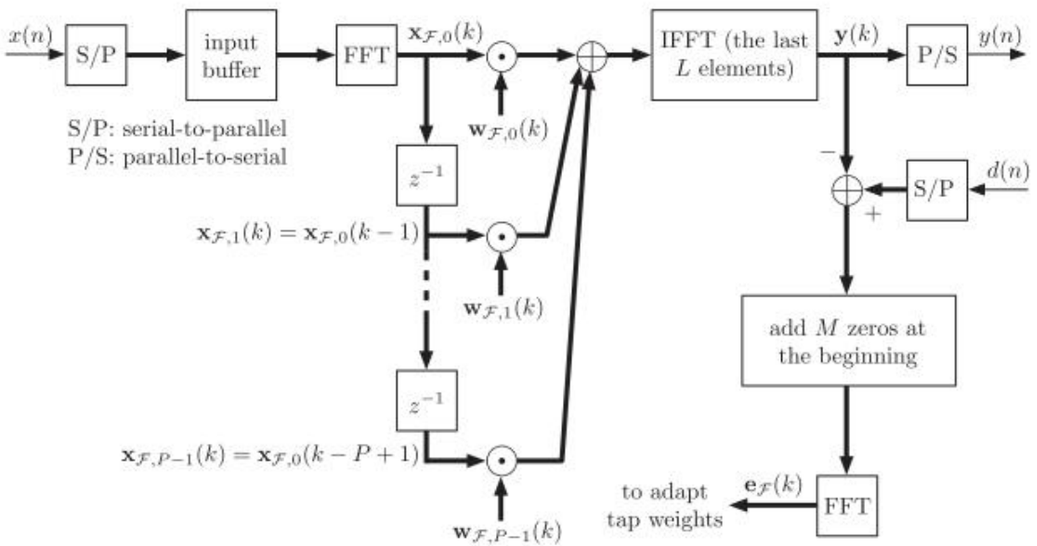
RLS算法



RLS算法

语音信号处理-回声消除

PFBLMS算法



Input: Tap-weight vectors, $\mathbf{w}_{\mathcal{F},l}(k)$, $l = 0, 1, \ldots, P-1$,
Extended input vector,
$\tilde{\mathbf{x}}_0(k) = [x(kL-M) \quad x(kL-M+1) \quad \cdots \quad x(kL+L-1)]^{\mathrm{T}}$,
The past frequency domain vectors of input, $\mathbf{x}_{\mathcal{F},0}(k-l)$, for $l = 1, 2, \ldots, (P-1)p$,
and desired output vector,
$\mathbf{d}(k) = [d(kL) \quad d(kL+1) \quad \cdots \quad d(kL+L-1)]^{\mathrm{T}}$.

Output: Filter output, $\mathbf{y}(k) = [y(kL)\ y(kL+1) \cdots y(kL+L-1)]^{\mathrm{T}}$,
Tap-weight vector update, $\mathbf{w}_{\mathcal{F},l}(k+1)$, $l = 0, 1, \ldots, P-1$.

---

1. Filtering:

$$\mathbf{x}_{\mathcal{F},0}(k) = \mathrm{FFT}(\tilde{x}_0(k))$$

$$\mathbf{y}(k) = \text{the last } L \text{ elements of IFFT}\left(\sum_{l=0}^{P-1} \mathbf{w}_{\mathcal{F},l}(k) \odot \mathbf{x}_{\mathcal{F},0}(k-pl)\right)$$

2. Error estimation:

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{y}(k)$$

3. Step-normalization:

$$\text{for } i = 0 \text{ to } M'-1$$

$$\hat{\sigma}^2_{x_{\mathcal{F},0,i}}(k) = \beta\hat{\sigma}^2_{x_{\mathcal{F},0,i}}(k-1) + (1-\beta)|x_{\mathcal{F},0,i}(k)|^2$$

$$\mu_i(k) = \mu_o/\hat{\sigma}^2_{x_{\mathcal{F},0,i}}(k)$$

$$\mu(k) = [\mu_0(k) \quad \mu_1(k) \quad \cdots \quad \mu_{M'-1}(k)]^{\mathrm{T}}$$

5. Tap-weight adaptation:

$$\mathbf{e}_{\mathcal{F}}(k) = \mathrm{FFT}\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{e}(k)\end{bmatrix}\right)$$

$$\text{for } l = 0 \text{ to } P-1$$

$$\mathbf{w}_{\mathcal{F},l}(k+1) = \mathbf{w}_{\mathcal{F},l}(k) + 2\mu(k) \odot \mathbf{x}^*_{\mathcal{F},0}(k-pl) \odot \mathbf{e}_{\mathcal{F}}(k)$$

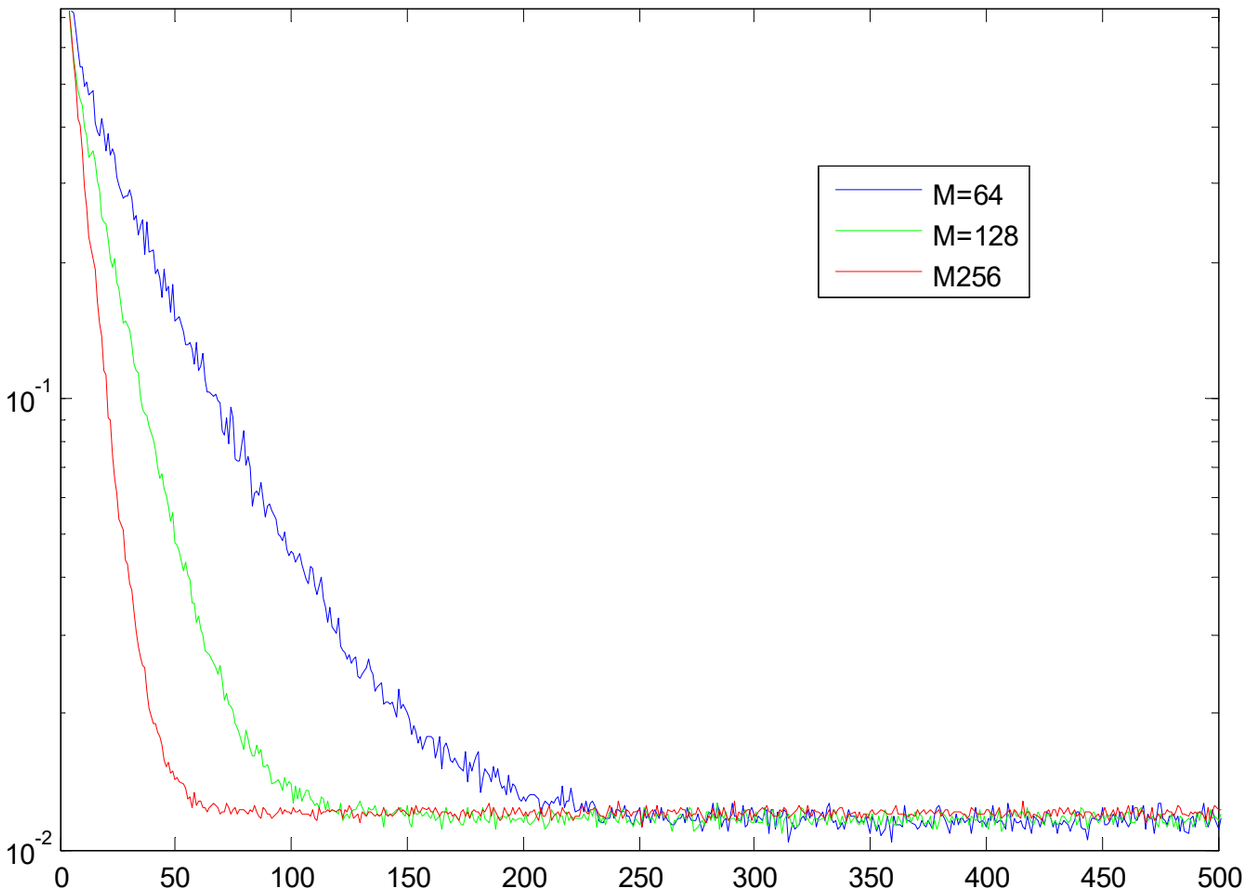
5. Tap-weight constraint:

$$\text{for } l = 0 \text{ to } P-1$$

$$\mathbf{w}_{\mathcal{F},l}(k+1) = \mathrm{FFT}\left(\begin{bmatrix}\text{first} \quad M \quad \text{elements of IFFT}(\mathbf{w}_{\mathcal{F},l}(k+1))\\ \mathbf{0}\end{bmatrix}\right)$$

语音信号处理-回声消除

PFBLMS算法性能

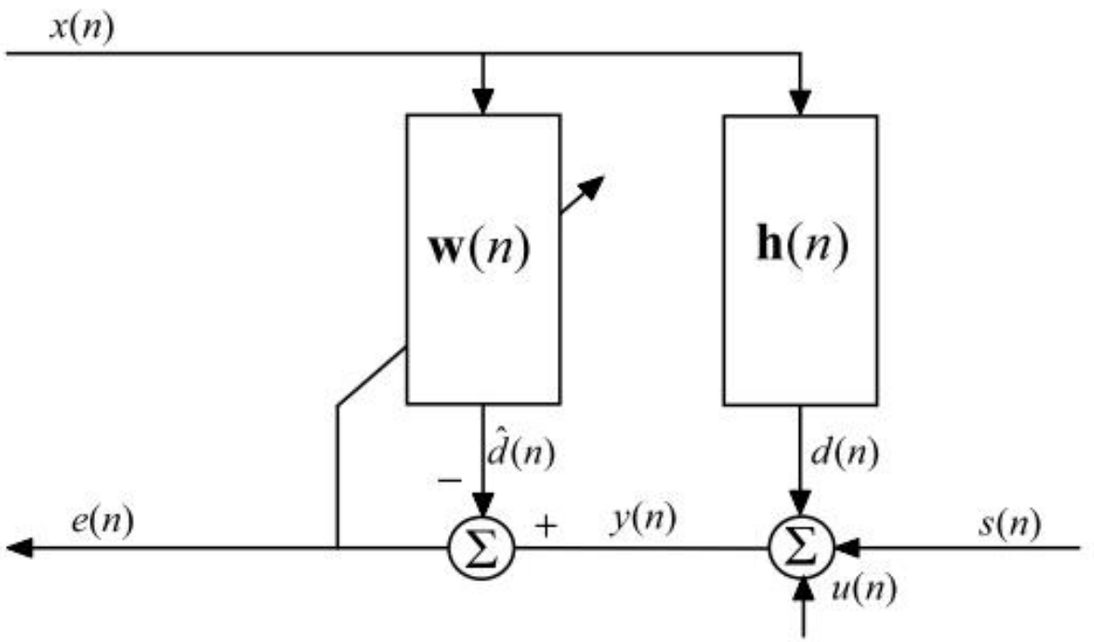


PFBLMS算法

$$M_{PFBLMS} = \mu_0 P/2$$

语音信号处理-回声消除

AEC中滤波器保护



$$\mu_{m\_opt}(k)=\frac{S_{\varepsilon\varepsilon}(k)}{S_{ee}(k)}=\frac{\left|S_{\hat{d}e}(k)\right|}{S_{\hat{d}\hat{d}}(k)*S_{ee}(k)}$$

$$\varepsilon(n)=d(n)-\hat{d}(n)$$

DoubleTalk检测

语音信号处理-回声消除

AEC后滤波算法

$$P_m(k)=\frac{S_{ll,m}(k)}{S_{ee,m}(k)}=\frac{S_{ll,m}(k)}{S_{ll,m}(k)+S_{\varepsilon\varepsilon,m}(k)}$$

Webrtc后滤波算法

$$P_m(k)=Min(r_{de},1-r_{xd})$$

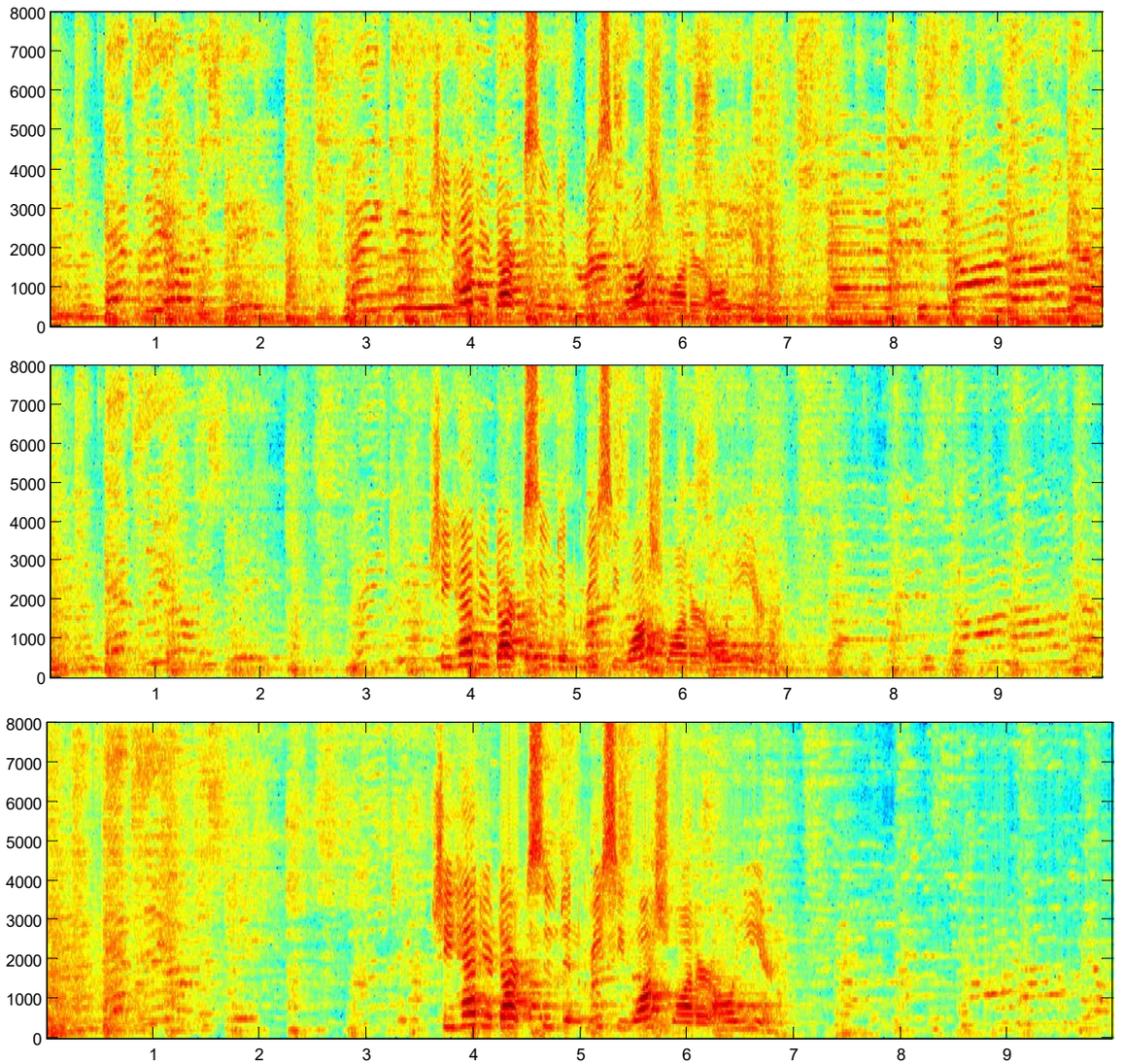
$$r_{de}=\frac{|S_{de}|^2}{S_{dd}S_{ee}} \qquad r_{xd}=\frac{|S_{xd}|^2}{S_{xx}S_{dd}}$$

Speex后滤波算法

$$P_m(k)=\frac{\hat{S}_{ll}(k)}{\hat{S}_{ll}(k)+\hat{S}_{\varepsilon\varepsilon}(k)+\hat{S}_{nn}(k)}$$

