对于滚动姿态的角加速度，则有

$$\ddot{\phi}_H(n) = u(n) + w(n) \tag{4.4.39}$$

式中 $u(n)$ 为 n 时刻系统喷气所产生的加速度，并且喷气时间和大小是已知的；$w(n)$ 是由外力干扰产生的加速度，假定它为零均值的平稳白噪声过程，且各时区之间互不相关，但具有某个已知的方差 $\mathrm{E}\{w^2(n)\} = \sigma_w^2$。

定义状态向量

$$\boldsymbol{x}(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{bmatrix} = \begin{bmatrix} \phi_H(n) \\ \dot{\phi}_H(n) \\ \ddot{\phi}_H(n) \end{bmatrix} \tag{4.4.40}$$

则式 (4.4.37) ~ 式 (4.4.39) 可以合并写作下面的状态方程：

$$\boldsymbol{x}(n) = \boldsymbol{F}\boldsymbol{x}(n-1) + \boldsymbol{u}(n) + \boldsymbol{w}(n) \tag{4.4.41}$$

式中，状态转移矩阵

$$\boldsymbol{F} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & 0 \end{bmatrix} \tag{4.4.42}$$

且系统输入向量为

$$\boldsymbol{u}(n) = [0, 0, u(n)]^{\mathrm{T}} \tag{4.4.43}$$

系统噪声向量为

$$\boldsymbol{w}(n) = [0, 0, w(n)]^{\mathrm{T}} \tag{4.4.44}$$

因此，系统噪声的相关矩阵可以计算为

$$\boldsymbol{Q}_w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix} \tag{4.4.45}$$

将状态方程 (4.4.41) 与 Kalman 滤波器的标准状态方程比较知，描述滚动姿态的状态转移矩阵与离散时间无关，并且多了一个已知的输入向量 $\boldsymbol{u}(n)$。

另一方面，观测方程可以写作

$$y(n) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}(n) + v(n) \tag{4.4.46}$$

式中，滚动姿态的角观测值 $y(n)$ 就是实际的滚动姿态角 $\phi_H(n)$，它等于 $\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}(n)$ 与一白噪声 $v(n)$ 的叠加，而观测矩阵 $\boldsymbol{C}(n)$ 简化为一行向量 $\boldsymbol{h}^{\mathrm{T}} = [1, 0, 0]$。令 $v(n)$ 的均值为零，方差 σ_v^2 为已知。

前面介绍的标准 Kalman 滤波器的基础是一步预测。与之不同，这里是一个纯滤波问题。

状态向量存在两种不同的估计子。第一种估计子 $\hat{\boldsymbol{x}}^{(1)}(n)$ 由状态方程式 (4.3.41) 给出：

$$\hat{\boldsymbol{x}}^{(1)}(n) = \boldsymbol{F}\hat{\boldsymbol{x}}(n-1) + \boldsymbol{u}(n) \tag{4.4.47}$$

再由观测方程 (4.4.46)，又可得到状态向量的第二种估计子 $\hat{\boldsymbol{x}}^{(2)}(n)$ 为

$$y(n) = \boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{x}}^{(2)}(n) \tag{4.4.48}$$

容易验证 $\hat{\boldsymbol{x}}^{(1)}(n)$ 和 $\hat{\boldsymbol{x}}^{(2)}(n)$ 相互独立。为了使它们给出一个统一的估计，可以采用它们的线性组合形式：

$$\hat{\boldsymbol{x}}(n) = [\boldsymbol{I} - \boldsymbol{A}(n)]\hat{\boldsymbol{x}}^{(1)}(n) + \boldsymbol{A}(n)\hat{\boldsymbol{x}}^{(2)}(n) \tag{4.4.49}$$

式中 $\boldsymbol{A}(n)$ 为一加权矩阵。也可采用另一种线性组合形式：$\hat{\boldsymbol{x}}(n) = \boldsymbol{A}(n)\hat{\boldsymbol{x}}^{(1)}(n) + [\boldsymbol{I} - \boldsymbol{A}(n)]\hat{\boldsymbol{x}}^{(2)}(n)$。根据最小均方误差准则，最优加权矩阵应使估计误差 $\boldsymbol{x}(n) - \hat{\boldsymbol{x}}(n)$ 的均方值最小。

