语音信号处理

--前端处理（信号的获取与降噪处理）

http://blog.csdn.net/shichaog/article/details/52514816

http://blog.csdn.net/shichaog/article/details/52403107

http://blog.csdn.net/shichaog/article/details/52399354

目录

第 0 章 信号处理

0.1 语音信号特点

在一段时间内（10ms~30ms），人的声带和声道形状是相对稳定的，可认为其特征是不变的。语音可以分为周期性的浊音和非周期的清音。浊音和清音经常在一个音节中同时出现。浊音部分和音质关系密切，在时域上呈现出明显的周期性，在频域上有共振峰结构，而且大部分能量集中在较低频段内，是语音中大幅度高能量的部分；清音则具有明显的时域和频域特征，类似于白噪声，能量较小，在强噪声中容易被掩盖，但在较高信噪比时能提供较多的信息。在语音增强中，可以利用浊音的周期性特征，采用梳状滤波器提取语音分量或者抑制非语音信号，而清音则难以与宽带噪声区分。

加性噪声大致上有：周期性噪声、脉冲噪声、宽带噪声和同声道的其它语音干扰等。

周期性噪声主要来源于发动机等 周期性运转的机械，电气干扰，特别是电源交流声也会引起周期性噪声，其特点是有许多离散的窄谱峰。脉冲噪声来源于爆炸、撞击和放电等，表现为时域波形中突然出现的窄脉冲。宽带噪声的来源很多，包括热噪声、气流（风、呼吸）噪声及各种随机噪声源，量化噪声也可视为宽带噪声。平稳的宽带噪声，通常也可以视为宽带噪声。平稳的宽带噪声，通常也可以视为高斯白噪声。

语音增强算法大致分为四种：参数法、非参数法、统计法和其它方法。

0.2 信号响应的意义

对于任何一个信号均可以使用冲击函数来表示，即：

$$x(n)=\sum_{-\infty}^{+\infty}x(m)\delta(n-m)$$

数字信号处理的意义就是通过运算来达到处理的目的，设这种运算关系为：

$\mathrm{T}[\bullet]$,则输出信号 $y(n)$ 和输入信号 $x(n)$ 之间的关系描述为：

$y(n)=T[x(n)]$。

0.3 卷积推导

设系统输入 $x(n)=\delta(n)$,系统的输出 $y(n)$ 的初始状态为零，这时系统输出用 $h(n)$ 表示为：

$h(n)=\mathrm{T}[\delta(n)]$。

则称 $h(n)$ 为系统的单位脉冲响应。则对任意输入信号 $x(n)$，系统输出为：

$$y(n)=\mathrm{T}\left[\sum_{-\infty}^{+\infty}x(n)\delta(n-m)\right]$$

,根据叠加原理可得：

$$y(n)=\mathrm{T}\left[\sum_{-\infty}^{+\infty}x(n)\delta(n-m)\right]=\sum_{-\infty}^{+\infty}x(n)T\left[\delta(n-m)\right]$$ 利用系统时不变性，可得下式：

$\mathrm{T}\left[\delta(n-m)\right]=h(n-m)$，因此可得：

$$y(n)=\sum_{-\infty}^{+\infty}x(n)\delta(n-m)=x(n)*h(n)$$

上述就是卷积公式的推导。

0.4 时域离散系统的输入输出描述法

描述一个系统可以不管系统内部的结构如何，将系统看成一个黑盒子，只描述系统的输出和输入之间的关系，这种描述法被成为输入输出描述法。在模拟系统中使微分方程描述系统的输入和输出之间的关系，在时域离散系统中使用差分方程描述系统的输入和输出关系。

点评：微分方程重在描述变化的趋势，差分方程的过程可以套用卷积的方法。

0.4.1 时域离散信号傅里叶变换(DTFT,Discrete-Time Fourier Transform)

定义：

$$X(\omega)=x\left(e^{j\omega}\right)=\sum_{-\infty}^{+\infty}x(n)e^{-j\omega n}$$

上述ω 的单位是弧度，范围是2π。

其傅里叶反变换由如下公式得到：

$$x(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}X\left(e^{j\omega}\right)e^{j\omega n}d\omega$$。

0.4.2 周期信号由傅里叶级数表示

傅里叶变换的一些性质：

时域卷积，频域相乘；时域相乘，频域卷积。

$$\sum_{-\infty}^{+\infty}\left|x(n)^2\right|=\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|x(n)^2\right|d\omega$$ (巴塞伐尔 parseval 定理)-----信号的功率也可以在频域求。

0.5 离散傅里叶变换（DFT）

将有限长时域离散信号变换到频域的变换，但变换的结果是对时域离散信号的频谱的等

间隔采样。

DFT 定义

设序列 $x(n)$ 的长度为 M，定义 $x(n)$ 的 N 点 DFT 为

$$X(k)=DFT\left[x(n)\right]_N=\sum_{0}^{N-1}x(n)e^{-j\frac{2\pi}{N}kn},0\le k\le N-1$$

式中，N 成为离散傅里叶变换区间长度，要求 $N\ge M$。DTFT 中 $\omega_k=2\pi k/N$ 即可得 DFT。为书写简单，令 $W_N=e^{-j\frac{2\pi}{N}}$,则可以简写为：

$$X(k)=DFT\left[x(n)\right]_N=\sum_{n=0}^{N-1}x(n)W_N^{kn},0\le k\le N-1$$

其反变换如下：

$$x(n)=IDFT\left[X(k)\right]_N=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn},0\le k\le N-1$$

DFT 和 FT 之间的关系：

$$X(k)=X\left(e^{j\omega}\right)|_{\omega=\frac{2\pi}{N}k},k=0,1,...,N-1$$

DFT 的主要性质：

1）线性性质

2）隐含周期性

3）循环移位性质

1）有限长序列的循环移位

设序列 $x(n)$ 的长度为 M，对 $x(n)$ 以 N($N\ge M$)为周期进行周期延拓，得到：

$$\tilde{x}_N(n)=x\left((n)\right)_N$$

定义 $x(n)$ 的循环移位序列为:

$$y(n)=\tilde{x}_N(n+m)R_N(n)=x\left((n+m)\right)_N R_N(n)$$

上式表示将序列 $x(n)$ 以 N 为周期进行周期延拓，再左移 m 个单位取主值序列，就得到 $x(n)$ 的循环移位序列 $y(n)$。

则有如下结论：

设序列 $x(n)$ 的长度为 M，其循环移位序列为

$$y_N(n)=x\left((n)\right)_N R_N(n),N\ge M$$

$$X(k)=DFT\left[x(n)\right]_N$$

则

$$Y(k)=DFT\left[y(n)\right]_N=W_{-km}^{N}X(k)$$

0.6 短时傅里叶变换（STFT，Short-Time Fourier Transform）

DTFT 针对平稳信号的变换，语音信号在长时间跨度上不平稳，但其每个 10-30ms 时间段内可看成是平稳的。

定义：

$X(n,\omega)=\sum_{m=-\infty}^{\infty}x(m)w(n-m)e^{-j\omega m}$ ，$x(m)$ 是输入信号，$w(m)$ 是分析窗口，$w(n-m)$是经过时域翻转并右移 n 个采样点。类似于 DFT，离散 STFT 定义如下：

$$X(n,\omega_k)\triangleq X(n,k)=\sum_{m=-\infty}^{\infty}x(m)w(n-m)e^{-j\frac{2\pi}{N}km},\omega_k=\frac{2\pi}{N}k$$

其含义是在时域用窗函数截取信号，对截取部分的信号进行傅里叶变换，即在 n 时刻得到 n 时刻该段信号的傅里叶变换，不断移动 n，即可得到不同的傅里叶变换，将这些傅里叶变换组合起来即得 $X(n,\omega_k)$。

0.6.1 Overlap-and-Add (OA,OLA) 计算 STFT

OA 在计算 $x(n)$ 和 FIR 滤波器 $h(n)$ 卷积效率较高。

$$y(n)=x(n)*h(n)\overset{def}{=}\sum_{m=-\infty}^{\infty}h(m)x(n-m)=\sum_{m=-1}^{M}h(m)x(n-m),m\in[1\quad M],h(m)\neq 0$$

OA 的基本思想是将 $x(n)$ 分段，将分段后的每段与 $h(n)$ 卷积：

$x_k(n)\overset{def}{=}\begin{cases}x(n+kL),n=1,2,...,L\\0,otherwise\end{cases}$ ，L 是任意的分段长度

$$x(n)=\sum_k x_k(n-kL)$$

$$y(n)=\left(\sum_k x_k(n-kL)\right)*h(n)=\sum_k\left(x_k(n-kL)*h(n)\right)=\sum_k y_k(n-kL)$$

0.6 FIR 数字滤波器

FIR(Finite Impulse Response Digital Filter)的最大优点是可以实现线性相位滤波。

线性相位 FIRDF

设 FIRDF 的单位脉冲响应 $h(n)$ 的长度为 N，则其频响函数为：

$$H\left(e^{j\omega}\right)=\sum_{0}^{N-1}h(n)e^{-j\omega n}$$

将$H\left(e^{j\omega}\right)$表示成如下形式：

$$H\left(e^{j\omega}\right)=H_g\left(\omega\right)e^{j\theta(\omega)}$$

式中，$H_g\left(\omega\right)$是ω的实函数，如果满足$\theta\left(\omega\right)$=-$\omega T$ 则相位满足线性关系。

线性相位对时域和频域的约束

$$H\left(e^{j\omega}\right)=\sum_{0}^{N-1}h(n)e^{-j\omega n}=H_g(\omega)e^{-j\omega T}$$ 展开可得：

$$\sum_{n=0}^{N-1}h\left(n\right)\left(\cos(\omega n)-j\sin\left(\omega n\right)\right)=H_g\left(\omega\right)\left(\cos\left(\omega T\right)-j\sin\left(\omega T\right)\right)$$

系数偶对称。

0.7 窗函数设计 FIRDF

其设计思想是使用 FIRDF 逼近希望的滤波特性。

基本方法：

（1） 构造希望逼近的频响函数$H_d\left(e^{j\omega}\right)$。如：

$$H_d e^{j\omega}=\begin{cases}e^{-j\omega T},|\omega|\le\omega_c\\0,\omega_c<|\omega|\le\pi\end{cases}$$

（2） 求出$h_d\left(n\right)$。对$H_d\left(e^{j\omega}\right)$进行 IFT 变换：

$$h_d\left(n\right)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d\left(e^{j\omega}\right)e^{j\omega n}d\omega=\frac{1}{2\pi}\int_{-\pi}^{\pi}e^{-j\omega\tau}e^{j\omega n}d\omega=\frac{\sin\left[\omega_c\left(n-\tau\right)\right]}{\pi\left(n-\tau\right)}$$

（3） 加窗得到 FIRDF 的单位脉冲响应$h\left(n\right)$。

$$h\left(n\right)=h_d\left(n\right)\omega\left(n\right)$$

如果要满足线性相位，窗函数同样需要满足线性相位。

频域自适应滤波的主要原因是，时域自适应滤波器需要很长的脉冲响应来处理同样长的回声持续时间。

第一章 重采样算法

1.1 信号重采样

假设信号 $x(t)$ 是连续时间信号，则以 $t=nT_s$ 为间隔对信号 $x(t)$ 采样，采样后信号为

$x(nT_s)$ ，n 是整数，T_s 是采样周期，根据奈奎斯特采样定理，当 $x(t)$ 是带限信号，且其频带范在 $\pm F_s/2$ 之间，这时采样并不会导致频谱混叠，采样率 $F_s = 1/T_s$ ，设 $x(\omega)$ 是 $x(t)$ 的傅里叶变换，则有 $X(\omega)=\int_{-\infty}^{+\infty} x(t)e^{-j\omega t}dt$ 。可以假定：

$$X(\omega)=0,|\omega|\geq \pi F_s \qquad (1.1)$$

根据香农定理，根据 $x(nT_s)$ 重构 $x(t)$ 的公式如下：

$$x(t)\triangleq \sum_{n=-\infty}^{\infty} x(nT_s)h_s(t-nT_s)\equiv x(t) \qquad (1.2)$$

此处，

$$h_s(t)\triangleq \sin c(F_s t)\triangleq \frac{\sin(\pi F_s t)}{\pi F_s t} \qquad (1.3)$$

以下仅讨论 $x(t)$ 在（$F_s^{'}=1/T_s^{'}$），仅讨论方程 1.2 是整数倍情况下的 $T_s^{'}$ 。

当新的采样率 $F_s^{'}$ 小于原始 F_s 时，低通滤波截止频率必须是新采样率的二分之一，即 $\pm F_s^{'}/2$ 。对于理想的低通滤波，则有：

$$h(t)=\min\left\{1,F_s^{'}/F_s\right\}\sin c\left(\min\left\{F_s,F_s^{'}\right\}t\right)$$

前面的增益因子保持通带内单位增益。

Sinc 函数 $\sin c(t)\triangleq \sin(\pi t)/\pi t$ 的波形如下：

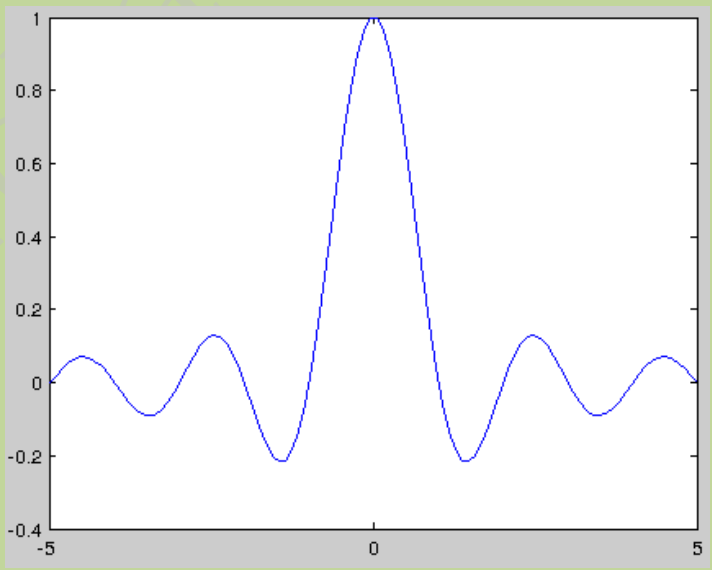


图 1.1 sinc 函数波形

公式 1.2 用卷积公式表示为：$(x*h_s(t))$ 。

该卷积过程可看成是 sinc 函数的平移叠加，每个 sinc 函数对应于一个采样点，并且幅

度被采样点调制过，所有 sinc 函数加在一起是原始信号。$\sin c(z)$ 值为零的地方位于整数处（除 Z=0）。这意味着 $t = nT_s$ 时刻（对应于一个采样点），为和的唯一贡献是采样信号 $x(nT_s)$ 。

sinc 函数是理想低通滤波器，实际使用时会加窗处理。

1.2 音频重采样

在音频上，常常遇到重采样问题，如将 44.1k、48k、32k 以及 16k 音源之间的转换，通常这一过程被称为 SRC（sample rate converter）。采样率转换的基本思想是抽取和内插。从信号角度看，音频重采样是滤波。滤波函数的窗口大小以及插值函数一旦被确定，其重采样的性能也就确定了。

抽取可能引起频谱混叠，内插则产生镜频。通常在抽取前将抗混叠滤波，在内插后加抗镜像滤波。在语音识别里所需要的语音信号采样率实际上由其 ASR（automatic speech recognition）训练模型决定的。

➢ 重采样过程

1 设音频信号的原始采样率是 L，新的采样率是 M，原始信号长度是 N，则新采样率下信号的长度 K 满足如下关系：

$$K = (M / L) * N$$

2 对每个离散时间值：$k(1 \le k \le K)$,则实值 n_k 的值为：

$$n_k = (L / M) * k$$

n_k 为在原始采样间隔的情况下，要进行插值的位置。

3 确定两个加权系数的值，利用第二步计算得到的 n_k 值，求两个权值。选择恰当的权值，才能让线性差值所要插取的值更加接近插值点的理想幅度值，让 w_1 和 w_2 分别代表两个重采样权值。

$$w_1 = n_k - 1$$

$$w_2 = 1 - w_1$$

4 根据两个权值，估计插入点的具体幅度值。

$$y(k) = w_{1x}(n+1) + w_{2x}(n)$$

上采样会产生镜频，下采样会产生混叠。为了消除镜频和混叠，就需要将信号通过低通滤波器。上采样时，产生的镜频通过低通滤波器去除；下采样时，为了防止产生混叠，先通过低通滤波器进行处理。

1.3 sinc 重采样

这里之所以使用 sinc 重采样，一是因为开源语音处理算法 speex 使用的就是加窗的 sinc 方法，加窗函数是凯撒窗。

以整数因子抽取为例来说明算法实现过程，设 $x(n)$ 是对模拟信号 $x_a(t)$ 以奈奎斯特速率 F_x 采样得到的信号，其频谱为 $X(e^{j\omega})$ ，则在频率区间 $0 \le |\omega| \le \pi$ （模拟频率是 $|f| \le F_x/2Hz$ ），$X(e^{j\omega})$ 是非零的。现在按整数倍 D 对 $x(n)$ 进行抽取。为了避免频谱混叠，必须先对 $x(n)$ 进行抗混叠低通滤波，将 $x(n)$ 的有效频带限制在折叠频率（ $F_x/2DHz$ ）以内，等效的数字频率为 π / D 弧度以内，然后再按整数因子 D 对 $x(n)$ 进行抽取，得到信号 $y(m)$ 。其原理框图如下：

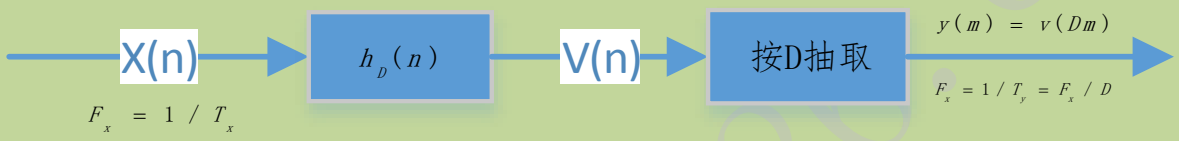


图 1.2 整数因子 D 抽取原理框图

理想情况下，抗混叠低通滤波器 $h_D(n)$ 的频率响应：

$$H_D(e^{j\omega}) = \begin{cases} 1, & |\omega| < \pi / D \\ 0, & \pi / D \le |\omega| \le \pi \end{cases} \tag{1.4}$$

经过抗混叠低通滤波器后输出为：

$$v(n) = h(n) * x(n) = \sum_{k=0}^{\infty} h_D(k)x(m-k) \tag{1.5}$$

考虑到 $h_D(n)$ 为因果稳定系统，所以式 1.5 中卷积求和从 0 开始。按整数因子 D 对 $v(n)$ 抽取得：

$$y(m) = V(Dm) = \sum_{k=0}^{\infty} h_D(k)x(Dm-k) \tag{1.6}$$

➢ 直接型 FIR 滤波器实现

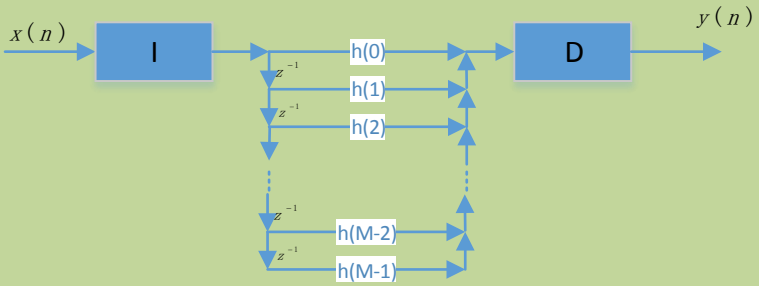


图 1.3 采样率转换系统的直接型 FIR 滤波器结构

图 1.3 中的结构很明白的表示了式 1.6 所表示的滤波过程，再经过 D 抽取器获得相应的

采样率，在 I 表示的插值过程中，会牵涉到在 $x(n)$ 相邻样本值之间插入 I-1 个零值，如果 I 值比较大，则进入 FIR 滤波器的信号大部分为零，因此乘法运算的结果也大部分是零，即多数乘法是无效的，此种运算效率比较低。

➢ 整数因子 D 抽取直接型 FIR 滤波器结构

整数因子 D 抽取系统的直接型 FIR 滤波器结构如图 2.3 a 所示，该结构中，FIR 滤波器以最高采样率 F_x 运行，但其输出的每 D 个样值中抽取一个作为最终输出，丢弃 D-1 个样值，这样效率较低。

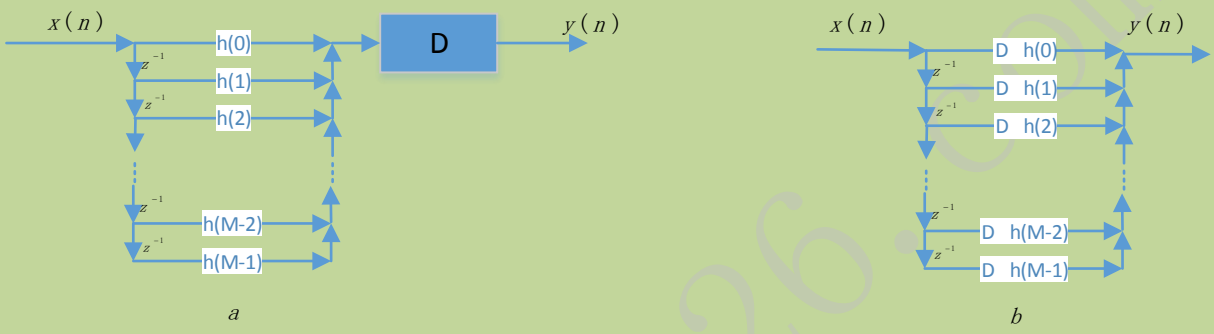


图 1.4 整数因子 D 抽取系统的直接型 FIR 滤波器

高效的 FIR 滤波器将抽取操作 D 嵌入 FIR 滤波器结构中，1.4 的 b 所示，a 的抽取器在 $n=Dm$ 时刻开通，选通 FIR 滤波器的一个输出作为抽取系统输出序列的一个样值 $y(m)$ 。

$$y(m)=\sum_{k=0}^{M-1}h(k)x(Dm-k) \qquad 2.4$$

b 的抽取器在 $n=Dm$ 时刻同时开通，选通 FIR 滤波器输入信号 $x(n)$ 的一组延迟：$x(Dm)$ ，$x(Dm-1)$ ，$x(Dm-2)$，…，$x(Dm-M+1)$，再进行乘法、加法运算，得到抽取系统输出序列的一个样值，$y(m)=\sum_{k=0}^{M-1}h(k)x(Dm-K)$ ，b 的运算量仅是 a 的 1/D 之一。

➢ 整数因子 I 内插系统直接型 FIR 滤波器结构

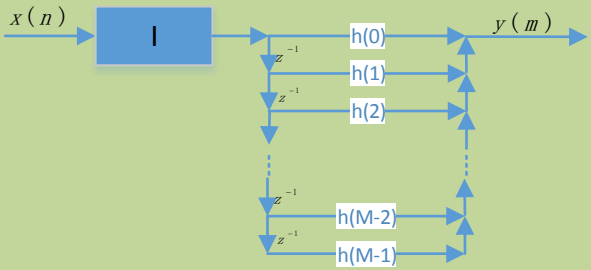


图 1.5 按整数因子 I 内插系统的直接型 FIR 滤波器

根据图 1.5 给出的结构，需要 FIR 滤波器以 IF_x 运行。该效率比较低。其高效的等效变换见图 1.6 b 所示。

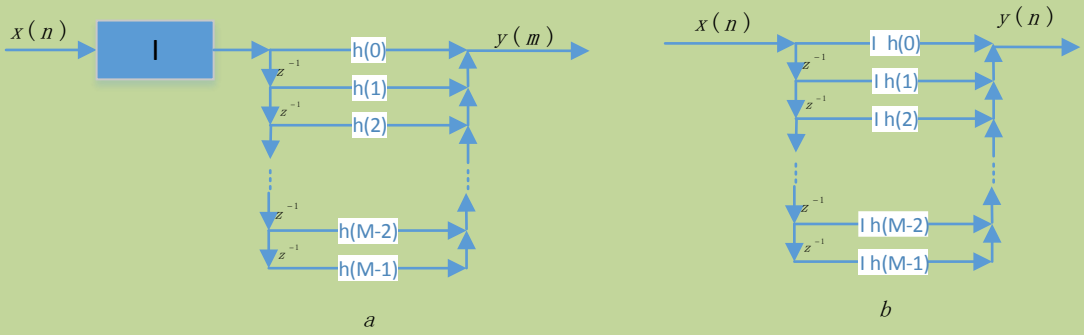


图 1.6 整数因子 I 内插系统的高效实现

➢ 多相滤波器结构

图 1.7 中 b 的实现结构可以采用较短的多项滤波器组来实现其内插功能，如果滤波器的总长度为 M=NI，则多相滤波器组有 I 个长度为 N=M/I 的短滤波器构成，且 I 个短滤波器轮流分时工作，所以称之为多相滤波器。整数因子 I 内插系统的直接型 FIR 滤波器的输出 $y(m)=h(m)*v(m)$ 。零值内插器的输出序列 $v(m)$ 是在输入序列 $x(n)$ 的两个相邻样值之间插入 I-1 个零样值得到，因此 $v(m)$ 进入 FIR 滤波器的 M 个样值中只有 N=M/I 个非零值。即在任意时刻 m，计算 $y(m)=h(m)*v(m)$ 时只有 N 个非零值与 $h(m)$ 中的 N 个系数相乘。

$$v(m)=\begin{cases}x(m/I), m=0,\pm I,\pm 2I,\pm 3I\\ 0, other\end{cases} \tag{1.7}$$

所以 $y(m)=\sum_{n=0}^{M-1}h(n)v(m-n)=\sum_{n=0}^{N-1}h(nI)x(m-n)$ ，当 $m=jI+k$ $k=0,1,2,\ldots I-1, j=0,1,\ldots$ 时，有：

$$y(m)=\sum_{n=0}^{M-1}h(n)v(m-n)=\sum_{n=0}^{N-1}h(k+nI)x(m-n) \tag{1.8}$$

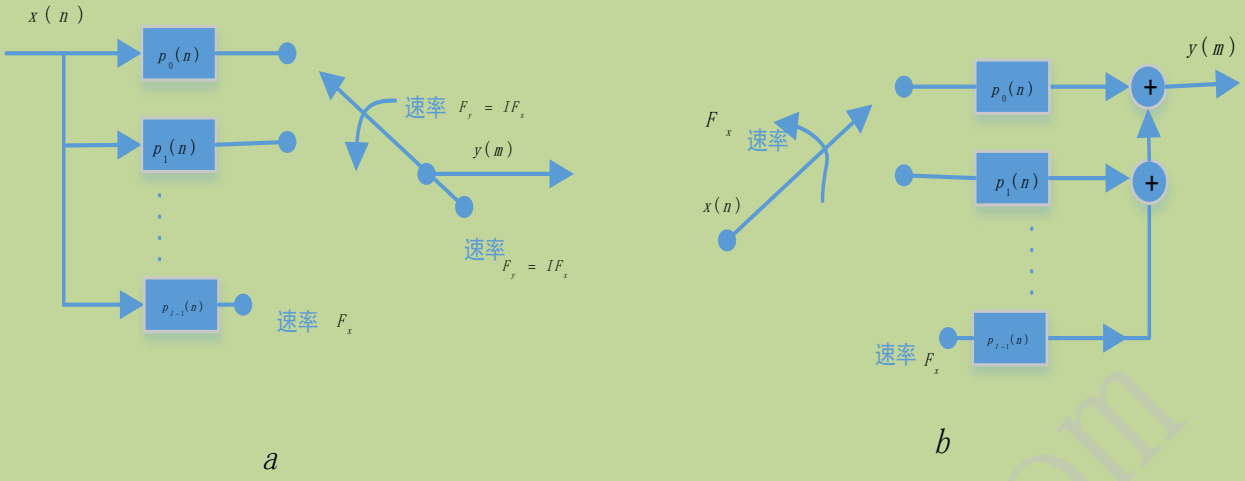
式 2.5 中的 $h(k+nI)$ 看做长度 N=M/I 的子滤波器的单位脉冲响应，并用 $p_k(n)$ 表示，则：

$$p_k(n)=h(k+nI), k=0,1,\ldots I-1, n=0,1,\ldots N-1$$

这样，从 $m=0$ 开始，整数因子 I 内插系统的输出序列 $y(m)$ 计算如下：

$$y(m)=\sum_{n=0}^{M-1}h(n)v(m-n)=p_k(n)*x(n) \tag{1.9}$$

当 $m=jI+k$ 从 0 开始增大时，k 从 0 开始以 I 为周期循环取值；j 表示循环周期数。所以式 2.6 对应的多相滤波器结构如图 2.6 所示。输出序列 $y(m)$ 就是从 $k=0$ 开始，依次循环选取 I 个子滤波器的输出所形成的序列。



a b

图 1.7 多相滤波结构，（a 整数因子 I 内插，b 整数因子 D 抽取）

多相滤波器的单位脉冲响应为：

$$p_k(n)=h(k+nD),k=0,1,2,\ldots,D-1;n=0,1,\ldots,N-1 \qquad 1.10$$

式中，N 为 $p_k(n)$ 的长度。一般选择原抗混叠 FIR 滤波器总长度 M=DN， N=M/D。电子开关以速率 F_x 逆时针旋转，从子滤波器 $p_0(n)$ 在 m=0 时刻开始，并输出 $y(0)$；然后以电子开关速率 F_x 逆时针旋转一周，即每次转到子滤波器 $p_0(n)$ 时，输出端就以速率 $F_y=F_x/D$ 送出一个 $y(m)$ 样值。