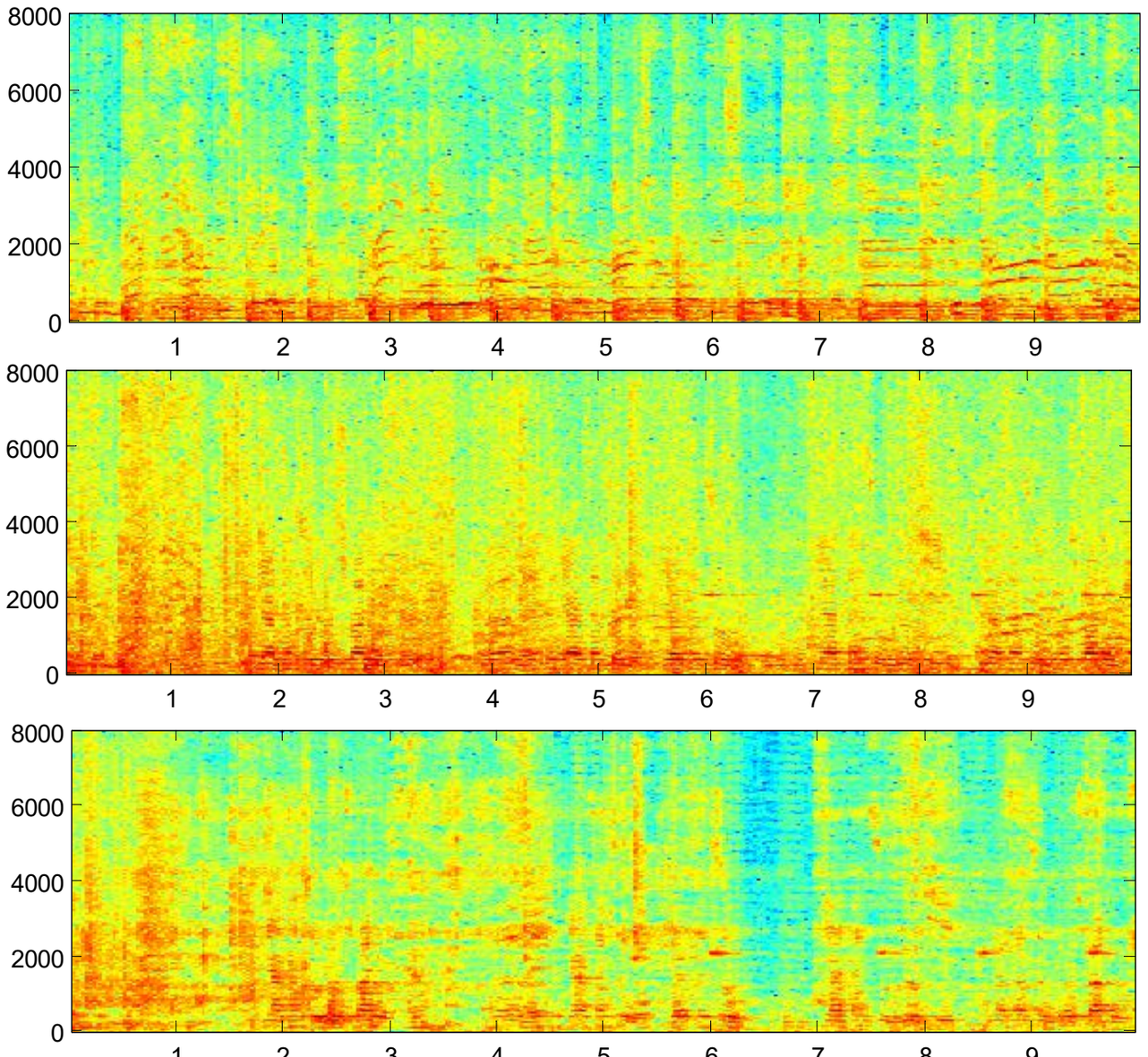
语音信号处理-回声消除

AEC后滤波算法



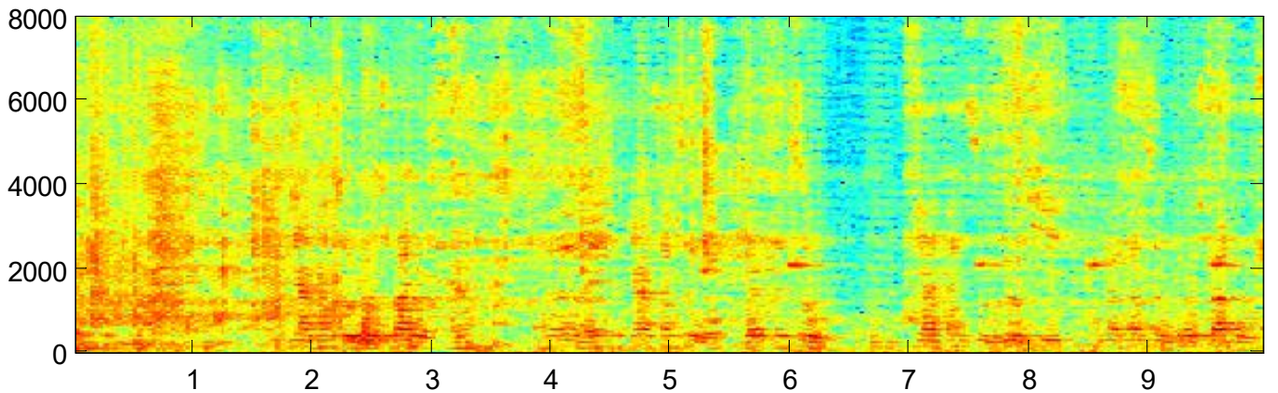
语音信号处理-回声消除

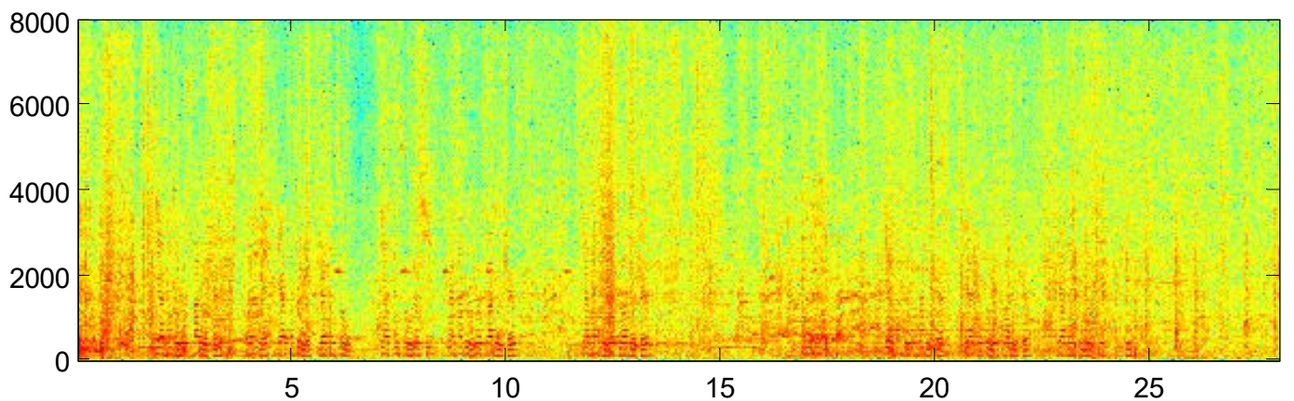
AEC后滤波算法



语音信号处理-回声消除

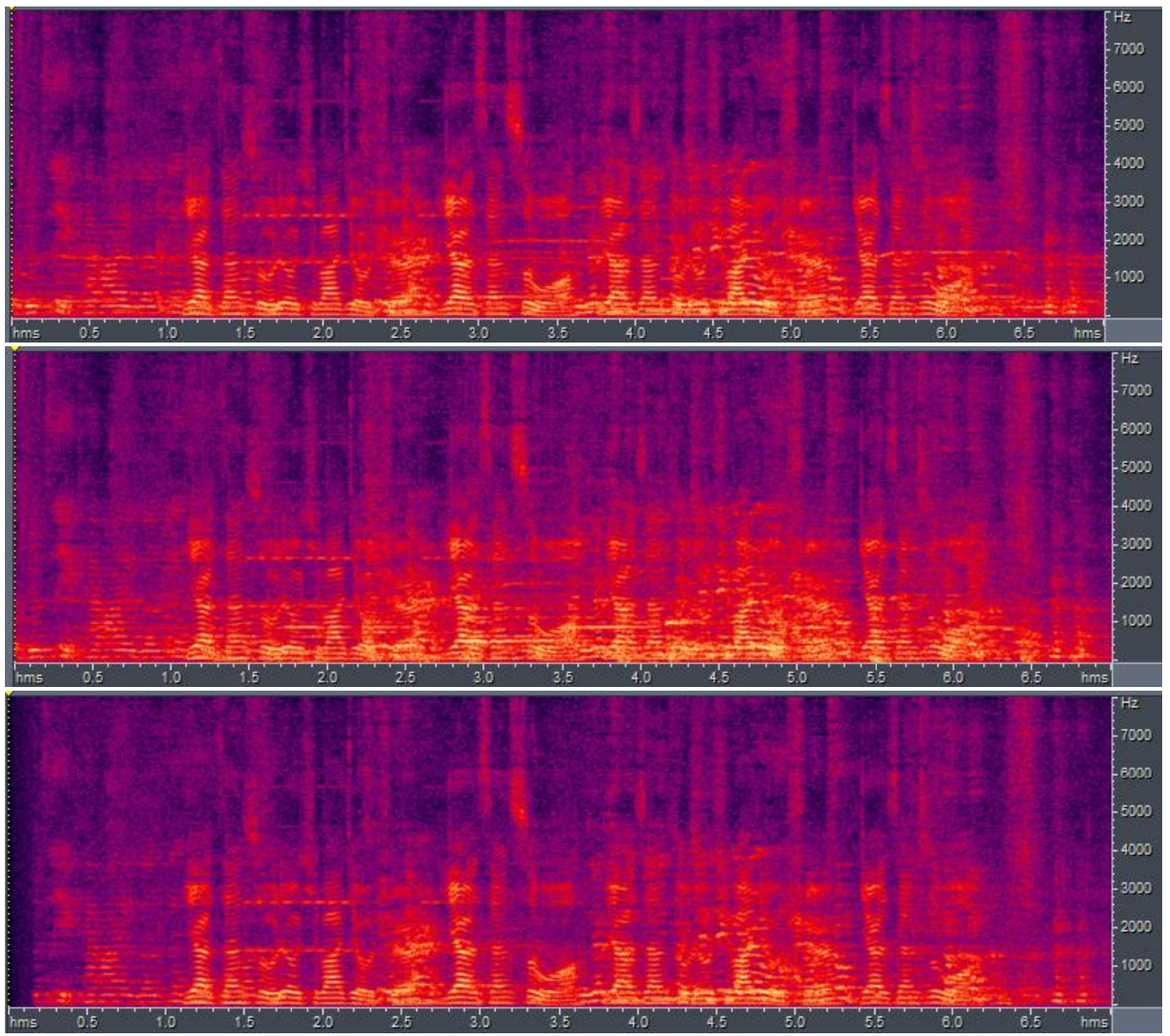
AEC后滤波算法





语音信号处理-回声消除

AEC滤波器发散



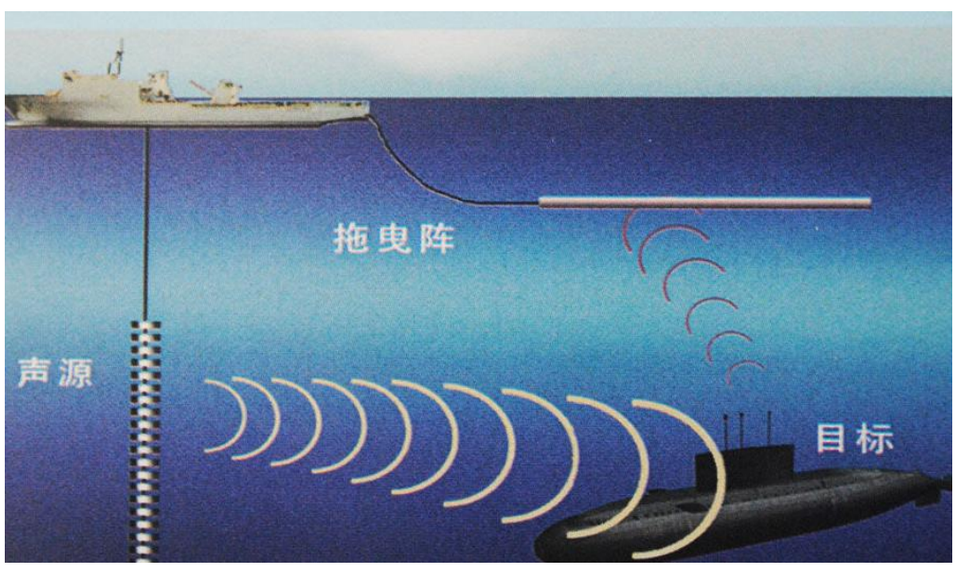
语音信号处理-波达方向估计

波达方向估计

语音信号处理-波达方向估计

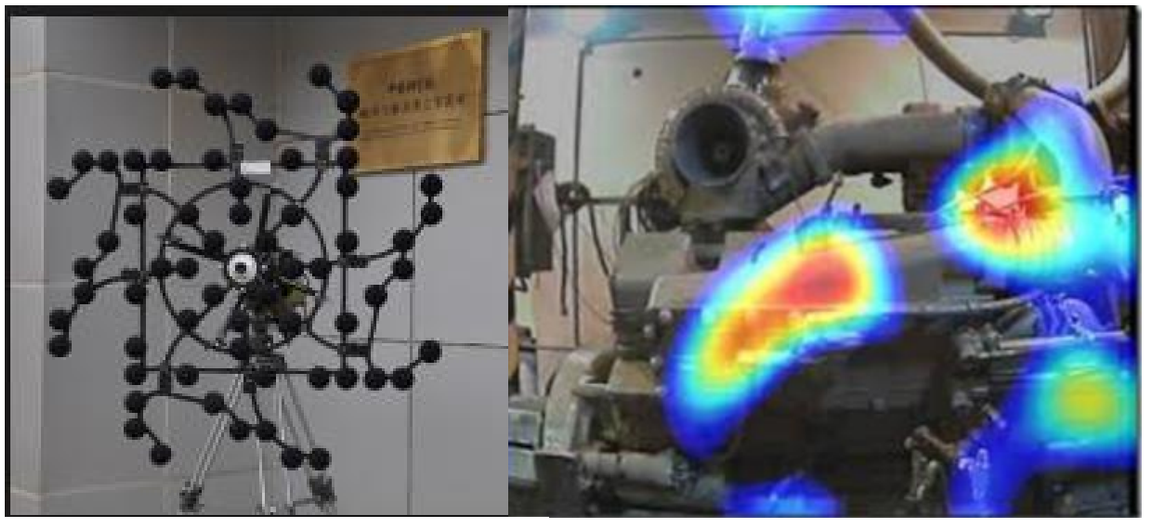
雷达

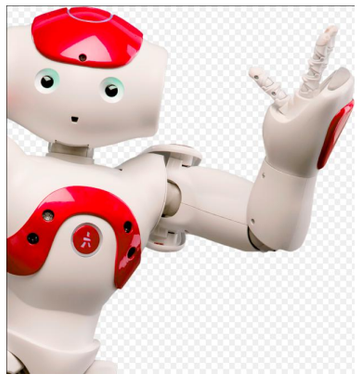
声呐

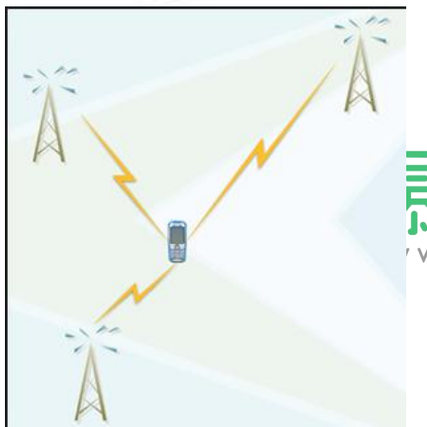


语音信号处理-波达方向估计


↓ 法国“皮勒尔”系统（右），该系统也是世界上广泛使用的声探测反狙击系统。系统还可连接视频输出终端（左），终端带有摄像机可立即转向弹丸来袭方向，显示相关区域视频图像。







语音信号处理-波达方向估计

麦克风阵列形式：分布式阵列和传统阵列

分布式阵列算法介绍

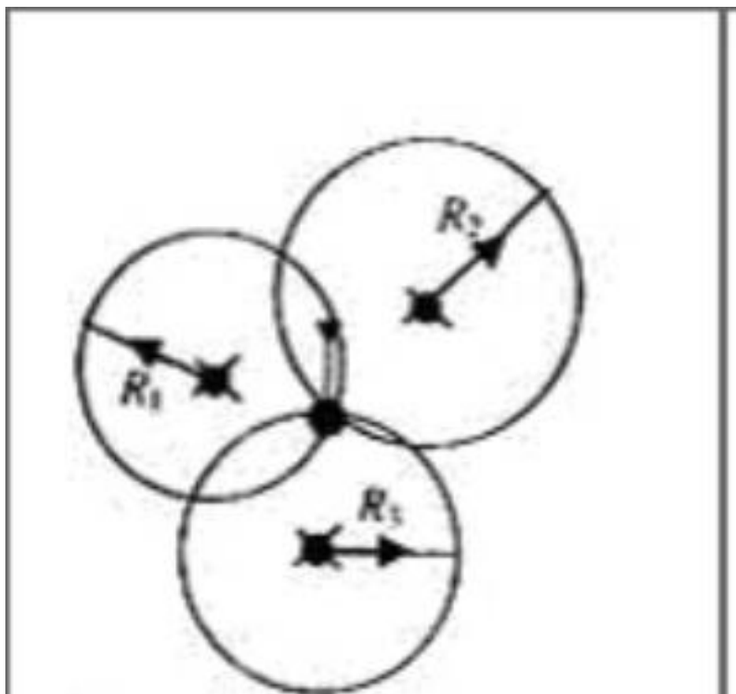


图 1　TOA 算法示意图

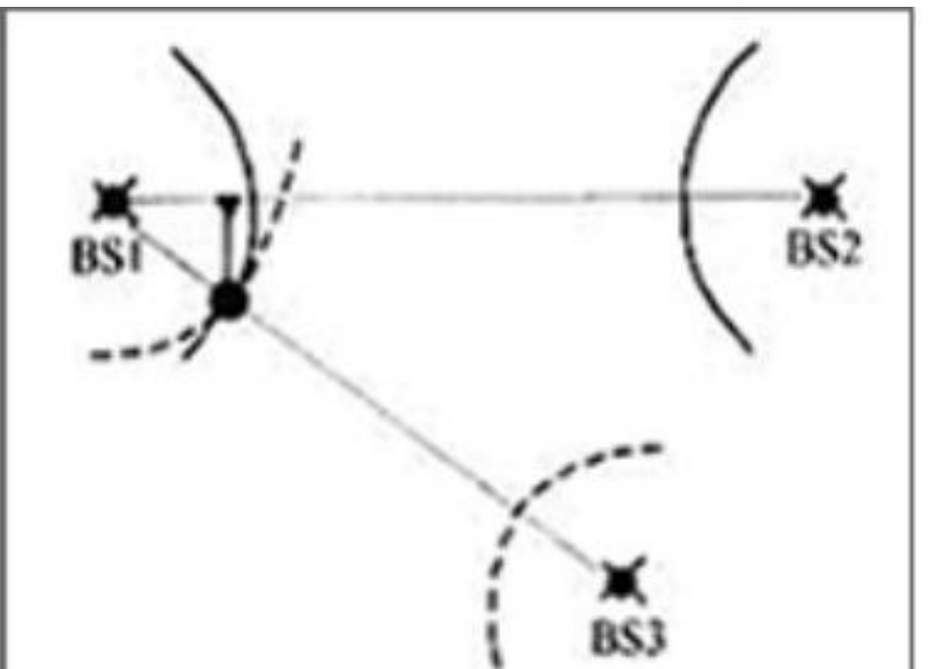
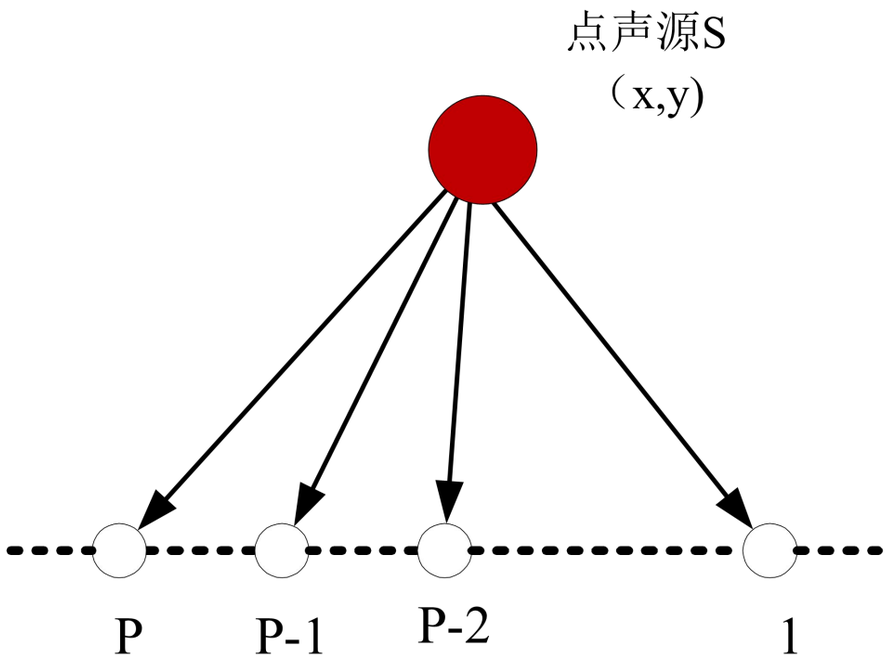


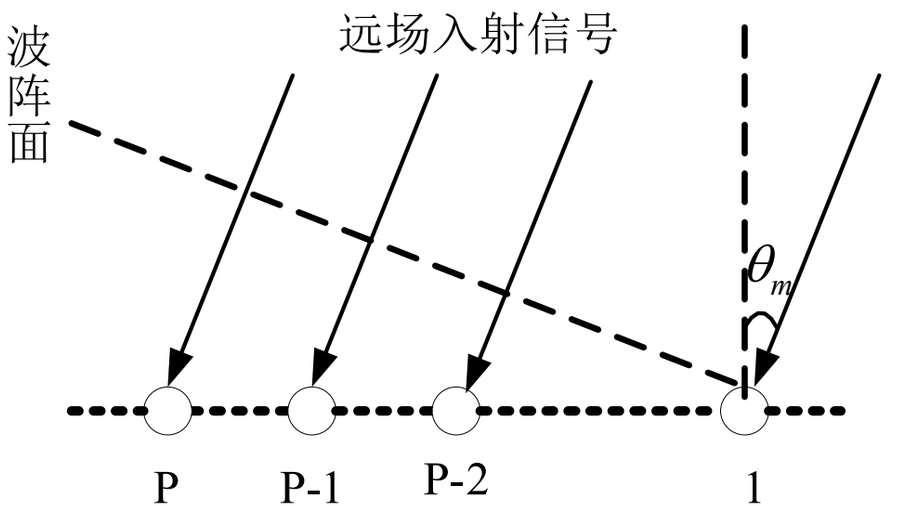
图 2　TDOA 算法示意图

语音信号处理-波达方向估计

麦克风阵列形式：分布式阵列和传统阵列

传统阵列算法介绍





$$\tau_{pm}=\left(\sqrt{(x-x_p)^2+(y-y_p)^2}-\sqrt{(x-x_m)^2+(y-y_m)^2}\right)/c$$

$$\tau_{pm}=\frac{(p-1)d\sin\theta_m}{c}\qquad p=1,2,\ldots,P$$

语音信号处理-波达方向估计

传统阵列算法介绍：窄带信号

阵列基本概念

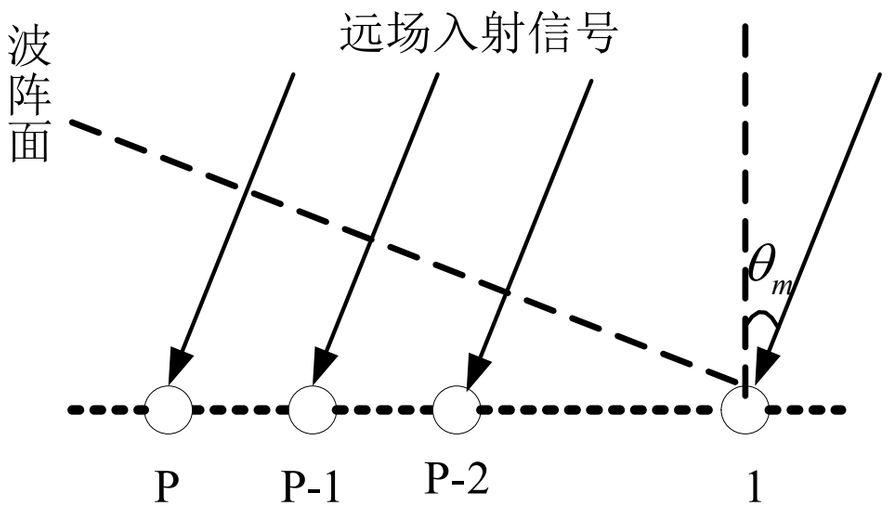
$$x_p(t)=\sum_{m=1}^{M}s_m(t)e^{-i2\pi f\tau_{mp}}+w_p(t)\quad p=1,2,...,P$$

$$\mathbf{x}(t)=\mathbf{A}\mathbf{s}(t)+\mathbf{w}(t)$$

$$\mathbf{x}(t)=\left[x_1(t),x_2(t),...,x_P(t)\right]^T$$

$$\mathbf{A}=\left[\mathbf{a}_1,\mathbf{a}_2,...,\mathbf{a}_M\right]$$

$$\mathbf{a}_m=\left[e^{-i2\pi f\tau_{1m}},e^{-i2\pi f\tau_{2m}},...,e^{-i2\pi f\tau_{Pm}}\right]^T$$



$$\mathbf{R}_x=\mathrm{E}\left\{\mathbf{x}(t)\mathbf{x}^H(t)\right\}=\mathbf{A}\,\mathrm{E}\left\{\mathbf{s}(t)\mathbf{s}^H(t)\right\}\mathbf{A}^H+\mathrm{E}\left\{\mathbf{w}(t)\mathbf{w}^H(t)\right\}$$

$$=\mathbf{A}\mathbf{R}_s\mathbf{A}^H+\mathbf{R}_w$$

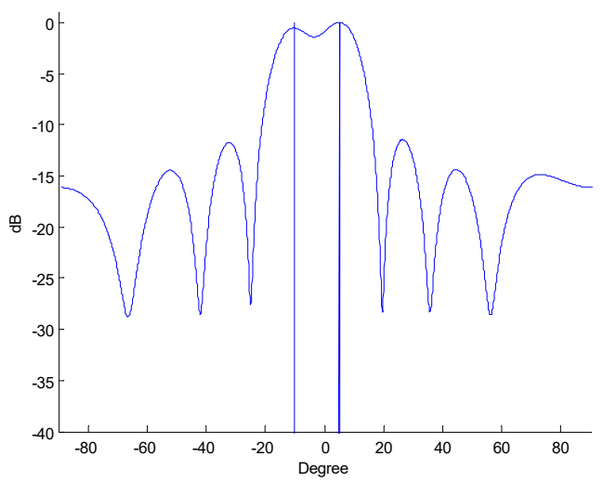
$$\mathbf{R}_x=\mathbf{A}\mathbf{R}_s\mathbf{A}^H+\sigma_w^2\mathbf{I}$$

语音信号处理-波达方向估计

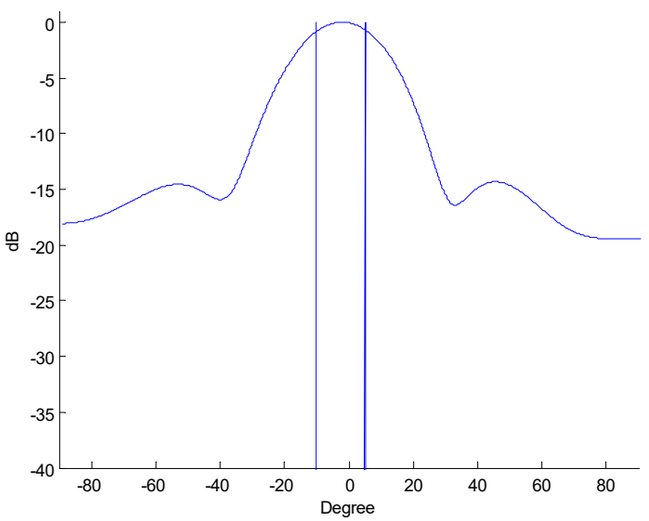
传统阵列算法介绍：窄带信号

波束形成算法

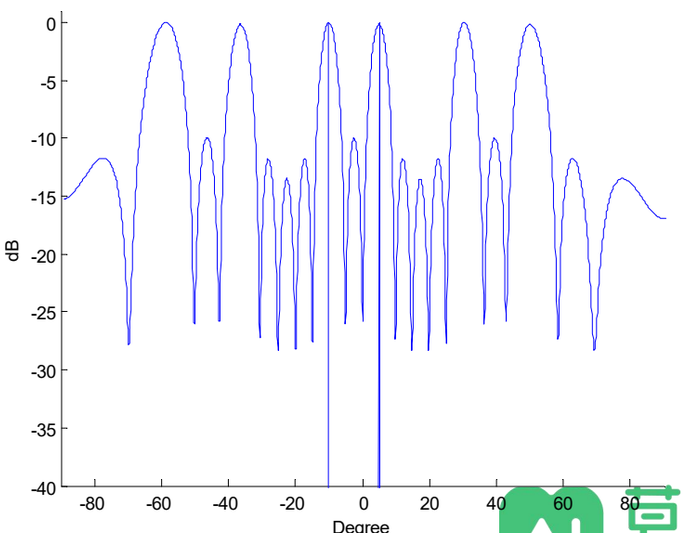
$$J = \mathbf{a}(\theta)\mathbf{R}\mathbf{a}(\theta)^H \qquad \mathbf{a}(\theta) = \left[1, e^{-j2\pi f d\sin(\theta)/c}, \ldots, e^{-j2\pi f(P-1)d\sin(\theta)/c}\right]^T$$



$d = \lambda / 2$



$d < \lambda / 2$

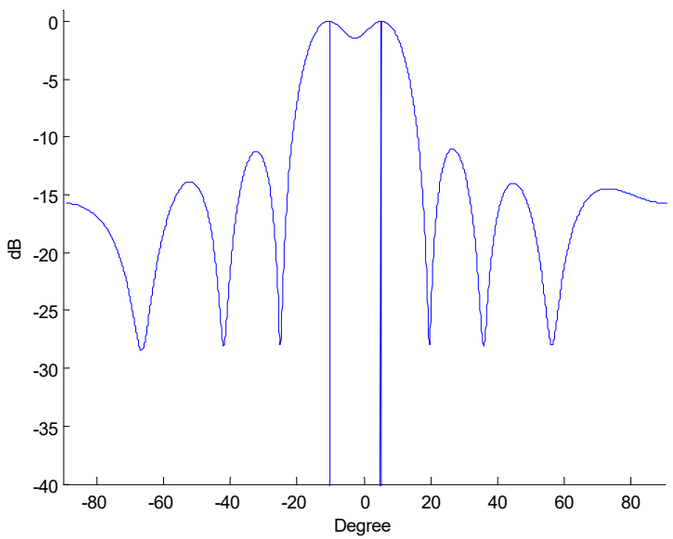


$d > \lambda / 2$

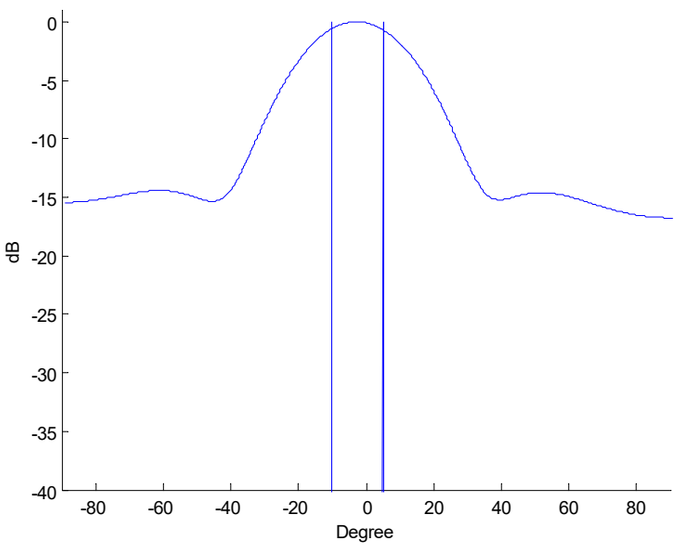
语音信号处理-波达方向估计

传统阵列算法介绍：窄带信号

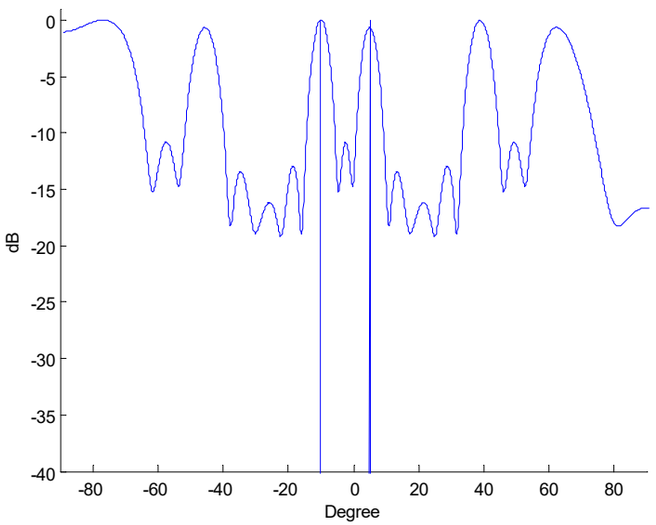
$$d = 0.17$$



$$f = 1000$$



$$f = 500$$



$$f = 2500$$

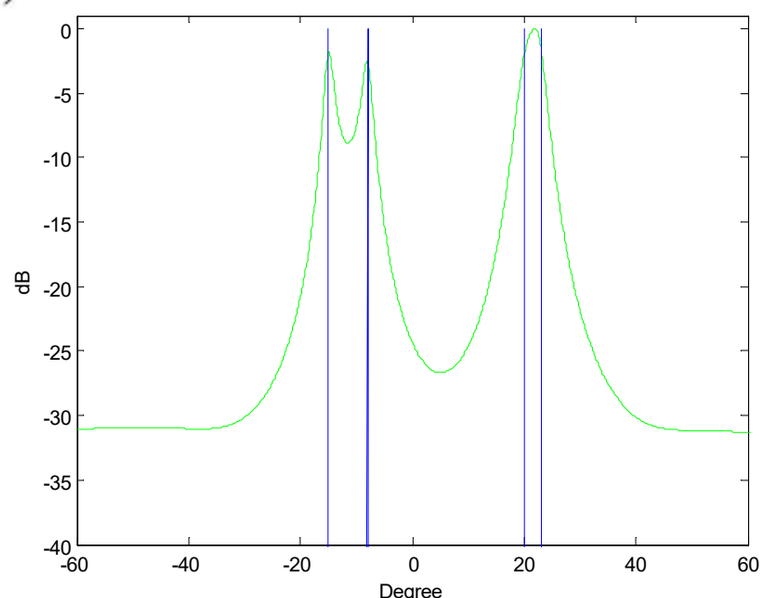
语音信号处理-波达方向估计

传统阵列算法介绍：窄带信号

Capon算法

$$\begin{cases} \min \mathbf{w}^H R\mathbf{w} \\ \mathbf{w}^H \mathbf{a}(\theta) = 1 \end{cases} \qquad \mathbf{w} = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta)}{\mathbf{a}(\theta)\mathbf{R}^{-1}\mathbf{a}(\theta)^H}$$

$$J = \frac{1}{\mathbf{a}(\theta)\mathbf{R}^{-1}\mathbf{a}(\theta)^H} \qquad \mathbf{a}(\theta) = \left[1, e^{-j2\pi f d\sin(\theta)/c}, \ldots, e^{-j2\pi f(P-1)d\sin(\theta)/c}\right]^T$$