为了确定最优加权矩阵 $\boldsymbol{A}(n)$，令

$$\boldsymbol{A}(n) = \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}} \tag{4.4.50}$$

将式 (4.4.50) 代入式 (4.4.49)，有

$$\begin{aligned}\hat{\boldsymbol{x}}(n) &= \left[\boldsymbol{I} - \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]\hat{\boldsymbol{x}}^{(1)}(n) + \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{x}}^{(2)}(n) & (4.4.51\text{a})\\ &= \left[\boldsymbol{I} - \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]\hat{\boldsymbol{x}}^{(1)}(n) + \boldsymbol{m}(n)y(n) & (4.4.51\text{b})\end{aligned}$$

另一方面，容易验证恒等式

$$\boldsymbol{x}(n) \equiv \left[\boldsymbol{I} - \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]\boldsymbol{x}(n) + \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}(n) \tag{4.4.52}$$

式 (4.4.52) 减去式 (4.4.51b)，得到信号的估计误差向量为

$$\begin{aligned}\boldsymbol{x}(n) - \hat{\boldsymbol{x}}(n) &= \left[\boldsymbol{I} - \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right][\boldsymbol{x}(n) - \hat{\boldsymbol{x}}^{(1)}(n)] + \boldsymbol{m}(n)[\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}(n) - y(n)]\\ &= \left[\boldsymbol{I} - \boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right][\boldsymbol{x}(n) - \hat{\boldsymbol{x}}^{(1)}(n)] + \boldsymbol{m}(n)v(n) \qquad (4.4.53)\end{aligned}$$

考虑估计误差向量的相关矩阵

$$
\begin{aligned}
\boldsymbol{P}(n) =& \mathrm{E}\left\{[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}(n)][\boldsymbol{x}(n)-\hat{\boldsymbol{x}}(n)]^{\mathrm{T}}\right\} \\
=& \mathrm{E}\left\{\left[\left(\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right)\left(\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right)-\boldsymbol{m}(n)v(n)\right]\times\right. \\
& \left.\left[\left(\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right)\left(\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right)-\boldsymbol{m}(n)v(n)\right]^{\mathrm{T}}\right\} \\
=& \left[\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]\mathrm{E}\left\{\left[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right]\left[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right]^{\mathrm{T}}\right\}\times \\
& \left[\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]^{\mathrm{T}}+\boldsymbol{m}(n)\sigma_v^2\boldsymbol{m}^{\mathrm{T}}(n) \\
=& \left[\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]\boldsymbol{P}_1(n)\left[\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]^{\mathrm{T}}+\boldsymbol{m}(n)\sigma_v^2\boldsymbol{m}^{\mathrm{T}}(n)
\end{aligned}
\tag{4.4.54}
$$

式中

$$
\boldsymbol{P}_1(n)=\mathrm{E}\left\{\left[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right]\left[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right]^{\mathrm{T}}\right\}
$$

表示第一个估计子的误差方差矩阵，而 σ_v^2 是第二种估计子的误差方差。

由于使均方误差最小等价于使误差方差矩阵 $\boldsymbol{P}$ 的迹 $\mathrm{tr}[\boldsymbol{P}(n)]$ 最小，对其求导数得

$$
\frac{\partial \mathrm{tr}[\boldsymbol{P}(n)]}{\partial \boldsymbol{m}(n)}=-2\boldsymbol{P}_1(n)\boldsymbol{h}+2\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\boldsymbol{P}_1(n)\boldsymbol{h}+2\boldsymbol{m}(n)\sigma_v^2 \tag{4.4.55}
$$

令其为零，则

$$
\boldsymbol{m}(n)=\boldsymbol{P}_1(n)\boldsymbol{h}\left[\boldsymbol{h}^{\mathrm{T}}\boldsymbol{P}_1(n)\boldsymbol{h}+\sigma_v^2\right]^{-1} \tag{4.4.56}
$$

将式 (4.4.56) 代入式 (4.4.54)，并加以整理，有下面的简洁结果：

$$
\boldsymbol{P}(n)=[\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}]\boldsymbol{P}_1(n) \tag{4.4.57}
$$