从频域角度看，线性重采样适合应用与下采样。由于自己所在的工程所需的是将 48K 的音频信号重采样到 16k，这就是一个抽取过程。

1.4 MATLAB 实现

前述的内容属于理论方面，现在基于 MATLAB 来阐述重采样实现的一些工程实现细节。首先使用 MATLAB 的 decimate 函数将 48K 的音源抽取到 16k，并且绘制它们的频谱如下：

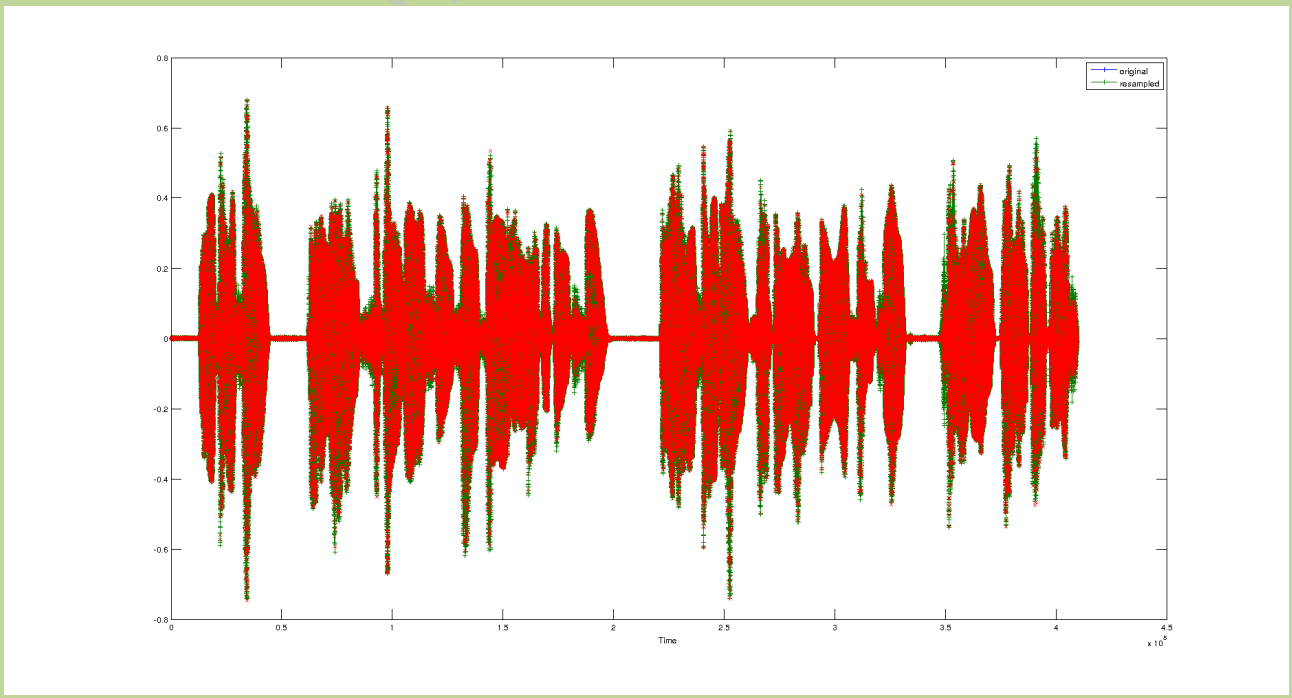


图 1.8 resample 抽取前后语音信号的时频域波形

上图深绿色的点数是 4096*100，蓝色的点数是 136500；4096*100/136500=3.0007，由时域可以看到信号确实数量降低到原来的 1/3 了。两种颜色叠加之后的波形如上，从时域波形上来看，两者的波形并未完全重叠，和其它地方相比，尖端处的差异大些。

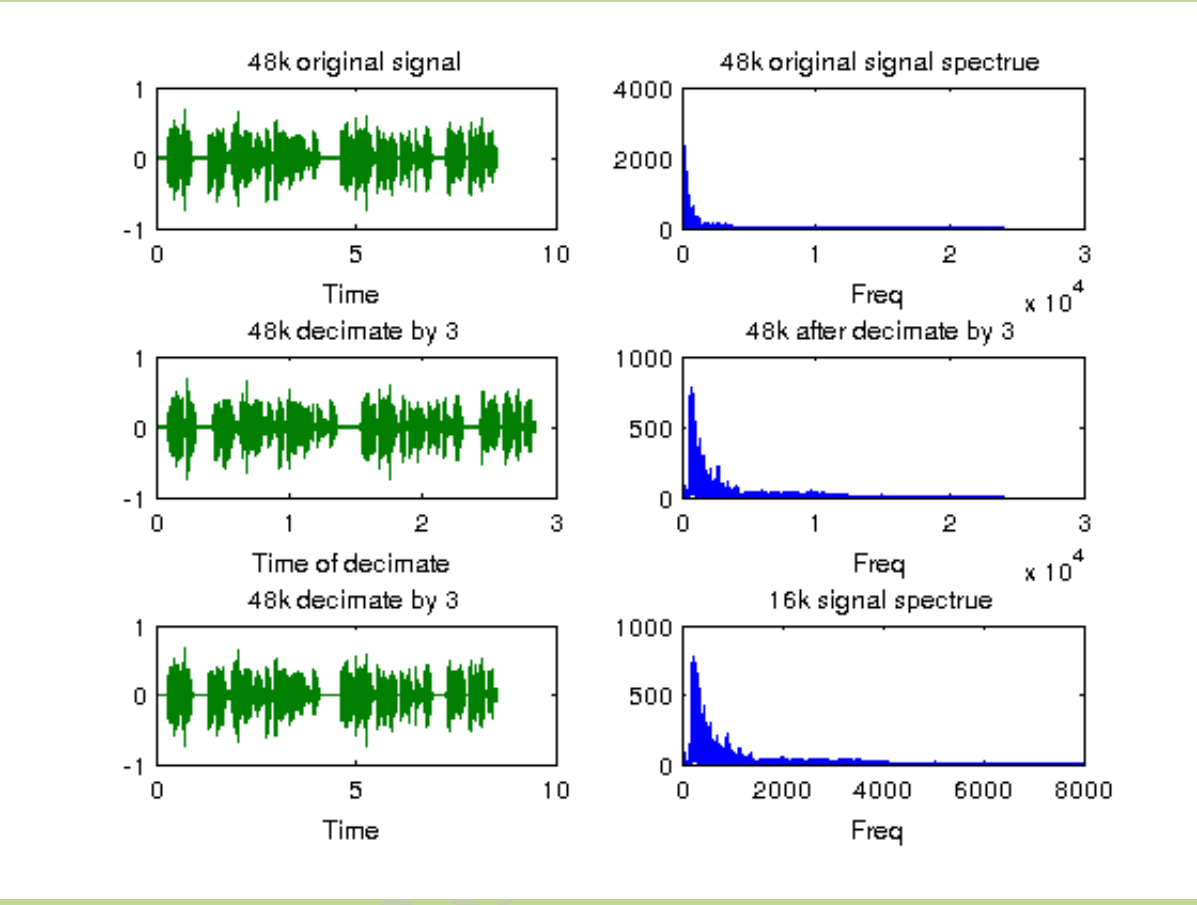


图 1.9 48k 和 16k 采样率下的信号时域和频域波形

图 1.9 的第一行是原始 48k 信号的时域和频域波形图，其频域波形在 24k 截止，第二行是 48K 采样率下抽取后信号的时域和频域波形图，对比时域和频域来看，它们的频谱区别还是有的，第三行是 16k 采样率下绘制重采样后信号的视频域波形，其频域截止频率在 8K，和第二行信号的频谱非常接近，但是它们还是存在差异的。

```
clear; close all; clc;
%read sound wave
[x,fs]=audioread('48k_sound.wav');
% sound(x,48000);
%sound(y,fs,nbits);
N=length(x);
tx=(0:N-1)/fs;%calc the time that correspond to original signal
xf=fft(x);
fx=(0:N/2)*fs/N;
figure(1);
subplot(321);plot(tx,x);xlabel('Time');title('48k original signal');
subplot(322);plot(fx,abs(xf(1:N/2+1)));xlabel('Freq ');title('48k original signal spectrue');
audiowrite('48k_sound.wav',x,48000);

%decimate signal
k=1:N/3;% Number to decimate 48k to 16k

%filter design kasier
fp=8000;fc=12000;as=100;ap=1;FS=48000;
wc=2*fc/FS; wp=2*fp/FS;%kaiser window
```

```
N=ceil((as-7.95)/(14.36*(wc-wp)/2))+1;
beta=0.1102*(as-8.7);
win=kaiser(N+1, beta);
wvtool(win);
B=fir1(N, wc,win);
freqz(B,1,512,fs);
y=filter(B,1,x)
y=downsample(y,3);
%实际上这里是每隔三个点取一个点，即先低通滤波再取点，和紧接着的下面的三行结果是一样的
% y=x(3*k);% 48k--->16k,and data store in y
%y=decimate(x,1,3);
% y=resample(x,1,3);
M=length(y);% M is the length of y
%analysis y under the frequency of 48k
ty=(0:M-1)/fs; %fs equ 48k
subplot(323);plot(ty,y);xlabel('Time of decimate');title('48k decimate by 3');
yf=fft(y);% fft of 48k downsample by 3
fy=(0:M/2)*fs/M;
subplot(324);plot(fy,abs(yf(1:M/2+1)));xlabel('Freq');title('48k after decimate by 3');

%decimate data under sample 16k
tz=(0:M-1)/(fs/3);
subplot(325);plot(tz,y);xlabel('Time');title('48k decimate by 3');
fz=(0:M/2)*(fs/3)/M;
subplot(326);plot(fz,abs(yf(1:M/2+1)));xlabel('Freq ');title('16k signal spectrue');

sound(y,16000);

audiowrite('16k_sound.wav',y,16000);

% %ellipord
% fp=8000;fc=12000;as=100;ap=1;fs=48000;
% wc=2*fc/fs;wp=2*fp/fs;
% [n,wn]=ellipord(wp,wc,ap,as);%
% [b,a]=ellip(n,ap,as,wn);
% freqz(b,a,512,fs);
```

从上面的 MATLAB 运行的结果来看，从 48K 数据重采样到 16K，每隔三个点取一个点和先经过低通滤波再抽取实际上几乎没有差别。上面 MATLAB 代码经过滤波后的各种波形和图 1.9 几乎没有差别。

1.5 开源重采样 C 语言实现

1.6 基于 FPGA 的多相滤波

1.7 重采样性能评估

音频信号的好坏常用信噪比来评估，同样可以根据重采样后音频信号的性噪比来评估重采样的质量。信噪比的定义如下：

$$SNR = 10\log_{10}\left\{\frac{\sum_{n=0}^{M} s^2(n)}{\sum_{n=0}^{M}\left(s(n)-\hat{s}(n)\right)^2}\right\} \tag{1.11}$$

式中 $s(n)$ 为原始信号，$\hat{s}(n)$ 是转换后的同频率音频信号，M 为音频信号的长度。信噪比越大意味着总体上原始音频信号和重采样后的信号之间的差距较小，接近程度较高，在功放的测试中就有信噪比这么一个测试项，现在信噪比在 80dB 以上的音响比较常见了。

由于语音信号是短时平稳随机信号，可将语音信号分帧计算各帧的信噪比，平均后可以得到信噪比。计算的公式如下：

$$SNR = \frac{1}{N}\sum_{m=0}^{M-1}\log_{10}\left\{\frac{\sum_{n=0}^{r}x_n^{\,2}}{\sum_{n=0}^{r}\left(x_n-\hat{x}_n\right)^2}\right\} \tag{1.12}$$

M 是总的帧数，r 是每一帧的长度。

第 2 章 回声消除（AEC ）原理及实现

2.1 回声消除原理

回声消除的基本原理是使用一个自适应滤波器对未知的回声信道 ω 进行参数辨识，根据扬声器信号与产生的多路回声的相关性为基础，建立远端信号模型，模拟回声路径，通过自适应算法调整，使其冲击响应和真实回声路径相逼近。然后将麦克风接收到的信号减去估计值，即可实现回声消除功能。

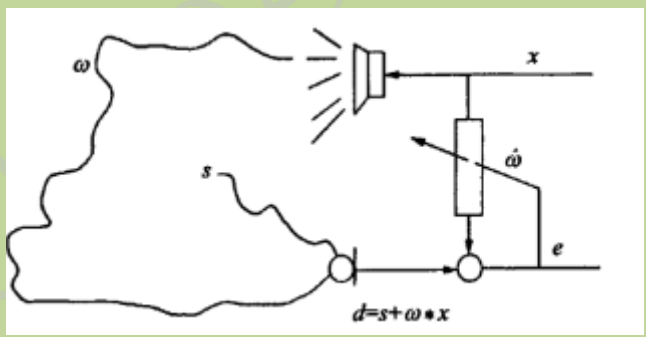


图 2.1 回声消除原理图

式中，ω 是回声通道的时域冲击响应函数；x 是远端语音；echo 是所得回声；s 是近端说话人语音，d 为麦克风采集到的信号；$\hat{y}$ 为对回声信号的估计值；e 为误差。在电话、视频会议中这里的 x 通信另一端的语音信号，而在机器语音识别中，这里的 x 则指机器自身发出的声音。

$$echo = x * \omega \tag{2.1.1}$$

$$d = s + echo \tag{2.1.2}$$

$$\hat{y} = x * \hat{\omega} \tag{2.1.3}$$

$$e = d - \hat{y} \tag{2.1.4}$$

为了消除较长时间的回声，需要 FIR 滤波器的阶数尽量的大。时域计算诸多不便，使用频域分块自适应滤波算法。

2.2 维纳滤波

均方误差（MSE， Mean Square Error）,对于离散时间系统，可定义期待响应 d_k 为一个希望自适应系统的输出 y_k 与之相接近的信号，k 为采样时刻。

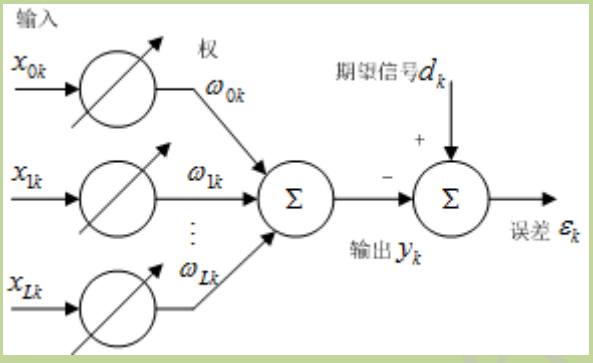


图 2.2 MSE 自适应系统

根据图 2.2，可以求得误差信号：

$$\varepsilon_k = d_k - y_k \tag{2.2.1}$$

自适应线性组合器输出：

$$y_k = W_k^T X_k \tag{2.2.2}$$

其中：

$$X_k = [x_{0k} x_{1k} \ldots x_{Lk}], W_k = [w_{0k} w_{1k} \ldots w_{Lk}]^T$$

分别为自适应系统在 k 时刻的输入信号向量和权向量，系统的均方误差为：

$$E(|\varepsilon_k|^2) = E[(d_k - y_k)^*(d_k - y_k)] = E(|d_k|^2) + \boldsymbol{W}_k^H E[\boldsymbol{X}_k^* \boldsymbol{X}_k^T]\boldsymbol{W}_k - 2\,\mathrm{Re}\{\boldsymbol{W}_k^T E[d_k^* \boldsymbol{X}_k]\} \tag{2.2.3}$$

$$\boldsymbol{R} = E[\boldsymbol{X}^* \boldsymbol{X}^T] = \begin{bmatrix} E[x_0^* x_0] & E[x_0^* x_1] & \cdots & E[x_0^* x_L] \\ E[x_1^* x_0] & E[x_1^* x_1] & \cdots & E[x_1^* x_L] \\ \vdots & \vdots & \ddots & \vdots \\ E[x_L^* x_0] & E[x_L^* x_1] & \cdots & E[x_L^* x_L] \end{bmatrix} \tag{2.2.4}$$

定义期待响应和输入信号之间的互相关向量为：

$$\boldsymbol{P} = E[d^* \boldsymbol{X}] = \begin{bmatrix} d^* x_0 \\ \vdots \\ d^* x_L \end{bmatrix} \tag{2.2.5}$$

将式 2.2.3 简化成下式：

$$\xi(\boldsymbol{w}) = E(|d_k|^2) + \boldsymbol{W}_k^H \boldsymbol{R}\boldsymbol{W}_k - 2\,\mathrm{Re}\{\boldsymbol{W}_k^T \boldsymbol{P}\} \quad 2.2.5$$

理想情况下 $E\left(|\varepsilon_k|^2\right)$ 等于零，这时估计值等于观测值，如果不能达到理想，则应该是越小越好，这样估计值和观测值最接近。

对 2.2.5 求偏导数，得：

$$\nabla = \frac{\partial}{\partial \boldsymbol{W}}[\xi(\boldsymbol{W})] = 2\boldsymbol{R}\boldsymbol{W} - 2\boldsymbol{P}^* \quad 2.2.6$$

最佳权向量处的梯度值为零，于是：

$$\nabla = 2\boldsymbol{R}\boldsymbol{W}_{opt} - 2\boldsymbol{P}^* = 0 \quad 2.2.7$$

最小均方误差输出情况下的最佳权向量 $\boldsymbol{W}_{opt}$ 满足维纳-霍夫方程：

$$\boldsymbol{W}_{opt} = \boldsymbol{R}^{-1}\boldsymbol{P}^* \quad 2.2.8$$

2.3 LMS 算法

$$\varepsilon_k = d_k - \boldsymbol{X}_k^T \boldsymbol{W}_k \quad 2.3.1$$

式中，$\boldsymbol{X}_k$ 为输入样本向量，使用单次采样数据 $|\varepsilon_k|^2$ 来代替均方误差 ξ_k，这样其梯度估计可表示为如下形式：

$$\begin{aligned}\hat{\nabla}_k &= \frac{\partial}{\partial \boldsymbol{W}_k}|\varepsilon_k|^2 = \frac{\partial}{\partial \boldsymbol{W}_k}[|d_k|^2 + \boldsymbol{W}_k^H \boldsymbol{X}_k^* \boldsymbol{X}_k^T \boldsymbol{W}_k - 2\,\mathrm{Re}(d_k^* X_k^T W_k)] \\ &= 2\boldsymbol{X}_k^* \boldsymbol{X}_k^T \boldsymbol{W}_k - 2d\boldsymbol{X}_k^* = -2\varepsilon_k \boldsymbol{X}_k^*\end{aligned} \quad 2.3.2$$

基于最速下降法的权向量迭代如下：

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k - \mu\hat{\nabla}_k = \boldsymbol{W}_k + 2\mu\varepsilon_k \boldsymbol{X}_k^* \quad 2.3.4$$

其中 μ 是步长因子，$0 < \mu < \frac{1}{\lambda_{\max}}$，$\lambda_{\max}$ 是 R_{XX} 的最大特征值。$W(k)$ 收敛于 W_{opt} 由比值 $d = \frac{\lambda_{\max}}{\lambda_{\min}}$ 决定，该比值叫做谱动态范围。大的 d 值意味着较长的时间才能收敛到最佳权值。

该算法用在语音增强的加性噪声消除功能上时，其工程实践并不完全按照式 2.3.1 意义来实现。

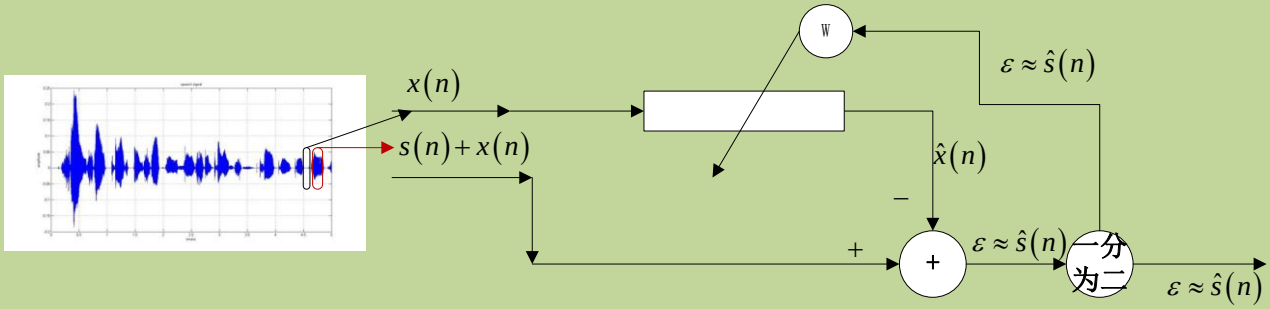


图 LMS 算法在语音增强中使用方法

在语音增强中，其目的是获得纯净的语音信号 $s(n)$，即上图中的最后输出信号，输入信号有两种，一种是带噪的语音信号 $s(n)+x(n)$，另一种是只有噪声的输入 $x(n)$，在没有人说话的情况下的输入信号，就仅仅是噪声输入。这里要使得噪声估计 $\hat{x}(n)$ 非常接近 $x(n)$，这样 $\varepsilon=\sum_n s(n)+x(n)-\hat{x}(n)$，这时如果 ε^2 最小，则可以知道估计出的 ε 最接近 $s(n)$。

上述过程可以概述如下：

A) 首先获取到噪声输入 $x(n)$，并存储下来，以 64 或者 128 点为总长度不断刷新存储噪声输入。

B) 采集带噪声的语音信号 $s(n)+x(n)$。

C) 用采集带噪语音信号减去估计到的噪声信号 $s(n)+x(n)-\hat{x}(n)$。

D) 用 C 的输出作为误差，调节噪声权向量 W。

MATLAB 实现具体包括如下三个部分：

```
% Loop over input vector
for ii = 1:length(signal_with_noise)
   % Update buffer
   noise_buf = obj.update_buf(noise_buf, noise(ii));//输入噪声估计
   % Filter this sample with current coefficient values
   filter_output = obj.data_filter(coefs, noise_buf);//通过权向量估计 x̂(n)
   % Compute error
   err = signal_with_noise(ii) - filter_output; //相减得到 ŝ(n)
   % Update coefficients
   coefs = obj.update_coefs(coefs, noise_buf, obj.filter_params.step_size,
obj.filter_params.leakage, err); //用 ŝ(n) 调节权向量
   % Build output vector
```

dout(ii) = err; //存储输入信号的估计值 $\hat{s}(n)$

2.3.1 NLMS

输入信号较大时，会遇到梯度噪声放大的问题，使得能量低的信号算法收敛速度较慢。将输入信号按照自身的平均能量进行归一化处理，即得到归一化 NLMS 算法。设输入带噪信号可表示为： $x(n)$ ，其迭代算法的 NLMS 表示公式如下：

$$\boldsymbol{W}_{n+1} = \boldsymbol{W}_n - \mu\hat{\nabla}_n = \boldsymbol{W}_n + \frac{\mu}{N}\frac{e(n)\boldsymbol{x}(n)}{\hat{\sigma}_x^2(n)}$$

其中 $\hat{\sigma}_x^2(n)=\frac{1}{N}\sum_{n=0}^{N-1}x^2(n-k)$ ，其中 N 是噪声消除器和回波抵消器的长度，（常取 512 或者 1024）； μ 是步长因子。当 $\hat{\sigma}_x^2(k)$ 较小时， $\frac{\mu}{\hat{\sigma}_x^2(n)}$ 的值可能较大，这时迭代算法变成如下形式：

$$\boldsymbol{W}_{n+1} = \boldsymbol{W}_n - \mu\hat{\nabla}_n = \boldsymbol{W}_n + \frac{\mu}{N}\frac{e(n)\boldsymbol{x}(n)}{\sigma + \hat{\sigma}_x^2(n)}$$

其计算过程如下：

参数： M=抽头系数（即 FIR 滤波器长度）

μ 自适应常数

$0<\mu<2\frac{E\left[\left|\boldsymbol{x}(n)\right|^2\right]E\left[\left|\boldsymbol{\varepsilon}(n)\right|^2\right]}{E\left[\left|\boldsymbol{e}(n)\right|^2\right]}$，其中 $E\left[\left|\boldsymbol{\varepsilon}(n)\right|^2\right]=E\left[\left|\boldsymbol{W}_{opt}-\hat{\boldsymbol{W}}(n)\right|^2\right]$，是权向量均方偏差， $\boldsymbol{W}_{opt}$ 是最优维纳解， $\hat{\boldsymbol{W}}(n)$ 是第 n 次迭代中得到的估值。 $E\left[\left|\boldsymbol{x}(n)\right|^2\right]$ 是带噪输入信号的功率， $E\left[\left|\boldsymbol{e}(n)\right|^2\right]$ 是误差信号功率。

初始化：

如果知道抽头权向量 $\hat{\boldsymbol{W}}(n)$ 的先验知识，则用其来初始化 $\hat{\boldsymbol{W}}(0)$，否则令 $\hat{\boldsymbol{W}}(0)=\boldsymbol{0}$

数据：

A）给定的 $\boldsymbol{X}(n)$ =第 n 时刻 $M\times1$ 抽头输入向量，

$d(n)$ =第 n 时间步的期望响应

B）要计算的： $\hat{\boldsymbol{W}}(n+1)$=第 n+1 步抽头权向量的估计

计算：

对 n=0,1,2，…计算

$$e(n)=d(n)-\hat{\boldsymbol{W}}^{H}(n)\boldsymbol{X}(n)$$

$$\boldsymbol{W}_{n+1}=\boldsymbol{W}_{n}-\mu\hat{\nabla}_{n}=\boldsymbol{W}_{n}+\frac{\mu}{N}\frac{e(n)\boldsymbol{x}(n)}{\hat{\sigma}_{x}^{2}(n)}$$

2.3.2 SE-LMS

Signed-errorLMS（SE-LMS）算法将误差$e(n)$用$[-1\quad 0\quad 1]$这三个量化值来代替，如果误差大于 0，则将$e(n)$赋值为 1，其它类推。这时式 2.3.4 退化成如下形式：

$$\boldsymbol{W}_{k+1}=\boldsymbol{W}_{k}-\mu\hat{\nabla}_{k}=\boldsymbol{W}_{k}+2\mu\boldsymbol{X}_{k}^{*}sign(e[n])$$