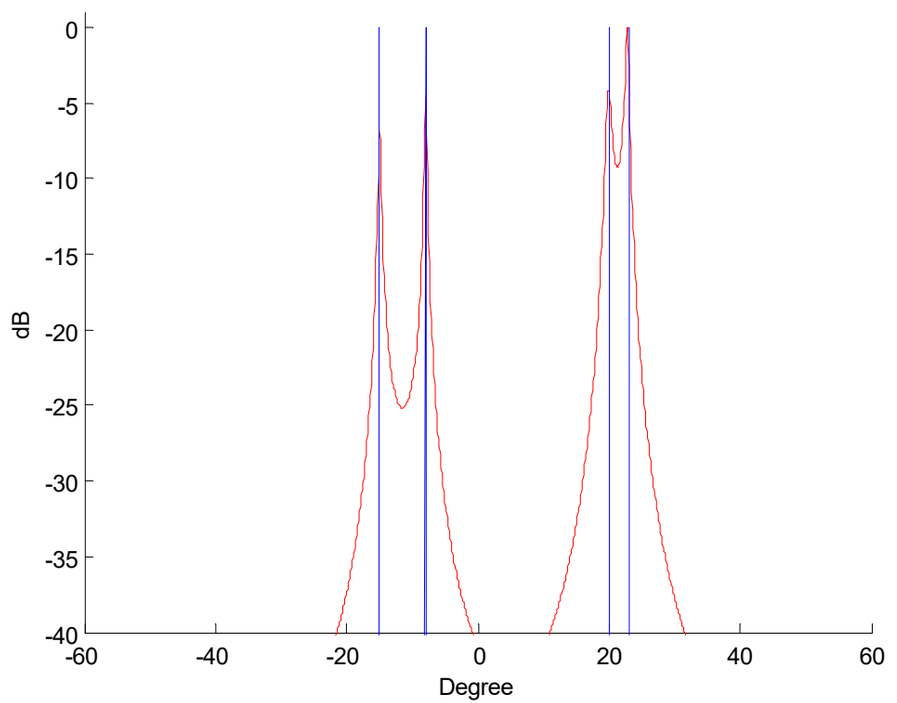
语音信号处理-波达方向估计

传统阵列算法介绍：窄带信号

MUSIC算法

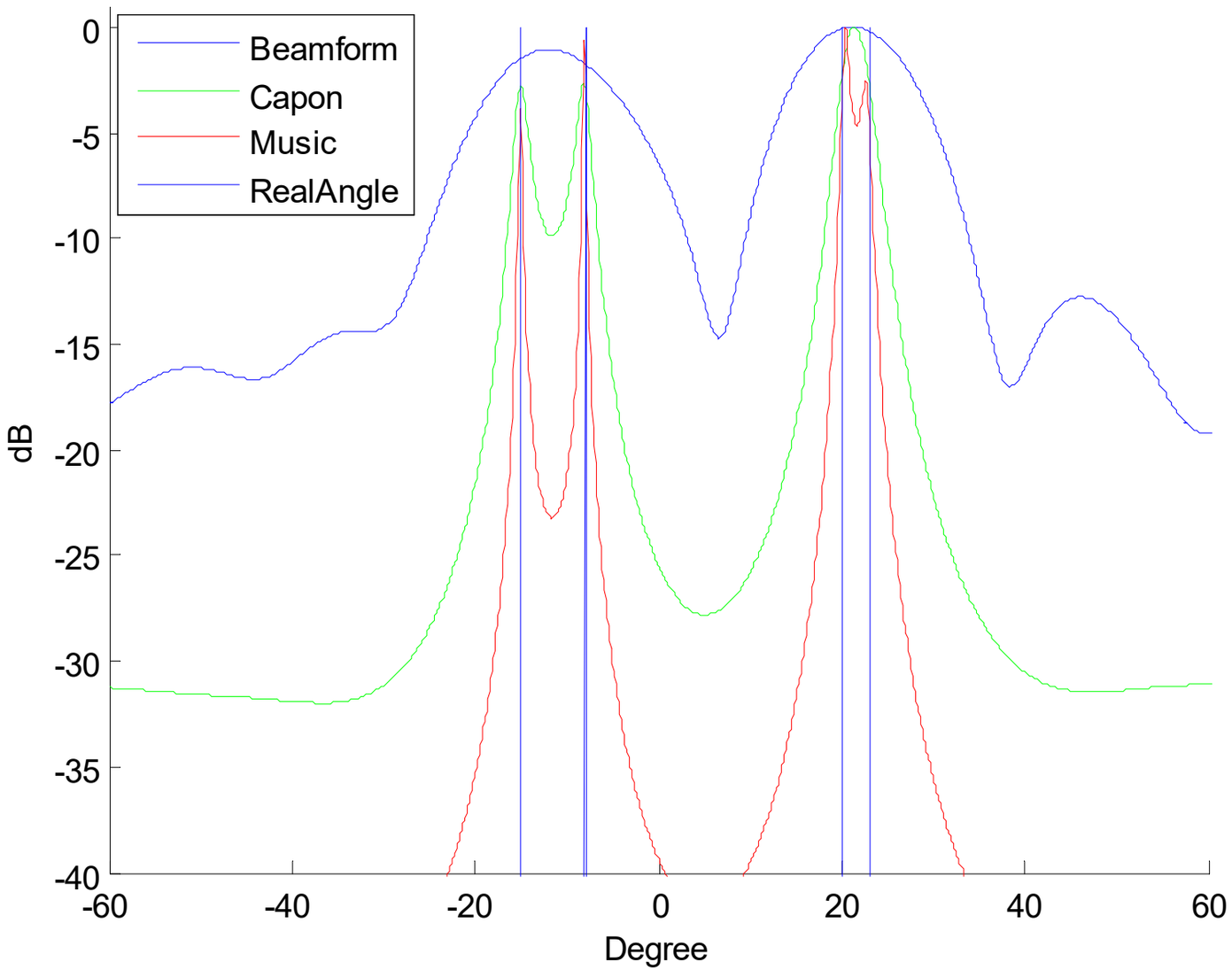
$$\mathbf{R}_x = \mathbf{U}_s \mathbf{\Lambda}_s \mathbf{U}_s^H + \mathbf{U}_n \mathbf{\Lambda}_n \mathbf{U}_n^H \qquad \mathbf{R} = \mathbf{U} diag\left[\sigma_1^2, \sigma_2^2, ..., \sigma_P^2\right] \mathbf{U}^H$$

$$J_{MUSIC}(\theta) = \frac{1}{\mathbf{a}(\theta)^H \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}(\theta)}$$



语音信号处理-波达方向估计

传统阵列算法介绍：窄带信号



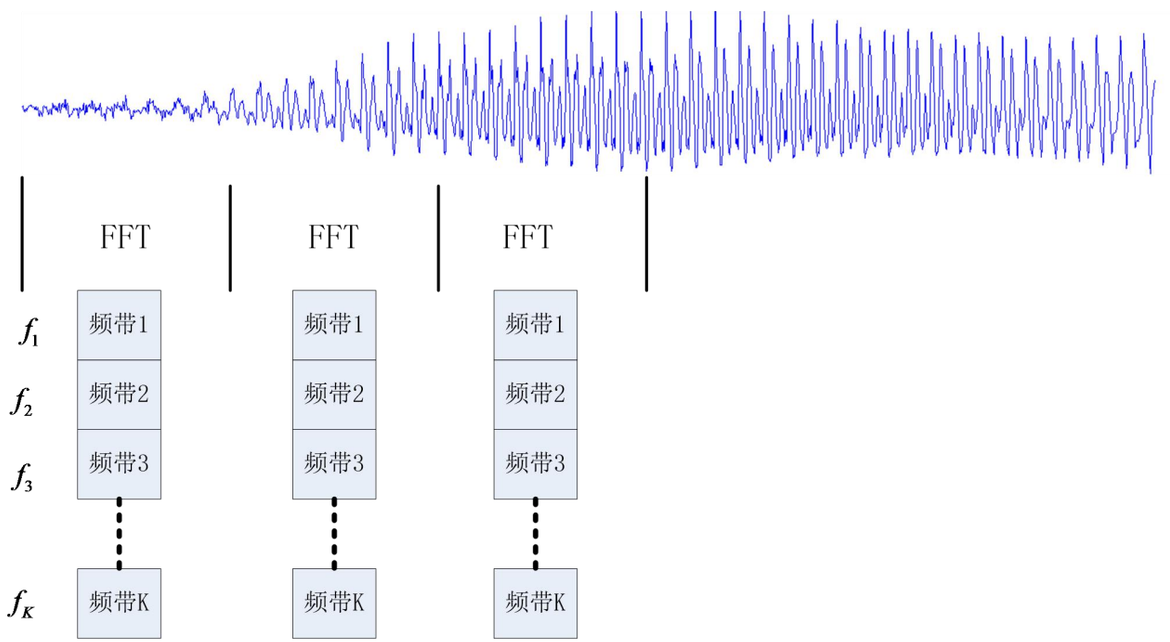
语音信号处理-波达方向估计

传统阵列算法介绍：宽带信号

$$x_p(t)=\sum_{m=1}^{M}s_m(t-\tau_{mp})+w_p(t)\quad p=1,2,...,P$$

$$X_p(f_j)=\sum_{m=1}^{M}S_m(f_j)e^{-i2\pi f_j\tau_{mp}}+W_p(f_j)\qquad p=1,2,...,P$$

$$\mathbf{X}(f_j)=\mathbf{A}(f_j)\mathbf{S}(f_j)+\mathbf{W}(f_j)$$



语音信号处理-波达方向估计

传统阵列算法介绍：宽带信号

宽带波束形成算法

$$J(\theta)=\sum_{k=1}^{K}\mathbf{a}(f_k,\theta)^H\mathbf{R}(f_k)\mathbf{a}(f_k,\theta)$$

宽带Capon算法

$$J(\theta)=\sum_{k=1}^{K}\frac{1}{\mathbf{a}(f_k,\theta)^H\mathbf{R}^{-1}(f_k)\mathbf{a}(f_k,\theta)}$$

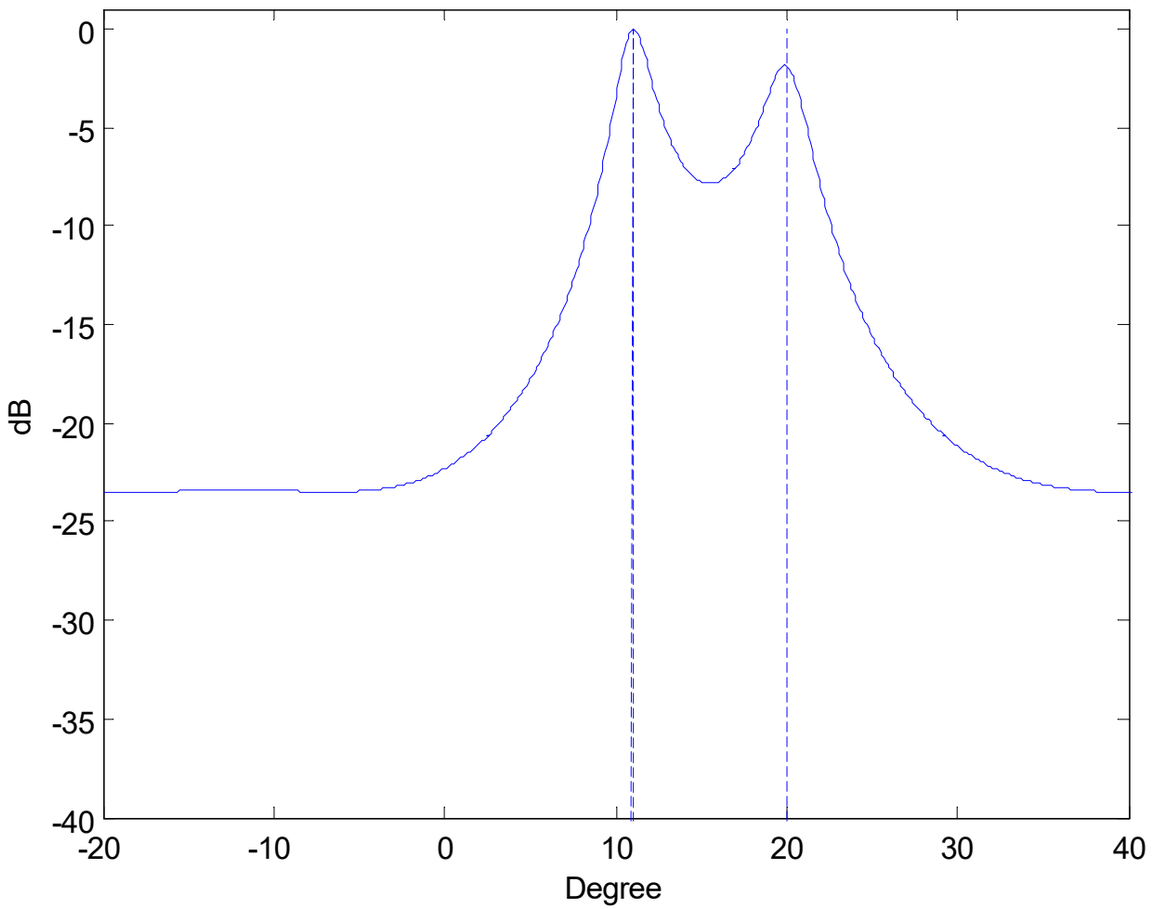
宽带MUSIC算法

$$J(\theta)=\sum_{k=1}^{K}\frac{1}{\mathbf{a}(f_k,\theta)^H\mathbf{U}_n(f_k)\mathbf{U}_n^H(f_k)\mathbf{a}(f_k,\theta)}$$

$$\mathbf{a}(f_k,\theta)=\left[1,e^{-j2\pi f_k d\sin(\theta)/c},\ldots,e^{-j2\pi f_k(P-1)d\sin(\theta)/c}\right]^T$$

语音信号处理-波达方向估计

传统阵列算法介绍：宽带信号



宽带MUSIC算法

语音信号处理-波达方向估计

传统阵列算法介绍：宽带信号

相干子空间方法

$$\mathbf{T}_j\mathbf{A}_j(\boldsymbol{\theta})=\mathbf{A}_0(\boldsymbol{\theta}) \qquad j=1,\ldots,K$$

$$\mathbf{Y}_j=\mathbf{T}_j\mathbf{X}_j \qquad j=1,\ldots,K$$

$$\mathbf{R}_{\mathbf{Y}}(f_j)=E\left[\mathbf{Y}_j\mathbf{Y}_j^H\right]=\mathbf{T}_j\mathbf{R}_{\mathbf{X}}(f_j)\mathbf{T}_j^H=\mathbf{T}_j\mathbf{A}_j\mathbf{R}_{\mathbf{S}}(f_j)\mathbf{A}_j^H\mathbf{T}_j^H+\sigma^2\mathbf{T}_j\mathbf{T}_j^H$$

$$=\mathbf{A}_0\mathbf{R}_{\mathbf{S}}(f_j)\mathbf{A}_0^H+\sigma^2\mathbf{T}_j\mathbf{T}_j^H$$

RSS算法

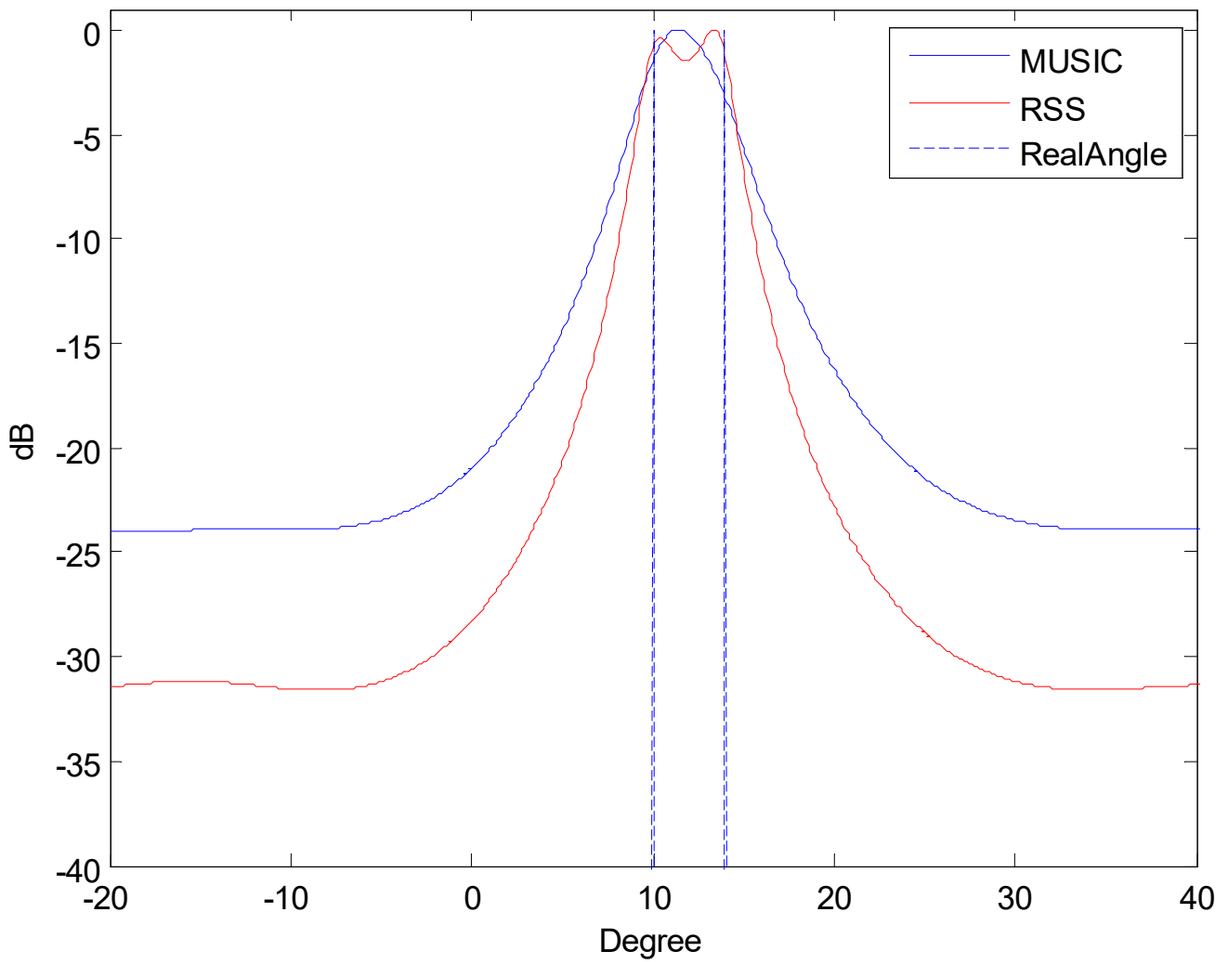
$$\begin{cases}\min\limits_{\mathbf{T}_j}\left\|\mathbf{A}_0(\boldsymbol{\beta})-\mathbf{T}_j\mathbf{A}_j(\boldsymbol{\beta})\right\|_F\\ s.t.\ \ \mathbf{T}_j^H\mathbf{T}_j=\mathbf{I}\end{cases}$$

$\mathbf{T}_j=\mathbf{V}_j\mathbf{U}_j^H$ 其中矩阵 $\mathbf{V}_j$ 和 $\mathbf{U}_j$ 的列分别为矩阵 $\mathbf{A}_j\mathbf{A}_0^H$ 的左奇异向量和右奇异向量

语音信号处理-波达方向估计

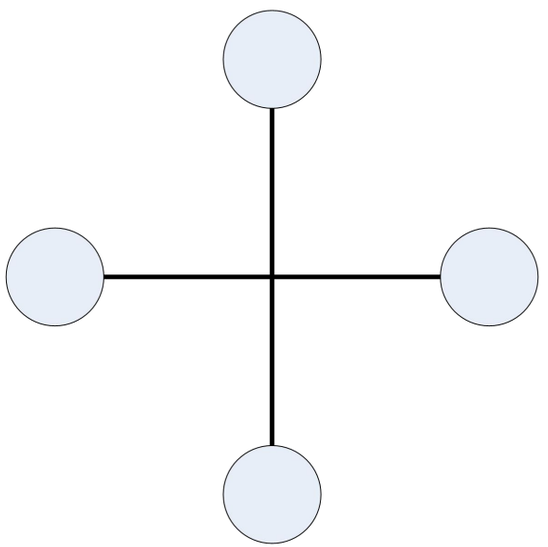
传统阵列算法介绍：宽带信号

相干子空间方法



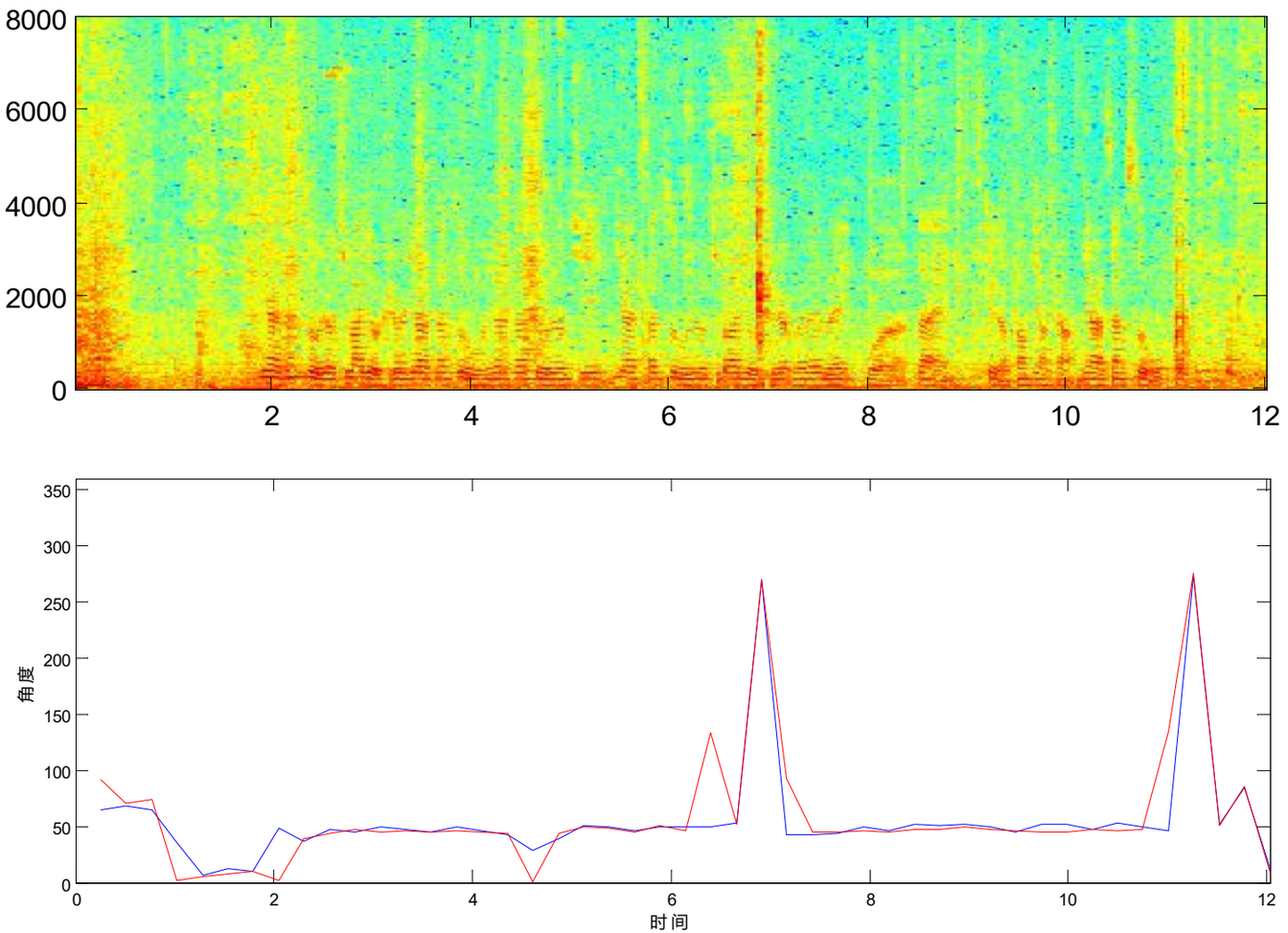
语音信号处理-波达方向估计

传统阵列算法介绍：宽带信号



语音信号处理-波达方向估计

传统阵列算法介绍：宽带信号



语音信号处理-波达方向估计

传统阵列算法介绍：GCC算法，波束形成算法，特征子空间算法

GCC算法

$$x_1(t) = s(t) + n(t)$$
$$x_2(t) = s(t - \tau_{12}) + n(t)$$

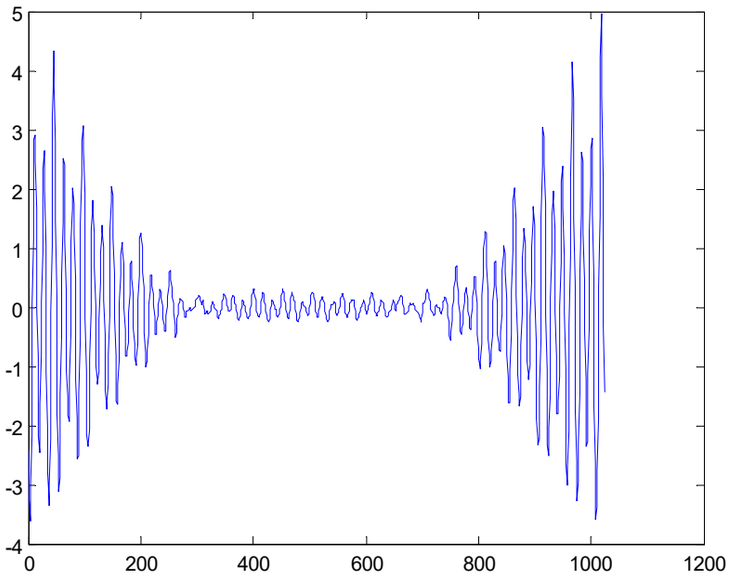
$$R_{12}(\tau) = \int_{-\infty}^{+\infty} x_1(t) x_2(t+\tau)$$