下面考虑 $\boldsymbol{P}_1(n)$ 的计算。根据定义，有

$$
\begin{aligned}
\boldsymbol{P}_1(n) =& \mathrm{E}\left\{\left[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right]\left[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}^{(1)}(n)\right]^{\mathrm{T}}\right\} \\
=& \mathrm{E}\left\{\left[\boldsymbol{F}\left(\boldsymbol{x}(n-1)-\hat{\boldsymbol{x}}^{(1)}(n-1)\right)+\boldsymbol{w}(n)\right]\times\right. \\
& \left.\left[\boldsymbol{F}\left(\boldsymbol{x}(n-1)-\hat{\boldsymbol{x}}^{(1)}(n-1)\right)+\boldsymbol{w}(n)\right]^{\mathrm{T}}\right\}
\end{aligned}
$$

由于 $\boldsymbol{F}$ 为常数矩阵，故上式可写作

$$\begin{aligned}\boldsymbol{P}_1(n) =& \boldsymbol{F}\mathrm{E}\left\{\left[\boldsymbol{x}(n-1)-\hat{\boldsymbol{x}}^{(1)}(n-1)\right]\left[\boldsymbol{x}(n-1)-\hat{\boldsymbol{x}}^{(1)}(n-1)\right]^{\mathrm{T}}\right\}\boldsymbol{F}^{\mathrm{T}}+\\ &\boldsymbol{F}\mathrm{E}\left\{\left[\boldsymbol{x}(n-1)-\hat{\boldsymbol{x}}^{(1)}(n-1)\right]\boldsymbol{w}^{\mathrm{T}}(n)\right\}+\\ &\mathrm{E}\left\{\boldsymbol{w}(n)\left[\boldsymbol{x}(n-1)-\hat{\boldsymbol{x}}^{(1)}(n-1)\right]^{\mathrm{T}}\right\}\boldsymbol{F}^{\mathrm{T}}+\mathrm{E}\left\{\boldsymbol{w}(n)\boldsymbol{w}^{\mathrm{T}}(n)\right\}\\ =&\boldsymbol{F}\boldsymbol{P}_1(n-1)\boldsymbol{F}^{\mathrm{T}}+\boldsymbol{Q}_w \end{aligned} \tag{4.4.58}$$

综合以上分析，求姿态角速度最小均方误差估计的 Kalman 滤波算法可以描述如下。

算法 4.4.2 (姿态角速度估计的 Kalman 滤波算法)

初始条件： $\hat{\boldsymbol{x}}(1)=0, \boldsymbol{P}=\boldsymbol{I}$

输入：

滚动姿态角观测值 $y(n)$

已知参数：

系统噪声 $w(n)$ 的方差 σ_w^2

观测噪声 $v(n)$ 的方差 σ_v^2

观测向量 $\boldsymbol{h}=[1,0,0]^{\mathrm{T}}$

状态转移矩阵 $\boldsymbol{F}$，如式 (4.4.42) 所示

对 $n=1,2,\cdots$ 计算

$\boldsymbol{P}_1(n)=\boldsymbol{F}\boldsymbol{P}(n-1)\boldsymbol{F}^{\mathrm{T}}+\boldsymbol{Q}_v$

$\boldsymbol{m}(n)=\boldsymbol{P}_1(n)\boldsymbol{h}\left[\boldsymbol{h}^{\mathrm{T}}\boldsymbol{P}_1(n)\boldsymbol{h}+\sigma_v^2\right]^{-1}$

$\boldsymbol{P}(n)=[\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}]\boldsymbol{P}_1(n)$

$\hat{\boldsymbol{x}}^{(1)}(n)=\boldsymbol{F}\hat{\boldsymbol{x}}(n-1)+\boldsymbol{u}(n)$

$\hat{\boldsymbol{x}}(n)=\left[\boldsymbol{I}-\boldsymbol{m}(n)\boldsymbol{h}^{\mathrm{T}}\right]\hat{\boldsymbol{x}}^{(1)}(n)+\boldsymbol{m}(n)y(n)$

用以上算法得到的状态估计向量 $\hat{\boldsymbol{x}}(n)$ 的第二个分量即为滚动状态角速度的估计 $\hat{\dot{\phi}}_H(n)$。