该算法在加快运算速度的同时简化了电路结构，μ 设置成 2 的指数时，通过移位就可以实现这里的乘法操作。降低了硬件实现的复杂度。

2.3.2 SD-LMS

Signal-dependent LMS 算法和 SE-LMS 很相似，误差也是只取$[-1\quad 0\quad 1]$这三个值，不同的是，其选择是以采样到的误差信号为参考的，如果$x(n)$大于 0，则$e(n)$用 1 代替，依次类推。

$$\boldsymbol{W}_{k+1}=\boldsymbol{W}_{k}-\mu\hat{\nabla}_{k}=\boldsymbol{W}_{k}+2\mu sign(x[n])$$

2.3.2 SS-LMS

Sign-sign LMS 算法既考虑输入信号的符号又考虑误差的信号。

$$\boldsymbol{W}_{k+1}=\boldsymbol{W}_{k}-\mu\hat{\nabla}_{k}=\boldsymbol{W}_{k}+2\mu sign(\boldsymbol{x}[n])sign(e[n])$$

2.3.2 LLMS

Leaky LMS 算法减轻了系数溢出问题。其不仅考虑了均方误差$e^{2}(n)$也考虑了滤波器系数。其权向量更新方程如下：

$$\boldsymbol{W}_{k+1}=(1-\mu\alpha)\boldsymbol{W}_{k}-\mu\hat{\nabla}_{k}=\boldsymbol{W}_{k}+2\mu e(n)\boldsymbol{x}(n)$$

2.3.2 LNLMS

Leaky NLMS 是归一化的 LLMS 算法。

$$\boldsymbol{W}_{n+1}=(1-\mu\alpha)\boldsymbol{W}_{n}-\mu\hat{\nabla}_{n}=(1-\mu\alpha)\boldsymbol{W}_{n}+\frac{\mu}{N}\frac{e(n)\boldsymbol{x}(n)}{\hat{\sigma}_{x}^{2}(n)}$$

2.4 块自适应滤波

2.4.1 块自适应滤波器

计算过程如下，对参考信号 x 分段并做 FFT 变换，分别对各段数据做频域滤波，累加后做 FFT 反变换，并只取后 L(L 是原始信号的分段后的长度)点为有效的线性卷积结果，得到的是估计信号，将估计信号从回声信号中去除，得残差信号。计算子带步长，调整各段滤波器系数。这一过程表示如下图。

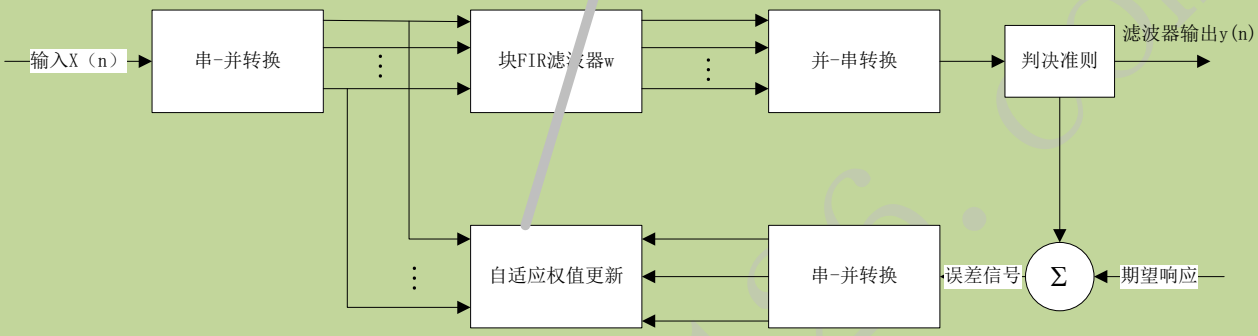


图 2.2 块自适应滤波器

设 n 时刻输入序列 $x(n)$ 如下：

$$X(n)=\left[x(n),x(n-1),...,x(n-M+1)\right]^T \tag{2.4.1}$$

对应于长度为 M 的 FIR 滤波器在 n 时刻的抽头权向量为：

$$\hat{W}(n)=\left[\hat{w}_0(n),\hat{w}_1(n-1),...,\hat{w}_{M-1}(n)\right]^T \tag{2.4.2}$$

根据 FIR 滤波器原理：

$$y(n)=x(n)\times\hat{w}_0(n)+x(n-1)\times\hat{w}_1(n)+...+x(n-M+1)\times\hat{w}_{M-1}(n) \tag{2.4.3}$$

用向量可以表示成如下：

$$y(n)=X(n)^T\hat{W}(n) \tag{2.4.4}$$

下面对 x(n)进行分块，设 k 表示块下标，它与原始样值时间 n 的关系为：

$$n=kL+i,i=0,1,...,L-1;k=1,2,... \tag{2.4.5}$$

其中 L 是块的长度。第 k 块的数据为 $\{X(kL+i)\}_{i=0}^{L-1}$,其矩阵表示形式如下：

$$A^T(k)=\left[x(kL),x(kL+1),...,x(kL+L-1)\right] \tag{2.4.6}$$

将滤波器对输入块 $A(k)$ 的响应表示如下：

$$y(kL+i)=\hat{W}^T(k)A(k)=\sum_{j=0}^{M-1}\hat{w}_j(k)x(kL+i-j),i=0,1,\ldots,L-1 \tag{2.4.7}$$

设 $d(kL+i)$ 表示期望信号，误差信号表示如下：

$$e(kL+i)=d(kL+i)-y(kL+i) \tag{2.4.8}$$

考虑滤波器长度 M=3，块长度 L=3,其三个相邻的数据块是 k-1，k， k+1，则

k-1 滤波结果如下：

$$\begin{vmatrix} y(3k-3) \\ y(3k-2) \\ y(3k-1) \end{vmatrix} = \begin{vmatrix} x(3k-3) & x(3k-4) & x(3k-5) \\ x(3k-2) & x(3k-3) & x(3k-4) \\ x(3k-1) & x(3k-2) & x(3k-3) \end{vmatrix} \begin{vmatrix} w_0(k-1) \\ w_1(k-1) \\ w_2(k-1) \end{vmatrix} \tag{2.4.9}$$

k 块滤波结果如下：

$$\begin{vmatrix} y(3k) \\ y(3k+1) \\ y(3k+2) \end{vmatrix} = \begin{vmatrix} x(3k) & x(3k-1) & x(3k-2) \\ x(3k+1) & x(3k) & x(3k-1) \\ x(3k+2) & x(3k+1) & x(3k) \end{vmatrix} \begin{vmatrix} w_0(k) \\ w_1(k) \\ w_2(k) \end{vmatrix} \tag{2.4.10}$$

k+1 块滤波器结果如下：

$$\begin{vmatrix} y(3k+3) \\ y(3k+4) \\ y(3k+5) \end{vmatrix} = \begin{vmatrix} x(3k+3) & x(3k+2) & x(3k+1) \\ x(3k+4) & x(3k+3) & x(3k+2) \\ x(3k+5) & x(3k+4) & x(3k+3) \end{vmatrix} \begin{vmatrix} w_0(k+1) \\ w_1(k+1) \\ w_2(k+1) \end{vmatrix} \tag{2.4.11}$$

上面的数据矩阵是托伯利兹矩阵，主对角线元素都相同。

2.4.2 块 LMS

权向量调整公式如下：

（权向量的调整）=（步长参数）*（抽头输入向量）*（误差信号）

因为在块 LMS 算法中误差信号随抽样速率而变，对于每一个数据块，我们有不同的用于自适应过程的误差信号值。因此，每一个块的抽头权向量更新公式如下：

$$\hat{w}(k+1)=\hat{w}(k)+\mu\sum_{i=0}^{L-1}x(kL+i)e(kL+i) \tag{2.4.12}$$

其梯度向量的估计如下：

$$\hat{\nabla}(k)=-\frac{2}{L}\sum_{i=0}^{L-1}x(kL+i)e(kL+i) \tag{2.4.13}$$

$\hat{\nabla}(k)$ 的无偏估计如下：

$$\hat{w}(k+1)=\hat{w}(k)-\frac{1}{2}\mu_B\hat{\nabla}(k) \tag{2.4.14}$$

2.4.3 块 LMS 算法收敛性

由于时间平均的缘故，它具有估计精度随快长度增加而大幅提高的特性。然而，长度的增加会导致其收敛速度进一步减慢。后文的快速 LMS 算法加速了这一过程。

➢ 平均时长数

$$\tau_{mse,av} = \frac{1}{2\mu_B \lambda_{av}} \tag{2.4.15}$$

其中 λ_{av} 是输入自相关矩阵 $R = E\left[x(n)x(n)^T\right]$

上式中为了使零阶公式成立，μ_B 必须小于 $1/\lambda_{\max}$ ，其中 $\lambda_{\max}$ 是相关矩阵的最大特征值。

➢ 失调

$$t\upsilon = \frac{\mu_B}{2L} tr[R] \tag{2.4.16}$$

$tr[B]$ 是相关矩阵的迹。

2.4.4 块长的选择

设滤波器长度 M 和块长度 L 的关系有三种可能：

1. L=M，从计算的复杂性上看，最佳

2. L<M，有降低延迟的好处。

3. L>M,将产生自适应过程冗余运算。

2.5 FLMS

FLMS（Fast LMS）的基本思想是将时域块 LMS 放到频域来计算。利用 FFT 算法在频域上完成滤波器系数的自适应。快速卷积算法用重叠相加法和重叠存储法。重叠相加法是将长序列分成大小相等的短片段，分别对各个端片段做 FFT 变换，再将变换重叠的部分相加构成最终 FFT 结果，重叠存储法在分段时，各个短的段之间存在重叠，对各个段进行 FFT 变换，最后将 FFT 变换得结果直接相加即得最终变换结果。当块的大小和权值个数相等时，运算效率达到最高。

根据重叠存储方法，将滤波器 M 个抽头权值用等个数的零来填补，并采用 N 点 FFT 进行计算，其中 $N = 2M$ ，因此，N*1 的向量：

$$\hat{W}(k) = FFT\begin{bmatrix} \hat{w}(k) \\ 0 \end{bmatrix} \tag{2.5.1}$$

表示 FFT 补零后的系数，抽头权向量为 $\hat{w}(k)$。值得注意的是频域权向量 $\hat{W}(k)$ 的长度是时域权向量 $\hat{w}(k)$ 长度的两倍。相应的令：

$$X(k)=diag\left\{FFT\begin{bmatrix}x(kM-M),\ldots,x(kM-1), & x(kM),\ldots,x(kM+M-1),\\ K-1,block & K,block\end{bmatrix}\right\} \quad 2.5.2$$

表示对输入数据的两个相继子块进行傅里叶变换得到一个 $N\times N$ 对角阵。

将重叠存储法应用于 2.4.7 得。

$$\begin{aligned}y^T(k)&=\left[y(kM),y(kM+1),...,y(kM+M-1)\right]\\&=IFFT\left[\boldsymbol{X}(k)\hat{\boldsymbol{W}}(k)\right],lastM\end{aligned} \quad 2.5.3$$

是 2.5.3 只有最后 M 个元素被保留，因为最前面的 M 个元素是循环卷积的结果。

设第 K 块的 $M\times 1$ 期望响应和误差信号分别如下：

$$\boldsymbol{d}(k)=\left[d(kM),d(kM+1),\ldots,d(kM+M-1)\right]^T \quad 2.5.4$$

$$\boldsymbol{e}(k)=\left[e(kM),e(kM+1),\ldots,e(kM+M-1)\right]^T=\boldsymbol{d}(k)-\boldsymbol{y}(k) \quad 2.5.5$$

根据式 2.5.3，可将 $\boldsymbol{e}(k)$ 变换到频域，即

$$E(k)=FFT\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{e}(k)\end{bmatrix} \quad 2.5.6$$

则在更新权值的相关矩阵如下：

$$\boldsymbol{\Phi}(k)=\sum_{i=0}^{L-1}x(kL+i)e(kL+i)=IFFT\left[\boldsymbol{X}^T(k)\boldsymbol{E}(k)\right] \quad ，的最前面 M 个元素 \quad 2.5.7$$

则抽头的更新过程在频域中的表现如下：

$$\hat{\boldsymbol{W}}(k+1)=\hat{\boldsymbol{W}}(k)+\mu FFT\begin{bmatrix}\Phi(k)\\ \boldsymbol{0}\end{bmatrix} \quad 2.5.8$$

2.6 MDF 自适应权值调整

时域解

对于 N 阶 NLMS 算法，其误差调节向量如下式：

$$e(n)=d(n)-\hat{y}(n)=d(n)-\sum_{k=0}^{N-1}\hat{w}_k(n)x(n-k) \quad 2.6.1$$

权值更新如下：

$$\begin{aligned}\hat{w}_k(n+1)&=\hat{w}_k(n)+\mu\frac{e(n)x^*(n-k)}{\sum_{i=0}^{N-1}|x(n-i)|^2}\\&=\hat{w}_k(n)+\mu\frac{\left(d(n)-\sum_i\hat{w}_i(n)x(n-i)\right)x^*(n-k)}{\sum_{i=0}^{N-1}|x(n-i)|^2}\end{aligned}\tag{2.6.2}$$

其中 $x(n)$ 是参考信号，$\hat{w}_k(n)$ 是 n 时刻和步长 μ 的权值更新。假设滤波后的误差为 $\sigma_k(n)=\hat{w}_k(n)-w_k(n)$ ，$d(n)=v(n)+\sum_k\hat{w}_k(n)x(n-k)$,则误差的迭代关系如下：

$$\delta_k(n+1)=\delta_k(n)+\mu\frac{\left(v(n)-\sum_i\delta_i(n)x(n-i)\right)x^*(n-k)}{\sum_{i=0}^{N-1}|x(n-1)|^2}\tag{2.6.3}$$

在每一次调节中，滤波器的误差估计为 $\Lambda(n)=\sum_k\delta_k^*(n)\delta_k(n)$ ，展开后得如下形式：

$$\mathrm{A}(n+1)=\sum_{k=0}^{N-1}\left|\delta_k(n)+\mu\frac{\left(v(n)-\sum_i\delta_i(n)x(n-i)\right)x^*(n-k)}{\sum_{i=0}^{N-1}|x(n-i)|^2}\right|^2\tag{2.6.4}$$

如果 $x(n)$ 和 $v(n)$ 是不相关的白噪声信号，则下式：

$$E\{\Lambda(n+1)\,|\,\Lambda(n),x(n)\}=\Lambda(n)\left[1-\frac{2\mu}{N}+\frac{\mu^2}{N}+\frac{\mu^2\sigma_v^2}{\Lambda(n)\sum_{i=0}^{N-1}|x(n-i)|^2}\right]\tag{2.6.5}$$

可以通过求解 $\partial E\{\Lambda(n+1)\}/\partial\mu=0,\Lambda\neq0$：

$$\frac{-2}{N}+\frac{2\mu}{N}+\frac{2\mu\sigma_v^2}{\Lambda(n)\sum_{i=0}^{N-1}|x(n-1)|^2}=0\tag{2.6.6}$$

求解后得到最优步长：

$$\mu_{opt}(n)=\frac{1}{1+\frac{\sigma_v^2}{\Lambda(n)(1/N)\sum_{i=0}^{N-1}|x(n-i)|^2}}\tag{2.6.7}$$

期望 $\Lambda(n)(1/N)\sum_{i=0}^{N-1}|x(n-i)|^2$ 等于剩余回声的方差 $\sigma_r^2(n)$ ，如果剩余回声的方差值等于 0，则步长因子等于 1，$r(n)=y(n)-\hat{y}(n)$ ，则有输出信号的方差是：

$$\sigma_e^2(n)=\sigma_v^2(n)+\sigma_r^2(n)\tag{2.6.8}$$

这样可以求得这种情况下的最优步长因子为：

$$\mu_{opt}(n)\approx\frac{\sigma_r^2(n)}{\sigma_e^2(n)}\tag{2.6.9}$$

则最优步长因子如下：

$$\hat{\mu}_{opt}(n)=\min\left(\frac{\hat{\sigma}_r^2(n)}{\hat{\sigma}_e^2(n)},1\right) \tag{2.6.10}$$

当 $\Lambda(n)\approx\dfrac{\sigma_v^2}{\sigma_x^2\left(\dfrac{2}{\mu}-1\right)}$ 时，式 2.6.5 的迭代将停止（滤波器系数不在更新，$E\{\Lambda(n+1)\}=\Lambda(n)$ ）。将 2.6.9 带入 2.6.10 得到在滤波器系数不更新情况下的剩余回声：

$$\sigma_r^2(n)\approx\min\left(\frac{1}{2}\hat{\sigma}_r^2(n),\sigma_v^2(n)\right) \tag{2.6.11}$$

频域解

和时域相比，频域可以使步长因子 $\mu(k,l)$ 按频域划分，$Y(k,l)$ 和 $E(k,l)$ 分别是频域中的记号，其和时域中的 $\hat{y}(n)$ 和 $e(n)$ 是对等的关系。k 是频域索引，l 是帧索引。和 2.6.9 类似，可得频域步长因子如下：

$$\mu_{opt}(k,l)\approx\frac{\sigma_r^2(k,l)}{\sigma_e^2(k,l)} \tag{2.6.12}$$

假设滤波器有一个和频谱无关的泄露（滤波器的误差）系数 $\eta(l)$ ，这将得到：

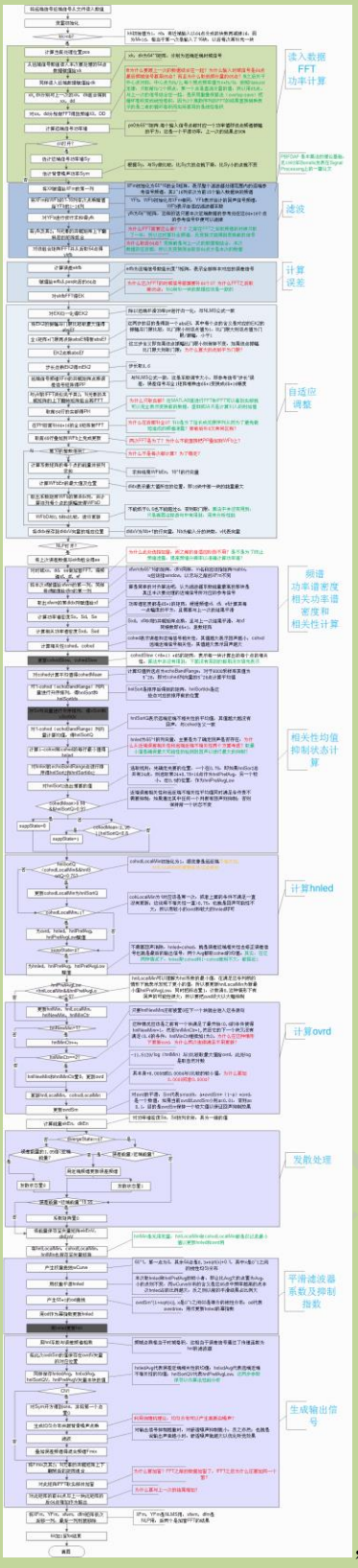
$$\hat{\sigma}_r^2(k,l)=\hat{\eta}(l)\hat{\sigma}_{\hat{Y}}^2(k,l) \tag{2.6.13}$$

$\eta(l)$ 实际上是滤波器的回声返回损失增强（ERLE）。

为了让步长因子调节的更快，使用瞬时估计，$\hat{\sigma}_y(k,l)=|Y(k,l)|^2$ 和 $\hat{\sigma}_e(k,l)=|E(k,l)|^2$，这将使 2.6.10 步长因子调整变为下式：

$$\hat{\mu}_{opt}(k,l)=\min\left(\hat{\eta}(l)\frac{|\hat{Y}(k,l)|^2}{|E(k,l)|^2},\mu_{\max}\right) \tag{2.6.14}$$

$\mu_{\max}$ 是小于等于 1 的数，以确保滤波器稳定。



第三章 语音阵列信号处理

和单麦克相比，麦克风阵列能够提供空域信息，这可以用来实现语音波束形成、盲信号分离以及声源定位。麦克风阵列主要分为两大类，它们是固定波束形成和自适应波束形成。固定波束形成包括：

- 延迟和波束形成
- 权重和波束形成
- 滤波和波束形成

自适应波束形成包括：

- LCMV 波束形成
- 广义互相关

这两类各有特点也各有用途。

麦克风阵列设计对语音信号的波束形成性能和噪声抑制性能影响较大。麦克风阵列的设计需要考虑的因数包括：麦克风个数，阵列的空间布局以及使用的麦克风类型。麦克风设计需要时刻考虑如下要数：

- 对噪声的抑制性能取决于信号的入射角和频率。
- 增加麦克风可以增加噪声的抑制能力。
- 在超过空域奈奎斯特定理的前提下，增加麦克风之间的间距能够增加噪声抑制能力，违反奈奎斯特准则时会导致空域频谱泄露，进而导致噪声抑制能力下降。
- 麦克风间的空间布局决定麦克风波束形成方向图。
- 和各向同性麦克相比，定向麦克风可以增加指向性。

麦克风阵列的语音波束方向图

- 波束方向图（指向图），通常针对远场语音模型而言。2D，3D 以及极坐标表示。
- 指向性指数，是频率的函数，常用 dB 为值来测量，指示的是麦克风阵列的指向性。
- 半功率波束宽度（波束宽度），指主瓣峰值的 1/2（3dB）的宽度。
- 第一空波束宽度，决定了空域分辨两个声源的能力，即两个声源接近到何程度，麦克风阵列无法分辨。
- 阵列增益。全方位信号和阵列输入信号之间的增益。

阵列设计的任务是在一个期望的方向图的前提下，设计在旁瓣幅度值允许的条件下，最小的波束宽度。阵列方向图设计的方法有如下几种：

- 延迟和技术
- 恒定束宽波束形成技术
- 最小方差无失真波束响应波束形成

- Frost 波束形成
- 广义旁瓣相消技术（GSC）

究竟如何选择合适的语音波束形成技术，需要理清如下问题：

✧ 设备使用场景，声源和麦克风的相对位置是运动还是静止的。

✧ 麦克风阵列的限制是什么？麦克风个数和它们的间距。

✧ 设备所在环境的噪声源通常是什么？

✧ 算法依赖平台的内存和处理能力。

3.1 阵列模型

3.1.1 线阵模型

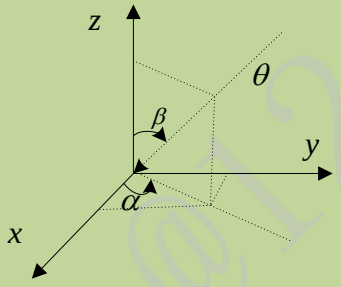


图 3.1 入射角三维坐标

入射方向的单位向量可表示为：

$$v(\theta)=\left[\sin(\beta)\cos(\alpha)\quad \sin(\beta)\sin(\alpha)\quad \cos(\beta)\right]^T \tag{3.1}$$

其中，α 和β 分别水平方位角和垂直俯仰角。是对于该坐标系中的任意一点，其可以使用 x，y，z 三个坐标轴表示出来。

$$p_m=\left[x_m\quad y_m\quad z_m\right]^T \tag{3.1}$$

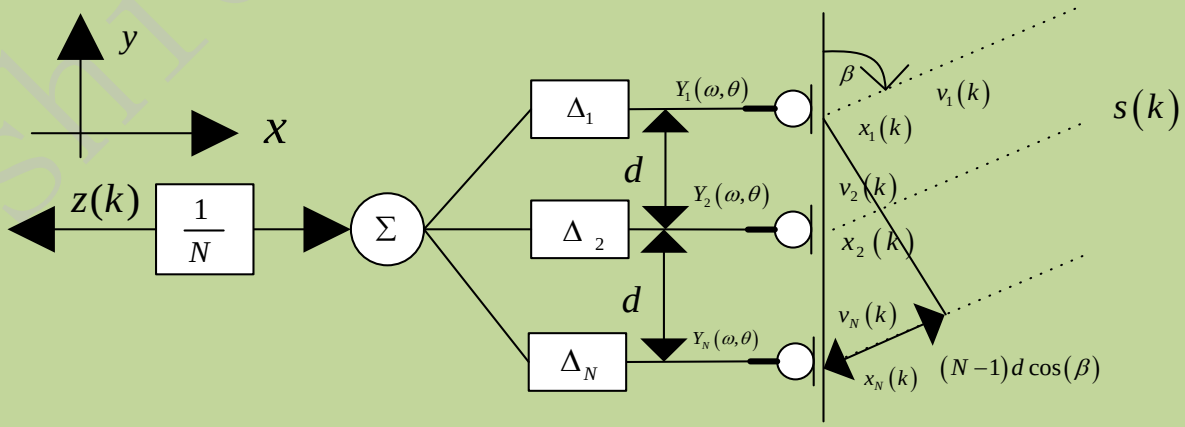


图 3.2 线阵模型

如上图所示线阵模型，设声波到麦克风阵列时，其波前可以被看成是平面波，入射角是$\theta=(\alpha,\beta)$，即于平面法线方向的夹角。麦克风阵列 TDOA（time difference of array）的计算

公式如下：

$$TDOA=\tau_n(\theta)=\frac{v^T(\theta)\,p_n}{c},n=1,\ldots,N \tag{3.1}$$

c 表示声速，n 是麦克风的索引，θ 是声源的入射角（麦克风阵列波束形成方向）。为简化阵列模型，现将阵列降为二维处理，

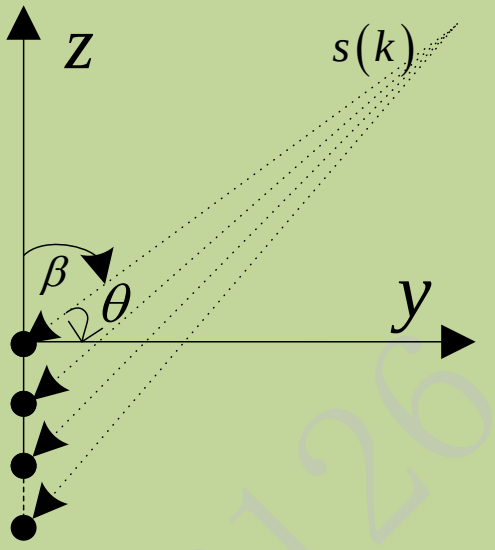


图 3.3 降维后入射角关系

虽然到达麦克风的声波并不平行，但在平面波模型下，可以认为同一声源发出的声音到达每个麦克风声波是平行的，即他们的入射角都是θ ，这时θ 退化为 yz 平面的角度，在 z 轴方向上和 y 轴方向上声源到各个麦克的时间均有差异。设麦克风之间的距离为 d，则在 z 轴上的时间差（式 3.1 退化）为：

$$\tau_n(\theta)=\frac{(n-1)d\cos\theta}{c} \tag{3.2}$$

则各个麦克风的输出$y_n(k)$ 为：

$$y_n(k)=s\left(k-\tau_n(\theta)\right)+v_n(k),n=1,2,\ldots,N \tag{3.3}$$

其中，$v_n(k)$为噪声，$s\left(k-\tau_n(\theta)\right)$是声源$s(k)$经过 TDOA 后的值。将阵列收到的信号使用向量表示如下：

$$x_s(t)=\begin{bmatrix}s_1(t) & s_2(t) & \ldots & s_N(t)\end{bmatrix} \tag{3.4}$$

对式 3.3 进行傅里叶变换后，得如下公式：

$$Y_n(\omega,\theta)=\int_{-\infty}^{+\infty}y_n(t)e^{-j\omega t}dt=\int_{-\infty}^{+\infty}y\left(t-\tau_n(\theta)\right)e^{-j\omega t}dt=e^{-j\omega\tau_n(\theta)}S(\omega) \tag{3.5}$$