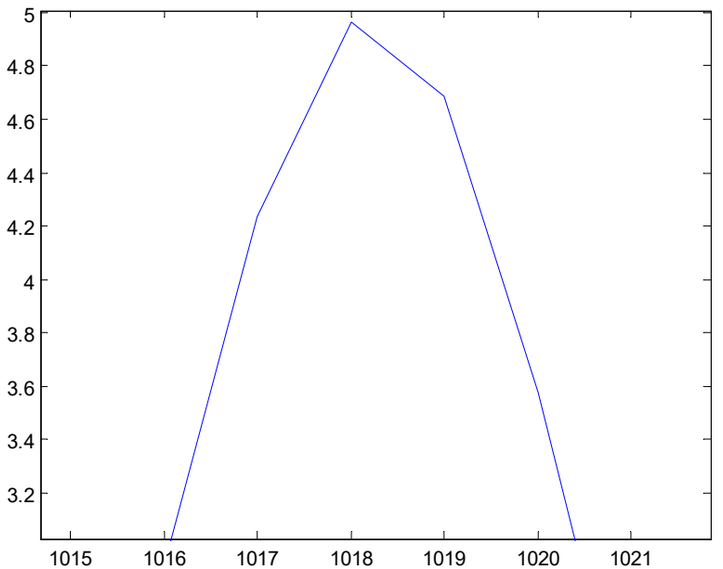
$$\hat{\tau}_{12} = \arg\max_{\tau \in D} R_{12}(\tau)$$

$$R_{12}(\tau) = IFFT\left(FFT(x_1(t)) * FFT(x_1(t))^*\right)$$

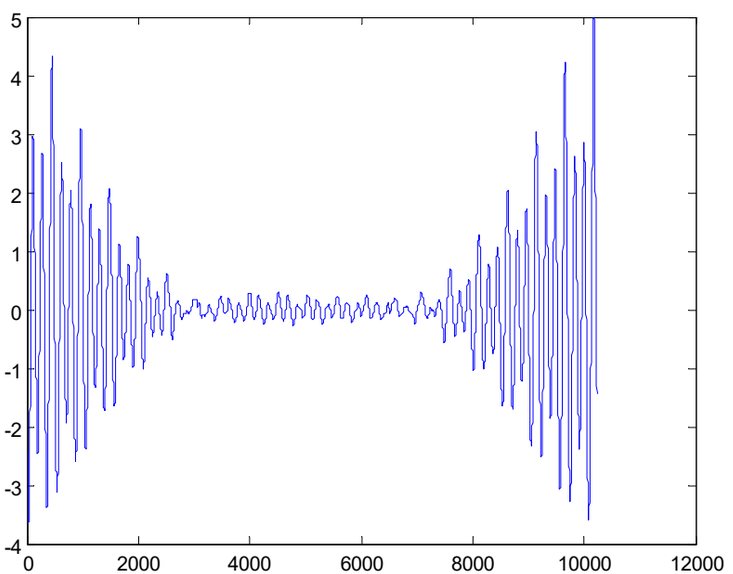
$$\begin{bmatrix} (x_1 - x_2)/c, (y_1 - y_2)/c \\ (x_2 - x_3)/c, (y_2 - y_3)/c \\ \vdots \\ (x_{P-1} - x_P)/c, (y_{P-1} - y_P)/c \end{bmatrix} \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} = \begin{bmatrix} \tau_{12} \\ \tau_{23} \\ \vdots \\ \tau_{(P-1)P} \end{bmatrix}$$

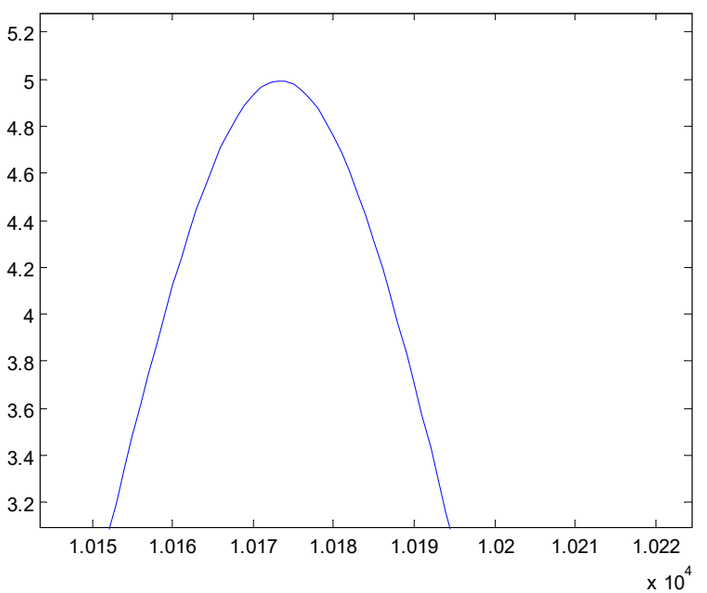
语音信号处理-波达方向估计





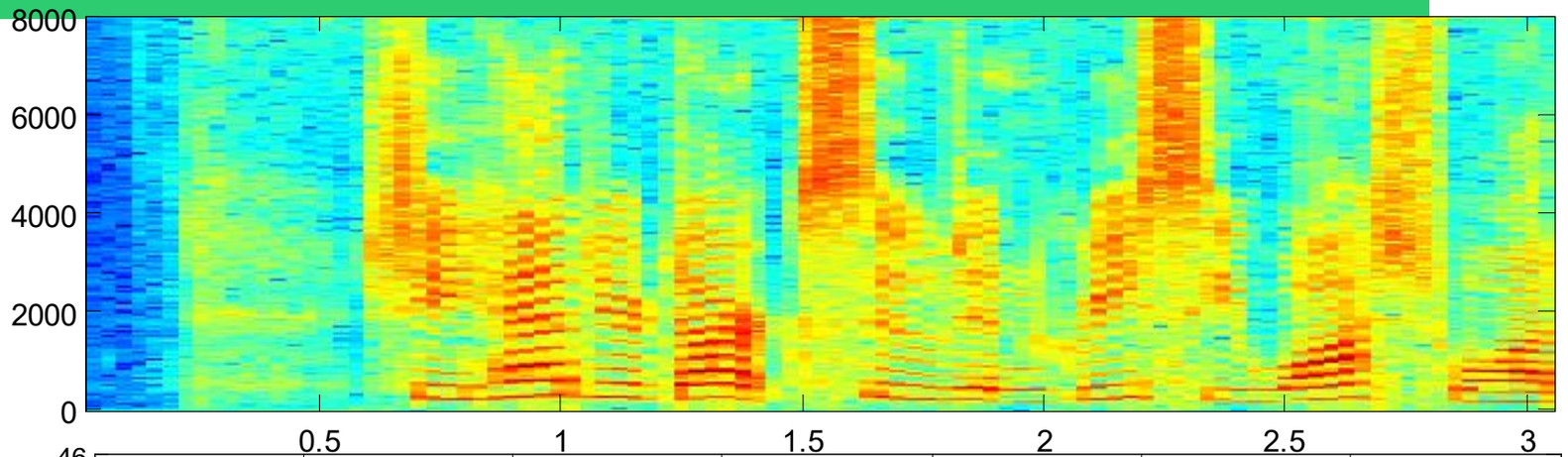
1017

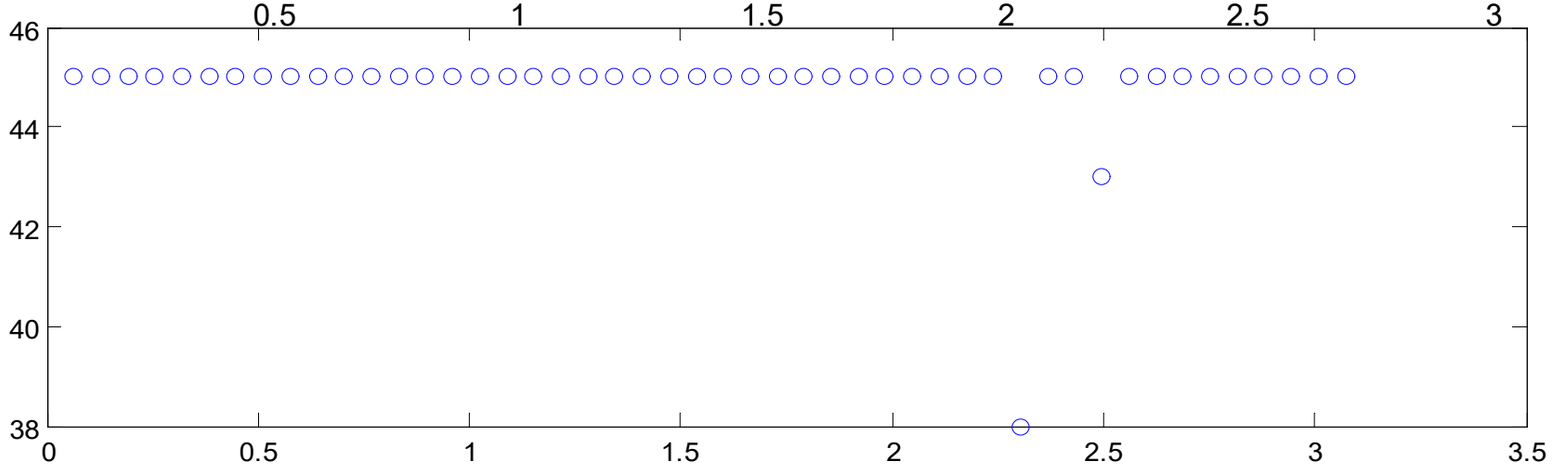


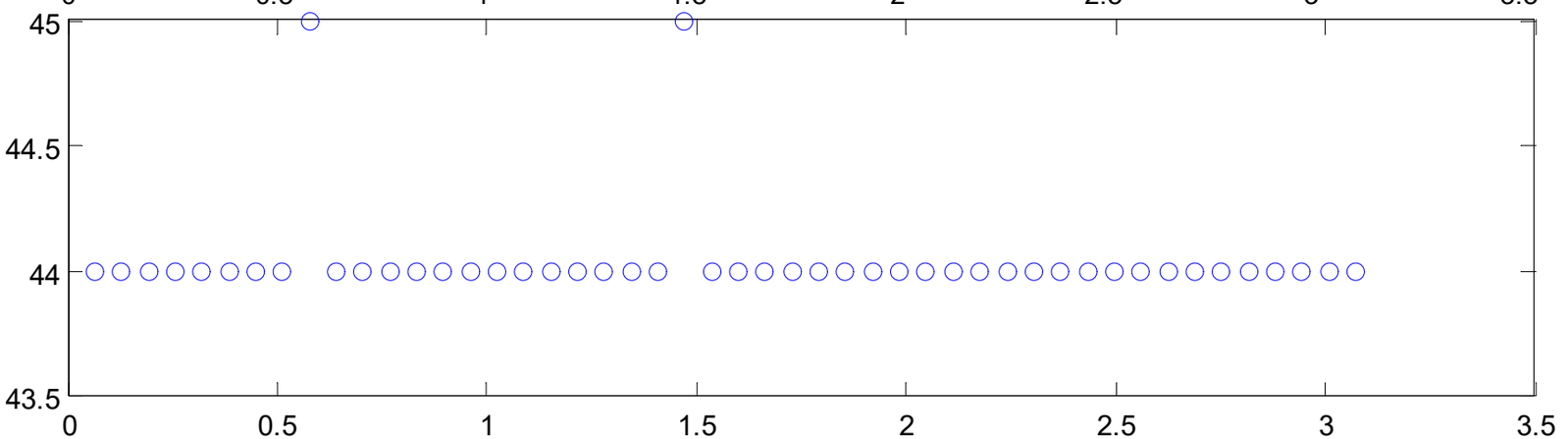


1017.3

语音信号处理-波达方向估计







语音信号处理-波达方向估计

传统阵列算法介绍：GCC算法，波束形成算法，特征子空间算法

GCC改进算法

$$R_{12}(\tau) = IFFT\left(G(w)X_1(\omega)X_2^*(\omega)\right)$$

Roth算法

$$G(\omega) = \frac{1}{S_1(\omega)}$$

SCOT算法

$$G(\omega) = \frac{1}{\sqrt{S_1(\omega)S_2(\omega)}}$$

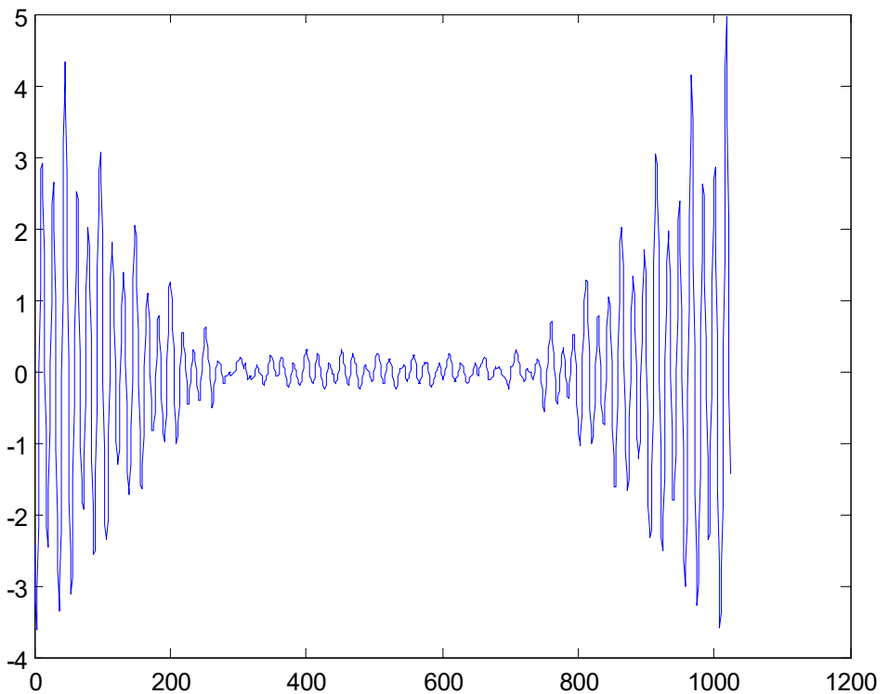
PHAT算法

$$G(\omega) = \frac{1}{|S_{12}(\omega)|}$$

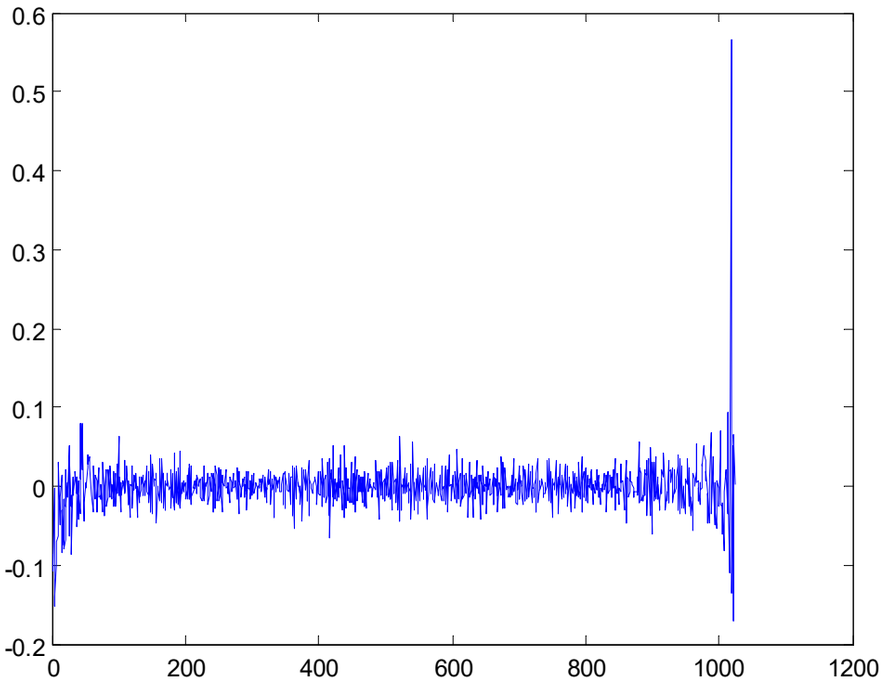
ML算法

$$G(\omega) = \frac{|\gamma_{12}|^2}{|S_{12}(\omega)|\left(1-|\gamma_{12}|^2\right)}$$