4.5 LMS 类自适应算法

考虑图 4.3.2 所示 FIR 滤波器的自适应实现。所谓自适应实现乃是指：M 阶 FIR 滤波器的抽头权系数 $w_0,\cdots,w_{M-1}$ 可以根据估计误差 $e(n)$ 的大小自动调节，使得某个代价函数最小。图 4.5.1 画出了自适应 FIR 滤波器的原理图。

滤波器设计最常用的准则是使滤波器实际输出 $y(n)=\boldsymbol{u}^{\mathrm{H}}(n)\boldsymbol{w}^*=\boldsymbol{w}^{\mathrm{H}}\boldsymbol{u}(n)$ 与期望响应 $d(n)$ 之间的均方误差 $\mathrm{E}\{|\epsilon(n)|^2\}$ 为最小，这就是最小均方误差 (MMSE) 准则。

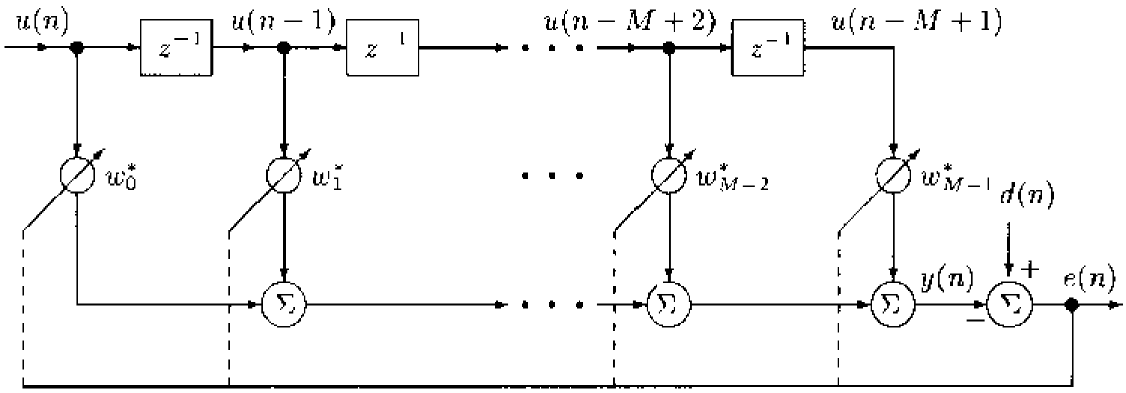

图 4.5.1 自适应 FIR 滤波器

令

$$\epsilon(n)=d(n)-\boldsymbol{w}^{\mathrm{H}}\boldsymbol{u}(n) \tag{4.5.1}$$

表示滤波器在 n 时刻的估计误差，并定义均方误差

$$J(n)\stackrel{\mathrm{def}}{=}\mathrm{E}\{|\epsilon(n)|^2\}=\mathrm{E}\left\{\left|d(n)-\boldsymbol{w}^{\mathrm{H}}\boldsymbol{u}(n)\right|^2\right\} \tag{4.5.2}$$

为代价函数。

由式 (4.3.10) 知，代价函数相对于滤波器抽头权向量 $\boldsymbol{w}$ 的梯度为

$$\begin{aligned}\nabla_k J(n)&=-2\mathrm{E}\{u(n-k)e^*(n)\}\\&=-2\mathrm{E}\left\{u(n-k)\left[d(n)-\boldsymbol{w}^{\mathrm{H}}\boldsymbol{u}(n)\right]^*\right\},\quad k=0,1,\cdots,M-1\end{aligned} \tag{4.5.3}$$

令 $w_i=a_i+\mathrm{j}b_i, i=0,1,\cdots,M-1$，并定义梯度向量

$$\boldsymbol{\nabla}J(n)\stackrel{\mathrm{def}}{=}[\nabla_0 J(n),\nabla_1 J(n),\cdots,\nabla_{M-1}(n)]^{\mathrm{T}} \tag{4.5.4}$$

$$=\begin{bmatrix}\dfrac{\partial J(n)}{\partial a_0(n)}+\mathrm{j}\dfrac{\partial J(n)}{\partial b_0(n)}\\\dfrac{\partial J(n)}{\partial a_1(n)}+\mathrm{j}\dfrac{\partial J(n)}{\partial b_1(n)}\\\vdots\\\dfrac{\partial J(n)}{\partial a_{M-1}(n)}+\mathrm{j}\dfrac{\partial J(n)}{\partial b_{M-1}(n)}\end{bmatrix} \tag{4.5.5}$$