再对 y 轴做一次类似的延迟变换求解:

$$z[k]=\frac{1}{N}\sum_{n=1}^{N}y_n[k+\Delta_n],\Delta_n=\frac{(n-1)d\cos(\beta)}{c} \tag{3.6}$$

对式 3.6 做傅里叶变换得：

$$Z(\omega,\theta)=\frac{1}{N}\sum_{n=1}^{N}e^{j\omega\Delta_n}Y_n(\omega,\theta)=\frac{1}{N}\sum_{n=1}^{N}e^{j\omega\Delta_n}S(\omega)e^{-j\omega\tau_n(\theta)} \tag{3.7}$$

根据$Z(\omega,\theta)$和$S(\omega)$可得到响应函数如下：

$$H(\omega,\theta)=\frac{Z(\omega,\theta)}{S(\omega)}=\frac{1}{N}\sum_{n=1}^{N}e^{-j\omega(n-1)\frac{d(\cos(\theta)-\cos(\beta))}{c}} \tag{3.8}$$

由此可求得天线的方向图(使用到三角中的 2 倍角公式)：

$$\left|H(\omega,\theta)\right|=\left|\frac{1-e^{-jN\omega\frac{d(\cos\theta-\cos\beta)}{c}}}{N(1-e^{-j\omega\frac{d(\cos\theta-\cos\beta)}{c}})}\right|=\left|\frac{1-\cos(N\omega\frac{d(\cos\theta-\cos\beta)}{c})}{N(1-\cos(\omega\frac{d(\cos\theta-\cos\beta)}{c})}\right| \tag{3.9}$$

$$=\left|\frac{\sin(\frac{N\omega}{2}\frac{d(\cos\theta-\cos\beta)}{c})}{N(\sin(\frac{\omega}{2}\frac{d(\cos\theta-\cos\beta)}{c})}\right|$$

根据麦克风个数，麦克风间距以及频率三个变量（同时设β 成 90°）可以画出如下三张图：

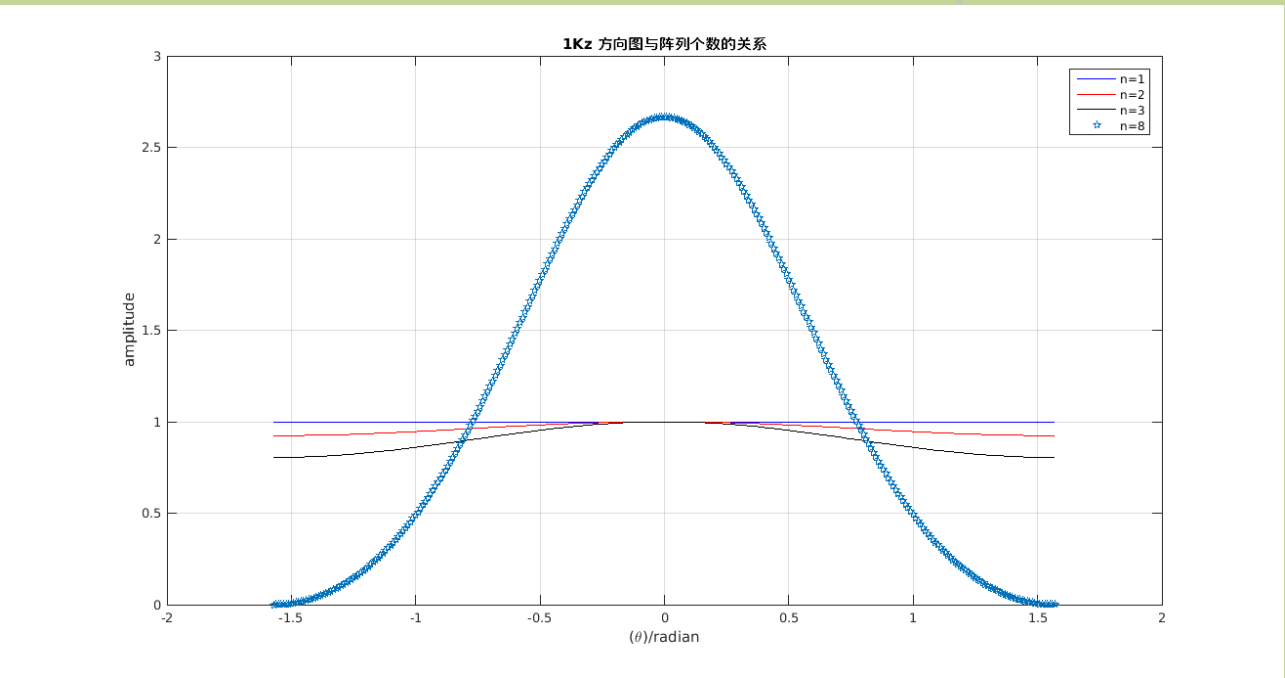


图 3.4 麦克风数量与波束方向图

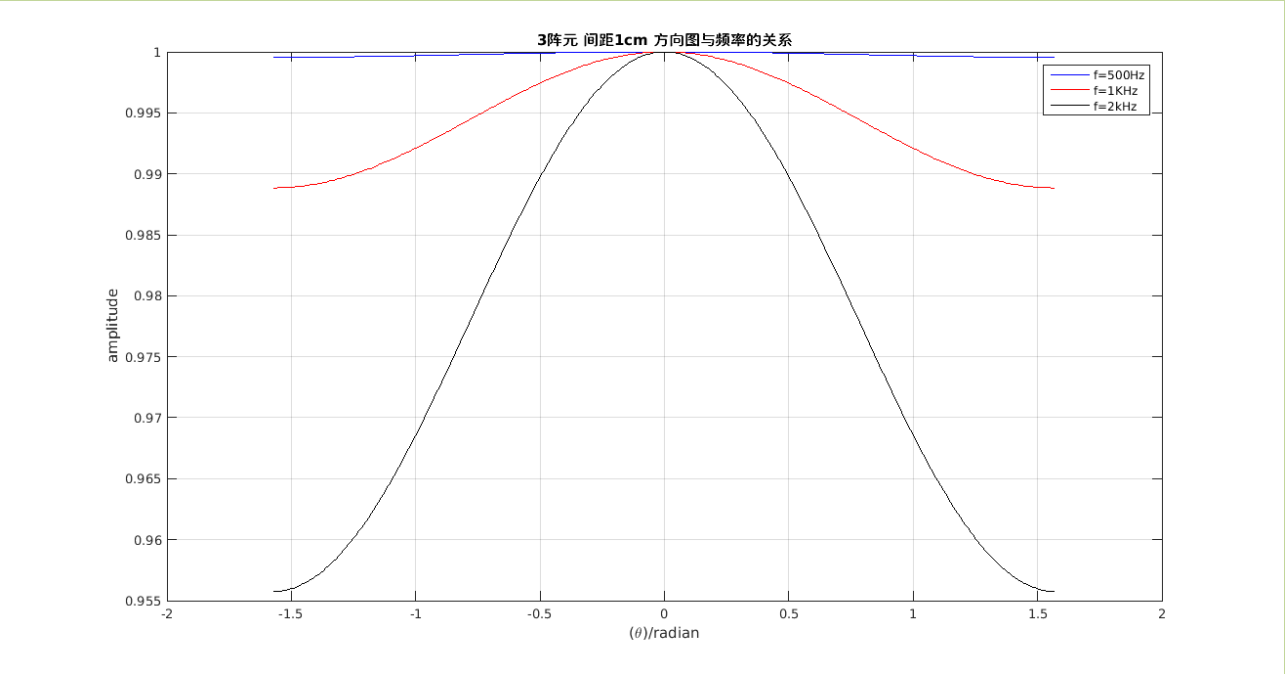


图 3.5 3 麦克与波束方向图关系

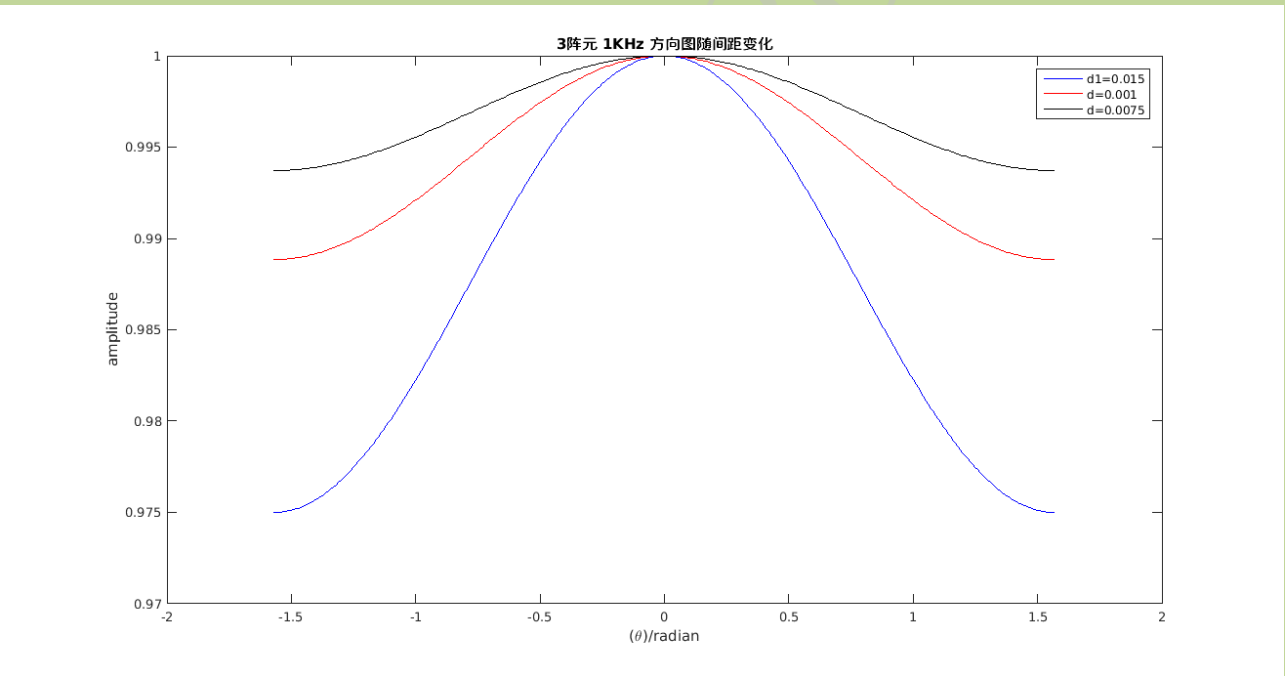


图 3.5 3 麦克间距与波束方向图关系

概括上述三张图可知，麦克风数量越多，其波束指向性越好，在满足频域奈奎斯特采样定理前提下，麦克风间距越大其空间分辨率越好。当信号频率越高时，麦克阵列的作用越明显。

图 3.5 3 麦克与波束方向图关系

```
clear;
v=343;f=1000;
%Number of element
sita=-pi:0.01:pi;
lamda=v/f;
d=lamda/8;
n1=1;
beta=2*pi*d*sin(sita)/lamda;
```

```
z11=(n1/2)*beta;
z21=(1/2)*beta;
f1=sin(z11)./(n1*sin(z21));
F1=abs(f1);
figure(1);
% plot(sita,F1,'b');
polar(sita, F1,'b');
hold on;
n2=2;
beta=2*pi*d*sin(sita)/lamda;
z12=(n2/2)*beta;
z22=(1/2)*beta;
f2=sin(z12)./(n2*sin(z22));
F2=abs(f2);
% plot(sita,F2,'r');
polar(sita, F2,'r');
hold on;
n3=3;
beta=2*pi*d*sin(sita)/lamda;
z13=(n3/2)*beta;
z23=(1/2)*beta;
f3=sin(z13)./(n3*sin(z23));
F3=abs(f3);
polar(sita, F3,'k');
% plot(sita,F3,'k');
hold on
n4=8;
beta=2*pi*d*sin(sita)/lamda;
z14=(n4/2)*beta;
z24=(1/2)*beta;
f4=sin(z14)./(n4*sin(z24));
F4=abs(f4);
% plot(sita, F4, 'p');
polar(sita, F4,'p');
hold off;
grid on;
xlabel('(\theta)/radian');
ylabel('amplitude');
title('1Kz 方向图与阵列个数的关系‘);
legend('n=1','n=2','n=3', 'n=8');

% Wave length
clear;figure;
v=343;f=1000;
sita=-pi/2:0.01:pi/2;
n=3;
d=0.01;
lamda1=v/(f/5);
beta=2*pi*d*sin(sita)/lamda1;
z11=(n/2)*beta;
z21=(1/2)*beta;
```

```
f1=sin(z11)./(n*sin(z21));
F1=abs(f1);
polar(sita, F1,'b');
hold on;
lamda2=(v/f);
beta=2*pi*d*sin(sita)/lamda2;
z12=(n/2)*beta;
z22=(1/2)*beta;
f2=sin(z12)./(n*sin(z22));
F2=abs(f2);
polar(sita, F2,'r');
hold on;
lamda3=v/(f*2);
beta=2*pi*d*sin(sita)/lamda3;
z13=(n/2)*beta;
z23=(1/2)*beta;
f3=sin(z13)./(n*sin(z23));
F3=abs(f3);
polar(sita, F3,'k');
% plot(sita,F1,'b',sita,F2,'r',sita,F3,'k');
hold off;
grid on;
xlabel('(\theta)/radian');
ylabel('amplitude');
title('3阵元 间距1cm 方向图与频率的关系');
legend('f=500Hz','f=1KHz','f=2kHz');


clear;figure;
v=343;f=1000;
sita=-pi:0.01:pi;
n=3;
lamda=v/f;
d1=0.015;
beta=2*pi*d1*sin(sita)/lamda;
z11=(n/2)*beta;
z21=(1/2)*beta;
f1=sin(z11)./(n*sin(z21));
F1=abs(f1);
polar(sita, 10*log(F1),'b');
% plot(sita,F1,'b');
hold on;
d2=0.01;
beta=2*pi*d2*sin(sita)/lamda;
z12=(n/2)*beta;
z22=(1/2)*beta;
f2=sin(z12)./(n*sin(z22));
F2=abs(f2);
% plot(sita,F2,'r');
polar(sita, 10*log(F2),'r');
hold on;
```

```
d3=0.0075;
beta=2*pi*d3*sin(sita)/lamda;
z13=(n/2)*beta;
z23=(1/2)*beta;
f3=sin(z13)./(n*sin(z23));
F3=abs(f3);
% plot(sita,F3,'k');
polar(sita, 10*log(F3),'k');
hold off;
hold on;
xlabel('(\theta)/radian');
ylabel('amplitude dB');
title('3阵元 1KHz 方向图随间距变化');
legend('d1=0.015','d=0.001','d=0.0075');
```

3.1.2 面阵模型

3.1.3 圆阵模型

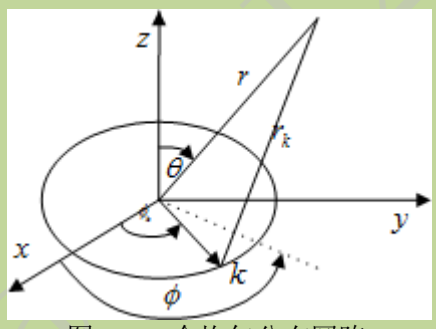


图 3.6 N 个均匀分布圆阵

在一个半径为 R 的圆周上，均匀分布 N 个相同的各向同性阵元。其中ϕ 是起始于 x 轴正向的方位角，θ 是起始于 z 轴正向的俯仰角。

假设每个阵元的权重为$W_n(n=0,1,...,N-1)$ ，则第 N 个阵元的方位角$\phi_n=2n\pi/N$ 。如果来自某一(θ,ϕ) 方向的平面波照射在圆形阵列上，则第 n 个阵元相对于阵列中心的相对相位为：

$$\beta_n=-kR\cos(\phi-\phi_n)\sin(\theta) \qquad 3.10$$

则 N 个均匀分布阵元所形成的圆形阵列因子的方向图函数为：

3.2 阵列波束形成技术

3.2.1 DS 模型

Delay and Sum（延迟和）

延迟调节的是接收天线的方向，和是滤波器系数，决定主瓣和旁瓣的大小和形状。

N 阶空域滤波器将有 N-1 个零点。可以设计何时的不同的权系数以抑制特定方向的干扰信号。

当麦克风阵列的间距增加时，阵列的波束宽度减小。所以如果当你需要一个尖锐的波束，可以简单的增加麦克风间距。但是这会增加麦克风阵列的孔径，这通常也意味着引入更多噪声。所以，在阵列设计时，我们期望麦克风的阵列的间距越大越好。但是当间距大于 $\frac{\lambda}{2}=\frac{c}{2f}$ 时，将会产生空域混叠。λ 是信号波长。

为了避免空域混叠，阵列间距必须满足 $d \le \frac{\lambda}{2}=\frac{c}{2f}$。人耳朵的听觉范围是 20Hz~20KHz，人发声的范围是 85Hz~1.1KHz，对应的阵列间距是 13.2cm。

N 个麦克风阵列的波束形成方程如下：

$$y_n(k)=\alpha_n s[k-t-\mathcal{F}_n(\tau)]+v_n(k)=x_n(k)+v_n(k), n=1,2,\ldots,N, \tag{3.11}$$

式中 α_n 取值介于 0 和 1 之间，是传输的衰减因子。$s(k)$ 是未知的语音信号（要识别的人说的话），t 是从语音信源传递到第一个传感器的时间。$v_n(k)$ 是第 n 个传感器的加性噪声。τ 是第一个和第二个麦克风的相对延迟，$\mathcal{F}_n(\tau)$ 是第一个麦克风相对第 n 个麦克风的传播延迟，$\mathcal{F}_1(\tau)=0$，$\mathcal{F}_2(\tau)=\tau$。

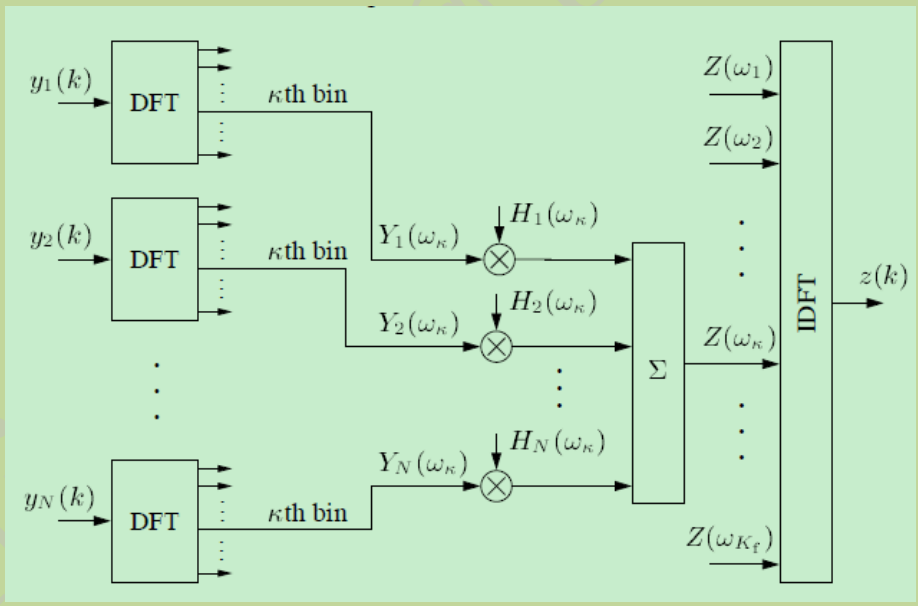


图 3.7 频域宽带波束形成

延迟和模型

将上式进行时移后得：

$$\begin{aligned} y_{a,n}(k) &= y_n[k+\mathcal{F}_n(\tau)]=\alpha_n s(k-t)+v_{a,n}(k) \\ &= x_{a,n}(k)+v_{a,n}(k), n=1,2,\ldots,N \end{aligned} \tag{3.11}$$

此时，$v_{a,n}(k)=v_n[k+\mathcal{F}_n(\tau)]$

接下来是将对其的信号进行累加得：

$$z_{DS}(k)=\frac{1}{N}\sum_{n=1}^{N}y_{a,n}(k)=\alpha_s s(k-t)+\frac{1}{N}v_s(k) \tag{3.12}$$

其中：

$$\alpha_s=\frac{1}{N}\sum_{n=1}^{N}\alpha_n \quad , \quad v_s(k)=\sum_{n=1}^{N}v_{a,n}(k)=\sum_{n=1}^{N}v_n[k+F_n(\tau)]$$

则输入和输出的信噪比为：

$$SNR=\frac{\sigma_{x_1}^2}{\sigma_{v_1}^2}=\alpha_1^2\frac{\sigma_s^2}{\sigma_{v_1}^2} \tag{3.13}$$

其中 $\sigma_{x_1}^2=E[x_1^2(k)]$， $\sigma_{v_1}^2=E[v_1^2(k)]$， $\sigma_s^2=E[s^2(k)]$ 。

则在 DS 后输出 SNR 可以表达如下：

$$oSNR=N^2\alpha_s^2\frac{E\left[s^2(k-t)\right]}{E\left(v_s^2(k)\right)}=N^2\alpha_s^2\frac{\sigma_s^2}{\sigma_{v_s}^2}=\left(\sum_{n=1}^{N}\alpha n\right)^2\frac{\sigma_s^2}{\sigma_{v_s}^2} \tag{3.14}$$

式 3.14 中 $\sigma_{v_s}^2=E\left\{\left[\sum_{n=1}^{N}v_n\left[k+\mathcal{F}_n(\tau)\right]\right]^2\right\}=\sum_{n=1}^{N}\sigma_{v_n}^2+2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\rho_{v_iv_j}$ ， $\rho_{v_iv_j}$

其中， $\sigma_{v_n}^2=E\left[v_n^2(k)\right]$ 是噪声 $v_n(k)$ 的方差， $\rho_{v_iv_j}=E\left\{v_i\left[k+\mathcal{F}_i(\tau)\right]v_j\left[k+\mathcal{F}_j(\tau)\right]\right\}$ 是噪声 $v_i(k)$ 和 $v_j(k)$ 的协方差。

特殊情况 1：

$$\rho_{v_iv_j}=0，\ oSNR=N\bullet SNR \tag{3.15}$$

特殊情况 2：

假设噪声的能量是一样的并且衰退因子等于 1。

$$oSNR=\frac{N}{1+\rho_s}\bullet SNR \tag{3.16}$$

$$\rho_s=\frac{2}{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\rho_{v_iv_j},\rho_{v_iv_j}=\frac{\mathrm{E}\left\{v_i\left[k+\mathcal{F}_i(\tau)\right]v_j\left[k+\mathcal{F}_j(\tau)\right]\right\}}{\sigma_{v_iv_j}} \tag{3.17}$$

3.2.2 最大信噪比

将 3.11 重写为如下式：

$$\boldsymbol{y}_a(k)=s(k-t)\boldsymbol{\alpha}+\boldsymbol{V}_a(k) \tag{3.18}$$

此处：

$$\boldsymbol{y}_a(k)=\left[y_{a,1}(k)\quad y_{a,2}(k)\quad \ldots\quad y_{a,N}(k)\right]^T,$$

$\boldsymbol{v}_a(k)=\left[v_{a,1}(k) \quad v_{a,2}(k) \quad \ldots \quad v_{a,N}(k)\right]^T$，

$\boldsymbol{\alpha}=\left[\alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_N\right]^T$

假设噪声和信号是不相关的，$\boldsymbol{y}_a(k)$的相关矩阵可以表示为如下形式：

$$\boldsymbol{R}_{y_a y_a}=\sigma_s^2\boldsymbol{\alpha}\boldsymbol{\alpha}^T+\boldsymbol{R}_{v_a v_a} \tag{3.19}$$

其中，$\boldsymbol{R}_{v_a v_a}=E\left[\boldsymbol{v}_a(k)\boldsymbol{v}_a^T(k)\right]$是噪声的相关矩阵。则波束形成的输出可写为如下形式：

$$z(k)=\boldsymbol{h}^T\boldsymbol{y}_a(k)=\sum_{n=1}^{N}h_n y_{a,n}(k)=s(k-t)\mathbf{h}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{h}^T\boldsymbol{v}_a(k) \tag{3.20}$$

此处：

$$h=\left[h_1 \quad h_2 \quad \ldots \quad h_N\right]^T \tag{3.21}$$

可得到信噪比如下：

$$SNR(\boldsymbol{h})=\frac{\sigma_s^2\left(\boldsymbol{h}^T\boldsymbol{\alpha}\right)^2}{\boldsymbol{h}^T\boldsymbol{R}_{v_a v_a}\boldsymbol{h}} \tag{3.22}$$

上述信噪比最大解，等价于求解下述特征值方程。

$$\sigma_s^2\boldsymbol{\alpha}\boldsymbol{\alpha}^T\boldsymbol{h}=\lambda\boldsymbol{R}_{v_a v_a}\boldsymbol{h} \tag{3.23}$$

设$\boldsymbol{R}_{v_a v_a}$可逆，则最优解是特征向量$\boldsymbol{h}_{\max}$和特征值$\lambda_{\max}$：

$$z_{\max}(k)=\boldsymbol{h}_{\max}^T y_a(k) \tag{3.24}$$

$$SNR(h_{\max})=\lambda_{\max} \tag{3.24}$$

3.2.3 最小方差无失真响应滤波器

Minimum variance distortionless response（MVDR），其基本思想是调节滤波系数$\boldsymbol{h}$最小化输出功率$E\left[z^2(k)\right]=\boldsymbol{h}^T\boldsymbol{R}_{y_a y_a}\boldsymbol{h}$，则 MVDR 可以表述为如下方程：

$$\min_{\boldsymbol{h}}\boldsymbol{h}^T\boldsymbol{R}_{y_a y_a}\boldsymbol{h},\boldsymbol{h}^T\boldsymbol{\alpha}=\alpha_1 \tag{3.25}$$

拉格朗日乘法器可以解 3.25，得：

$$h_{opt}=\alpha_1\frac{\boldsymbol{R}_{y_a y_a}^{-1}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T\boldsymbol{R}_{y_a y_a}^{-1}\boldsymbol{\alpha}} \tag{3.26}$$

3.2.4 线性约束最小方差

LCMV(linearly constrained minimum variance)，室内的麦克风阵列通常会接受到来自墙

壁、地板、天花板以及家具的回声，这一现象可以使用混响模型来表示：

$$y_n(k) = g_n * s(k) + v_n(k) = x_n(k) + v_n(k) \tag{3.27}$$

g_n 是第 n 个麦克风对语音信号的脉冲响应函数。上述模型的矩阵表示方式是：

$y_n(k) = \boldsymbol{g}_n^T \boldsymbol{s}(k) + v_n(k)$，这里：

$$\boldsymbol{g}_n = [g_{n,0} g_{n,1} \ldots g_{n,L_g-1}]^T,$$

$$\boldsymbol{s}(k) = [s(k) s(k-1) \ldots s(k-L_h+1)]^T$$

麦克风阵列可以重写为如下形式：

$$\boldsymbol{y}_n(k) = \boldsymbol{G}_n \boldsymbol{s}_L(k) + \boldsymbol{v}_n(k), n = 1, 2, \ldots, N \tag{3.28}$$

这里：

$$\boldsymbol{y}_n(k) = [y_n(k) \quad y_n(k-1) \quad \ldots \quad y_n(k-L_h+1)]^T,$$

$$\boldsymbol{v}_n(k) = [v_n(k) \quad v_n(k-1) \quad \ldots \quad v_n(k-L_h+1)]^T,$$

$$\boldsymbol{S}_L(k) = [s(k) \quad s(k-1) \quad \ldots \quad s(k-L+1)]^T,$$

$$G_n = \begin{bmatrix} g_{n,0} & \cdots & g_{n,L_g-1} & 0 & 0 & \cdots & 0 \\ 0 & g_{n,0} & \cdots & g_{n,L_g-1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & g_{n,0} & \cdots & g_{n,L_g-1} \end{bmatrix}$$