语音信号处理-波达方向估计



GCC



PHAT

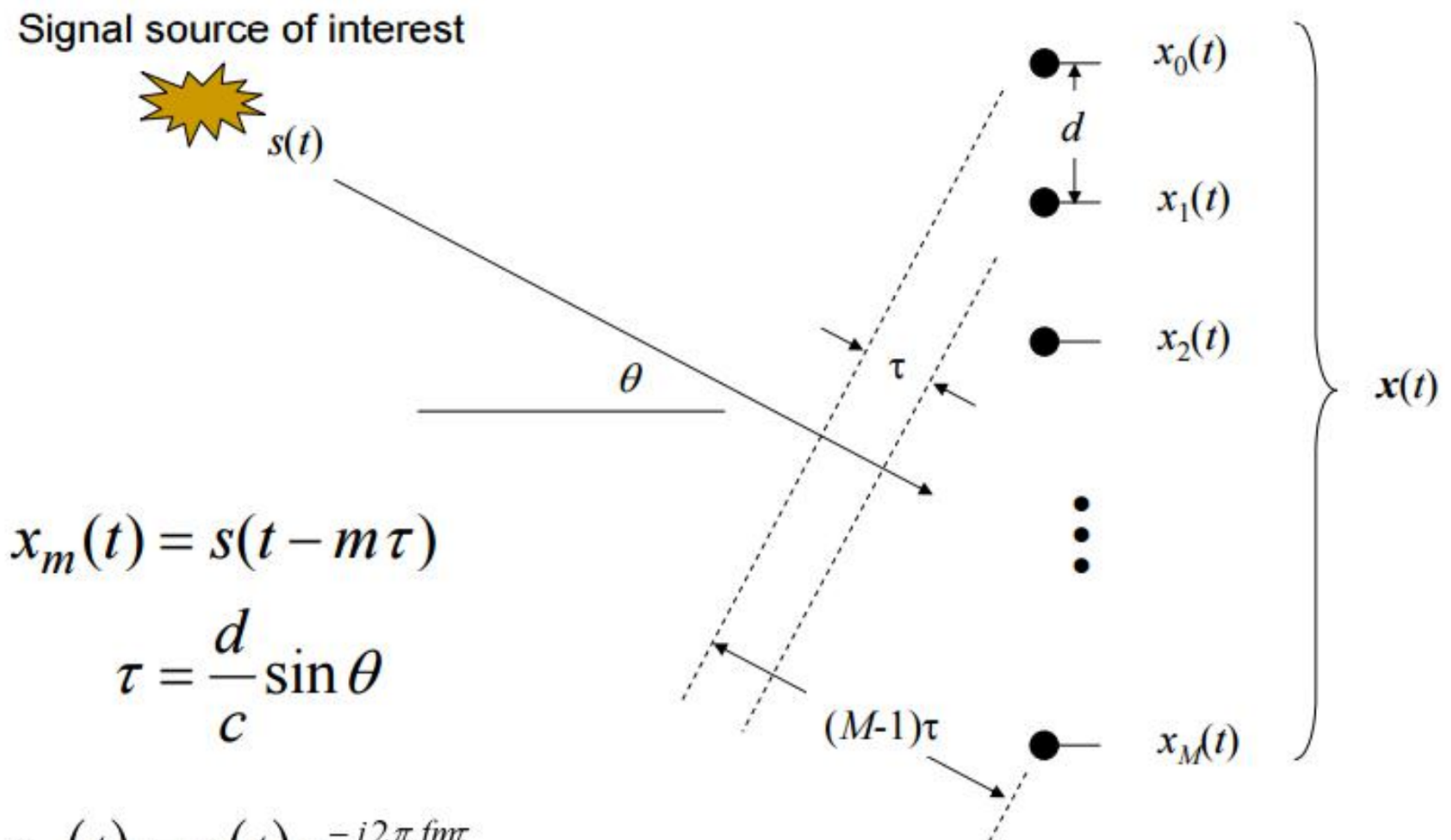
语音信号处理-波达方向估计

总结：

- 提高分辨率
- 降低估计误差
- 增加阵列的自由度
- 阵列自校准
- DOA估计的鲁棒性

语音信号处理-波束形成

语音信号处理-波束形成

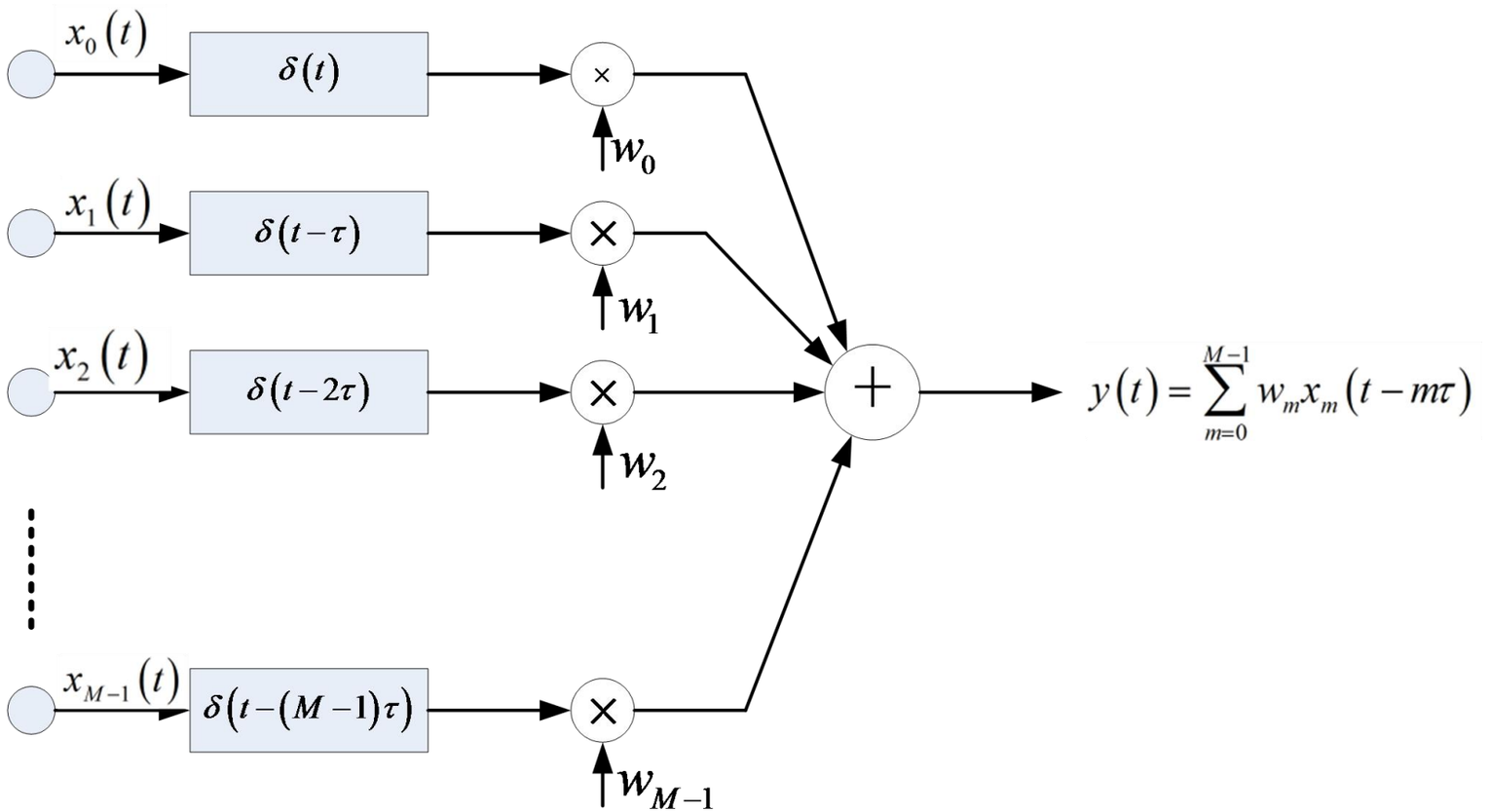


$$x_m(t) = s(t - m\tau)$$

$$\tau = \frac{d}{c}\sin\theta$$

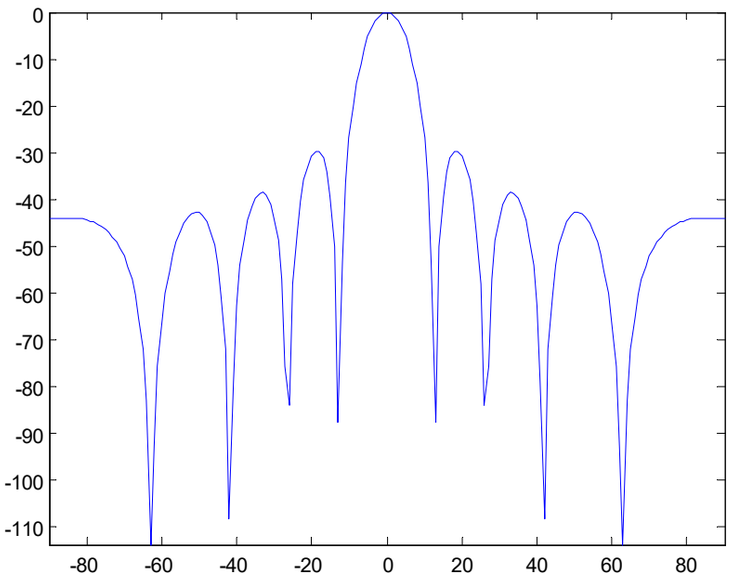
$$x_m(t) = s(t)e^{-j2\pi fm\tau}$$

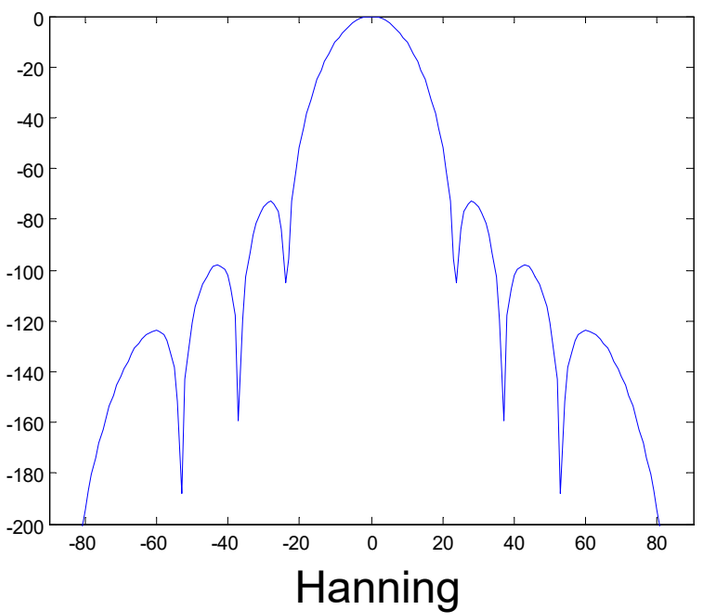
语音信号处理-波束形成

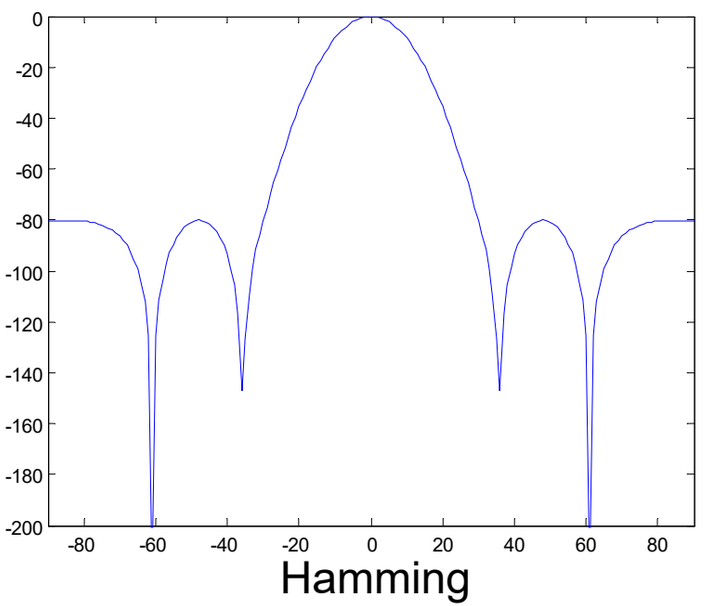


- 改变 τ 的值可以改变波束的指向
- W是加权值，可以改变旁瓣的高度
- 波束形成与频域滤波原理一样

语音信号处理-波束形成







语音信号处理-波束形成

MVDR算法

$$x_p(t)=\sum_{m=1}^{M}s_m(t)e^{-i2\pi f\tau_{mp}}+w_p(t)\quad p=1,2,...,P$$

$$\mathbf{x}(t)=\mathbf{A}\mathbf{s}(t)+\mathbf{w}(t)$$

$$\mathbf{x}(t)=\left[x_1(t),x_2(t),...,x_P(t)\right]^T$$

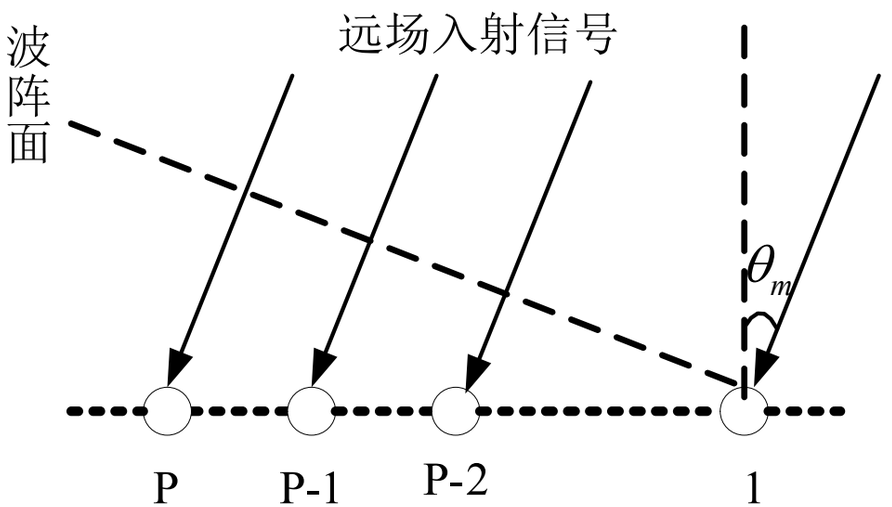
$$\mathbf{A}=\left[\mathbf{a}_1,\mathbf{a}_2,...,\mathbf{a}_M\right]$$

$$\mathbf{a}_m=\left[e^{-i2\pi f\tau_{1m}},e^{-i2\pi f\tau_{2m}},...,e^{-i2\pi f\tau_{Pm}}\right]^T$$

$$\mathbf{R}_{\mathbf{x}}=\mathrm{E}\left\{\mathbf{x}(t)\mathbf{x}^H(t)\right\}=\mathbf{A}\,\mathrm{E}\left\{\mathbf{s}(t)\mathbf{s}^H(t)\right\}\mathbf{A}^H+\mathrm{E}\left\{\mathbf{w}(t)\mathbf{w}^H(t)\right\}$$

$$=\mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A}^H+\mathbf{R}_{\mathbf{w}}$$

$$\mathbf{R}_{\mathbf{x}}=\mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A}^H+\sigma_{\mathbf{w}}^2\mathbf{I}$$

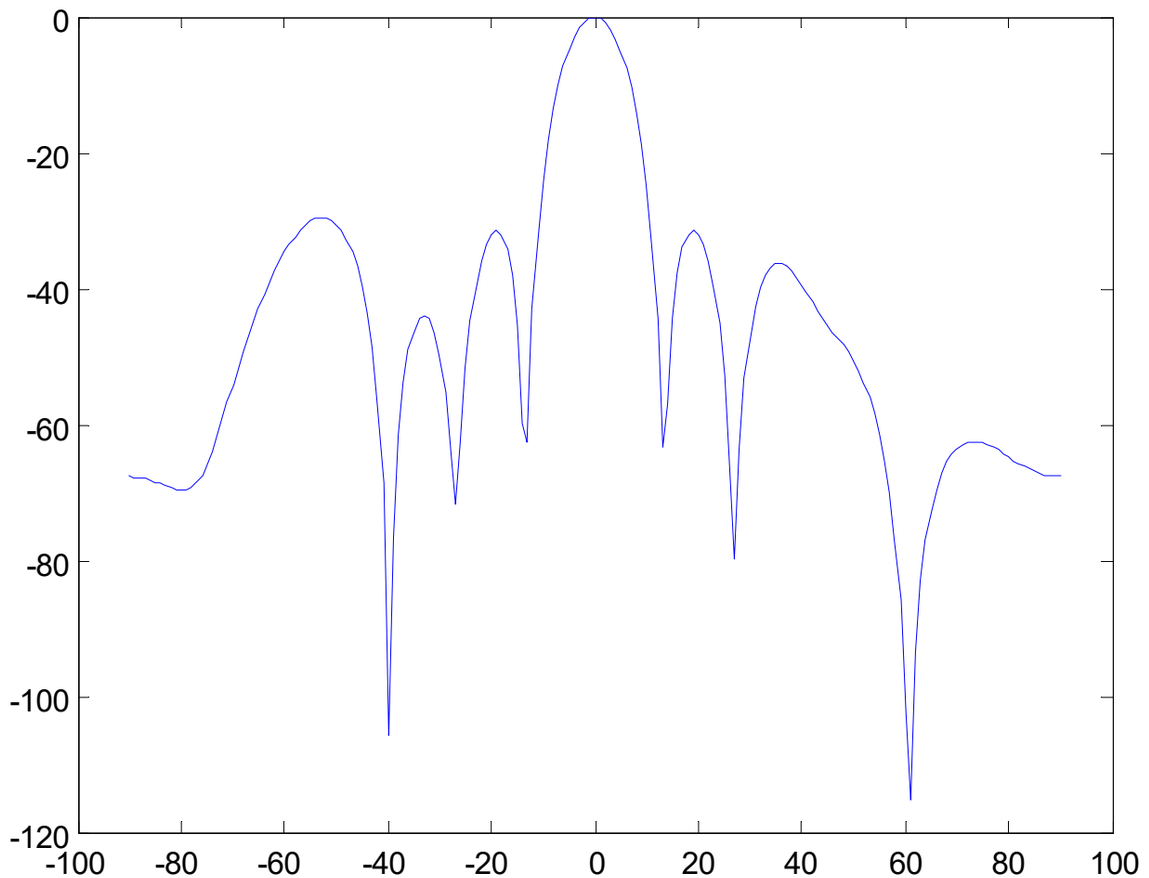


$$\min\mathbf{w}^H\mathbf{R}_{\mathbf{x}}\mathbf{w}\qquad s.t.\ \mathbf{w}^H\mathbf{a}(\theta_s)$$

$$\mathbf{w}=\frac{\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}(\theta_s)}{\mathbf{a}(\theta_s)^H\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}(\theta_s)}$$

语音信号处理-波束形成

MVDR算法



语音信号处理-波束形成

MVDR算法

对角加载法

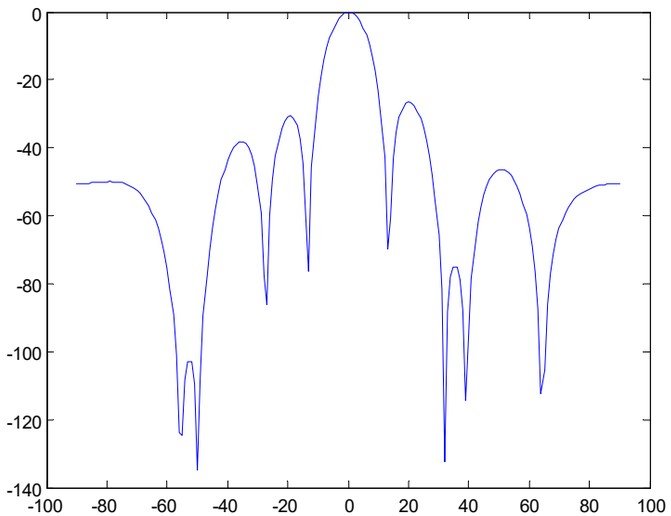
$$\mathbf{w}=\frac{\left(\mathbf{R}_{\mathbf{x}}+\sigma^{2}\mathbf{I}\right)^{-1}\mathbf{a}\left(\theta_{s}\right)}{\mathbf{a}\left(\theta_{s}\right)^{H}\left(\mathbf{R}_{\mathbf{x}}+\sigma^{2}\mathbf{I}\right)^{-1}\mathbf{a}\left(\theta_{s}\right)}$$

- 加载的能量越大，滤波器越稳定，但噪声抑制能力变差
- 极限情况，加载的能量无穷大，则退化到原始波束形成

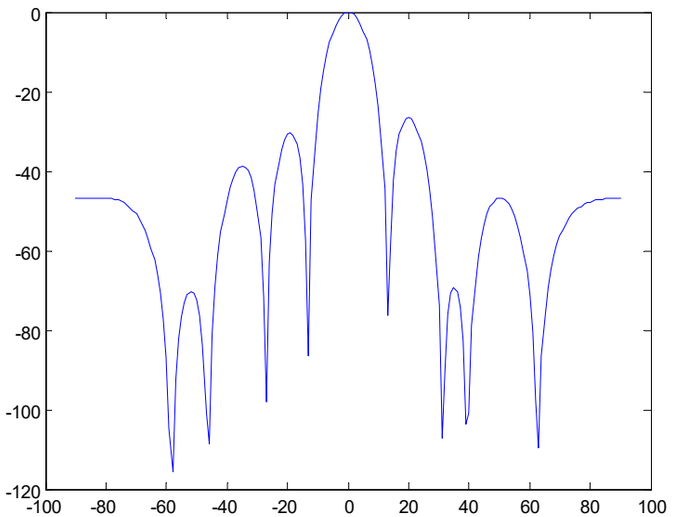
语音信号处理-波束形成

MVDR算法

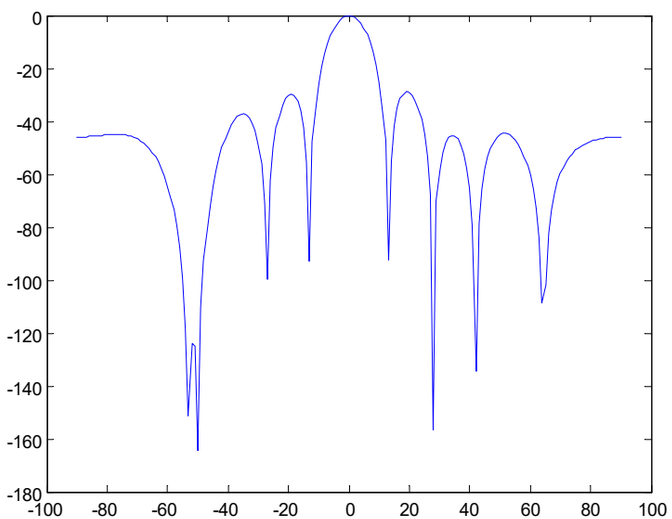
对角加载法



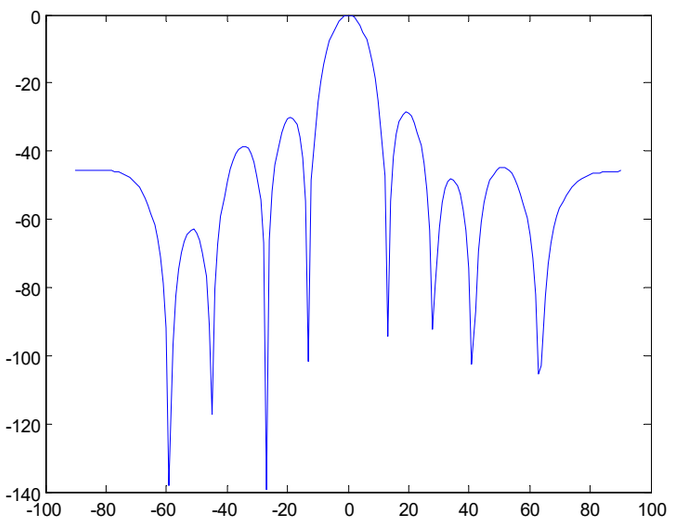
10dB



13dB



16dB



20dB

语音信号处理-波束形成

超指向波束形成

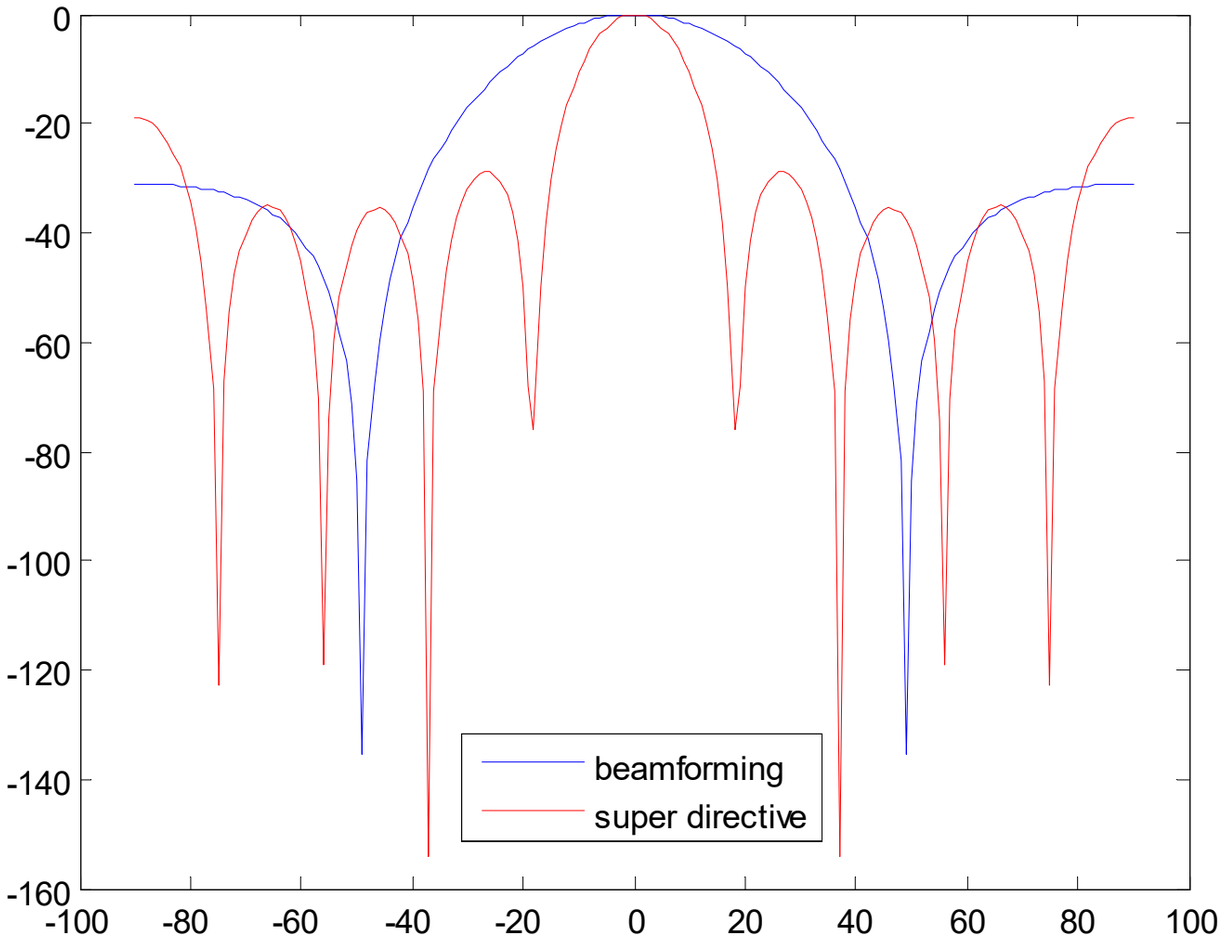
$$\mathbf{w} = \frac{\Gamma_{\mathbf{v}}^{-1}\mathbf{a}(\theta_s)}{\mathbf{a}(\theta_s)^H \Gamma_{\mathbf{v}}^{-1}\mathbf{a}(\theta_s)}$$

$$\Gamma_{\mathbf{v}} = \begin{bmatrix} 1 & \Gamma_{v_1v_2} & \Gamma_{v_1v_3} & \cdots & \Gamma_{v_1v_P} \\ \Gamma_{v_2v_1} & 1 & \Gamma_{v_2v_3} & \cdots & \Gamma_{v_2v_P} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Gamma_{v_Pv_1} & \Gamma_{v_Pv_2} & \Gamma_{v_Pv_3} & \cdots & 1 \end{bmatrix}$$