以及输入向量和抽头权向量:

$$\boldsymbol{u}(n)=[u(n),u(n-1),\cdots,u(n-M+1)]^{\mathrm{T}} \tag{4.5.6}$$

$$\boldsymbol{w}(n)=[w_0(n),w_1(n),\cdots,w_{M-1}(n)]^{\mathrm{T}} \tag{4.5.7}$$

则式 (4.5.5) 可以写作以下向量形式:

$$\nabla J(n)=-2\mathrm{E}\left\{\boldsymbol{u}(n)\left[d^*(n)-\boldsymbol{u}^{\mathrm{H}}(n)\boldsymbol{w}(n)\right]\right\} \tag{4.5.8a}$$

$$=-2\boldsymbol{r}+2\boldsymbol{R}\boldsymbol{w}(n) \tag{4.5.8b}$$

式中

$$\boldsymbol{R}=\mathrm{E}\{\boldsymbol{u}(n)\boldsymbol{u}^{\mathrm{H}}(n)\} \tag{4.5.9a}$$

$$\boldsymbol{r}=\mathrm{E}\{\boldsymbol{u}(n)d^*(n)\} \tag{4.5.9b}$$

最广泛使用的自适应算法形式为“下降算法”:

$$\boldsymbol{w}(n)=\boldsymbol{w}(n-1)+\mu(n)\boldsymbol{v}(n) \tag{4.5.10}$$

式中 $\boldsymbol{w}(n)$ 为第 n 步迭代 (也即时刻 n) 的权向量, $\mu(n)$ 为第 n 次迭代的更新步长, 而 $\boldsymbol{v}(n)$ 为第 n 次迭代的更新方向 (向量)。

下降算法 (4.5.10) 有两种主要实现方式。一种是自适应梯度算法, 另一种是自适应高斯–牛顿算法。自适应梯度算法包括 LMS 算法及其各种变型和改进 (统称 LMS 类自适应算法), 自适应高斯–牛顿算法则包括 RLS 算法及其变型和改进。本节介绍 LMS 类自适应算法, 下一节介绍后一类自适应算法。

4.5.1 LMS 算法及其基本变型

最常用的下降算法为梯度下降法, 常称最陡下降法。在这种算法里, 更新方向向量 $\boldsymbol{v}(n)$ 取作第 $n-1$ 次迭代的代价函数 $J[\boldsymbol{w}(n-1)]$ 的负梯度, 即最陡下降法 (也称梯度算法) 的统一形式为

$$\boldsymbol{w}(n)=\boldsymbol{w}(n-1)-\frac{1}{2}\mu(n)\nabla J(n-1) \tag{4.5.11}$$

系数 1/2 是为了使得到的更新公式更简单。

将式 (4.5.8b) 代入式 (4.5.11), 即可得到抽头权向量 $\boldsymbol{w}(n)$ 的更新公式为

$$\boldsymbol{w}(n)=\boldsymbol{w}(n-1)+\mu(n)[\boldsymbol{r}-\boldsymbol{R}\boldsymbol{w}(n-1)],\quad n=1,2,\cdots \tag{4.5.12}$$

更新公式 (4.5.12) 表明:

(1) $[\boldsymbol{r}-\boldsymbol{R}\boldsymbol{w}(n-1)]$ 为误差向量，它代表了 $\boldsymbol{w}(n)$ 每步的校正量；

(2) 参数 $\mu(n)$ 与校正量相乘，它是控制 $\boldsymbol{w}(n)$ 每步的实际校正量的参数，因此 $\mu(n)$ 称为在时间 n 的"步长参数"。这一参数决定了更新算法 (4.5.12) 的收敛速度。