G_n 是 $L_h \times L, L = L_h + L_g - 1$

将 N 维观测的向量写作：

$$\boldsymbol{y}(k) = [y_1^T \quad y_2^T \quad \cdots \quad y_N^T]^T = \boldsymbol{G}\boldsymbol{S}_L(k) + \boldsymbol{v}(k) \tag{3.29}$$

其中：

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \\ \vdots \\ \boldsymbol{G}_N \end{bmatrix}_{NL_h XL}$$

$$\boldsymbol{V}(k) = \left[\boldsymbol{v}_1^T(k) \quad \boldsymbol{v}_2^T(k) \quad \cdots \quad \boldsymbol{v}_N^T(k)\right]^T$$

在信号和噪声不相关时，$NL_h \times NL_h$ 的协方差系数 $\boldsymbol{y}(k)$ 是：

$$\boldsymbol{R}_{yy} = E\left[\boldsymbol{y}(k)\boldsymbol{y}^T(k)\right] = \boldsymbol{G}\boldsymbol{R}_{SS}\boldsymbol{G}^T + \boldsymbol{R}_{vv} \tag{3.30}$$

LCMV 滤波器的解可按下式获得：

在满足 $\boldsymbol{G}^T\boldsymbol{h} = \boldsymbol{u}$ 条件下的 $\min_{\boldsymbol{h}} \boldsymbol{h}^T \boldsymbol{R}_{yy} \boldsymbol{h}$

其中：

$\boldsymbol{h}=\left[h_1^T \quad h_2^T \quad \cdots \quad h_N^T\right]^T$ ，$\boldsymbol{u}=[1 \quad 0 \quad \cdots \quad 0]^T$ ，这里的意义是在去混响的前提下，使噪声最小。

上式的最优解是：

$$\boldsymbol{h}_R = \boldsymbol{R}_{yy}^{-1}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{R}_{yy}^{-1}\boldsymbol{G})\boldsymbol{u}^{-1} \quad , \qquad 3.31$$

其中 R 表示的是混响模型，假设 $\boldsymbol{R}_{yy}$ 和 $\boldsymbol{R}_{vv}$ 是正定矩阵，$\boldsymbol{G}^T\boldsymbol{R}_{yy}^{-1}\boldsymbol{G}$ 非奇异的条件是 $NL_h \geq L$ ，这就意味着：

$$L_h \geq \frac{L_g - 1}{N - 1} \qquad 3.32$$

其解如下：

$$\boldsymbol{h}_R = \boldsymbol{R}_{vv}^{-1}\boldsymbol{G}\left(\boldsymbol{G}^T\boldsymbol{R}_{vv}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{u} \qquad 3.33$$

3.2.3 广义旁瓣相消

广义旁瓣相消技术将约束的 LCMV 最优问题变成了一个无约束问题。在广义旁瓣相消中，有：

$\boldsymbol{h} = \boldsymbol{f} - \boldsymbol{B}\boldsymbol{w}$ （7.23）

此处：

$\boldsymbol{f} = \boldsymbol{G}_{:1}\left[\boldsymbol{G}_{:1}^T \quad \boldsymbol{G}_{:1}^T\right]^{-1}\boldsymbol{u}$ （7.24）

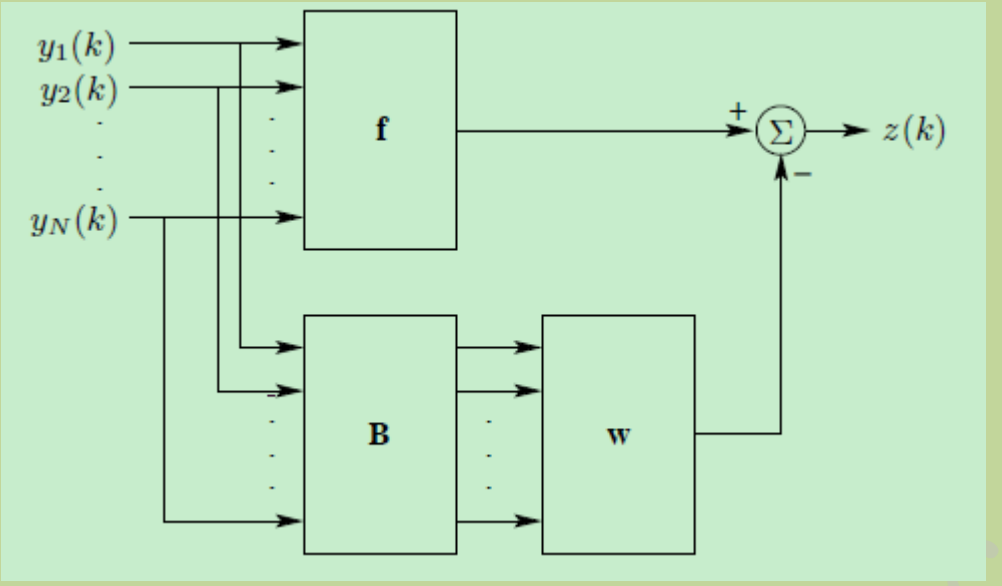
上式是 $\boldsymbol{G}_{:1}^T\boldsymbol{f} = \boldsymbol{u}$ 的最小范数解。B 是阻塞矩阵，W 是权向量。广义旁瓣是下式无限制的最优解：

$\min_{w}(\boldsymbol{f} - \boldsymbol{B}\boldsymbol{w})^T\boldsymbol{R}_{yy}(\boldsymbol{f} - \boldsymbol{B}\boldsymbol{w})$ （7.25）

其解是：

$w_{GSC} = \left[\boldsymbol{B}^T\boldsymbol{R}_{yy}\boldsymbol{B}\right]^{-1}\boldsymbol{B}^T\boldsymbol{R}_{yy}\boldsymbol{f}$ （7.26）

其中 f 代表 fixed Beamformer，B 代表 Blocking matrix，W 是自适应权向量。



广义旁瓣相消

$$h_{LCMV} = \boldsymbol{R}_{yy}^{-1}\boldsymbol{G}_{:1}\left[\boldsymbol{G}_{:1}^{T}\boldsymbol{R}_{yy}^{-1}\boldsymbol{G}_{:1}\right]^{-1}\boldsymbol{u} = \left\{\boldsymbol{I}_{2Lh\times 2Lh} - \boldsymbol{B}\left[\boldsymbol{B}^{T}\boldsymbol{R}_{yy}\boldsymbol{B}\right]^{-1}\boldsymbol{B}^{T}\boldsymbol{R}_{yy}\right\}\boldsymbol{f}$$
$$= h_{GSC}$$

since every 4 - 5dB improvement of the Signal to Noise Ratio (SNR) may raise the speech intelligibility by 50% [1]

3.3 基于阵列定位和跟踪技术

麦克风阵列在空域实现了噪声抑制，此外，还能定位声源。即两个问题，一个是 DOA（direction of arrival），另一个是定位声源空间坐标定位。

DOA 算法的主要思想是找到各个麦克风收集到的信号的相位信息关系。主要有三类 DOA 算法： 第一类是高分辨谱估计技术，这类技术在计算时需要用到相关矩阵，计算量较大，第二类是最大输出功率可控波速形成技术，利用加权函数求和得到，如 MVB（maximum variance beamformer），DSB（Delay and Sum Beamformer），多声源定位问题可以使用 MUSIC 和 ESPRIT 算法。第三类是基于时间差的定位技术，这种技术利用声源到各个麦克风阵列的时间差，然后求出角度，其计算量相对较小，对噪声处理也比较好，但只使用与单个声源的定位，对多个声源的定位效果比较差。

3.3.1 互相关方法

考虑只有两个麦克风的场景，互相关函数如下：

$$
\begin{aligned}
R_{y_1y_2}(p) &= E\left[y_1(k)y_2(k+p)\right] \\
&= E\left[\left(\alpha_1 s(k-t)+v_1(k)\right)\left(\alpha_2 s(k-t-\tau+p)+v_2(k+p)\right)\right] \\
&= \alpha_1\alpha_2 R_{ss}(p-\tau)+\alpha_1 R_{sv_2}(p+t)+\alpha_2 R_{sv_1}(p-t-\tau)+R_{v_1v_2}(p)
\end{aligned}
\qquad 3.3.1.1
$$

如果噪声之间不相关，并且和输入信号也不相关，则在 $p=\tau$ 时，信号互相关函数达到极大值 $\hat{\tau}=\arg\max_p R_{y_1y_2}(p), p\in\left[-\tau_{\max} \quad \tau_{\max}\right]$ 3.3.1.2

这种方法估计 DOA 简单，但是也有显而易见的缺点，在有反射，混响的模型中，这一方法并不是很适用。在使用 TDOA 算法时，通常要注意空域混叠。

➢ 互相关计算过程

$$
r_{xy}(m)=\sum_{n=-\infty}^{\infty}x(n)y(n+m) \qquad 3.3.1.2
$$

其计算过程是保持 $x(n)$ 不动，$y(n)$ 左移 m（m 可正可负）个抽样点后，两个序列相乘的结果。

```
x1 =

     1     2     3     4     5

>> x2=[1 0 2 3 4]

x2 =

     1     0     2     3     4

>> xcorr(x1,x2)

ans =

    4.0000   11.0000   20.0000   29.0000   39.0000   25.0000   13.0000    4.0000    5.0000
```

图 3.3.1.1 MATLAB 计算互相关结果

当 m=-5 时：

$$
r_{xy}(4)=\sum_{n=-\infty}^{\infty}x(n)y(n+4)=\cdots+x(-1)y(3)+x(0)y(4)+\ldots=x(0)y(4)=4
$$

其它值依次类推可以计算出来。

对于截取长度为 N 的两路麦克风信号，m 可取值从 $-(N-1)$ 到 $(N-1)$，对于 N 点信号，共可以计算出 2N-1 点互相关，m 值越大，对于 N 点有限长信号，其有意义计算的长度越短，实际中通常选择 $m\ll N$。其计算复杂度是 $O(N^2)$，当数据量较大时，计算较为耗时，可以采用 FFT 和 IFFT 来计算互相关函数以减少运算量，其计算复杂度是 $\frac{N}{2}\log_2 N$。

设 $y(n)$ 是 $x(n)$ 与 $h(n)$ 的互相关函数，即

$$y(m)=\frac{1}{N}\sum_{n=0}^{N-1}x(n)h(n+m) \qquad 3.3.1.3$$

则，

$$Y\left(e^{j\omega}\right)=X^{*}\left(e^{j\omega}\right)H\left(e^{j\omega}\right) \qquad 3.3.1.4$$

互相关可以表示成下式：

$$r_{xy}(m)=\sum_{n=-\infty}^{\infty}x(n)y(n+m)=x(-m)*y(m) \qquad 3.3.1.5$$

相关函数在频域也不是完全乘积，是一个信号的共轭乘以令一个信号的原始频谱。

3.3.2 广义互相关（GCC Generalized Cross-Correlation）

GCC 算法用于远场语音 DOA 估计时，其表达式和互相关法类似：

$$\hat{\tau}=\arg\max_{p}R_{y_1y_2}(p),p\in\left[-\tau_{\max}\quad\tau_{\max}\right] \qquad 3.3.2.1$$

$$R_{y_1y_2}(p)=F^{-1}\left[\Psi_{y_1y_2}(f)\right]=\int_{-\infty}^{\infty}\Psi_{y_1y_2}(f)e^{j2\pi fp}df=\int_{-\infty}^{\infty}\vartheta(f)\phi_{y_1y_2}(f)e^{j2\pi fp}df \qquad 3.3.2.2$$

$F^{-1}[\bullet]$ 是离散傅里叶反变换（IDTFT）。

$$\phi_{y_1y_2}(f)=E\left[Y_1(f)Y_2^{*}(f)\right] \qquad 3.3.2.3$$

是互相关谱

$$Y_n(f)=\sum_{k}y_n(k)e^{-j2\pi fk},n=1,2, \qquad 3.3.2.4$$

$\vartheta(f)$ 是频域权向量。选择不同的权向量可以产生不同的 GCC 方法。

3.3.2.1 经典互相关

如果 $\vartheta(f)=1$，则 GCC 退化为类似 3.3.2.1 节的互相关情况，区别是互相关函数用了离散傅里叶变换（DFT）和离散傅里叶反变换（IDFT）。对于远场模型：

$$Y_n(f)=\alpha_nS(f)e^{-j2\pi f\left[t-\tau(n)\right]}+V_n(f),n=1,2 \qquad 3.3.2.5$$

则可得:

$$\psi_{y_1y_2}(f)=\alpha_1\alpha_2e^{-j2\pi f\tau}E\left[\left|S(f)\right|^2\right] \qquad 3.3.2.6$$

$\psi_{y_1y_2}(f)$ 取决于声源的这一性质可以用来估计 TDOA。

3.3.2.2 平滑相关变换

为了克服语音信号在估计 TDOA 时的波动性，一个有效的方法是在计算交叉谱之前先白化麦克风阵列的输出，这等价于下式：

$$\vartheta(f)=\frac{1}{\sqrt{E\left[\left|Y_1(f)\right|^2\right]E\left[\left|Y_2(f)\right|^2\right]}} \tag{3.3.2.7}$$

由此可得：

$$\begin{aligned}\psi_{y_1y_2}(f)&=\frac{\alpha_1\alpha_2e^{-j2\pi f\tau}E\left[\left|S(f)\right|^2\right]}{\sqrt{E\left[\left|Y_1(f)\right|^2\right]E\left[\left|Y_2(f)\right|^2\right]}}\\&=\frac{\alpha_1\alpha_2e^{-j2\pi f\tau}E\left[\left|S(f)\right|^2\right]}{\sqrt{\alpha_1^2E\left[\left|S(f)\right|^2\right]+\sigma_{v_1}^2(f)}\cdot\sqrt{\alpha_2^2E\left[\left|S(f)\right|^2\right]+\sigma_{v_2}^2(f)}}\\&=\frac{e^{-j2\pi f\tau}}{\sqrt{1+\frac{1}{SNR_1(f)}}\cdot\sqrt{1+\frac{1}{SNR_2(f)}}}\end{aligned} \tag{3.3.2.8}$$

这就意味着 TDOA 估计的性能和 SNR 相关，当性噪比足够大时：

$$\psi_{x_1x_2}(f)=e^{-j2\pi f\tau}$$

3.3.2.3 相位变换 PATH

TDOA 依赖于信号的相位信息而非信号的幅度，所以可以简单的丢掉幅度信息而保留相位信息。

$$\vartheta(f)=\frac{1}{\left|\phi_{y_1y_2}(f)\right|} \tag{3.3.2.9}$$

此时：

$$\psi_{y_1y_2}(f)=e^{-j2\pi f\tau} \tag{3.3.2.10}$$

这时可得广义旁瓣互相关函数为：

$$R_{y_1y_2}(p)=\int_{-\infty}^{\infty}e^{j2\pi f(p-\tau)}df=\begin{cases}\infty, p=\tau\\0, otherwise\end{cases}$$

3.3.2.4 多通道互相关系数法

使用多通道自相关系数法，将新的信号向量表示成如下形式：

$$y_a(k,p)=\left[y_1(k)\quad y_2\left(k+\tau_2(p)\right)\quad\cdots\quad y_N\left(k+\tau_N(p)\right)\right]^T \tag{3.3.2.11}$$

由此可得空域互相关矩阵：

$$R_a(p)=E\left[y_a(k,p)y_a^T(k,p)\right]=\begin{bmatrix}\sigma_{y_1}^2 & r_{y_1y_2}(p) & \cdots & r_{y_1y_N}(p)\\ r_{y_1y_2}(p) & \sigma_{y2}^2 & \cdots & r_{y_1y_2}(p)\\ \vdots & \vdots & \ddots & \vdots\\ r_{y_1y_2}(p) & r_{y_1y_2}(p) & \cdots & \sigma_{y_N}^2\end{bmatrix} \qquad 3.3.2.12$$

空域自相关矩阵可以

3.3.3 基于特征向量的方法

此方法最先用于雷达的 DOA 估计，该方法已经用于宽带麦克风阵列信号处理，考虑 N 元含高斯白噪声的麦克风阵列。

3.3.3.1 窄带 MUSIC（Multiple Signal Classification）算法

MUSIC 最早用在雷达上的超分辨技术上，且常用在多声源定位中。MUSIC 算法是正交法，其基本思想是信号子空间与和噪声的子空间域进行正交，这样就能构造一个空间函数，再利用计算机技术搜素信号的谱峰值，最后检查语音信号的 DOA。对于第 n 个麦克的输出，其频域表示如下：

$$Y_n(f)=X_n(f)+V_n(f)=S(f)e^{-j2\pi[t+\tau_n]f}+V_n(f) \qquad 3.3.3.1$$

可以定义如下频域向量：

$$\vec{y}=\left[Y_1(f)\quad Y_2(f)\quad\cdots\quad Y_N(f)\right]^T \qquad 3.3.3.2$$

这样可以用向量表示接收到的信号 $\vec{y}$ ：

$$\vec{y}=\vec{x}+\vec{v}=\varsigma(\tau)S(f)e^{-j2\pi tf}+\vec{v} \qquad 3.3.3.3$$

此处可得：

$$\varsigma(\tau)=\left[e^{-j2\pi\tau_1 f}\quad e^{-j2\pi\tau_2 f}\quad\cdots\quad e^{-j2\pi\tau_N f}\right]^T \qquad 3.3.3.4$$

则输出的协方差可以写为如下形式：

$$\boldsymbol{R}_Y=E\left(\vec{y}\vec{y}^H\right)=\boldsymbol{R}_X+\sigma_V^2\boldsymbol{I} \qquad 3.3.3.5$$

其中

$R_X=\sigma_s^2\varsigma(\tau)\varsigma^H(\tau)$ ， $\sigma_S^2=E\left[\left|S(f)\right|^2\right]$ ， $\sigma_V^2=E\left[\left|V_1(f)\right|^2\right]=\cdots=E\left[\left|V_N(f)\right|^2\right]$ 使用特征向量分解矩阵：

$$\boldsymbol{R}_Y=B\Lambda B^H \qquad 3.3.3.6$$

这里：

$$\Lambda = diag\left[\lambda_{Y,1} \quad \lambda_{Y,2} \quad \cdots \quad \lambda_{Y,N}\right] = diag\left[\lambda_{X,1} \quad \sigma_V^2 \quad \cdots \quad \sigma_V^2\right] \qquad 3.3.3.7$$

是矩阵 R_Y 的特征值矩阵，$\boldsymbol{B} = \left[\boldsymbol{b}_1 \quad \boldsymbol{b}_2 \quad \cdots \quad \boldsymbol{b}_N\right]$，$b_n$ 是特征值 $\lambda_{Y,n}$ 对应的特征向量，$\lambda_{Y,1}$ 是 R_X 的唯一半正定特征值。即有：

$$\boldsymbol{R}_Y \boldsymbol{b}_n = \lambda_{Y,n} \boldsymbol{b}_n = \sigma_V^2 \boldsymbol{b}_n \qquad 3.3.3.8$$

同时有：

$$\boldsymbol{R}_Y \boldsymbol{b}_n = \left[\sigma_s^2 \varsigma(\tau) \varsigma(\tau)^H \quad \sigma_V^2 \boldsymbol{I}\right] \boldsymbol{b}_n \qquad 3.3.39$$

由 3.3.38 和 3.3.3.9 可得：

$$\sigma_s^2 \boldsymbol{\varsigma}(\tau) \boldsymbol{\varsigma}(\tau)^H \boldsymbol{b}_n = \boldsymbol{0} \qquad 3.3.3.10$$

这等价于：

$$\boldsymbol{\varsigma}(\tau)^H \boldsymbol{b}_n = \boldsymbol{0} \qquad 3.3.3.11$$

这意味着 R_Y 的 N-1 个特征向量和实际的 TDOA 向量正交。假设代价函数如下：

$$J_{MUSIC}(p) = \frac{1}{\sum_{n=2}^{N} \left| b_n^H \varsigma(p) \right|^2} \qquad 3.3.3.12$$

则有 TDOA 的估计值如下：

$$\hat{\tau} = \arg\max_p J_{MUSIC}(p) \qquad 3.3.3.13$$

3.3.3.2 宽带 MUSIC 算法

窄带 MUSIC 算法并在语音中效果并不好，这是因为语音是非平稳信号，语音信号的中的频谱成分随着语音变换而变化，一个抵消非平稳的方法是将式 3.3.3.12 的代价函数在整个频谱上去求代价函数。

$$y_{1:N}(k,p) = \left[y_1\left[k+\tau_1(p)\right] \quad y_2\left[k+\tau_2(p)\right] \quad \cdots \quad y_N\left[k+\tau_N(p)\right]\right] \qquad 3.3.3.14$$

空域相关矩阵如下：

$$\boldsymbol{R}_a(p) = E\left[\boldsymbol{y}_{1:N}(k,p)\boldsymbol{y}_{1:N}^T(k,p)\right] = \boldsymbol{R}_s(p) + \sigma_v^2 \boldsymbol{I} \qquad 3.3.3.15$$

原始信号的协方差如下：

$$R_s(p) = \begin{bmatrix} \sigma_s^2 & r_{ss,12}(p,\tau) & \cdots & r_{ss,1N}(p,\tau) \\ r_{ss,21}(p,\tau) & \sigma_s^2 & \cdots & r_{ss,2N}(p,\tau) \\ \vdots & \vdots & \ddots & \vdots \\ r_{ss,N2}(p,\tau) & r_{ss,N2}(p,\tau) & \cdots & \sigma_s^2 \end{bmatrix} \qquad 3.3.3.16$$

$$r_{ss,ij}(p,\tau)=E\left\{s\left[k-t-\tau_i+p_i\right]s\left[k-t-\tau_j+p_j\right]\right\} \quad 3.3.3.17$$

如果$\tau=p$ 则有：

$$R_s(p)=\sigma_s^2\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1\end{bmatrix} \quad 3.3.3.18$$

该矩阵的秩是 1，如果$\tau\neq p$，则矩阵的秩取决于输入信号的自身的特性。如果输入信号是白化过程的信号，则输入信号的自相关矩阵$R_s(p)=diag\left[\sigma_s^2 \quad \sigma_s^2 \quad \cdots \quad \sigma_s^2\right]$。在这一特殊情况下，$\boldsymbol{R}_s(p)$ 是满秩的。通常来说，当$\tau\neq p$，$\boldsymbol{R}_s(p)$是半正定的。对$\boldsymbol{R}(p)$和$\boldsymbol{R}_s(p)$进行特征值分解，假设$\lambda_{s,1}(p)\geq\lambda_{s,2}(p)\geq\cdots\geq\lambda_{s,N}(p)$ 是$\boldsymbol{R}_s(p)$的 N 个特征值。则$\boldsymbol{R}(p)$的 N 个特征向量可表示为如下：

$$\lambda_{y,n}(p)=\lambda_{s,n}(p)+\sigma_v^2 \quad 3.3.3.19$$

设$\boldsymbol{b}_1(p),\boldsymbol{b}_2(p),\ldots,\boldsymbol{b}_N(p)$ 是$\boldsymbol{R}(p)$特征向量，$\boldsymbol{R}(p)$是对称的拓普利兹矩阵，其所有特征向量都是实数组成的，则有：

$$\boldsymbol{R}(p)\boldsymbol{B}(p)=\boldsymbol{B}(p)\Lambda(p) \quad 3.3.3.20$$

此处：

$\boldsymbol{B}(p)=\left[b_1(p) \quad b_2(p) \quad \ldots \quad b_N(p)\right]$

$\Lambda(p)=diag\left[\lambda_{y,1}(p) \quad \lambda_{y,2}(p) \quad \cdots \quad \lambda_{y,N}(p)\right]$

此处$p=\tau$ ，由于$\boldsymbol{R}_s(p)$的秩等于 1，所以对于$n\geq 2$ 情况，有下式成立：

$$\boldsymbol{R}(\tau)\boldsymbol{b}_n(\tau)=\left[\boldsymbol{R}_s(\tau)+\sigma_v^2\boldsymbol{I}\right]b_n(\tau)=\sigma_v^2 b_n(\tau) \quad 3.3.3.21$$

这意味着：

$$\boldsymbol{b}_n^T(p)\boldsymbol{R}(p)\boldsymbol{b}_n(p)=\begin{cases}\sigma_v^2, p=\tau\\ \lambda_{y,n(p)}\geq\sigma_v^2, p\neq\tau\end{cases} \quad 3.3.3.22$$

所以可以使用如下的代价函数：

$$J_{BMUSIC}(p)=\frac{1}{\sum_{n=2}^{N}\boldsymbol{b}_n^T(p)\boldsymbol{R}(p)\boldsymbol{b}_n(p)} \quad 3.3.3.23$$

该代价函数的峰值对于 TDOA 值τ 。

3.3.4 最小熵法

基于高阶统计的 TDOA 估计法，熵是一个对随机或者不确定变量的一个统计度量，由香农在信息论中最先提出来，对于一个随机变量 y，其概率密度函数（PDF，probability density function）是 $p(y)$ ，其熵定义为：

$$H(y)=-\int p(y)\ln p(y)dy=-E\left[\ln p(y)\right] \tag{3.3.4.1}$$

对于语音接收到的信号 $y(k,p)$ 而言，其熵如下：

$$H\left[y(k,p)\right]=-\int py(k,p)\ln py(k,p)dy(k,p) \tag{3.3.4.2}$$

其最小值（熵），对应的解 p 就是 TDOA：

$$\hat{\tau}^{ME}=\arg\min_{p} H\left[y(k,p)\right] \tag{3.3.4.3}$$

ME 是 minimum enropy 的缩写。

3.3.4.1 高斯信号源

如果信源是高斯类型的且麦克风输出没有噪声。假设麦克信号是零均值联合随机高斯信号，则它们的联合概率密度函数是：

$$p\left[\boldsymbol{y}(k,p)\right]=\frac{e^{-\eta(k,p)}/2}{\sqrt{(2\pi)^N\det\left[\boldsymbol{R}(p)\right]}} \tag{3.3.4.4}$$

此处：

$$\eta(k,p)=\boldsymbol{y}^T(k,p)\boldsymbol{R}^{-1}(p)\boldsymbol{y}(k,p) \tag{3.3.4.5}$$

将式 3.3.4.5 代入 3.3.4.2 得：

$$H\left[y(k,p)\right]=\frac{1}{2}\ln\left\{(2\pi e)^N\det\left[\boldsymbol{R}(p)\right]\right\} \tag{3.3.4.6}$$

则 3.3.4.3 将变为：

$$\hat{\tau}^{ME}=\arg\min_{p}\det\left[\boldsymbol{R}(p)\right] \tag{3.3.4.7}$$

3.3.4.2 语音信号源

虽然语音信号是复杂的随机信号，并没有明确的数学表达式表示其熵，但是语音信号可以用拉普拉斯模型来近似描述。单变量拉普拉斯零均值，方差为 σ_y^2 的分布如下：

$$p(y)=\frac{\sqrt{2}}{2\sigma_y}e^{-\sqrt{2}|y|/\sigma_y} \tag{3.3.4.8}$$

其熵可表示如下：

$$H[y]=1+\ln\left(\sqrt{2}\sigma_y\right) \qquad 3.3.4.9$$

对于零均值多变量协方差为 $\boldsymbol{R}(p)$ 的拉普拉斯累计分布如下：

$$p[\boldsymbol{y}(k,p)]=\frac{2\left[e^{-\eta(k,p)}/2\right]^{Q/2}K_Q\left[\sqrt{2\eta(k,p)}\right]}{\sqrt{(2\pi)^N\det[\boldsymbol{R}(p)]}} \qquad 3.3.4.10$$