$$\Gamma_{v_nv_m} = \frac{\sin(2\pi fl_{mn}/c)}{2\pi fl_{mn}/c}$$

语音信号处理-波束形成

超指向波束形成



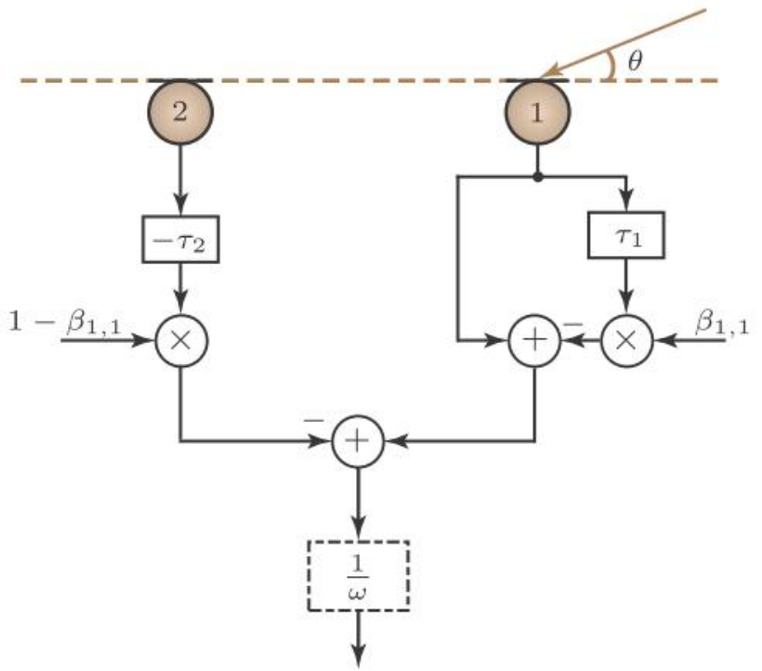
Beamforming_gain = 0.12

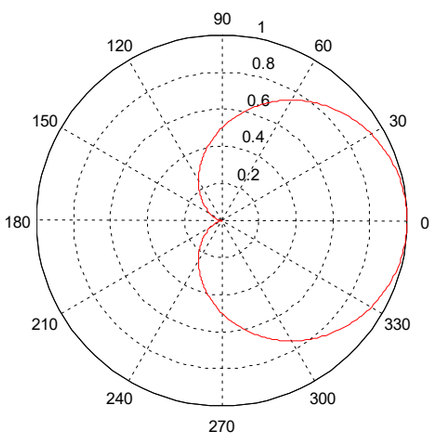
SuperDirective_gain = 4.3557e+07

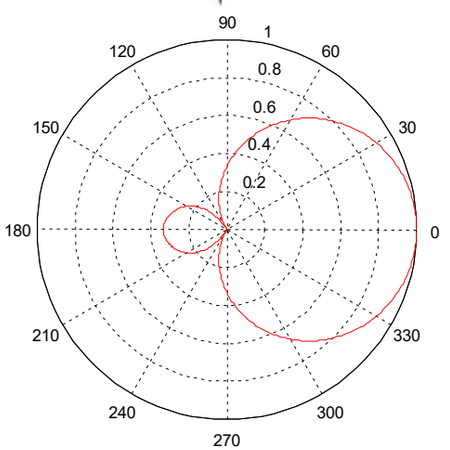
语音信号处理-波束形成

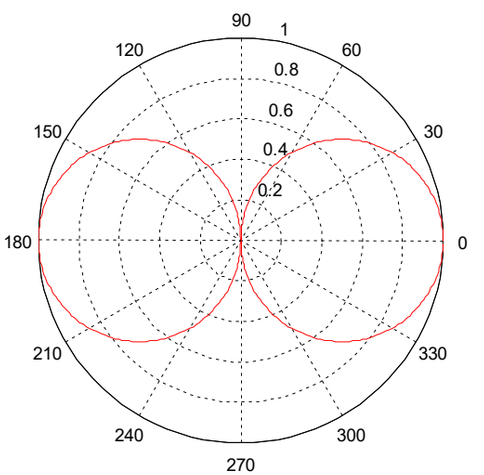
差分阵列

一阶差分



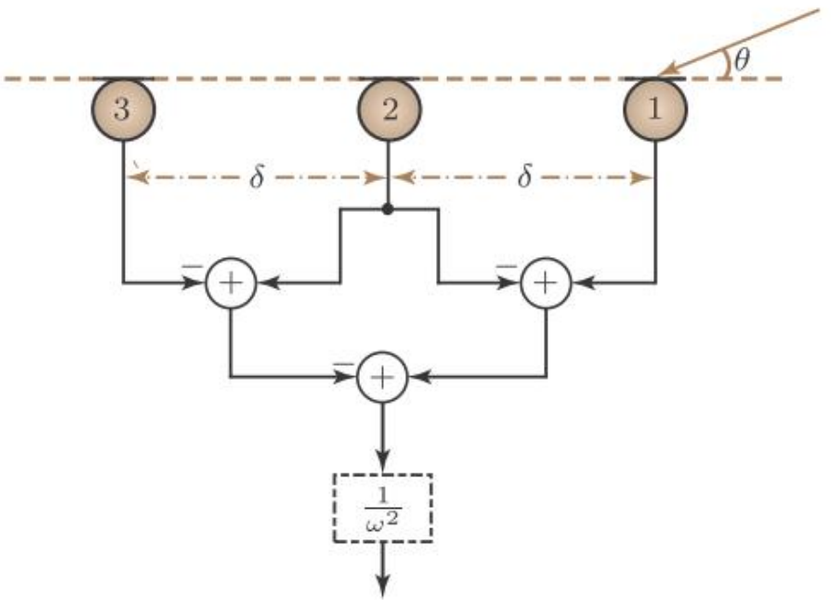


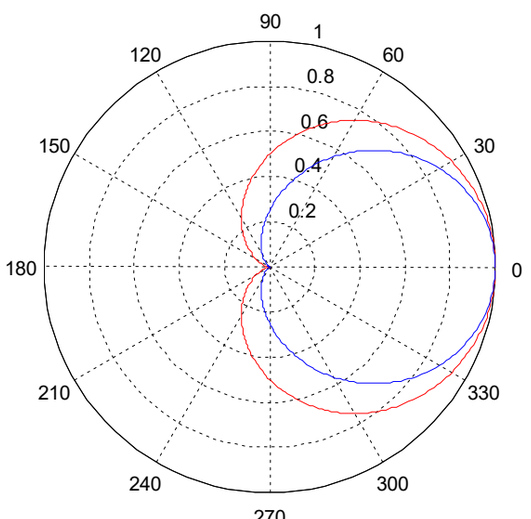


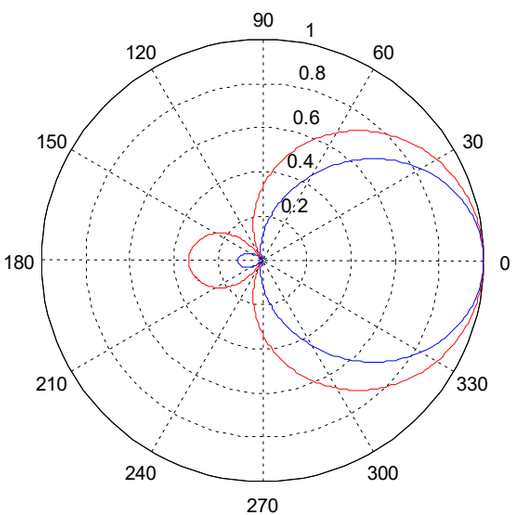


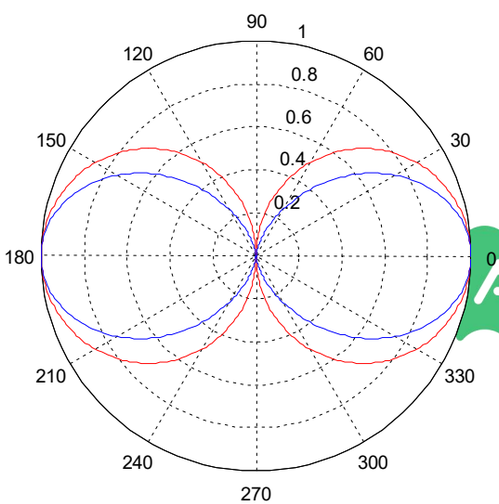
语音信号处理-波束形成

差分阵列
二阶差分



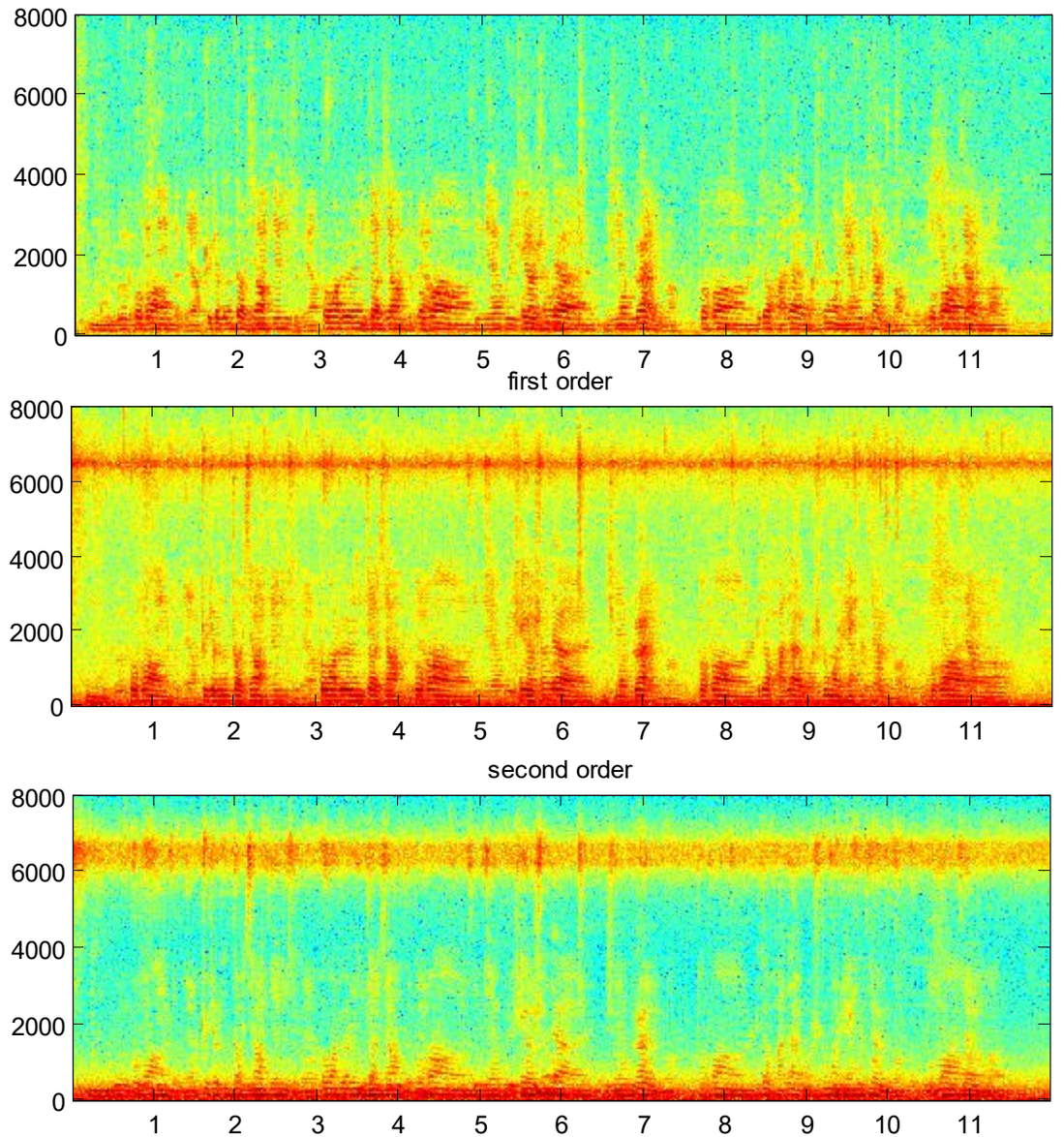






语音信号处理-波束形成

差分阵列



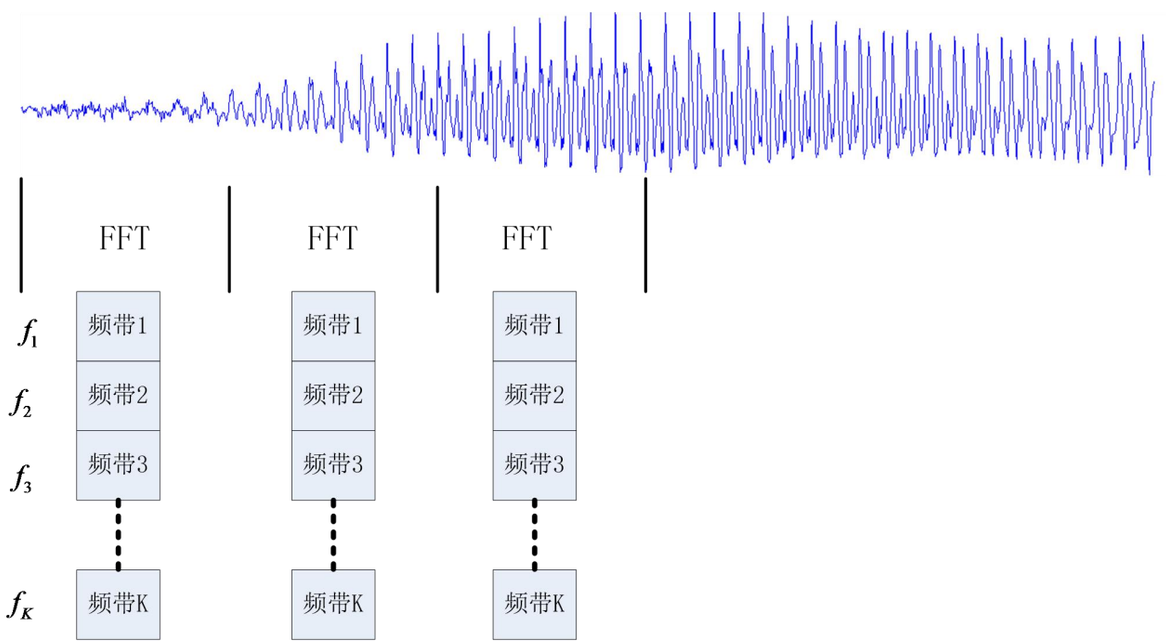
语音信号处理-波束形成

宽带信号处理

$$x_p(t)=\sum_{m=1}^{M}s_m(t-\tau_{mp})+w_p(t)\quad p=1,2,...,P$$

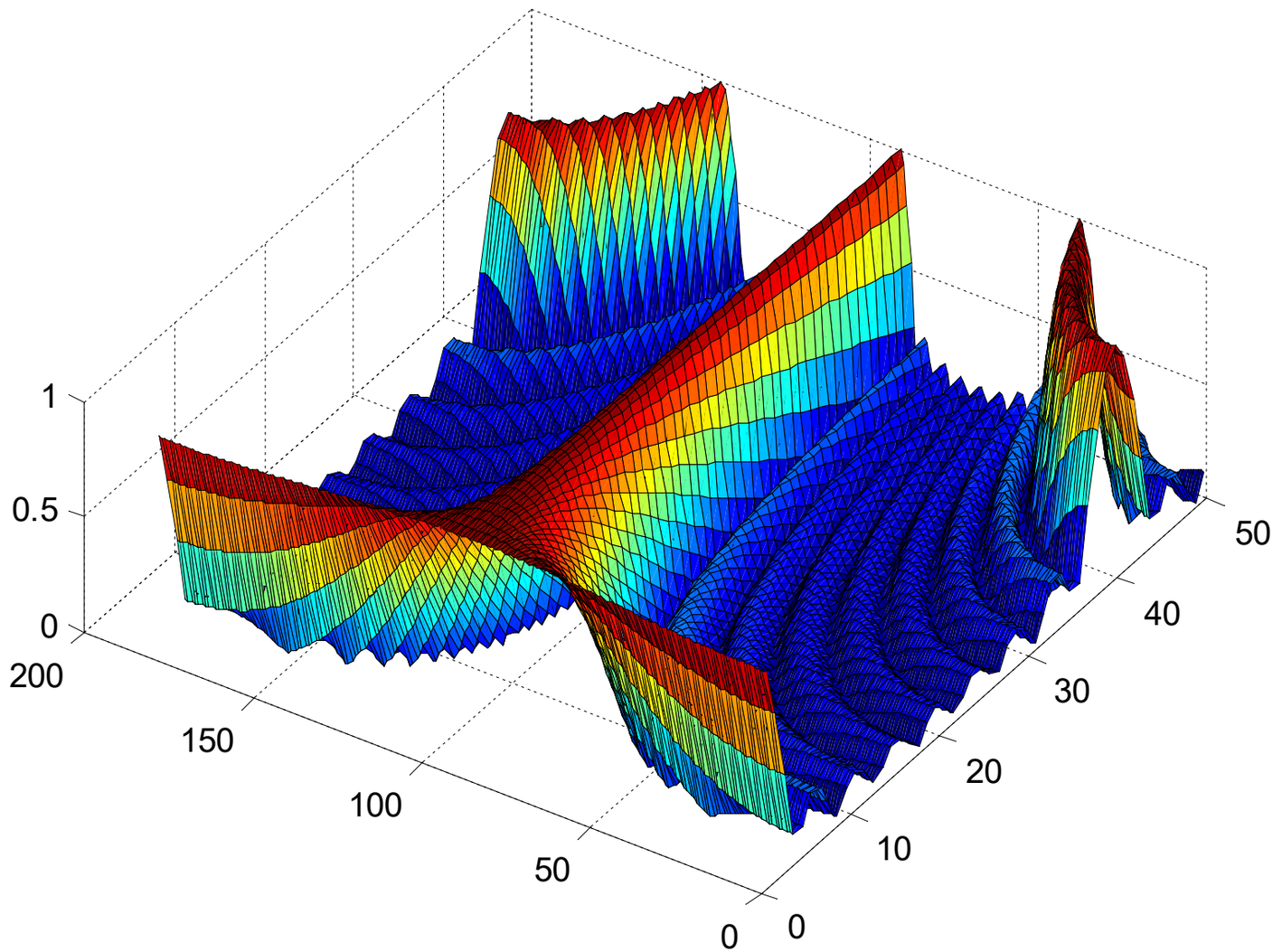
$$X_p(f_j)=\sum_{m=1}^{M}S_m(f_j)e^{-i2\pi f_j\tau_{mp}}+W_p(f_j)\qquad p=1,2,...,P$$

$$\mathbf{X}(f_j)=\mathbf{A}(f_j)\mathbf{S}(f_j)+\mathbf{W}(f_j)$$



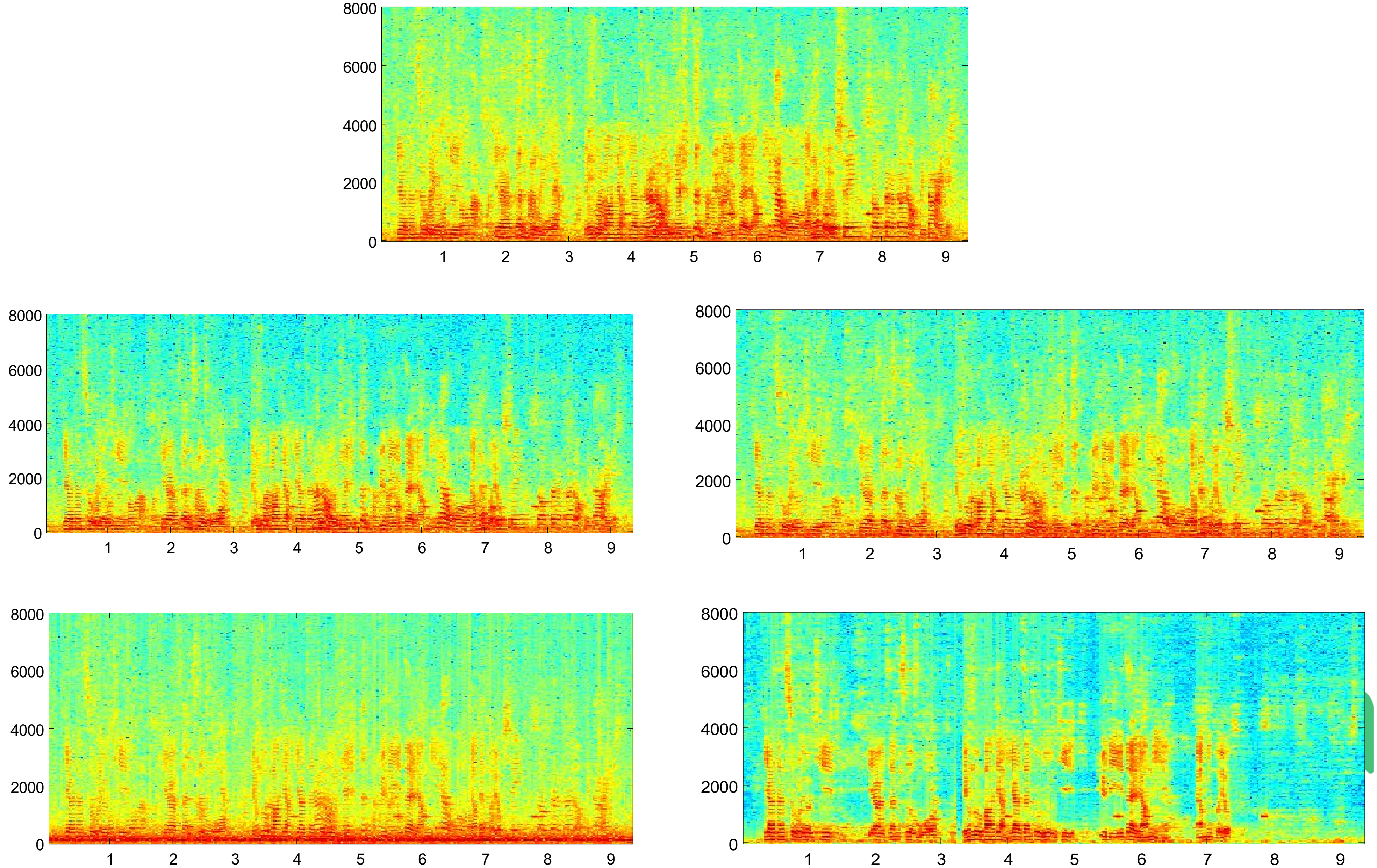
语音信号处理-波束形成

宽带信号处理



语音信号处理-波束形成

宽带信号处理



语音信号处理-波束形成

多通道维纳滤波器

维纳滤波器准则

$$\mathbf{x}=\mathbf{a}(\theta)s+\mathbf{v}$$

$$\phi_{ee}(i)=E\left[\left(s-\mathbf{w}^{H}\mathbf{x}\right)\left(s-\mathbf{w}^{H}\mathbf{x}\right)^{*}\right]$$

$$\mathbf{w}=\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{r}_{\mathbf{x}s}$$

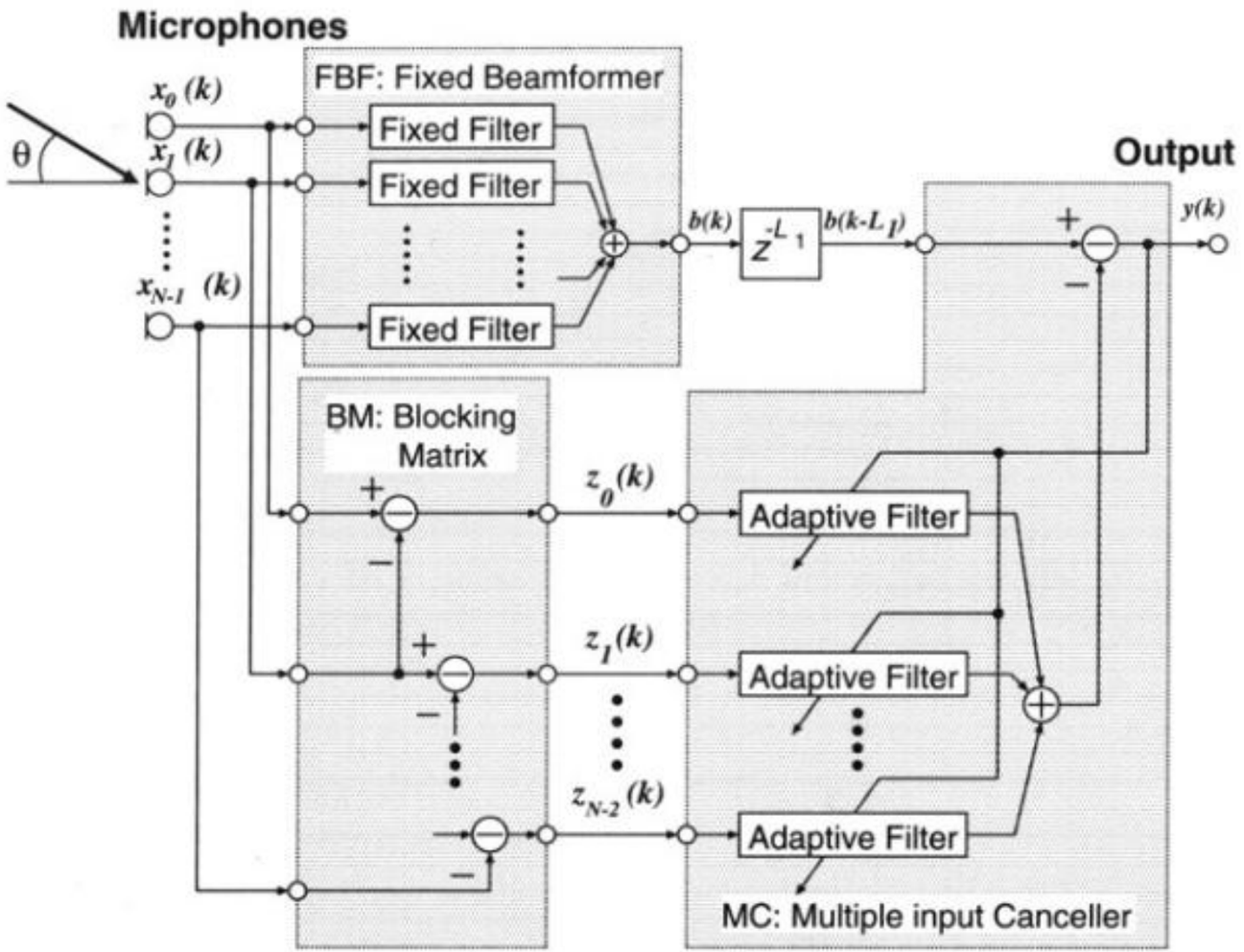
$$\mathbf{w}=\frac{\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}(\theta_s)}{\mathbf{a}(\theta_s)^{H}\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}(\theta_s)}\left[\frac{\phi_{s_0}}{\phi_{s_0}+\phi_{v_0}}\right]$$

多通道维纳滤波器等价于MVDR算法级联一个单通道语音增强

$$\left(\mathbf{R}_{\mathbf{x}}+\alpha\mathbf{x}\mathbf{x}^{T}\right)^{-1}=\mathbf{R}_{\mathbf{x}}^{-1}-\frac{\alpha\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{x}\mathbf{x}^{T}\mathbf{R}_{\mathbf{x}}^{-1}}{1+\alpha\mathbf{x}^{T}\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{x}}$$

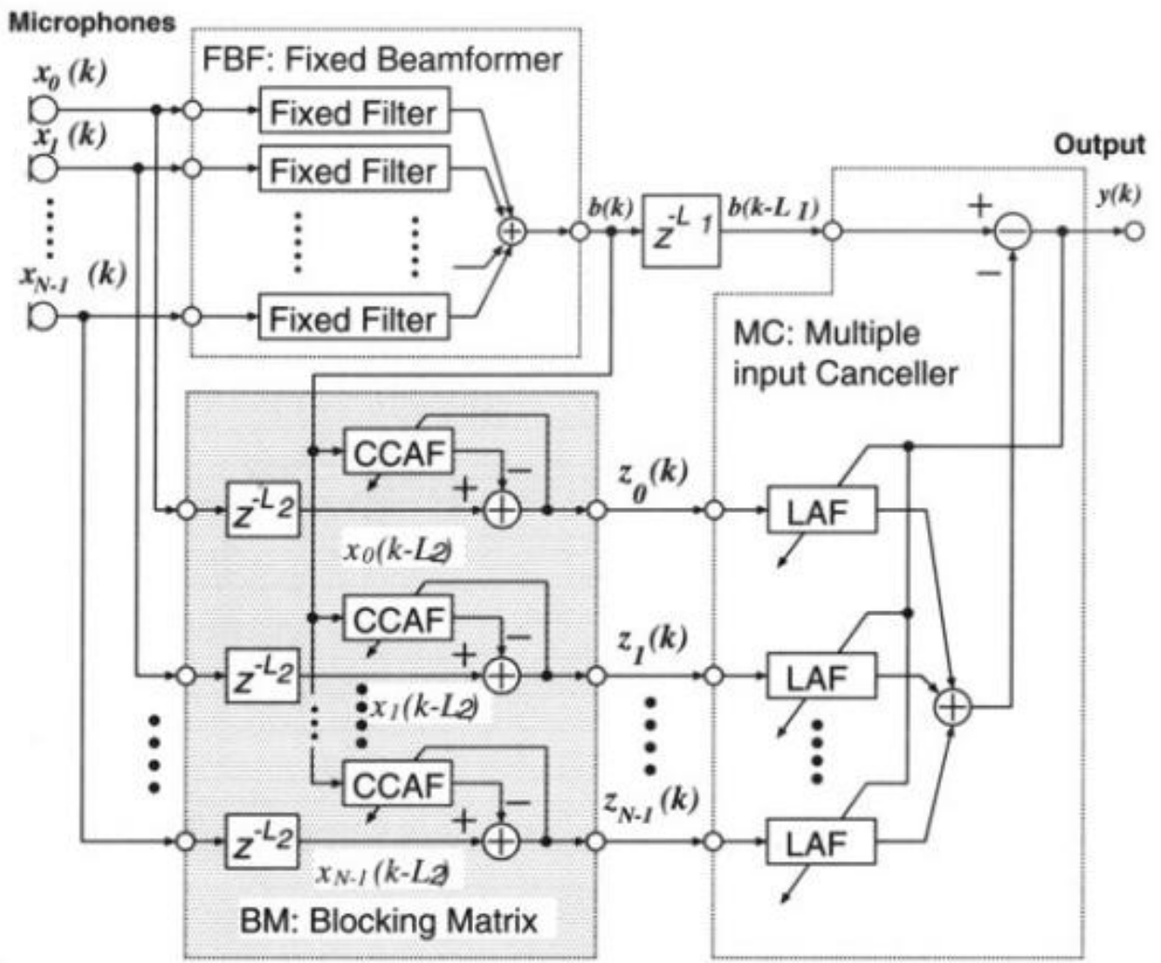
语音信号处理-波束形成

GSC算法



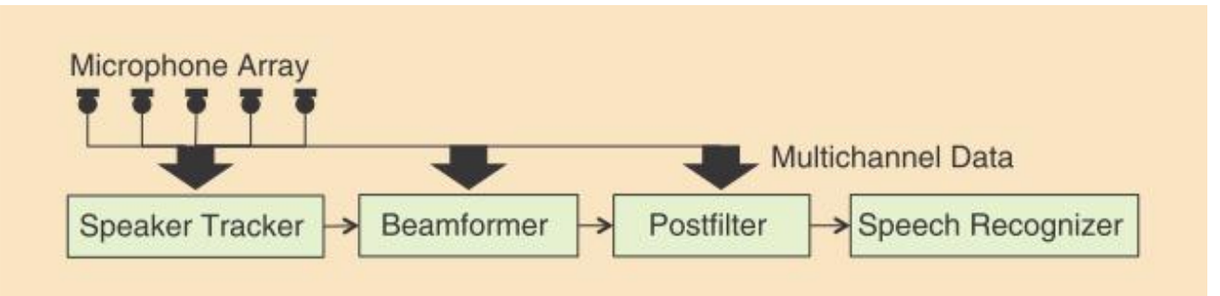
语音信号处理-波束形成

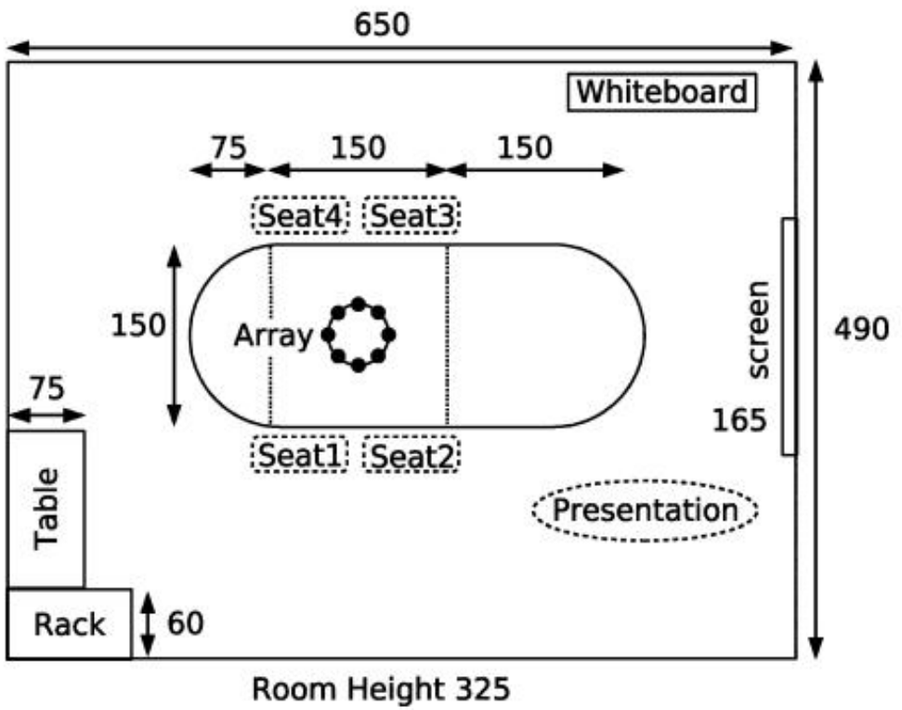
GSC算法
Robust Adaptive Beamforming



语音信号处理-波束形成

波束形成对识别率的影响





Beamforming Algorithm	Pass (%WER)			
	1	2	3	4
D&S BF	80.1	39.9	21.5	17.8
D&S BF with PF	79.0	38.1	20.2	16.5
MMSE BF	78.6	35.4	18.8	14.8
GEV BF	78.7	35.5	18.6	14.5
MN BF with Gamma pdf	75.6	34.9	19.8	15.8
MN BF with GG pdf (1)	75.1	32.7	16.5	13.2
MN BF with GG pdf (2)	79.0	37.2	20.0	16.7
SDM	87.0	57.1	32.8	28.0
CTM	52.9	21.5	9.8	6.7

语音信号处理-波束形成

波束形成算法总结

- 多通道维纳滤波器等效于MVDR算法级联单通道语音增强
- MVDR算法与RLS算法，GSC算法是等效的
- 计算量

MVDR：$O\left(P^3\right)$

RLS：$O\left(P^2\right)$

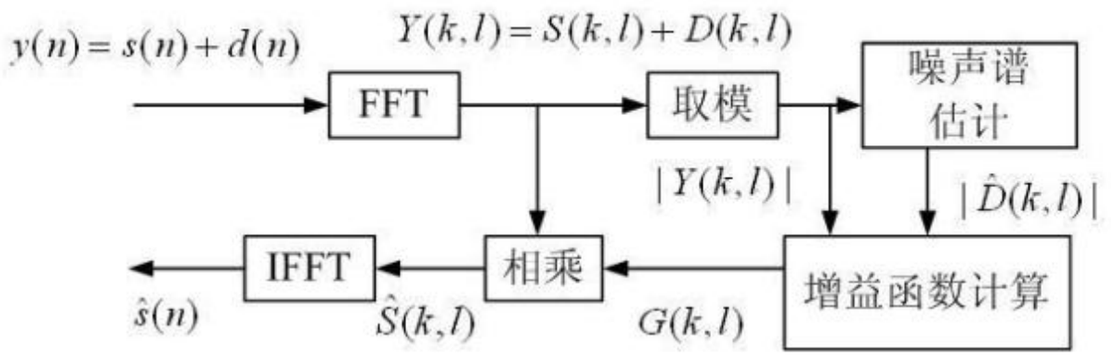
GSC：$O\left(P\right)$

语音信号处理-单通道语音增强

谱减法

$$y(n) = s(n) + d(n)$$

$$Y(k,l) = S(k,l) + D(k,l)$$



维纳滤波器 $E\left[\left(\left|\hat{S}(k,l)\right| - \left|\hat{S}(k,l)\right|\right)^2\right]$

$$G(k,l) = \frac{\lambda_s}{\lambda_s + \lambda_D}$$

语音信号处理-单通道语音增强

谱减法

$$Y(k,l)=S(k,l)+D(k,l)$$

维纳滤波器 $$E\left[\left(\left|\hat{S}(k,l)\right|-\left|\hat{S}(k,l)\right|\right)^2\right]$$

$$G(k,l)=\frac{\lambda_s}{\lambda_s+\lambda_D}=\frac{\xi(k,l)}{\xi(k,l)+1}$$

先验信噪比 $$\xi(k,l)=E\left[\frac{S(k,l)^2}{D(k,l)^2}\right]$$

后验信噪比 $$\gamma(k,l)=E\left[\frac{Y(k,l)^2}{D(k,l)^2}\right]$$

语音信号处理-单通道语音增强

谱减法

$$Y(k,l)=S(k,l)+D(k,l)$$

MMSE-SATA

$$G(k,l)=\Gamma(1.5)\frac{\sqrt{v(k,l)}}{\hat{\gamma}(k,l)}\exp\left(-v(k,l)\right)\left[(1+v(k,l))I_0\left(\frac{v(k,l)}{2}\right)+v(k,l)I_1\left(v(k,l)\right)\right]$$

MMSE-LSA

$$G(k,l)=\frac{\hat{\xi}_{DD}(k,l)}{1+\hat{\xi}_{DD}(k,l)}\exp\left[\frac{1}{2}\int_{v(k,l)}^{\infty}\frac{\exp(-t)}{t}dt\right]$$

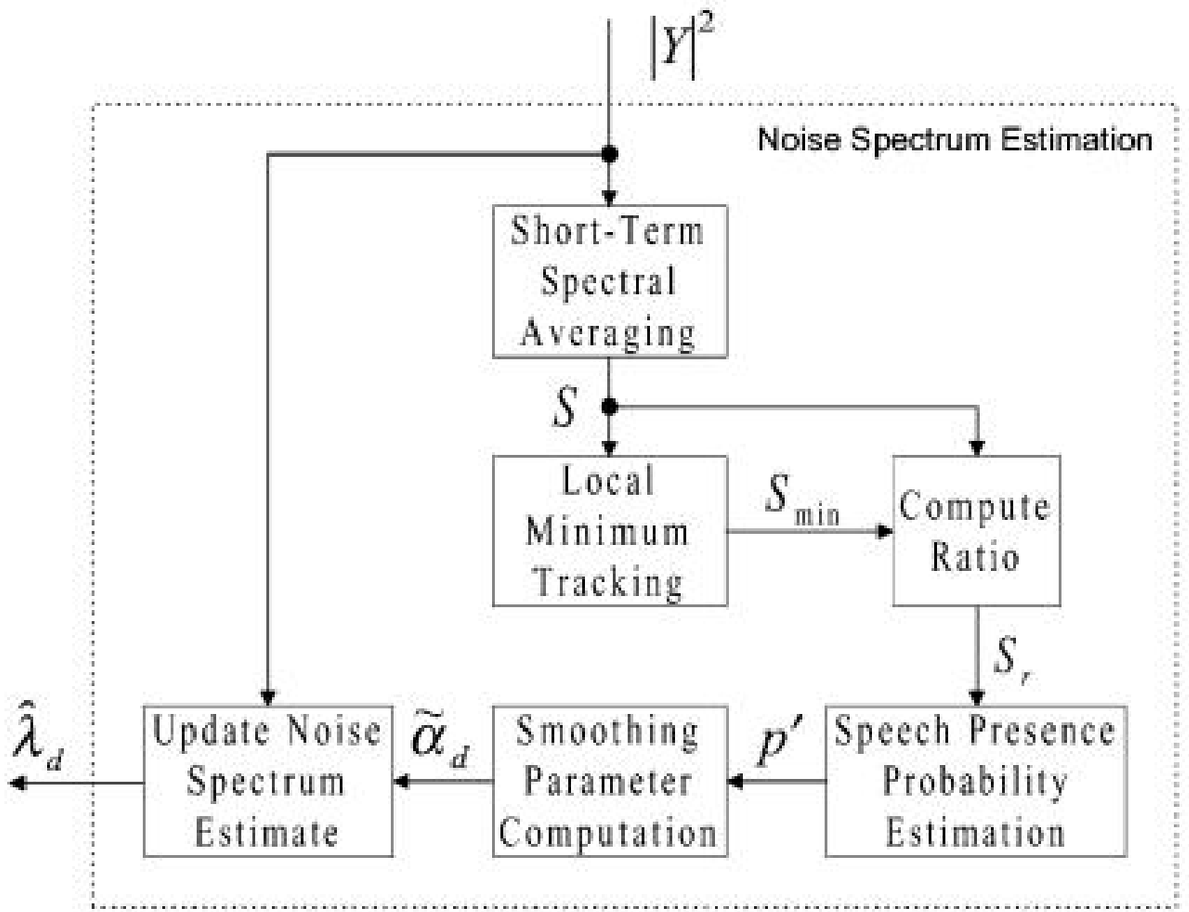
$$\hat{\xi}_{DD}(k,l)=\alpha_{DD}G^2(k,l-1)\hat{\gamma}(k,l-1)+(1-\alpha_{DD})\max\left\{\hat{\gamma}(k,l)-1,0\right\}$$

$$v(k,l)=\frac{\hat{\xi}_{DD}(k,l)\hat{\gamma}(k,l)}{1+\hat{\xi}_{DD}(k,l)}$$

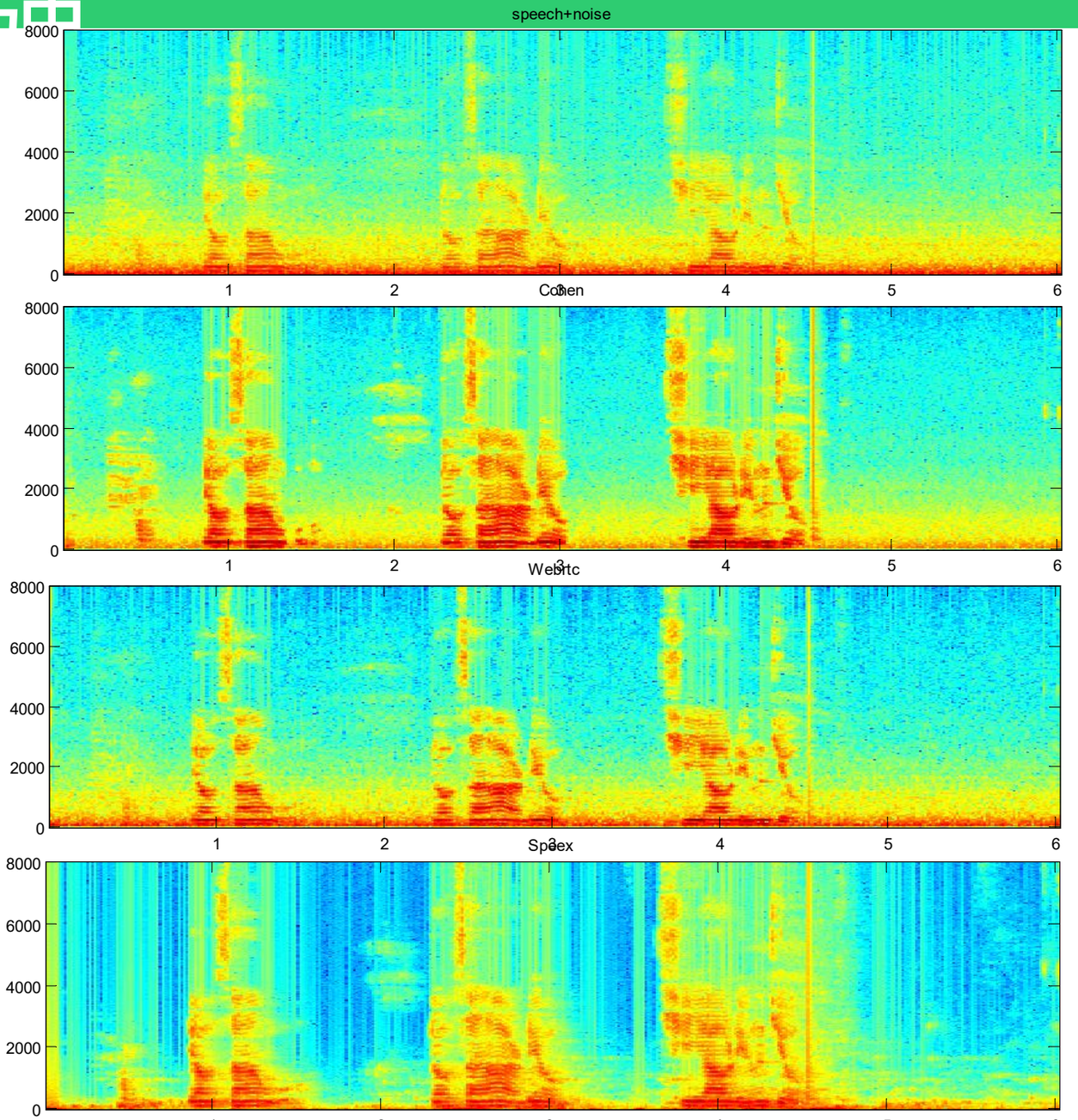
语音信号处理-单通道语音增强

谱减法

噪声谱估计，cohen，MCRA

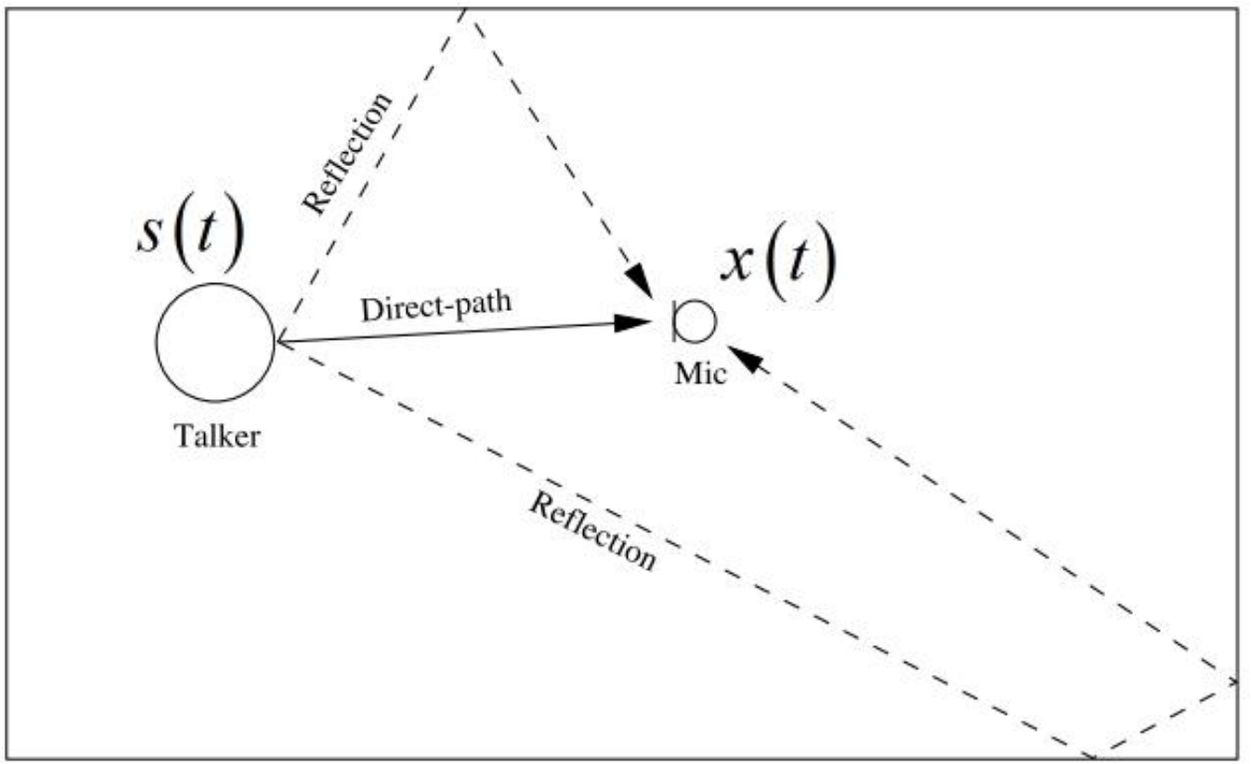


语音信号处理-单通道语音增强



语音信号处理-去混响

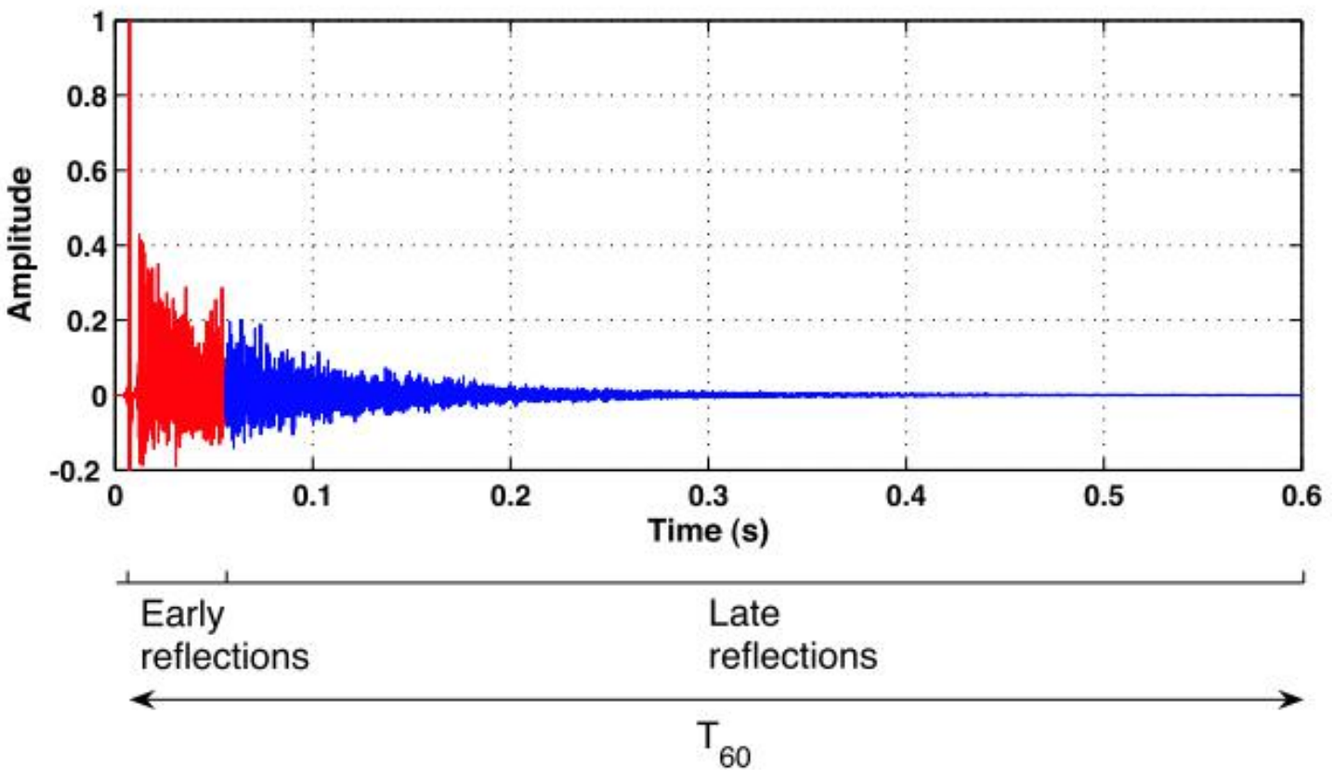
去混响算法



$$x(t) = s(t) * h(t) + v(t)$$

语音信号处理-去混响

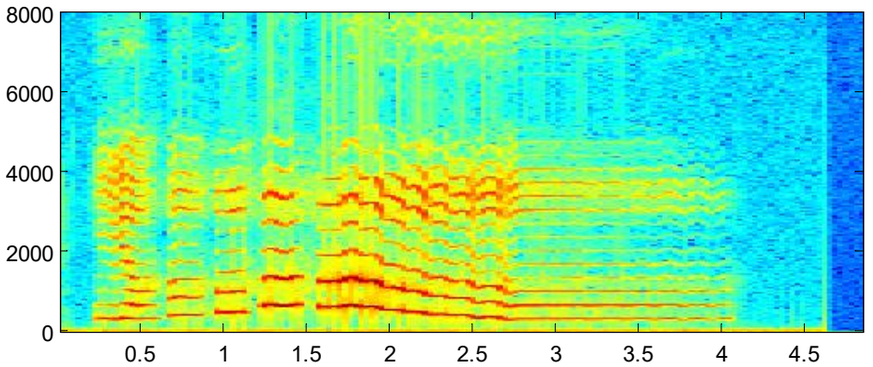
去混响算法

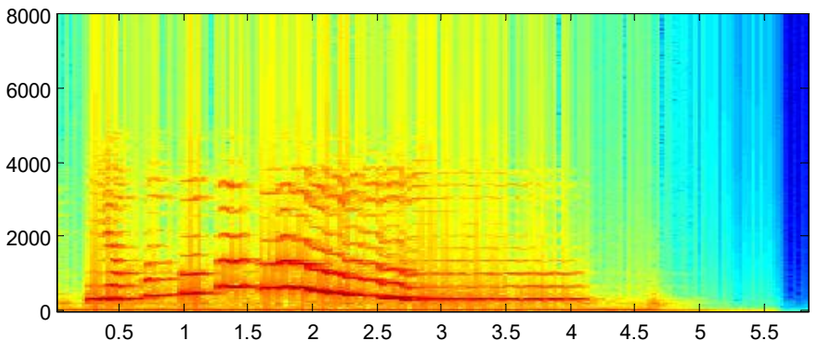


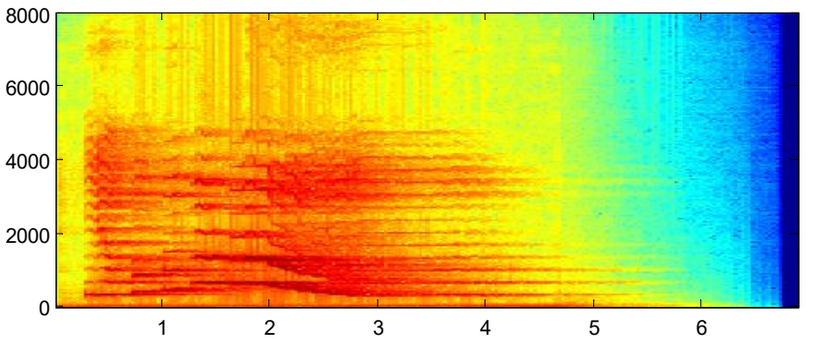
$$T_{60} = \frac{24\ln(10)}{c}\frac{V}{\alpha_{\text{Sabine}}A} \quad \text{s.}$$