这里：

$Q=(2-N)/2$ ， $K_Q(\cdot)$ 是第三类修正贝塞尔函数：

$$K_Q(a)=\frac{1}{2}\left(\frac{a}{2}\right)^Q\int_0^\infty z^{-Q-1}e^{-z-\frac{a^2}{4z}}dz, a>0 \qquad 3.3.4.11$$

其联合熵函数如下：

$$H[\boldsymbol{y}(k,p)]=\frac{1}{2}\ln\left\{\frac{(2\pi)^N\det[\boldsymbol{R}(p)]}{4}\right\}-\frac{Q}{2}E\left\{\ln\left[\frac{\eta(k,p)}{2}\right]\right\}-E\left\{\ln K_Q\left[\sqrt{2\eta(k,p)}\right]\right\}$$

3.3.5 自适应特征向量分解法

该方法对于混响模型下，具有较好的健壮性，在加性噪声情况下：

$$y_1(k)*g_2=x_1(k)*g_1*g_2=x_2(k)*g_1=y_2(k)*g_1 \qquad 3.3.4.12$$

则 k 时刻可得下式：

$$\boldsymbol{y}^T(k)\boldsymbol{w}=\boldsymbol{y}_1^T(k)\boldsymbol{g}_2-\boldsymbol{y}_2^T(k)\boldsymbol{g}_1=0 \qquad 3.3.4.13$$

将上式左乘以 $\boldsymbol{y}(k)$ 可得：

$$\boldsymbol{R}_{yy}\boldsymbol{W}=\boldsymbol{0}_{2L\times1} \qquad 3.3.4.14$$

$\boldsymbol{W}$ 是 $\boldsymbol{R}_{yy}$ 对应的特征值 0 的特征向量，实际上噪声总是存在的， $\boldsymbol{R}_{yy}$ 实际上是正定矩阵，所以， $\boldsymbol{W}$ 是相对 $\boldsymbol{R}_{yy}$ 于最小的特征值的特征向量。即有：

$$\widehat{\boldsymbol{W}}=\boldsymbol{arg}\min_{w}W^TR_{yy}W,\|w\|=1 \qquad 3.3.4.15$$

式 3.3.4.15 可以使用约束性 LMS 算法来求解：

初始化

$$\hat{\boldsymbol{g}}_n(0)=\left[\frac{\sqrt{2}}{2}\quad 0\quad\cdots\quad 0\right],n=1,2$$

$$\hat{\boldsymbol{w}}(0)=\left[\hat{g}_2^T(0)-\hat{g}_1^T(0)\right]^T$$

按下式计算：

$$e(k)=\hat{\boldsymbol{w}}^{T}(k)\boldsymbol{y}(k)$$

$$\hat{w}(k+1)=\frac{\hat{w}(k)-\mu e(k)\boldsymbol{y}(k)}{\left\|\hat{w}(k)-\mu e(k)\boldsymbol{y}(k)\right\|}$$

在上述方法收敛后，

$$\hat{\tau}=\arg\max_{l}\left|\hat{g}_{1,l}\right|-\arg\max_{l}\left|\hat{g}_{2,l}\right|$$

3.3.6 自适应盲信号分离（BSS，Blind Source Separation）

独立分量分析法（ICA，Independent Component Analysis）是盲源分离采用最多的算法。

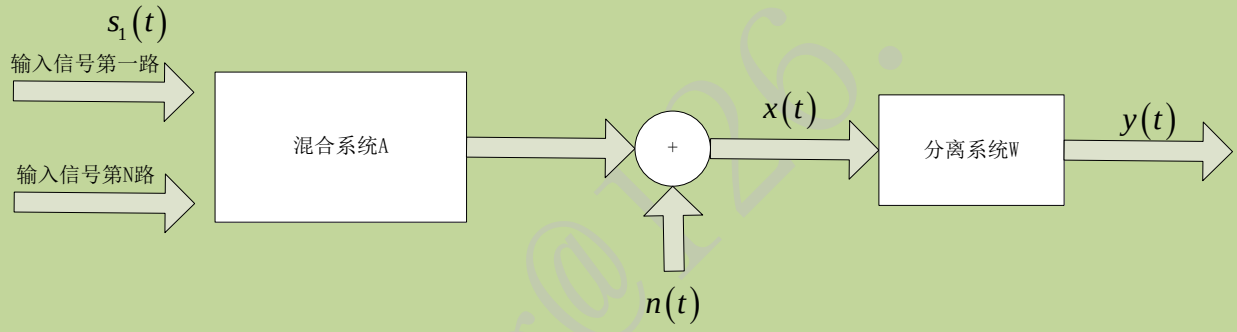


图 3.3.6.1 盲源分离原理

BSS 分离的数学模型如上图，

$\boldsymbol{s}(t)=\left[s_1(t)\ \cdots\ s_N(t)\right]$ 是输入 N 维未知信源，A 为未知混合系统，$\boldsymbol{x}(t)=\left[x_1(t)\ \cdots\ x_M(t)\right]$是观测到的信号向量，$\boldsymbol{n}(t)=\left[n_1(t)\ \cdots\ n_M(t)\right]^{T}$，盲源分离的目的就是在源信号 s 和混合系统 A 均未知前提下，调节分离系统 W，使得输出信号 y 是 s 的估计，即：

$$\boldsymbol{y}(t)=\boldsymbol{x}\bullet W=\hat{s} \qquad 3.3.6.1$$

FastICA 算法

1. 对观测的数据 X 去均值，使其均值为 0，其目的是简化 ICA 算法。
2. 对数据进行白化 $X\rightarrow Z$ ，去除观测信号之间的相关性，简化后续独立分量的提取过程。
3. 更新准则，采用随机梯度法，定义自然梯度：

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-\theta(k)\frac{\partial J(\boldsymbol{W}(k))}{\partial \boldsymbol{W}(k)}\boldsymbol{W}^{T}(k)\boldsymbol{W}(k) \qquad 3.3.6.2$$

根据信息最大或者互信息最大可以得出如下代价函数：

$$J(w,y)=-\log\left(\det(w)\right)-\sum_{i=1}^{n}\log\left(p_i(y_i)\right) \qquad 3.3.6.3$$

$p_i(y_i)$是y_i信号的概率密度函数，$\det(W)$ 是矩阵 W 的行列式的值。将代价函数代入 3.3.6.2 得：

$$\Delta W(k)=\theta(k)\left[I-f\left(y(k)y^T(k)\right)\right]W(k) \qquad 3.3.6.4$$

FastICA：

1. 初始化非零权向量 W
2. 向量叠加，$w_p^+=E\left\{xg\left(w_p^Tx\right)\right\}-E\left\{g'\left(w_p^Tx\right)\right\}w_p$ ，g 是非线性函数，g' 是其一阶时域导数。
3. 归一化成酉空间向量

$$w_p=\frac{w_p^+}{\left\|w_p^+\right\|}$$

4. 根据施密特正交，解其和之前向量的相关性。

$$w_p=w_p-\sum_{j=1}^{p-1}w_p^Tw_jw_j;w_p=\frac{w_p}{\left\|w_p\right\|} \qquad 3.3.6.5$$

5.如果 W 还是没收敛，则返回步骤一。

3.3.1 TDOA

一般实时系统使用 TDOA（time difference of arrival）技术，该技术主要是估计声源的入射角，该方法的基本步骤是：首先根据各麦克风之间的相对延迟，然后利用相对之间的相位延迟估计声源的位置。在 3.1.1 节中，在入射角是θ 的情况下，信号到达麦克风之间的延迟$\tau=\frac{d\sin\theta}{c}$，则将对$\theta$ 的估计转换为对τ的估计。由于噪声和混响的存在，精确估计延迟比较困难，常采用的时延估计法主要有广义互相关（GCC，Generalized Cross Correlation）和 LMS 自适应滤波器法。考虑两麦克风接收信号为：

$$y_1(n)=s_1(n)+n_1(n)$$

$y_2(n)=s_2(n+\tau)+n_2(n)$

其中 τ 是时延，$s_1(n)$ 和 $s_2(n+\tau)$ 是语音信号，$n_1(n)$ 和 $n_2(n)$ 是背景噪声，它们的互相关函数为：

$$R_{12}(p)=F^{-1}\left[\psi_{y_1y_2}(f)\right]=\int_{-\infty}^{+\infty}\psi_{y_1y_2}(f)e^{j2\pi fp}df=\int_{-\infty}^{+\infty}\phi_{y_1y_2}(f)e^{j2\pi fp}df \quad 3.3.1.1$$

其中 $\phi_{y_1y_2}(f)=E\left[Y_1(f)Y_2^*(f)\right]$，$Y_n(f)=\sum_k y_n(k)e^{-j2\pi fk}, n=1,2$，$\tau$ 为延迟，f 为频率。

该麦克风对的时延就是函数 $R_{12}(\tau)$ 的峰值对应的 τ 值。通常为了使时延 τ 估计的更准确，通常会采用加权的方式进行计算：

$$R_{12}(p)=\int_{-\infty}^{+\infty}W_{12}Y_1(f)Y_2^*(f)e^{j2\pi fp}df \quad 3.3.1.2$$

W_{12} 是广义互相关加权函数。

● 平滑相关变换

为了克服语音信号在估计 TDOA 时的波动影响，一个有效的方法是在计算麦克风之间的互相关前白化麦克风输出，即：

$$W_{12}=\frac{1}{\sqrt{E\left[|Y_1|^2\right]E\left[|Y_2|^2\right]}} \quad 3.3.1.3$$

则可得：

$$\begin{aligned}\psi_{y_1y_2}(f)&=\frac{\alpha_1\alpha_2e^{-j2\pi fp}E\left[|S(f)|^2\right]}{\sqrt{E\left[|Y_1(f)|^2E\left[|Y_2(f)|^2\right]\right]}}\\&=\frac{\alpha_1\alpha_2e^{-j2\pi fp}E\left[|S(f)|^2\right]}{\sqrt{\alpha_1^2E\left[|s(f)|^2\right]+\sigma_{v_1}^2(f)}\cdot\sqrt{\alpha_2^2E\left[|s(f)|^2\right]+\sigma_{v_2}^2(f)}}\\&=\frac{e^{-j2\pi fp}}{\sqrt{1+\frac{1}{SNR_1(f)}}\cdot\sqrt{1+\frac{1}{SNR_2(f)}}}\end{aligned} \quad 3.1.3.4$$

可得到信噪比如下：

$$SNR_n(f)=\frac{\alpha_n^2E\left[|S(f)|^2\right]}{E\left[|V_n(f)|^2\right]}, n=1,2 \quad 3.1.3.5$$

● 相位变化法

由式 3.3.1.2 可知，TDOA 实际上是相位的估计，所以可以将 W_{12} 设置为如下形式：

$$W_{12}=\frac{1}{\left|\phi_{y_1y_2}(f)\right|} \tag{3.1.3.6}$$

则有：

$$\psi_{y_1y_2}(f)=e^{-j2\pi f\tau} \tag{3.1.3.7}$$

这样可以求得：

$$R_{12}(p)=\int_{-\infty}^{+\infty}e^{j2\pi f(p-\tau)}df=\begin{cases}\infty, p=\tau\\ 0, otherwise\end{cases} \tag{3.1.3.8}$$

3.3.3 空域线性预测法

其基本思想是使用冗余的麦克风来增加 TDOA 估计的准确度。假设有三个麦克风，它们之间的延迟分别是τ_{12} τ_{13} 和τ_{23}，它们之间的关系是$\tau_{13}=\tau_{12}+\tau_{23}$ ，考虑 N 元线阵远场语音模型：

$$y_n(k)=\alpha_n s(k-t-\tau_{n1})+v_n(k),1\le n\le N \tag{3.1.3.10}$$

忽略噪声的影响，则有下式成立：

$\alpha_n s(k-t)=y_n(k+\tau_{n1}),1\le n\le N$，如果取$y_1(k)$ 和$y_n(k+\tau_{n1})$ 对齐，则可得前向预测误差：

$$e_1(k,p)=y_1(k)-\boldsymbol{y}_{a,2:N}^T(k,p)\boldsymbol{\alpha}_{2:N}(p) \tag{3.1.3.11}$$

其中p 代表了时延。

$$\boldsymbol{y}_{a,2:N}(k,p)=\left[y_2(k+p_{21})\quad\cdots\quad y_N(k+p_{N1})\right]^T \tag{3.1.3.12}$$

根据最小均方误差准则，可以求得延迟 p。

$$J_1(p)=E\left[e_1^2(k,p)\right] \tag{3.1.3.13}$$

$$R_{2:N}(p)\alpha_{2:N(p)}=r_{2:N(p)} \tag{3.1.3.14}$$

此处：

$$R_{2:N}(p)=E\left[\boldsymbol{y}_{2:N}(k,p)\boldsymbol{y}_{2:N}^T(k,p)\right]=\begin{bmatrix}\sigma_{y_2}^2 & r_{y_2y_3}(p) & \cdots & r_{y_2y_N}(p)\\ r_{y_3y_2}(p) & \sigma_{y_3}^2 & \cdots & r_{y_3y_N}(p)\\ \vdots & \vdots & \vdots & \vdots\\ r_{y_Ny_2}(p) & r_{y_Ny_3}(p) & \cdots & \sigma_{y_N}^2\end{bmatrix} \tag{3.1.315}$$

是空域自相关矩阵。

$$\sigma_{y_n}^2=E\left[y_n^2(k)\right],1\le n\le N \tag{3.1.3.16}$$

$$r_{y_i y_j}(p)=E\left\{y_i\left[k+\tau_i(p)\,y_j\left[k+\tau_j(p)\right]\right]\right\},i,j=1,2\cdots N \qquad 3.1.3.17$$

并且：

$$\boldsymbol{r}_{2:N}(p)=E\left\{r_{y_1y_2}(p)\quad r_{y_1y_3}(p)\quad\cdots\quad r_{y_1y_N}(p)\right\},i,j=1,2\cdots N \qquad 3.1.3.18$$

将上式代入 3.1.3.14 得：

$$e_{1,\min}(k,p)=y_1(k)-\boldsymbol{r}_{2:N}^T(p)\boldsymbol{R}_{2:N}^T(p)r_{2:N}(p)$$

可以得：

$$J_{1,\min}(p)=E\left\{e_{1,\min}^2(k,p)\right\}=\sigma_{y_1}^2-\boldsymbol{r}_{2:N}^T(p)\boldsymbol{R}_{2:N}^{-1}(p)\boldsymbol{r}_{2:N}(p)$$

然后可得：

$\hat{\tau}=\arg\min_p J_{1,\min(p)}$ ，如果只有两个麦克风，则退化为互相关算法，当麦克风数量增加时，该算法将利用冗余信息提高 TDOA 的估计值。

当信噪比达到 10dB 时，两个麦克风可以达到准确的 TDOA 估计，当信噪比到-5dB 时，基于两个麦克风的 TDOA 的估计将不准确。但是当使用 4 或者更多的麦克风时，系统的结果将是正确的。

3.3.

3.3.2 SRP-PHAT

SRP-PHAT（steering response power-phase transform）可控功率响应和相位变换技术，基于粒子滤波需要的运算量太大，使用 SRC（stochastic region contraction）随机区域收缩方法简化计算量。SRP-PHAT 方法的波束输出功率（帕斯瓦尔定律）为：

$$P(\vec{q})=\sum_{m=1}^{M}\sum_{l=1}^{M}\int_{-\infty}^{+\infty}\psi_{ml}(k)X_m(k)X_l^*(k)e^{j2\pi k(\tau_1-\tau_m)}dk$$

其中 τ_1 和 τ_m 为阵列的可控时延，$\vec{q}$ 为声源的空间位置矢量，$X_m(k)$ 为第 m 个麦克风信号 $x_m(t)$ 的加窗傅里叶变换。上式即第 m 和第 l 个麦克风的交叉功率谱，在多信道下 PHAT 加权系数为：

$$\psi_{ml}(k)=\frac{1}{\left|\left[X_m(k)X_1^*(k)\right]\right|}$$

设 $R_{ml}(\tau)$ 为第 m 和第 l 个麦克风接收到信号的 PHAT 加权广义互相关函数，则 SRP-PHAT 算法的时域表示为：

$$P(\vec{q})=\sum_{m=1}^{M}\sum_{l=1}^{M}\int_{-\infty}^{+\infty}R_{ml}(\tau_1(\vec{q})-\tau_m(\vec{q}))$$

该定位方法就是在所有可能覆盖声源的位置中，$P(\vec{q})$ 最大值所对应的 $\vec{q}$ 值。也就是

$$\hat{\vec{q}} = \arg\max_{\vec{q}} P(\vec{q})$$

随机区域收敛技术 SRC（stochastic region contraction）

其基本思想是：给定一个起始的长方体作为搜索区域，并且假定全局最优解（说话人的位置）就在这个长方体里，在求解最优解的过程中通过迭代的方式缩小长方体的体积直到体积小到需要的范围。

假设 V_{peak} 是要缩小的范围，V_{room} 是起始考察的范围。则为了使其 miss 概率（左栏）小于该栏时计算的次数列在了对应的表中。人脸的面积约 330~400 平方厘米，实际按 800 平方厘米作为搜索区域，则 8000 平方分米（8 平方米，3*3 米）。

$\frac{v_{peak}}{v_{room}} \mid P(_{miss} v_{peak})$	0.1	0.01	0.001	0.0001
1%	44	459	4063	46050
0.1%	66	688	6905	69,075
0.001%	88	917	9206	92099

表：计算 $\hat{P}(\vec{q})$ 的次数

定义 J_i 是第 i 次迭代的点数；N_i 是定义新空间的点数；V_{i+1} 有一个长方体边界向量 $\vec{B}_{i+1} \equiv \left[x_{\max}(i+1) x_{\min}(i+1) y_{\max}(i+1) y_{\min}(i+1) z_{\max}(i+1) z_{\min}(i+1)\right]$，$I$ 迭代的次数，FE_i 是第 i 次迭代的所有 fe（functional evaluation，任一点的 $\hat{P}(\vec{q})$）的估计次数之和，Φ 是 fe 的最大个数，则 SRC 算法如下：

1. $i = 0$

2.初始化 J_0、N_0 以及 $V_0 = V_{room}$

3．对 J_i 点计算 $\hat{P}(\vec{q})$

4.排序，找出最优的 $N_i \le J_i$ 点

5 将原始区域缩小为 V_{i+1}，$\vec{B}_{i+1}$ 包括这些 N_i 个点。

6 IF：$V_{i+1} < V_u$，或者 $FE_i > \Phi$ 并且 $V_{i+1} < T_1 V_u$，T_1 是一个大约等于 10 的参数，V_u 是目标区域大小；这时求 $\hat{x}_s^n(i^*)$，$I = i$，Return x

7 ELSE IF：$FE_i > \Phi$，stop，return NULL

8 ELSE：在 N_i 点中选取大于等于均值 μ_i（N_i 的均值）的子集 G_i

9 在 V_{i+1} 中评估 J_{i+1}

10 将 N_{i+1} 点作为 G_i 和刚刚计算的 J_{i+1}（源于 N_{i+1}-G_i 空间）的集合，

11 $i=i+1$ ，跳至第 5 步。

计算开销：

对于 M 个麦克风， $\hat{P}(\bar{q})$ 的计算次数 $Q=M(M-1)/2$ 相位变换，对于 DFT 的长度是 L

则有：DFT： $M*\frac{5}{2}L\log_2^L$ ，频谱计算：约 10QL， IDFT：

语音阵列处理的其它问题

说话人数量估计

背景噪声下信号分离和盲信号分离

生理语音学

下篇-语音识别

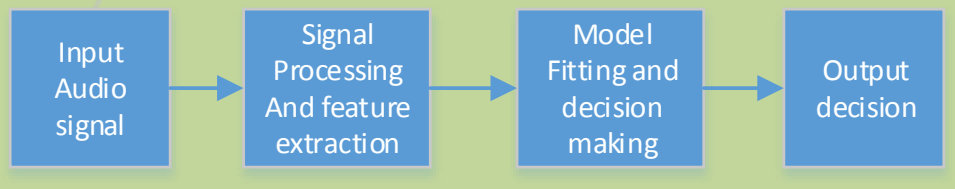
语音信号预加重

语音信号随着频谱增加幅度值越小，对于频率相差两边的语音信号，其功率谱差 6dB；需要对高频信号进行预加重，一般是将语音信号通过一个高通滤波器1-aZ^{-1},即预加重滤波器，

其目的是滤除 50Hz，60Hz 的工频干扰。

VAD 算法

当前 VAD 采用的算法依赖于以下几种特征：基于能量变换，周期性，频谱差异。其可以用在语音增强，语音编码和语音识别。VAD 算法模型有基于神经元模型和基于统计模型。比较流行的统计模型是高斯分布模型和拉普拉斯分布模型。



VAD 处理框图

第五章 ASR（automatic speech recognition）

语音识别问题就是模式分类问题。将音素以及音素组用离散的类来模拟。语音识别的目标是预测正确的类序列。如果 z 表示从声波提取的特征向量序列，则语音识别系统根据最优分类方程来工作：

$$\hat{\omega} = \arg\max_{\omega \in W} P(\omega \mid z) \tag{5.1}$$

$\hat{\omega}$ 是识别系统判断的说话的词序列，W 是所有可能的说话词的序列集合。实际上用贝叶斯准则来计算该值。

$$\hat{\omega} = \arg\max_{\omega \in W} \frac{P(Z \mid \omega)P(\omega)}{P(Z)} \tag{5.2}$$

$P(Z \mid \omega)$ 是声学似然（声学打分），代表了在词 w 被说了的情况下，语音序列 Z 出现的概率。$P(\omega)$ 是语言打分，是语音序列出现的先验概率，其计算依赖于语言模型。在忽略语音序列出现概率的情况下，式 5.2 可以简化为：

$$\hat{\omega} = \arg\max_{\omega \in W} P(Z \mid \omega)P(\omega) \tag{5.3}$$

这样语音识别可以分为两个主要步骤，特征提取和解码。

ASR 主要包括四个部分：信号处理和特征提取，声学模型（AM acoustic model），语言模型（LM language model），假设搜索（hypothesis search）。

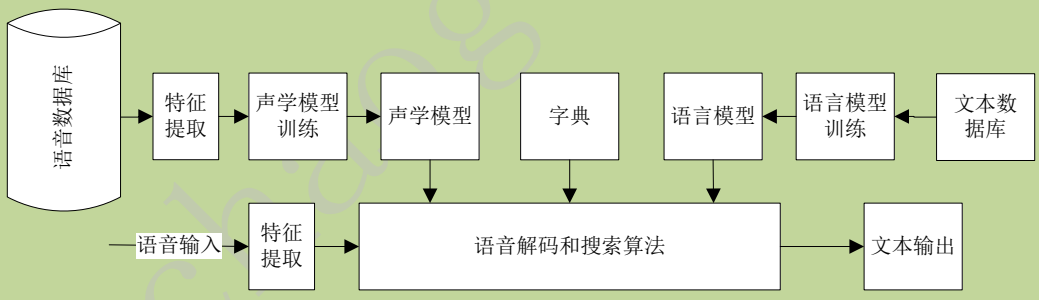


图 5.1 连续语音识别框图

在任意一个 ASR（automatic speech recognition）系统中第一步是特征提取，声道的形状（舌头，牙齿）在语音短时功率谱的包络中显示出来。而 MFCC（Mel Frequency Cepstral Coefficent，梅尔频率倒谱系数）就是一种准确描述该包络的一种特征向量。目前业界广泛使用的是基于 deep Learning 的特征学习，确切的说是深度神经元网络（Deep Neural Network（DNN））。

5.1 ASR 模型

5.1.1 高斯混合模型 GMM(Gaussian mixture model)

对于说话人识别，语言特性降噪以及语音识别适用。

若随机变量 X 服从一个位置参数为 μ、尺度参数为 σ 的概率分布，其概率密度函数

是：

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad 5.4$$

则称该变量是高斯分布(正态分布)的。记作 $X \sim N(\mu, \sigma^2)$ ；正态随机向量 $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)^T$ 的高斯分布是：

$$f(\boldsymbol{x}) = \frac{1}{\sqrt[D]{2\pi}\sqrt{|\Sigma|}} e\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \quad 5.5$$

也常记作：$X \sim N(\boldsymbol{\mu} \in R^D, \Sigma \in R^{DxD})$ ，一个连续标量 x 的混合高斯分布的概率密度函数是：

$$f(x) = \sum_{m=1}^{M} \frac{c_m}{\sqrt{2\pi}\sigma_m} e^{\left[-\frac{1}{2}(\frac{x-\mu_m}{\sigma_m})\right]} = \sum_{m=1}^{M} c_m N(x; \mu_m, \sigma_m^2), (-\infty < x < +\infty; \sigma_m > 0; c_m > 0) \quad 5.6$$

混合权重的和等于一，即 $\sum_{m=1}^{M} c_m = 1$ 。和单峰高斯分布相比，5.6 是一个具有多个峰值的分布（混合高斯分布），体现在 M>1。

混合高斯分布（5.6）随机变量 x 的期望是 $E(x) = \sum_{m=1}^{M} c_m \mu_m$ 。

多元混合高斯分布的联合概率密度函数是：

$$f(x) = \sum_{m=1}^{M} \frac{c_m}{\sqrt[D]{2\pi}\sqrt{|\Sigma_m|}} e^{\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^T \Sigma_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m)\right]} = \sum_{m=1}^{M} c_m N(x; \mu_m, \Sigma_m), (c_m > 0) \quad 5.7$$

5.1.2 参数估计

$\Theta = \{c_m, \mu_m, \Sigma_m\}$

使用最大似然估计法估计混合高斯分布的参数，

$$c_m^{(j+1)} = \frac{1}{N} \sum_{t=1}^{N} h_m^{(j)}(t) \quad 5.8$$

$$\mu_m^{(j+1)} = \frac{\sum_{t=1}^{N} h_m^{(j)}(t) x^{(t)}}{\sum_{t=1}^{N} h_m^{(j)}(t)} \quad 5.9$$

$$\Sigma_m^{(j+1)} = \frac{\sum_{t=1}^{N} h_m^{(j)} \left[x^t - \mu_m^j\right]\left[x^t - \mu_m^j\right]^T}{\sum_{t=1}^{N} h_m^{(j)}(t)} \quad 5.10$$

后验概率的计算如下：

$$h_m^j(t) = \frac{c_m^{(j)} N(x^{(t)}; \mu_m^{(j)}, \Sigma_m^{(j)})}{\sum_{i=1}^{n} c_i^{(j)} N(x^{(t)}; \mu_i^{(j)}, \Sigma_i^{(j)})} \quad 5.11$$

基于当前（第 j 次）的参数估计，$x^{(t)}$ 的条件概率取决于每一个的采样。

5.1.2 隐马尔可夫模型 HMM（hidden Markov model）

马尔科夫链

$X_1, X_2, X_3 \ldots$ 马尔科夫链（Markov Chain），描述一种状态序列，其每个状态值取决于前面有限个状态。马尔科夫链是具有马尔科夫性质的随机变量的一个数列。这些变量的范围，就是它们所有可能的集合（又称状态空间），X_n 是在状态 n 的状态。如果 X_{n+1} 对于过去状态的条件概率分布仅是 X_n 的一个函数，则：

$$P(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(X_{n+1} = x \mid X_n = x_n)$$

这里 x 为过程中的某个状态。

隐马尔可夫模型是统计模型，其被用来描述一个含有隐含位置参数的马尔科夫过程。其难点是从可观察的参数中确定该过程的隐含参数。然后利用这些参数来作进一步的分析，例如模式识别。

5.2 MFCC

MFCC 将语音被分为很多帧，每帧语音对应于一个频谱（通过短时 FFT 计算），频谱表示频率与能量的关系。在实际使用中，频谱图有三种，即线性振幅谱、对数振幅谱、自功率谱（对数振幅谱中各谱线的振幅都作了对数计算，所以其纵坐标的单位是 dB（分贝）。这个变换的目的是使那些振幅较低的成分相对高振幅成分得以拉高，以便观察掩盖在低幅噪声中的周期信号）。