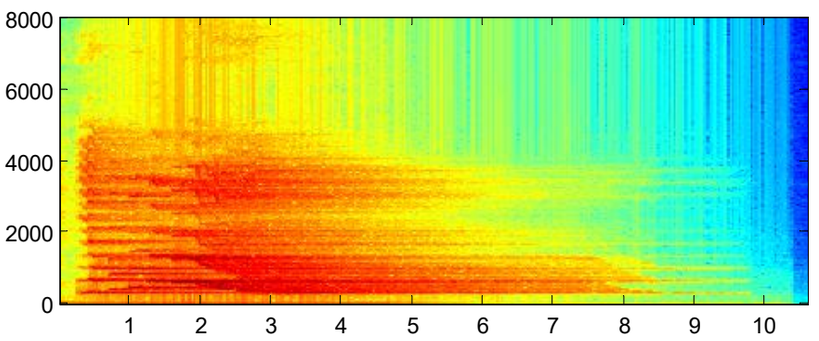
语音信号处理-去混响

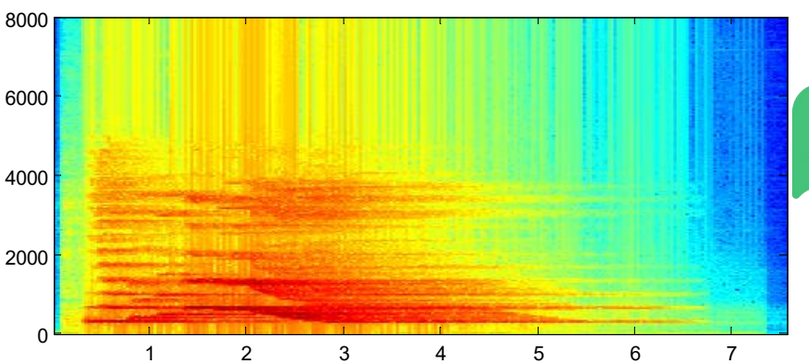
去混响算法











语音信号处理-去混响

去混响算法

混响对语音识别的影响

$$\tilde{h}(n,T,A) = \begin{cases} h(n) & \text{for} \quad 0 \le n < T\, f_s\,, \\ 10^{-\frac{A}{20\,\mathrm{dB}}}\, h(n) & \text{otherwise}\,, \end{cases}$$

$$\tilde{x}(n,T,A) = \sum_{m=0}^{N-1} \tilde{h}(n,T,A)\; s(n-m)\;.$$

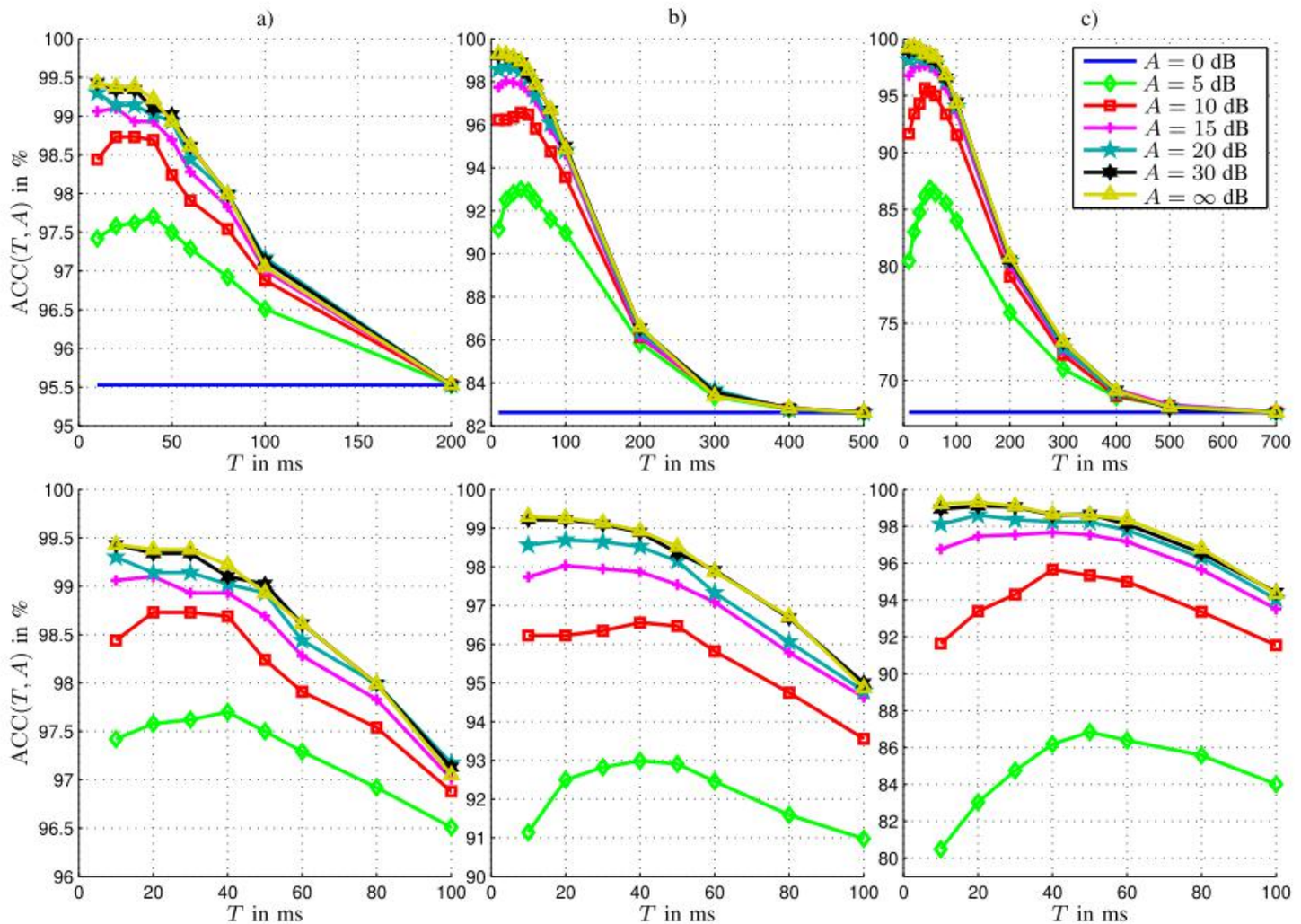
	Room		
	A	B	C
T_{60}	300 ms	780 ms	900 ms
N/f_s	200 ms	500 ms	700 ms
d	2.0 m	2.0 m	4.0 m
D_{50}	97.0 %	81.4 %	74.9 %

$$D_{50}(T,A) = \frac{\sum_{n=0}^{N_{50}-1} \tilde{h}^2(n,T,A)}{\sum_{n=0}^{N-1} \tilde{h}^2(n,T,A)} \cdot 100\,\%\,,$$

语音信号处理-去混响

去混响算法

混响对语音识别的影响



	Room		
	A	B	C
T_{60}	300 ms	780 ms	900 ms
N/f_s	200 ms	500 ms	700 ms
d	2.0 m	2.0 m	4.0 m
D_{50}	97.0 %	81.4 %	74.9 %

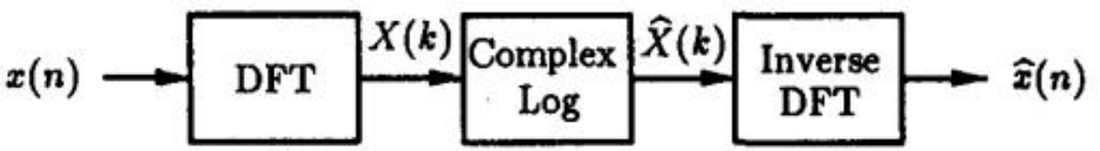
$$\tilde{h}(n, T, A) = \begin{cases} h(n) & \text{for} \quad 0 \le n < T f_s \,, \\ 10^{-\frac{A}{20\,\mathrm{dB}}}\, h(n) & \text{otherwise} \,, \end{cases}$$

语音信号处理-去混响

去混响算法

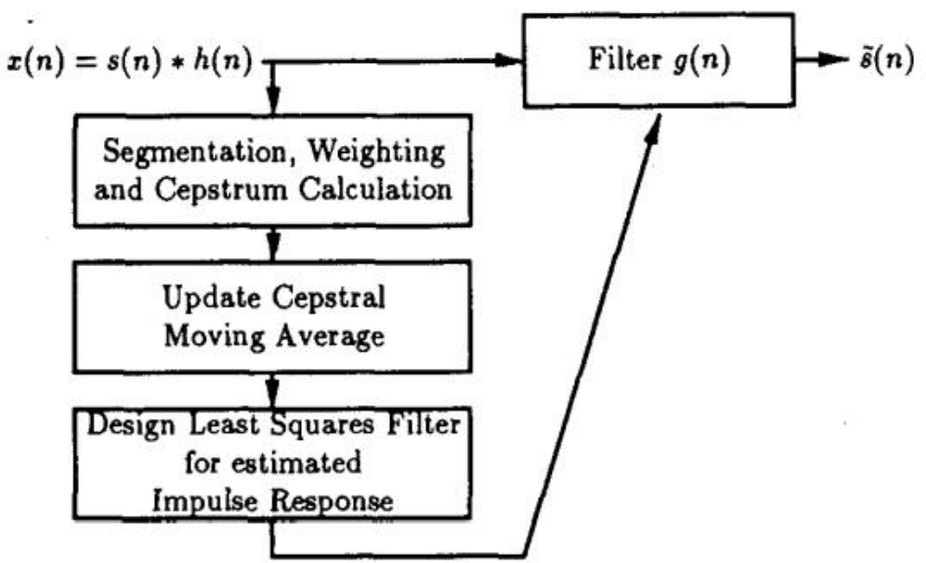
倒谱平均法

$$x(t) = s(t) * h(t) + v(t)$$



忽略噪声

$$\hat{x}(n) = \hat{s}(n) + \hat{h}(n)$$



语音信号处理-去混响

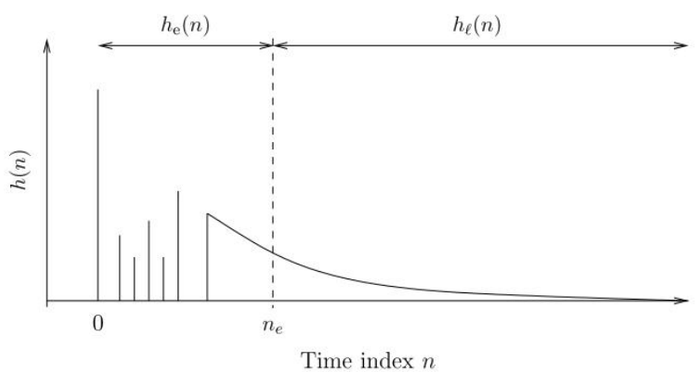
去混响算法

Statistical Reverberation Models

$$h(n) = \begin{cases} h_{\mathrm{d}}(n), & \text{for } 0 \le n < n_{\mathrm{d}}, \\ h_{\mathrm{r}}(n), & \text{for } n \ge n_{\mathrm{d}}, \\ 0, & \text{otherwise.} \end{cases} \qquad h_{\mathrm{d}}(n) = b_{\mathrm{d}}(n)\mathrm{e}^{-\bar{\zeta}n}, \qquad h_{\mathrm{r}}(n) = b_{\mathrm{r}}(n)\mathrm{e}^{-\bar{\zeta}n}, \qquad \bar{\zeta} \triangleq \frac{3\ln(10)}{T_{60} f_{\mathrm{s}}},$$

$$\mathscr{E}\{b_{\mathrm{d}}(n) b_{\mathrm{r}}(n+\tau)\} = 0$$

$$\mathscr{E}\{h^2(n)\} = \begin{cases} \sigma_{\mathrm{d}}^2 \mathrm{e}^{-2\bar{\zeta}n}, & \text{for } 0 \le n < n_{\mathrm{d}} \\ \sigma_{\mathrm{r}}^2 \mathrm{e}^{-2\bar{\zeta}n}, & \text{for } n \ge n_{\mathrm{d}} \\ 0, & \text{otherwise,} \end{cases}$$

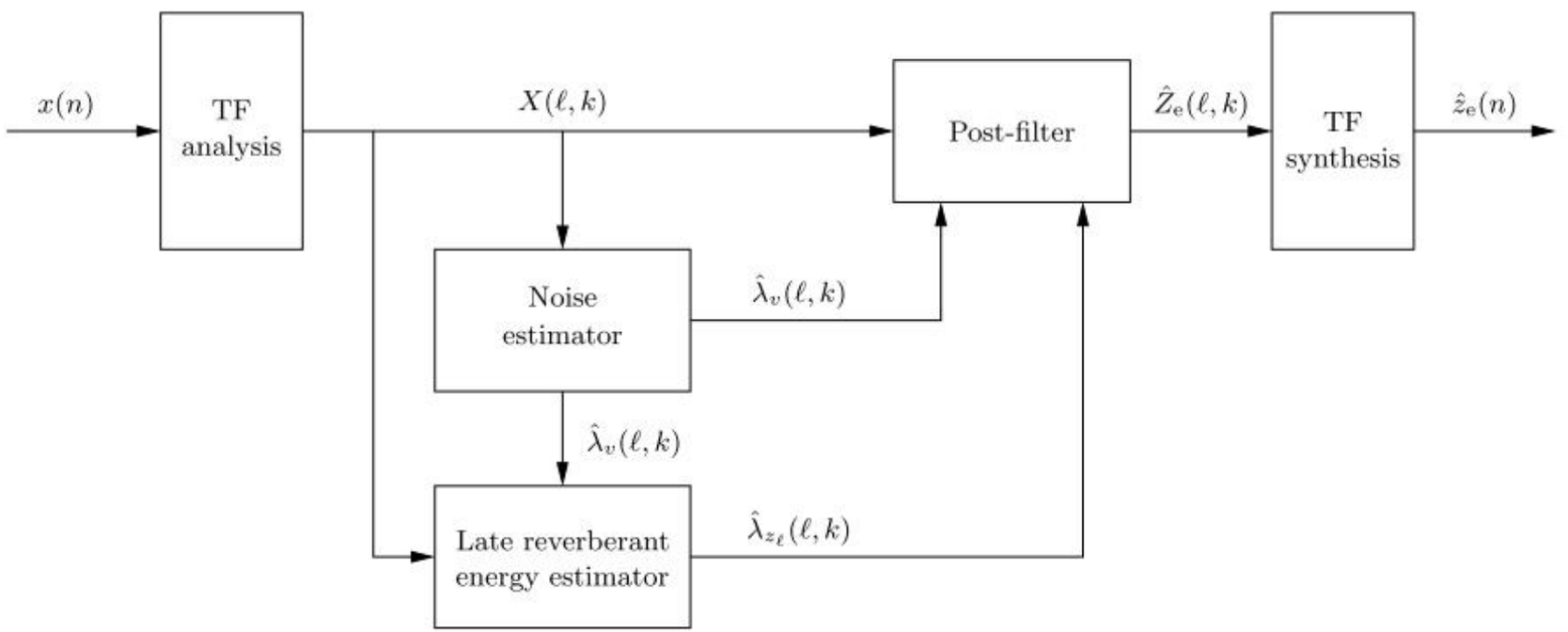


语音信号处理-去混响

去混响算法

Statistical Reverberation Models

$$x(n) = \underbrace{\sum_{l=n-n_e+1}^{n} s(l)h_e(n-l)}_{z_e(n)} + \underbrace{\sum_{l=-\infty}^{n-n_e} s(l)h_\ell(n-l)}_{z_\ell(n)} + v(n),$$



语音信号处理-去混响

去混响算法

Statistical Reverberation Models

$$x(n) = \underbrace{\sum_{l=n-n_e+1}^{n} s(l)h_e(n-l)}_{z_e(n)} + \underbrace{\sum_{l=-\infty}^{n-n_e} s(l)h_\ell(n-l)}_{z_\ell(n)} + v(n),$$

$$\lambda_{z_l}(l,k) = \mathrm{e}^{-2\bar{\delta}(k)\left(T_l - \frac{R}{f_s}\right)} \lambda_{z_r}(l - N_l + 1, k),$$

$$\lambda_{z_r}(l,k) = \mathrm{e}^{-2\bar{\delta}(k)\frac{R}{f_s}} \left(1 - \kappa(k)\right) \lambda_{z_r}(l-1,k) + \kappa(k)\, \mathrm{e}^{-2\bar{\delta}(k)\frac{R}{f_s}} \lambda_z(l-1,k).$$

$$G(k,l) = \frac{S_{ze}}{S_{ze} + S_{zl} + S_v}$$

语音信号处理-去混响

去混响算法

Statistical Reverberation Models

语音信号处理-去混响

去混响算法

WPE算法

CHIME-3rd

NTT, Japan [1]	✓	✓		✓		✓	✓		4.5	7.4	4.5	6.2	5.2	**5.8**
MERL/SRI [2]		✓	✓	✓	✓		✓	✓	8.6	13.5	7.7	7.1	8.1	**9.1**
UST China + GIT, US [3]		✓		✓	✓	✓		✓	7.0	13.8	11.4	9.3	7.8	**10.6**
Inria, France [4]	✓	✓		✓	✓		✓		6.2	16.2	9.6	12.3	7.2	**11.3**
Fraunhofer + U. Oldenburg [5]	✓			✓	✓		✓		6.4	13.5	13.5	10.6	9.2	**11.7**
Hitachi R\&D, Japan [6]		✓	✓	✓	✓			✓	9.8	16.6	11.8	10.0	8.8	**11.8**
NTU, Singapore [7]	✓	✓		✓	✓				8.2	14.5	11.7	11.5	10.0	**11.9**
Rolls-Royce, NTU, Singapore [8]		✓	✓		✓		✓		8.5	17.6	12.1	8.5	9.6	**11.9**
A*Star, Singapore [9]	✓	✓		✓	✓		✓		8.6	18.6	10.7	9.7	9.6	**12.1**
U.Paderborn, Germany [10]	✓	✓					✓		9.0	17.5	10.5	11.0	10.0	**12.3**

Dereverberation challenge2014

语音信号处理-去混响

去混响算法

WPE算法

$$x(t)=\sum_{k=0}^{L_h-1}h(k)s(t-k)+v(t)$$

$$x(t)=d(t)+r(t)$$

$$d(t)=\sum_{k=0}^{D-1}h(k)s(t-k)\qquad r(t)=\sum_{k=D}^{L_h-1}h(k)s(t-k)$$

$$r(t)=\sum_{k=0}^{L_c-1}c(k)x(t-D-k)$$

$$d(t)=x(t)-\sum_{k=0}^{L_c-1}c(k)x(t-D-k)=x(t)-\mathbf{c}^T\mathbf{x}(t-D)$$

语音信号处理-去混响

去混响算法

WPE算法

$$E\left[\left|\hat{d}(t)-d(t)\right|^2\right]=E\left[\left(x(t)-\mathbf{c}^T\mathbf{x}(t-D)-d(t)\right)\left(x(t)-\mathbf{c}^T\mathbf{x}(t-D)-d(t)\right)^*\right]$$

$$\mathbf{c}=\mathbf{R}_{\mathbf{x}(t-D)}^{-1}\mathbf{r}_{\mathbf{x}(t-D)x(t)}$$

1) Update $\hat{\mathbf{c}}$ as follows

$$\hat{\mathbf{c}}=\mathbf{R}_{\mathbf{x}(t-D)}^{-1}\mathbf{r}_{\mathbf{x}(t-D)x(t)}$$

where

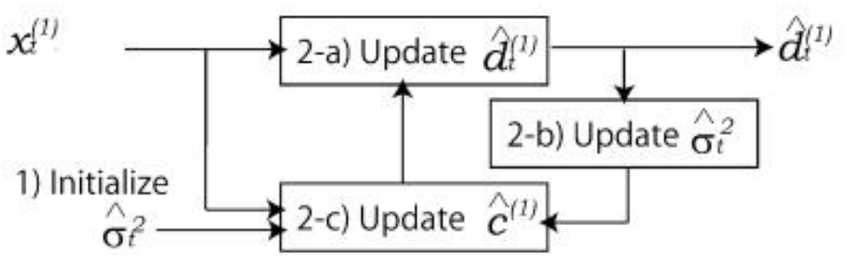
$$\mathbf{R}_{\mathbf{x}(t-D)}=\sum_{t=1}^{T}\frac{\mathbf{x}(t-D)\mathbf{x}(t-D)^T}{\hat{\sigma}_t^2}$$

$$\mathbf{r}_{\mathbf{x}(t-D)x(t)}=\sum_{t=1}^{T}\frac{\mathbf{x}(t-D)x(t)}{\hat{\sigma}_t^2}$$

2) Update $\hat{d}(t)=x(t)-\hat{\mathbf{c}}^T\mathbf{x}(t-D)$

3) Update $\hat{\sigma}_t^2$

$$\hat{\sigma}_t^2=\max\left[\frac{1}{L}\sum_{t=t-L/2+1}^{t=t+L/2+1}\left|\hat{d}(t)\right|^2,\varepsilon\right]$$



语音信号处理-去混响

去混响算法

WPE算法

$$\mathbf{x}_{n,f}^{(1)} = \bar{\mathbf{c}}_f^{*T} \bar{\mathbf{x}}_{n-D',f} + \mathbf{d}_{n,f}$$

$$\mathbf{d}_{n,f} = \sum_{k=0}^{D'-1} \left(\mathbf{h}_{k,f}^{(1)}\right)^{*T} \mathbf{s}_{n-k,f}.$$

1) Initialize $\hat{\rho}_{n,f}^2$ as

$$\hat{\rho}_{n,f}^2 = \max\{|\mathbf{x}_{n,f}|^2, \epsilon_f\}$$

where $\epsilon_f > 0$ is a lower bound of $\hat{\rho}_{n,f}^2$ at f.

2) Repeat the following until convergence.

a) Update $\hat{\bar{\mathbf{c}}}_f$ as follows:

$$\hat{\bar{\mathbf{c}}}_f = \left(\sum_n \frac{\bar{\mathbf{x}}_{n-D',f} \bar{\mathbf{x}}_{n-D',f}^{*T}}{\rho_{n,f}^2}\right)^+ \sum_n \frac{\bar{\mathbf{x}}_{n-D',f} \mathbf{x}_{n,f}^{(1)*}}{\rho_{n,f}^2}.$$

b) Update $\hat{\mathbf{d}}_{n,f}$ as $\hat{\mathbf{d}}_{n,f} = \mathbf{x}_{n,f}^{(1)} - \hat{\bar{\mathbf{c}}}_f^{*T} \bar{\mathbf{x}}_{n-D',f}$.

c) Update $\hat{\rho}_{n,f}^2$ as follows:

$$\hat{\rho}_{n,f}^2 = \max\{|\hat{\mathbf{d}}_{n,f}|^2, \epsilon_f\}.$$

语音信号处理-去混响

去混响算法

WPE算法

语音信号处理-总结