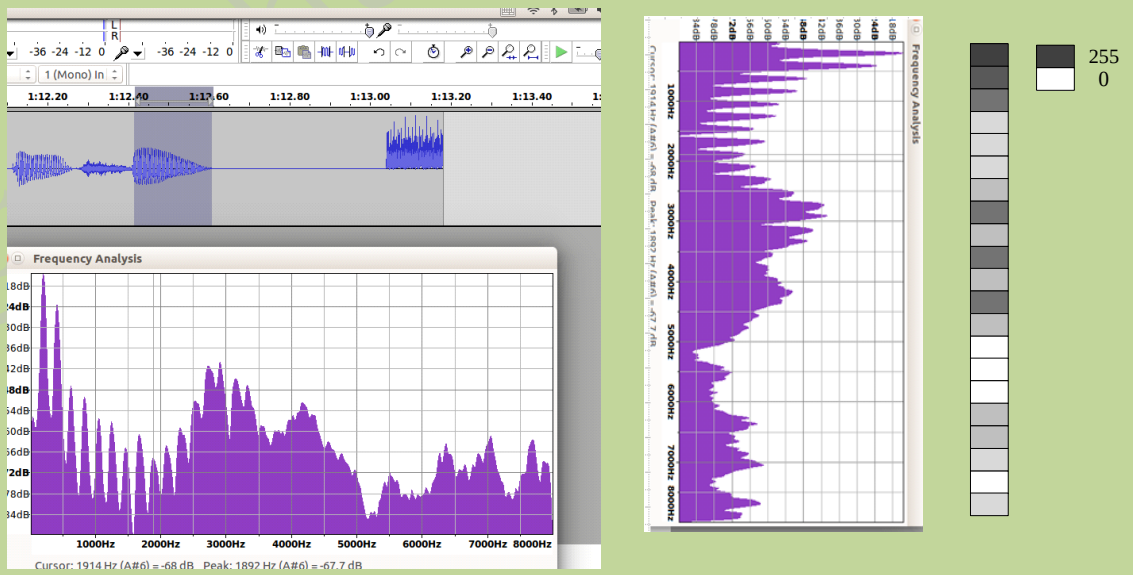


图 5.2 一帧信号的频谱变换

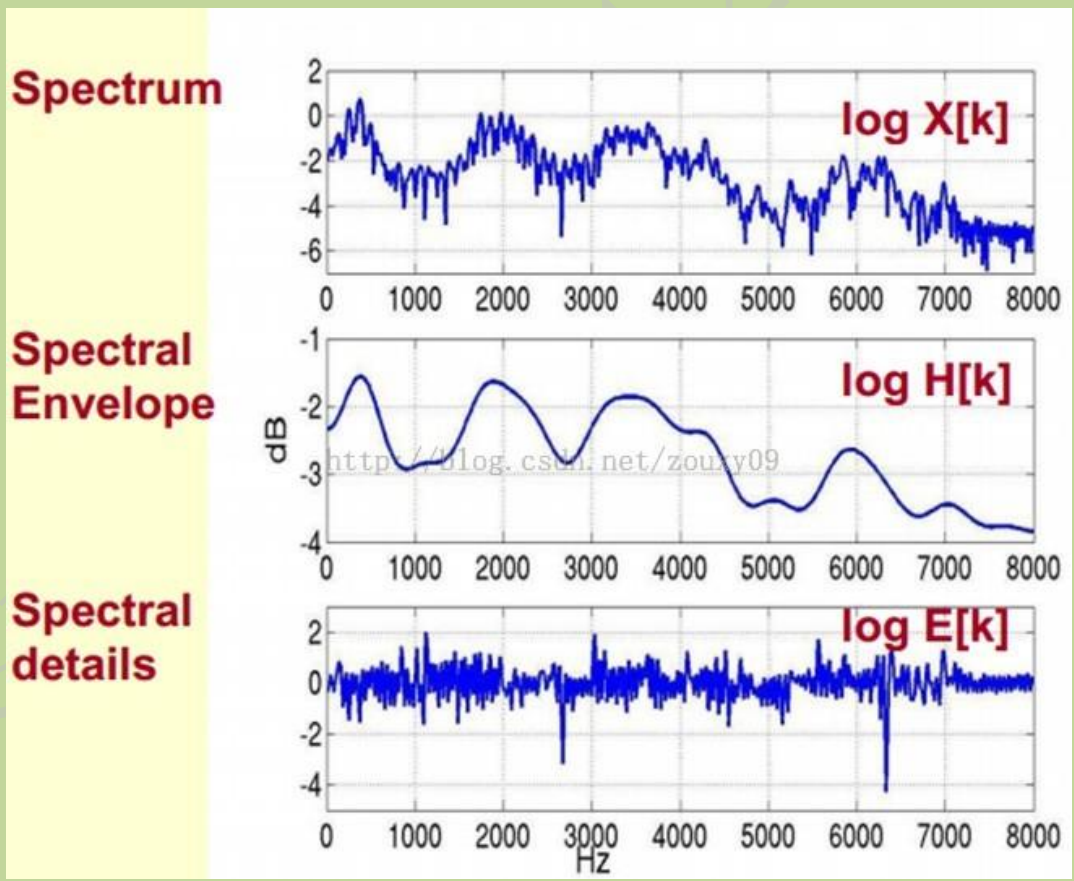
先将其中一帧语音（蓝色）的频谱（红色）通过坐标表示出来，如上图。现在将频谱旋

转 90 度。然后把这些幅度映射到一个灰度级表示，255 表示黑， 0 表示白色。幅度值越大，相应的区域越黑。这样我们会得到一个随着时间变化的频谱图，这个就是描述语音信号的 spectrogram 声谱图。

音素（Phones）的属性可以更好的在声谱图里观察出来。另外，通过观察共振峰和它们的转变可以更好的识别声音。隐马尔科夫模型（Hidden Markov Models）就是隐含地对声谱图进行建模以达到好的识别性能。还有一个作用就是它可以直观的评估 TTS 系统（text to speech）的好坏，直接对比合成的语音和自然的语音声谱图的匹配度即可。图 5.2 是中文“区”这个字的时域和频域语音的频谱图。峰值就表示语音的主要频率成分，这些峰值称为共振峰（formants），而共振峰就是携带了声音的辨识属性。用它就可以识别不同的声音。

要提取的不仅仅是共振峰的位置，还得提取它们转变的过程。所以提取的是频谱的包络（Spectral Envelope）。这包络就是一条连接这些共振峰点的平滑曲线。

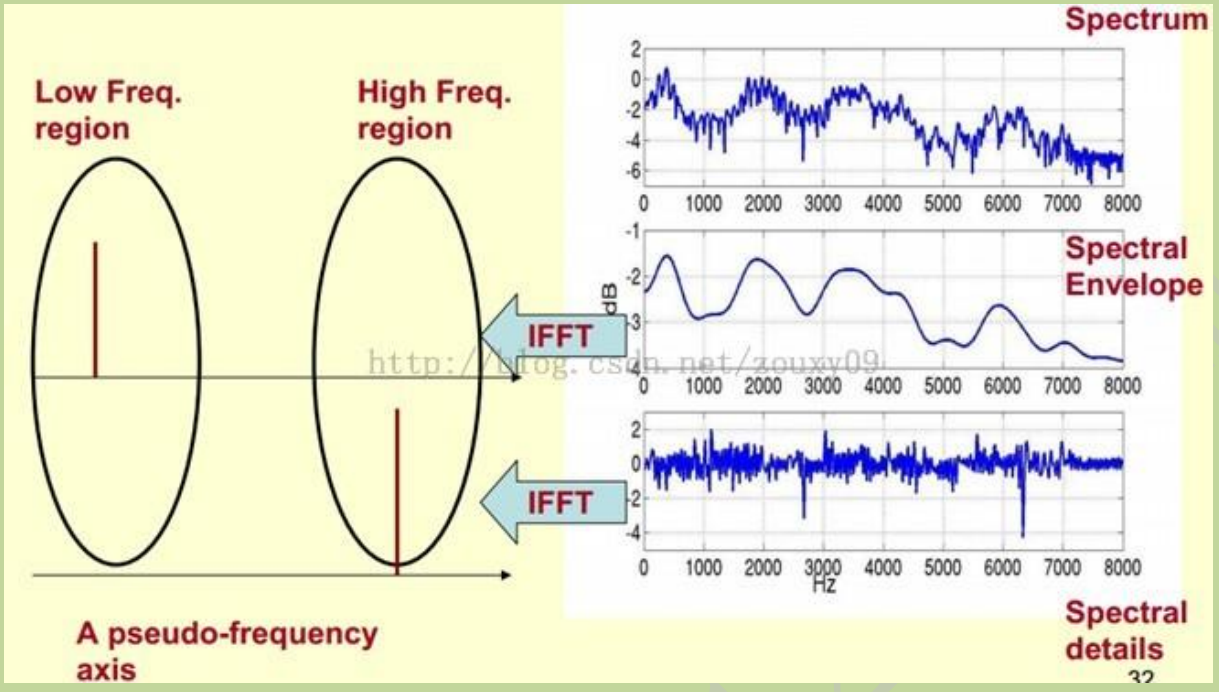
可以这么理解，原始的频谱由两部分组成：包络和频谱的细节。这里用到的是对数频谱，所以单位是 dB。那现在我们需要把这两部分分开，这样我们就可以得到包络了。



那怎么把他们分离开呢？也就是，怎么在给定 log X[k]的基础上，求得 log H[k] 和 log E[k]以满足 log X[k] = log H[k] + log E[k]。

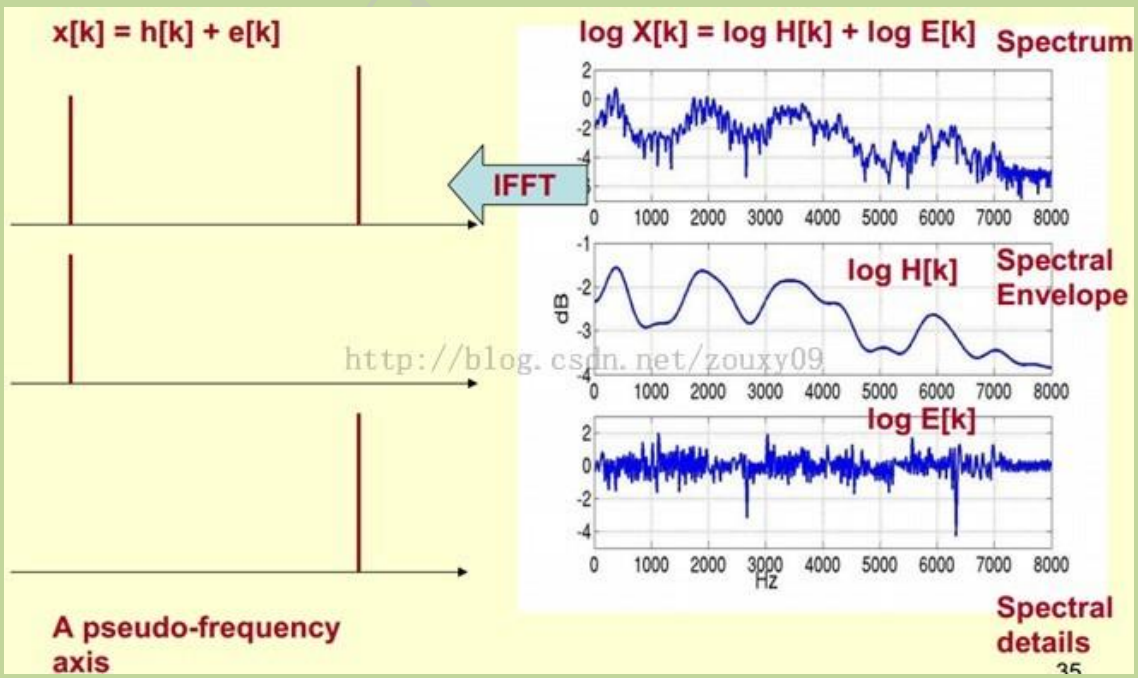
为了达到这个目标，需要 Play a Mathematical Trick。这个 Trick 是什么呢？就是对频谱做 FFT。在频谱上做傅里叶变换就相当于逆傅里叶变换 Inverse FFT (IFFT)。需要注意的一点

是，是在频谱的对数域上面处理的，这也属于 Trick 的一部分。这时候，在对数频谱上面做 IFFT 就相当于在一个伪频率（pseudo-frequency）坐标轴上面描述信号。



由上面这个图可以看到，包络是主要是低频成分（这时候需要转变思维，这时候的横轴就不要看成是频率了，可以看成时间），把它看成是一个每秒 4 个周期的正弦信号。这样在伪坐标轴上面的 4Hz 的地方给它一个峰值。而频谱的细节部分主要是高频。把它看成是一个每秒 100 个周期的正弦信号。这样在伪坐标轴上面的 100Hz 的地方给它一个峰值。

把它俩叠加起来就是原来的频谱信号了。



在实际中已经知道 log X[k]，所以可以得到了 x[k]。那么由图可以知道，h[k]是 x[k]的低频部分，那么将 x[k]通过一个低通滤波器就可以得到 h[k]了。

x[k]实际上就是倒谱 Cepstrum（这个是一个新造出来的词，把频谱的单词 spectrum 的前面四个字母顺序倒过来就是倒谱的单词了）。h[k]就是倒谱的低频部分。h[k]描述了频谱的包络，它在语音识别中被广泛用于描述特征。

现在总结下倒谱分析，它实际上是这样一个过程：

1）将原语音信号经过傅里叶变换得到频谱：X[k]=H[k]E[k]；

只考虑幅度就是：|X[k] |=|H[k]||E[k] |；

2）我们在两边取对数：log||X[k] ||= log ||H[k] ||+ log ||E[k] ||。

3）再在两边取逆傅里叶变换得到：x[k]=h[k]+e[k]。

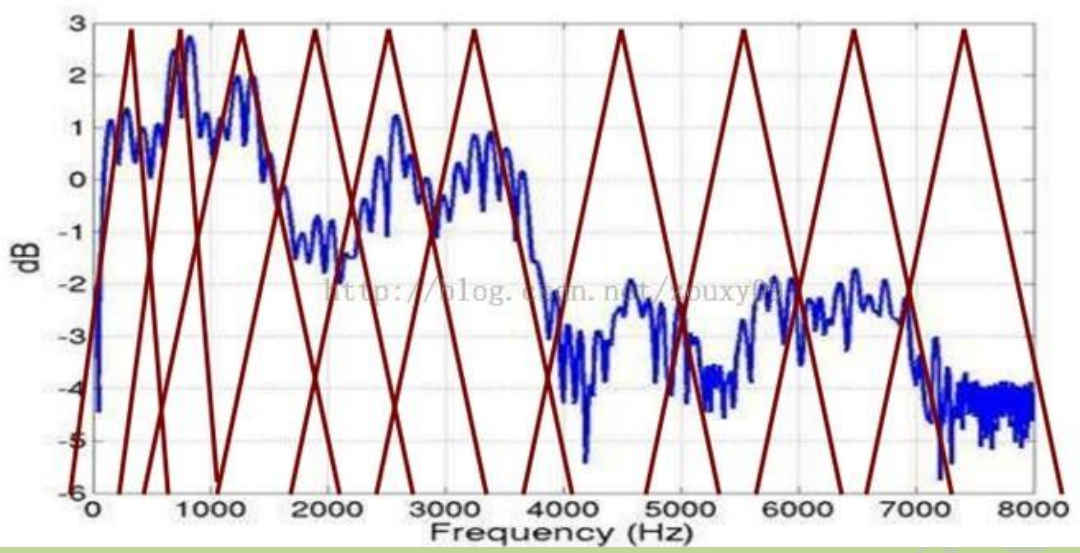
这实际上有个专业的名字叫做同态信号处理。它的目的是将非线性问题转化为线性问题的处理方法。对应上面，原来的语音信号实际上是一个卷性信号（声道相当于一个线性时不变系统，声音的产生可以理解为一个激励通过这个系统），第一步通过卷积将其变成了乘性信号（时域的卷积相当于频域的乘积）。第二步通过取对数将乘性信号转化为加性信号，第三步进行逆变换，使其恢复为卷性信号。这时候，虽然前后均是时域序列，但它们所处的离散时域显然不同，所以后者称为倒谱频域。

总结下，倒谱（cepstrum）就是一种信号的傅里叶变换经对数运算后再进行傅里叶反变换得到的谱。它的计算过程如下：

5.2.1Mel 频率分析（Mel-Frequency Analysis）

给一段语音，可以得到了它的频谱包络（连接所有共振峰值点的平滑曲线）了。但是，对于人类听觉感知的实验表明，人类听觉的感知只聚焦在某些特定的区域，而不是整个频谱包络。

而 Mel 频率分析就是基于人类听觉感知实验的。实验观测发现人耳就像一个滤波器组一样，它只关注某些特定的频率分量（人的听觉对频率是有选择性的）。也就说，它只让某些频率的信号通过，而压根就直接无视它不想感知的某些频率信号。但是这些滤波器在频率坐标轴上却不是统一分布的，在低频区域有很多的滤波器，他们分布比较密集，但在高频区域，滤波器的数目就变得比较少，分布很稀疏。



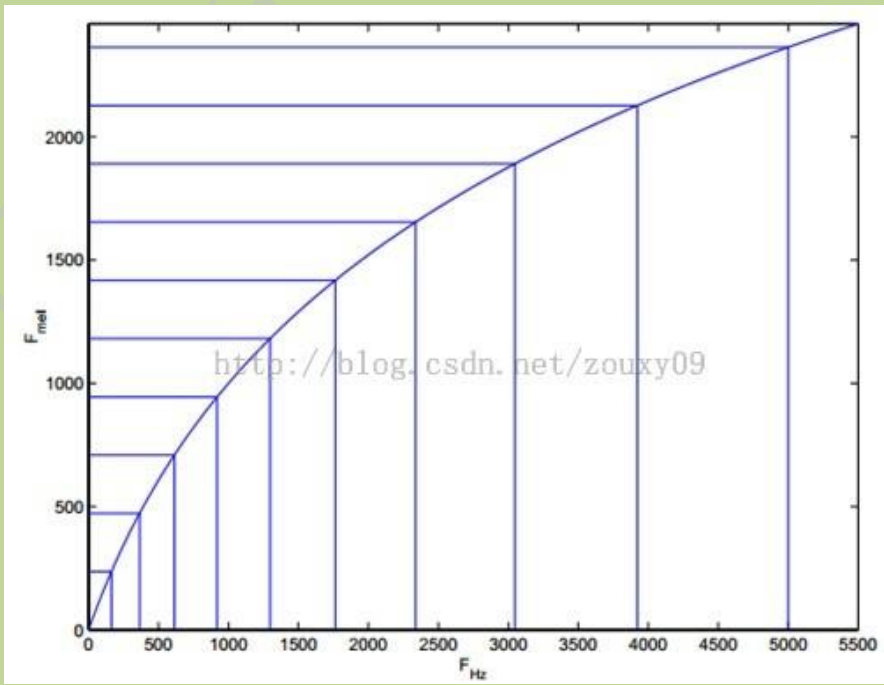
人的听觉系统是一个特殊的非线性系统，它响应不同频率信号的灵敏度是不同的。在语音特征的提取上，人类听觉系统做得非常好，它不仅能提取出语义信息，而且能提取出说话人的个人特征，这些都是现有的语音识别系统所望尘莫及的。如果在语音识别系统中能模拟人类听觉感知处理特点，就有可能提高语音的识别率。

梅尔频率倒谱系数（Mel Frequency Cepstrum Coefficient，MFCC）考虑到了人类的听觉特征，先将线性频谱映射到基于听觉感知的 Mel 非线性频谱中，然后转换到倒谱上。

将普通频率转化到 Mel 频率的公式是：

$$mel(f) = 2595 * \log_{10}(1 + f/700)$$

由下图可以看到，它可以将不统一的频率转化为统一的频率，也就是统一的滤波器组。



在 Mel 频域内，人对音调的感知度为线性关系。举例来说，如果两段语音的 Mel 频率相差两倍，则人耳听起来两者的音调也相差两倍。

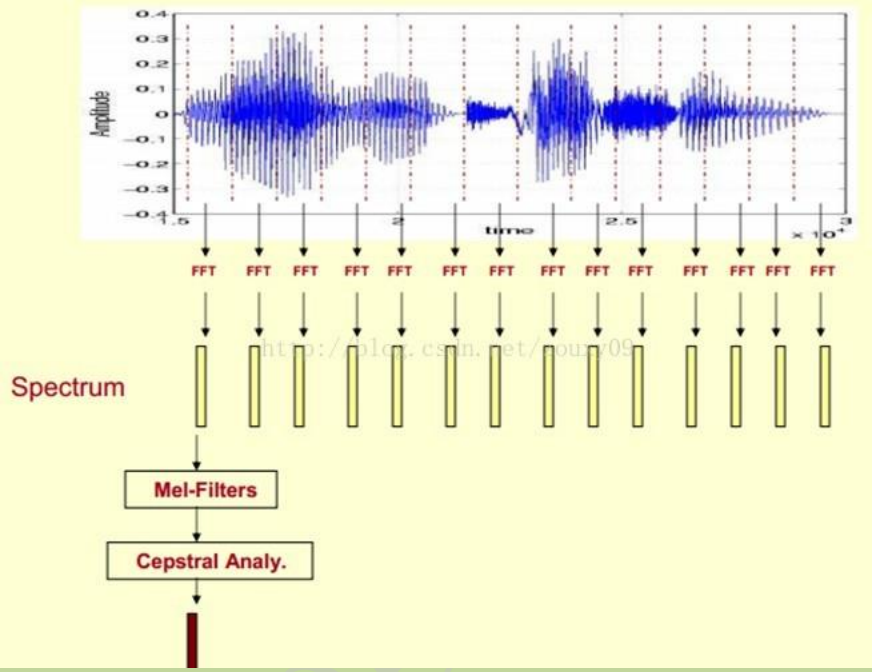
5.2.2 Mel 频率倒谱系数（Mel-Frequency Cepstral Coefficients）

将频谱通过一组 Mel 滤波器就得到 Mel 频谱。公式表述就是：log X[k] = log (Mel-Spectrum)。这时候我们在 log X[k]上进行倒谱分析：

1）取对数：log X[k] = log H[k] + log E[k]。

2）进行逆变换：x[k] = h[k] + e[k]。

在 Mel 频谱上面获得的倒谱系数 h[k]就称为 Mel 频率倒谱系数，简称 MFCC。



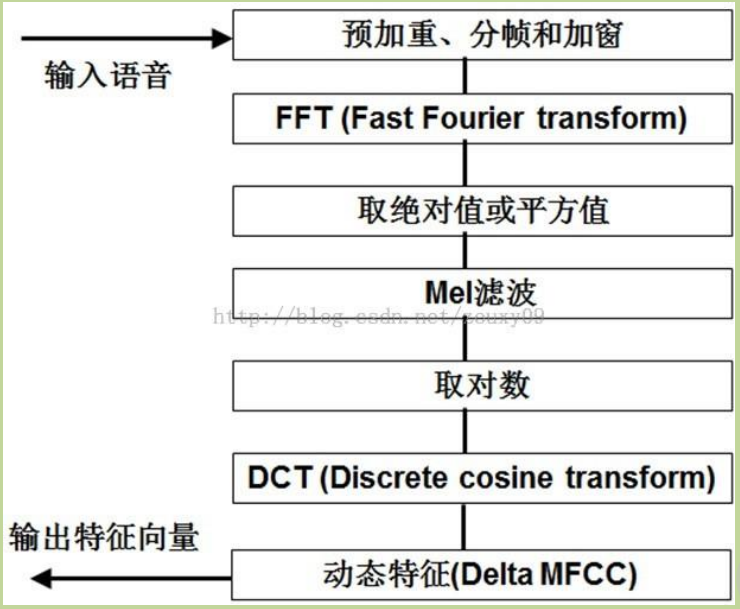
总结下提取 MFCC 特征的过程：（具体的数学过程网上太多了，这里就不想贴了）

1）先对语音进行预加重、分帧和加窗；

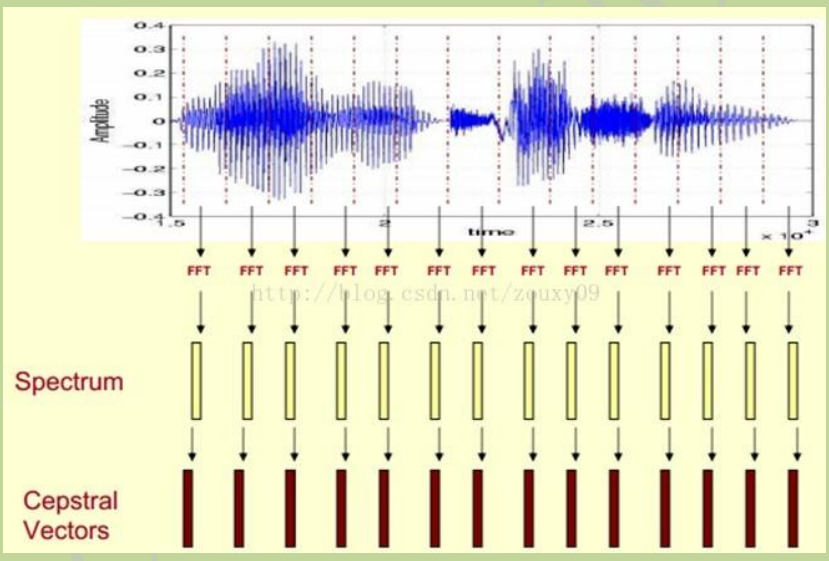
2）对每一个短时分析窗，通过 FFT 得到对应的频谱；

3）将上面的频谱通过 Mel 滤波器组得到 Mel 频谱；

4）在 Mel 频谱上面进行倒谱分析（取对数，做逆变换，实际逆变换一般是通过 DCT 离散余弦变换来实现，取 DCT 后的第 2 个到第 13 个系数作为 MFCC 系数），获得 Mel 频率倒谱系数 MFCC，这个 MFCC 就是这帧语音的特征；



这时候，语音就可以通过一系列的倒谱向量来描述了，每个向量就是每帧的 MFCC 特征向量。



这样就可以通过这些倒谱向量对语音分类器进行训练和识别了。

DNN（Deep neutral network）深度神经网络

15.1 深度神经元网络架构

深度神经元网络是传统的多层感知系统（MLP，multiplayer perceptron）。式 15.1 是一个五层神经元网络，一个输入层，三个隐藏层和一个输出层。

$$\boldsymbol{V}^l = f(\boldsymbol{z}^l) = f(\boldsymbol{W}^l \boldsymbol{v}^{l-1} + \boldsymbol{b}^l), 0 < l < L \qquad 15.1$$

此处，$\boldsymbol{z}^l = \boldsymbol{W}^l \boldsymbol{v}^l + \boldsymbol{b}^l \in \Re^{N_l \times 1}$ ，$\boldsymbol{v}^l \in \Re^{N_l \times 1}$ ，$\boldsymbol{W}^l \in \Re^{N_l \times N_{l-1}}$ ，$\boldsymbol{b}^l \in \Re^{N_l \times 1}$ ，$N_l \in \Re$；它们

分别是激励向量，activation 向量，权重矩阵，偏移向量以及 l 层神经元数。$\boldsymbol{V}^0 = \mathbf{0} \in \Re^{N_0 \times 1}$ 是观测（特征）向量。$N_0 = D$ 是特征维度。$f(\bullet): \Re^{N_l \times 1} \to \Re^{N_l \times l}$ 是激励函数。在大多数应用中，激励函数是 s 型函数是：

$$\sigma(z) = \frac{1}{1+e^{-z}}, \quad 15.2$$

也可能是双曲正切函数：

$$\tanh(z) = \frac{e^z - e^z}{e^z + e^z} \quad 15.3$$

对于回归型问题，线性层常采用下式模型：

$$\boldsymbol{v}^L = \boldsymbol{z}^L = \boldsymbol{W}^L \boldsymbol{v}^{L-1} + \boldsymbol{b}^L \quad 15.4$$

生成输出向量 $\boldsymbol{v}^L \in \Re^{N_L}$，$N_L$ 是 输出维数。

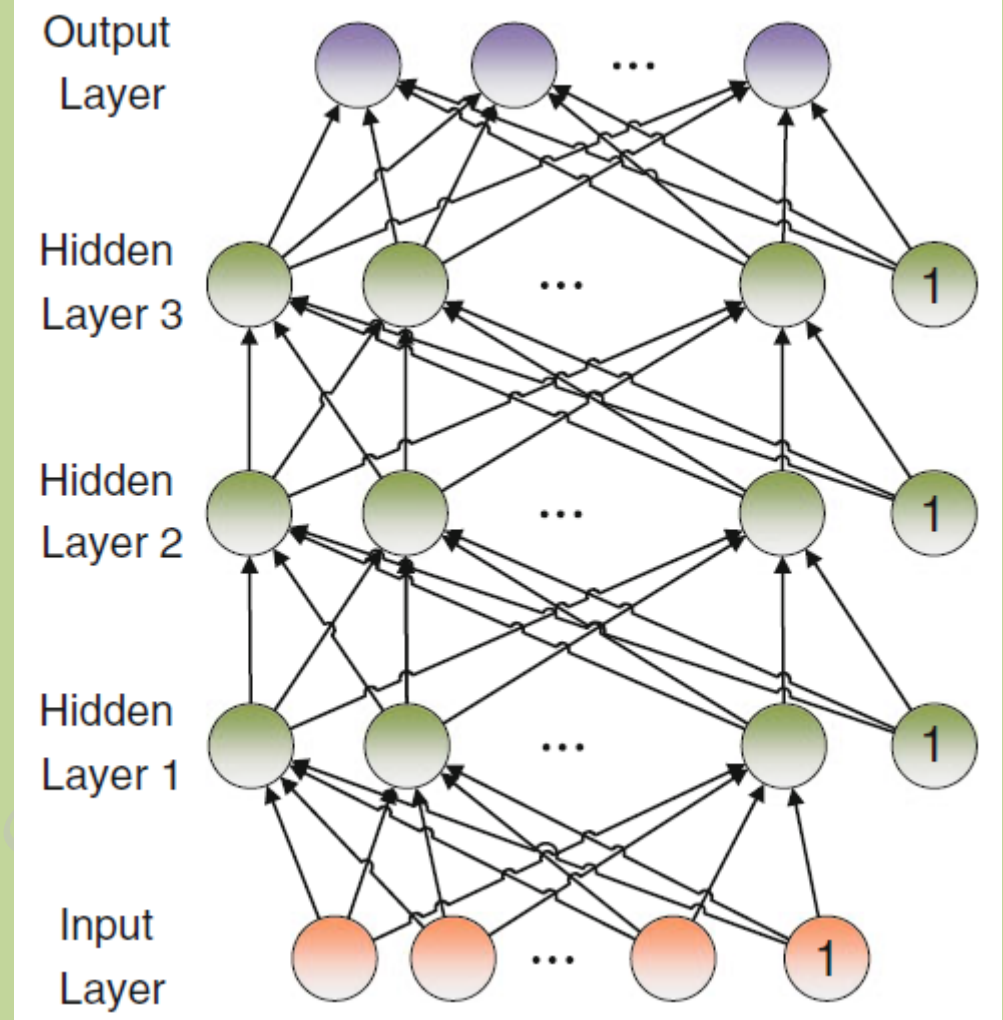


图 15.1 深度神经元网络

对于分类问题，神经元的每一个输出代表一个类 $i \in \{1,...,C\}$，$C = N_L$ 是类的总数。第 i 个输出 v_i^L 代表了观测向量 O 的观测概率 $P_{dnn}(i \mid \boldsymbol{o})$ 。

DNN 前向计算过程：

$\boldsymbol{V}^0 \leftarrow \boldsymbol{o}$

for $l \leftarrow 1;\ l < L; l \leftarrow l+1$ do

$\boldsymbol{Z}^l \leftarrow \boldsymbol{W}^l \boldsymbol{V}^{l-1} + \boldsymbol{B}^l$

$\boldsymbol{V}^l \leftarrow f(\boldsymbol{Z}^l)$

end for

$\boldsymbol{Z}^L \leftarrow \boldsymbol{W}^L \boldsymbol{V}^{L-1} + \boldsymbol{B}^L$

if regression then

$\boldsymbol{V}^l \leftarrow \boldsymbol{Z}^l$

else

$\boldsymbol{V}^l \leftarrow soft\max(\boldsymbol{Z}^l)$

return $\boldsymbol{V}^l$

end procedure

15.2 误差反向传播参数估计

DNN 的模型参数$\{\boldsymbol{W},\boldsymbol{b}\}$是未知的，可以从训练样本$S=\{(\boldsymbol{o}^m,\boldsymbol{y}^m)\,|\,0\le m<M\}$中估计，M 是训练的样本数，$\boldsymbol{o}^m$是 m 阶观测向量，$\boldsymbol{y}^m$是期望的输出向量。这一过程通常称为训练过程或者参数估计过程。

15.2.1 训练评价准则

最小期望损准则：

$$J_{EL} = E\big(J(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y})\big) = \int_o J(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y})\,p(\boldsymbol{o})\,d(\boldsymbol{o}) \quad 15.7$$

$\{\boldsymbol{W},\boldsymbol{b}\}$是参数模型，$\boldsymbol{o}$是观测向量，$\boldsymbol{y}$是输出向量，$p(o)$是观测向量$\boldsymbol{o}$概率密度函数，$J(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y})$是损耗函数。

回归问题常采用均方误差准则：

$$J_{MSE}(\boldsymbol{W},\boldsymbol{b};S) = \frac{1}{M}\sum_{m=1}^{M} J_{MSE}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o}^m,\boldsymbol{y}^m) \qquad 15.8$$

$$J_{MSE}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) = \frac{1}{2}\left\|\boldsymbol{v}^L-\boldsymbol{y}\right\|^2 = \frac{1}{2}(\boldsymbol{v}^L-\boldsymbol{y})^T(\boldsymbol{v}^L-\boldsymbol{y}) \quad 15.9$$

对于分类问题，$\boldsymbol{y}$是概率分布，则交叉熵准则是：

$$J_{CE}(\boldsymbol{W},\boldsymbol{b};S) = \frac{1}{M}\sum_{m=1}^{M} J_{CE}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o}^m,\boldsymbol{y}^m) \qquad 15.10$$

$$J_{CE}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) = \sum_{i=1}^{C} y_i \log v_i^L \qquad 15.11$$

$y_i = P_{emp}(i|\boldsymbol{o})$ 是观测向量 O 所属分类 i 的经验概率，$v_i^L = P_{dnn}(i|o)$ 是和上式一样的 DNN 的概率。

15.2.2 训练算法

误差反向传递算法：

$$\boldsymbol{W}_{t+1}^l \leftarrow \boldsymbol{W}_t^l - \varepsilon \Delta \boldsymbol{W}_t^l \qquad 15.14$$

$$\boldsymbol{b}_{t+1}^l \leftarrow \boldsymbol{b}_t^l - \varepsilon \Delta b_t^l \qquad 15.15$$

$\boldsymbol{W}_t^l$ 和 $\boldsymbol{b}_t^l$ 分别是 l 层的第 t 次更新的权向量和偏移向量。

$$\Delta W_t^l = \frac{1}{M_b}\sum_{m=1}^{M_b} \nabla_{W_t^l} J(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o}^m,\boldsymbol{y}^m) \qquad 15.16$$

$$\Delta b_t^l = \frac{1}{M_b}\sum_{m=1}^{M_b} \nabla_{b_t^l} J(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o}^m,\boldsymbol{y}^m) \qquad 15.17$$

分别是 t 时刻的平均权重梯度和平均偏移向量。

最上层的权重矩阵和偏移向量的梯度取决于选择的准则，对于回归问题，MSE 训练准则（15.9 且其线性输出层是式 15.5，则输出层的权重矩阵的梯度是：

$$\nabla_{W_t^L} J_{MSE}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) = \nabla_{z_t^L} J_{MSE}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) \frac{\partial \boldsymbol{z}_t^L}{\partial \boldsymbol{W}_t^L} = e_t^L (v_t^{L-1})^T$$

15.3 实际处理中的问题

15.3.1 数据预处理

在语音识别中，CMN(Cepstral Normalization)倒谱系数归一化，即减去每个音节 MFCC 特征的均值能减小通道引入的畸变。CMN 的归一化可以通过首先估算样本的均值，然后用样本值减去估算均值而得到。

15.3.2 模型初始化

15.3.3 权重衰减

控制过拟合的方法，过拟合是最最大限度减少预期损失带来的，因为实际上是减少的训练集的损失。控制过拟合的方法是规范训练准则使参数模型并不是十分适合样本。常用的规范标准包括：基于 L_1 规范

$$R_1(\boldsymbol{W}) = \left\|vec(\boldsymbol{W})\right\|_1 = \sum_{l=1}^{L}\left\|vec(\boldsymbol{W}^l)\right\|_1 = \sum_{l=1}^{L}\sum_{i=1}^{N_l}\sum_{j=1}^{N_{l-1}}\left|\boldsymbol{W}_{ij}^l\right| \qquad 15.37$$

基于 L_2 规范，

$$R_2(\boldsymbol{W}) = \left\|vec(\boldsymbol{W})\right\|_2^2 = \sum_{l=1}^{L}\left\|vec(\boldsymbol{W}^l)\right\|_2^2 = \sum_{l=1}^{L}\sum_{i=1}^{N_l}\sum_{j=1}^{N_{l-1}}\left(\left|\boldsymbol{W}_{ij}^l\right|\right)^2 \qquad 15.38$$

$\boldsymbol{W}_{ij}$ 是矩阵 $\boldsymbol{W}$ 的 (i,j) 的第 n 个值，$vec(\boldsymbol{W}) \in \Re^{[N_l \times N_{l-1}]\times 1}$ 是矩阵 $\boldsymbol{W}^l$ 所有列的组合。$\left\|vec(\boldsymbol{W}^l)\right\|_2$ 等于 $\left\|\boldsymbol{W}^l\right\|_F$，即 $\boldsymbol{W}^l$ 的弗罗贝尼乌斯范数。

当引入规范准则，训练模型准则变为：

$$\ddot{J}(\boldsymbol{W},\boldsymbol{b};S) = J(\boldsymbol{W},\boldsymbol{b};S) + \lambda R(\boldsymbol{W}) \qquad 4.39$$

$J(\boldsymbol{W},\boldsymbol{b};S)$ 是 $J_{MSE}(\boldsymbol{W},\boldsymbol{b};S)$ 或者 $J_{CE}(\boldsymbol{W},\boldsymbol{b};S)$ 在训练集 S 上对应的经验损失最优。λ 是插入权重。

$$\nabla_{W_t^l} \ddot{J}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) = \nabla_{W_t^l} J(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) + \lambda \nabla_{W_t^l} R(\boldsymbol{W}) \qquad 4.40$$

$$\nabla_{\boldsymbol{b}_t^l} \ddot{J}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) = \nabla_{\boldsymbol{b}_t^l} J(\boldsymbol{W},\boldsymbol{b};\boldsymbol{o},\boldsymbol{y}) \qquad 4.40$$

此处：

$\nabla_{W_t^l} R_1(\boldsymbol{W}) = \mathrm{sgn}(\boldsymbol{W}_t^l)$，$\nabla_{W_t^l} R_2(\boldsymbol{W}) = 2\boldsymbol{W}_t^l$

权重衰减在样本数较少时很有用。在语音识别类型的问题中权重矩阵通常有一百万个参数，当样本集很大时，插入权重 λ 应当很小甚至是零。

15.3.4 丢弃

过拟合问题也会采用丢弃的方式解决问题，丢弃的基本思想是对每一个隐藏层随机忽略一部分神经元。这就要求检测模式的神经元之间的依赖性较弱。

15.4 高级模型初始化方法

15.4.1 受限玻尔兹曼机

RBM（Restricted Boltzmann machine）是一个随机神经生成网络。本质上是一个无向图，该图包括一层随机可见的神经和一层随机不可见神经。对于二值 RBM，$\boldsymbol{v} \in \{0,1\}^{N_v \times 1}$，$\boldsymbol{h} \in \{0,1\}^{N_h \times 1}$，对每一个可见向量 $\boldsymbol{v}$ 和不可见向量 $\boldsymbol{h}$ RBM 注入能量如下：

$$E(\boldsymbol{v},\mathbf{h}) = -\boldsymbol{a}^T\boldsymbol{v} - \boldsymbol{b}^T\boldsymbol{h} - \boldsymbol{h}^T\boldsymbol{W}\boldsymbol{v} \quad 15.4.1$$

N_v 和 N_h 分别是可见和隐藏神经元数。$W \in \Re^{N_h \times N_v}$ 是权重矩阵，a 和 b 是偏移矩阵。如果可见矩阵是实数，则 RBM 变成高斯-伯努利分布（Gaussian-Bernoulli），注入能量变成：

$$E(\boldsymbol{v},\mathbf{h}) = \frac{1}{2}(\boldsymbol{v}-\boldsymbol{a})^T(\boldsymbol{v}-\boldsymbol{a}) - \boldsymbol{b}^T\boldsymbol{h} - \boldsymbol{h}^T\boldsymbol{W}\boldsymbol{v} \qquad 15.4.2$$

根据能量，每一种情况分配了一个概率：

$$P(\boldsymbol{v},\boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z} \quad 15.4.3$$

其中 $Z=\sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$ 是已知分区函数的归一化因子。

在 RBM 算法中， $P(\boldsymbol{v}\mid\boldsymbol{h})$ 和 $P(\boldsymbol{h}\mid\boldsymbol{v})$ 的后验概率是很容易计算的。例如在伯努利-伯努利模型中：

$$P(\boldsymbol{v}\mid\boldsymbol{h})=\frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}}=\frac{e^{\boldsymbol{a}^T\boldsymbol{v}+\boldsymbol{b}^T\boldsymbol{h}+\boldsymbol{h}^T\boldsymbol{W}\boldsymbol{v}}}{\sum_{\tilde{\boldsymbol{h}}} e^{\boldsymbol{a}^T\boldsymbol{v}+\boldsymbol{b}^T\tilde{\boldsymbol{h}}+\tilde{\boldsymbol{h}}^T\boldsymbol{W}\boldsymbol{v}}}=\frac{\prod_i e^{b_i h_i+h_i W_{i,*}\boldsymbol{v}}}{\sum_{\tilde{\boldsymbol{h}}}\cdots\sum_{\tilde{\boldsymbol{h}}N}\prod_i e^{b_i\tilde{h}_i+\tilde{h}_i W_{i,*}\boldsymbol{v}}}$$

$$=\prod_i\frac{e^{b_i h_i+h_i W_{i,*}\boldsymbol{v}}}{\sum_{\tilde{\boldsymbol{h}}N} e^{b_i\tilde{h}_i+\tilde{h}_i W_{i,*}\boldsymbol{v}}}=\prod_i P(h_i\mid\boldsymbol{v}) \qquad 15.4.4$$

$W_{i,*}$ 是矩阵 W 的第 i 行。15.4.4 表明相对于可见向量隐藏神经元是有条件独立的。

$$P(h_i\mid\boldsymbol{v})=\frac{e^{b_i 1+1W_{i,*}\boldsymbol{v}}}{e^{b_i 1+1W_{i,*}\boldsymbol{v}}+e^{b_i 0+0W_{i,*}\boldsymbol{v}}}=\sigma(b_i+\boldsymbol{W}_{i,*}v) \qquad 15.4.5$$

15.4.2 RBM 性能

定义如下自由能量：

$$F(\boldsymbol{v})=-\log\left(\sum_h e^{-E(\boldsymbol{v},\boldsymbol{h})}\right) \quad 15.4.6$$

使用自由能量，可以将边际概率 $P(v)$ 写成如下：

$$P(v)=\sum_{\boldsymbol{h}} P(\boldsymbol{v},\boldsymbol{h})=\sum_h\frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z}=\frac{\sum_h e^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z}=\frac{e^{-F(v)}}{\sum_v e^{-F(v)}} \qquad 15.4.7$$

RBM 参数学习

训练 RBM 时，采用了随机梯度下降法（SGD）来最小化负对数似然（NLL，negative log likelihood）的方法。

$$J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v})=-\log P(\boldsymbol{v})=F(\boldsymbol{v})+\log\sum_v e^{-F(\boldsymbol{v})} \qquad 15.4.8$$

参数更新方法如下：

$$\boldsymbol{W}_{t+1}\leftarrow\boldsymbol{W}_t-\varepsilon\Delta\boldsymbol{W}_t$$

$$\boldsymbol{a}_{t+1}\leftarrow\boldsymbol{a}_t-\varepsilon\Delta\boldsymbol{a}_t$$

$$\boldsymbol{b}_{t+1}\leftarrow\boldsymbol{b}_t-\varepsilon\Delta\boldsymbol{b}_t$$

这里 ε 是学习速率：

$$\Delta\boldsymbol{W}_t=\rho\Delta\boldsymbol{W}_{t-1}+(1-\rho)\frac{1}{M_b}\sum_{m=1}^{M_b}\nabla_{W_t}J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}^m)$$

$$\Delta\boldsymbol{a}_t=\rho\Delta a_{t-1}+(1-\rho)\frac{1}{M_b}\sum_{m=1}^{M_b}\nabla_{a_t}J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}^m)$$

$$\Delta \boldsymbol{b}_t = \rho \Delta \boldsymbol{b}_{t-1} + (1-\rho)\frac{1}{M_b}\sum_{m=1}^{M_b} \nabla_{\boldsymbol{b}_t} J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}^m)$$

ρ 是动量参数，M_b 是分批处理量的大小，$\nabla_{\boldsymbol{W}_t} J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}^m)$ ，$\nabla_{\boldsymbol{a}_t} J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}^m)$，$\nabla_{\boldsymbol{b}_t} J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}^m)$ 分别是 NLL 准则模型参数 W ，$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的梯度。

NLL 的倒数如下：

$$\nabla_\theta J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}) = -\left[\left\langle \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \theta}\right\rangle_{data} - \left\langle \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \theta}\right\rangle_{model}\right] \quad 15.4.9$$

θ 是一些模型参数，$\langle x\rangle_{data}$ 和 $\langle x\rangle_{model}$ 分别是从数据和模型估计的 x 期望。对于可见和隐藏权重的更新有下式：

$$\nabla_{w_{ij}} J_{NLL}(W,a,b;v) = -\left[\left\langle v_i h_j\right\rangle_{data} - \left\langle v_i h_j\right\rangle_{model}\right] \quad 15.4.10$$

$\left\langle v_i h_j\right\rangle_{data}$ 是样本集中可见神经元和隐藏神经元激发的频率，$\left\langle v_i h_j\right\rangle_{model}$ 是对应的模型激发频率。当隐藏值未知时，需要计算指数次才能得到 $\langle\bullet\rangle_{model}$ 。所以常常会采用接近的算法。

常采用对比分歧（CD，contrastive divergence）快速算法来接近，可见隐藏权重梯度的一步对比分歧接近估计如下：

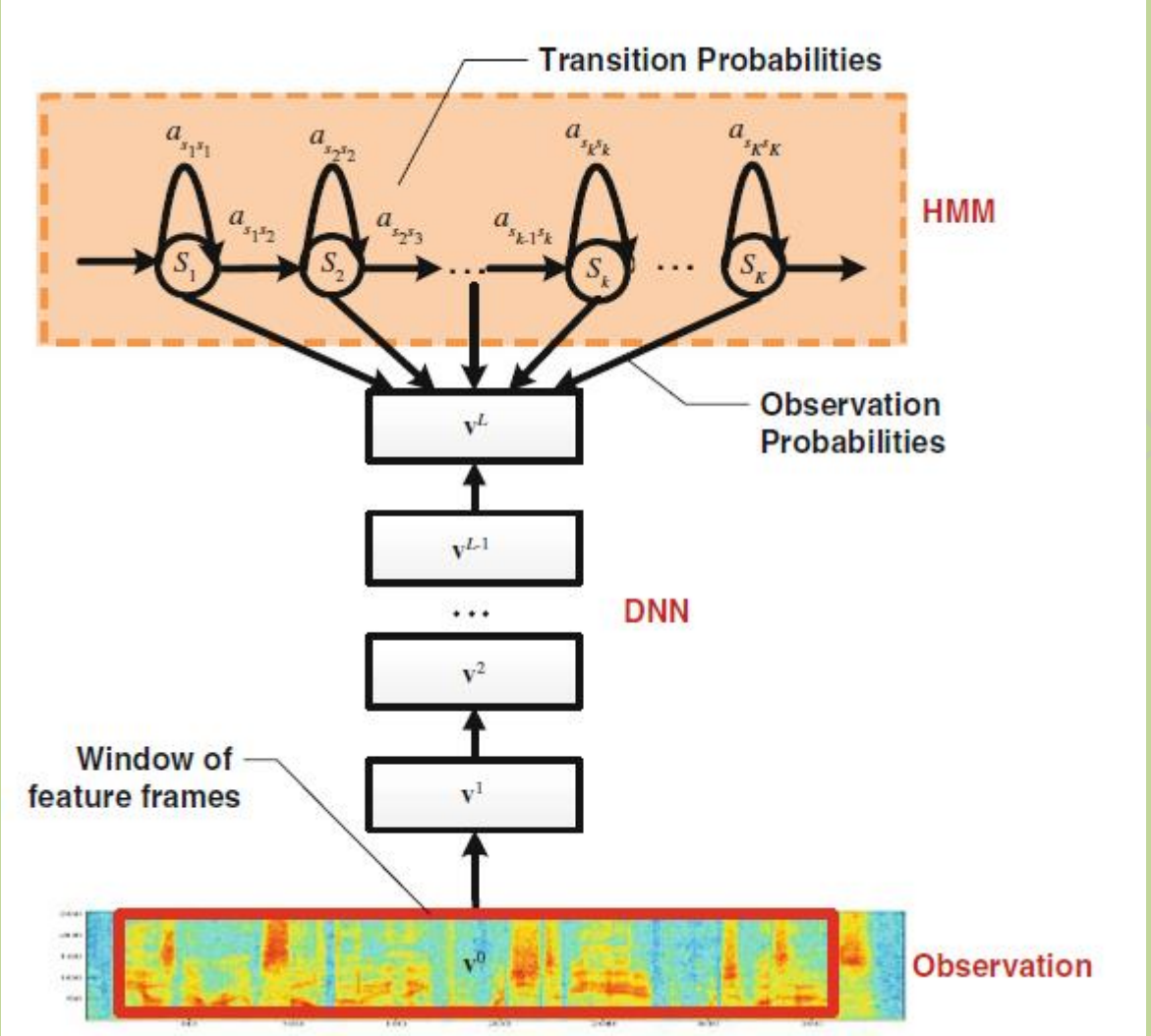
$$\nabla_{w_{ji}} J_{NLL}(\boldsymbol{W},\boldsymbol{a},\boldsymbol{b};\boldsymbol{v}) = -\left[\left\langle v_i h_j\right\rangle_{data} - \left\langle v_i h_j\right\rangle_{\infty}\right] \approx -\left[\left\langle v_i h_j\right\rangle_{data} - \left\langle v_i h_j\right\rangle_{1}\right] \quad 15.4.11$$

$\langle\bullet\rangle_\infty$ 和 $\langle\bullet\rangle_1$ 分别是运行无限步和一步吉布斯采样器的期望，

15.4.4

15.5 深度神经元和马尔科夫混合模型（DNN-HMM）

在 DNN-HMM 算法中，语音的力度变化由 HMM 模型来匹配，观测概率使用 DNN 方法来计算。这一过程如下图。



DNN-HMM 混合系统架构

DNN 的每个神经元输出

Deep Learning 理论

Deep learning 概念源于人造神经元网络研究。分为三类：

1 无监督或者生成学习，用于模式的分析和综合，找到观测数据的高阶相关性。

2 有监督学习，用于模式分类。

3 混合学习，

有监督的识别方法是后项传递神经网络（BPNN）。

NLP（native language processing）

TTS（text to speech）

AEC 算法：

AEC 算法主要包括如下几个重要的模块：

1 回声延迟估计

2 NLMS（normalize least mean square）最小均方误差

3 NLP（）非线性滤波

4 CNG 舒适噪声产生，一般经典 aec 算法还包括双端检测（DT）

回声延迟估计

回声延迟长短对回声抵消器的性能有较大影响，过长的的滤波器抽头会带来较大的延迟，并且语音信号是短时平稳信号，过长的滤波器抽头也不适合短时平稳信号特征。

基于相关的延迟算法。该算法的主要思想是：

设 1 表示有说话音，0 表无说话声（或者很弱的说话声），参考端（远端）信号 $x(t)$ 和接收端信号可能的组合方式如下：

(0, 0); (0, 1); (1, 0); (1, 1)

webrtc 默认(1, 0)和（0，1）是不可能发生的。设在时间间隔 p 上，即 $p=1,2,...,p$,频带 q, $q=1,2,...,Q$,输入信号 x 加窗后的功率谱用 $X_{w(p,q)}$ 表示，其角标表示其加了窗函数。对每个频带的功率谱设定一个门限 $X_{w(p,q)}$，

如果 $X_w(p,q)>=X_w(p,q)threshold$,则 $X_w(p,q)=1$；

如果 $X_w(p,q)$ < $X_w(p,q)threshold$,则 $X_w(p,q)=0$；

同理，对于信号 $y(t)$,加窗信号功率谱 $Y_W(p,q)$ 和门限 $Y_W(p,q)threshold$,

如果 $Y_w(p,q)$ >= $Y_w(p,q)threshold$, 则 $Y_w(p,q)$=1；

如果 $Y_w(p,q)$< $Y_w(p,q)threshold$, 则 $Y_w(p,q)$=0；考虑到实际处理的方便，在 webrtc 的 c 代码中，将经过 fft 变换后的频域功率谱分为 32 个子带，这样每个特定子带 $X_w(p,q)$ 的值可以用 1 个比特来表示，总共需要 32 个比特，只用一个 32 位数据类型就可以表示。

2）NLMS 归一化最小均方自适应算法

LMS/NLMS/RLS 都是经典自适应滤波算法，设远端信号为 $x(n)$,近端信号为 $d(n)$ ，则误差信号 $e(n)=d(n)-w(n)x(n)$,此处'表示转置，因为信号一般使用列向量表示的。NULMS 对滤波器的系数更新使用变步长方法，即步长 $u=u0/(gamma+x'(n)*x(n))$;其中 u0 为更新

步长因子，gamma 是稳定因子，则滤波器系数更新方程为$W(n+1)=W(n)+u*e(n)*x(n)$;

3)NLP(非线性滤波)

webrtc 采用了维纳滤波器，此处只给出传递函数的表达式，设估计的语音信号的功率谱为$P_s(\omega)$,噪声的功率谱为$P_n(\omega)$，则滤波器的传递函数为$H(\omega)=P_s(\omega)/(P_s(\omega)+P_n(\omega))$。

4）CNG（舒适噪声产生）

首先生成在[0, 1]上均与分布随机噪声矩阵，再用噪声的功率谱开方后去调制噪声的幅度。

fullaec 算法需要注意两点：

1）延迟要小，因为算法默认滤波器长度是分为 12 块，每块 64 点，按照 8000 采样率，也就是 12×8ms=96ms 的数据，超过这个长度就处理不了了。

2）延迟抖动要小，因为算法是默认 10 块也计算一次参考数据的位置（即滤波器能量最大的那一块），所以如果抖动很大的话，找参考数据不准确的，这样回声就消除不掉了。

PBFDAF（基于分段块频域自适应滤波算法的长延时回声消除, partitioned Multiple Block Frequency Domain Adaptive Filter）