(3) 当自适应算法趋于收敛时，则有 $\boldsymbol{r}-\boldsymbol{R}\boldsymbol{w}(n-1)\rightarrow 0$ (若 $n\rightarrow\infty$)，即有

$$\lim_{n\to\infty}\boldsymbol{w}(n-1)=\boldsymbol{R}^{-1}\boldsymbol{r}$$

即抽头权向量收敛为式 (4.3.25) 所示的 Wiener 滤波器。

当式 (4.5.8a) 中的数学期望项 $\mathrm{E}\{\boldsymbol{u}(n)d^*(n)\}$ 和 $\mathrm{E}\{\boldsymbol{u}(n)\boldsymbol{u}^{\mathrm{H}}(n)\}$ 分别用它们各自的瞬时值 $\boldsymbol{u}(n)d^*(n)$ 和 $\boldsymbol{u}(n)\boldsymbol{u}^{\mathrm{H}}(n)$ 代替时，便得到真实梯度向量的估计值

$$\hat{\boldsymbol{\nabla}}J(n)=-2[\boldsymbol{u}(n)d^*(n)-\boldsymbol{u}(n)\boldsymbol{u}^{\mathrm{H}}(n)\boldsymbol{w}(n)] \tag{4.5.13}$$

习惯称之为瞬时梯度。

若梯度算法 (4.5.11) 中的真实梯度向量 $\boldsymbol{\nabla}J(n-1)$ 用瞬时梯度向量 $\hat{\boldsymbol{\nabla}}J(n-1)$ 代替后，即得到瞬时梯度算法如下：

$$\begin{aligned}\boldsymbol{w}(n)&=\boldsymbol{w}(n-1)+\mu(n)\boldsymbol{u}(n)[d(n)-\boldsymbol{u}^{\mathrm{T}}(n)\boldsymbol{w}^*(n-1)]^*\\&=\boldsymbol{w}(n-1)+\mu(n)e^*(n)\boldsymbol{u}(n)\end{aligned} \tag{4.5.14a}$$

式中

$$e(n)=d(n)-\boldsymbol{u}^{\mathrm{T}}(n)\boldsymbol{w}^*(n-1)=d(n)-\boldsymbol{w}^{\mathrm{H}}(n-1)\boldsymbol{u}(n) \tag{4.5.14b}$$

注意，虽然 $e(n)$ 与式 (4.5.3) 定义的 $\epsilon(n)$ 都是表示滤波器在 n 时刻的估计误差，但它们是不同的：$e(n)$ 由 $\boldsymbol{w}(n-1)$ 决定，而 $\epsilon(n)$ 则由 $\boldsymbol{w}(n)$ 决定。为区别计，常称 $e(n)$ 为先验估计误差，$\epsilon(n)$ 为后验估计误差。

式 (4.5.14) 所示的算法就是著名的最小均方误差自适应算法，简称 LMS 算法，它是 Widrow 在 20 世纪 60 年代初提出的 [206]。

容易验证，瞬时梯度向量是真实梯度向量的无偏估计：

$$\begin{aligned}\mathrm{E}\{\hat{\boldsymbol{\nabla}}J(n)\}&=-2\mathrm{E}\{\boldsymbol{u}(n)[d^*(n)-\boldsymbol{u}^{\mathrm{H}}(n)\boldsymbol{w}(n-1)]\}\\&=-2[\boldsymbol{r}-\boldsymbol{R}\boldsymbol{w}(n-1)]\\&=\boldsymbol{\nabla}J(n)\end{aligned} \tag{4.5.15}$$

为了方便读者使用，这里将 LMS 自适应算法以及它的几种基本变型归纳如下。

算法 4.5.1 (LMS 自适应算法及其基本变型)

步骤 1 初始化： $w(0)=0$;

步骤 2 更新： $n=1,2,\cdots$

$$e(n)=d(n)-\boldsymbol{w}^{\mathrm{H}}(n-1)\boldsymbol{u}(n)$$

$$\boldsymbol{w}(n)=\boldsymbol{w}(n-1)+\mu(n)\boldsymbol{u}(n)e^{*}(n)$$

下面是关于 LMS 算法的几点注释。

注释 1: 若取 $\mu(n)=$ 常数，则称之为基本 LMS 算法。

注释 2: 若取 $\mu(n)=\dfrac{\alpha}{\beta+\boldsymbol{u}^{\mathrm{H}}(n)\boldsymbol{u}(n)}$，其中 $\alpha\in(0,2)$, $\beta>0$, 则得到归一化 LMS 算法。

注释 3: 在功率归一化 LMS 算法中，取 $\mu(n)=\dfrac{\alpha}{\sigma_u^2(n)}$，其中 σ_u^2 表示 $u(n)$ 的方差，可由 $\sigma_u^2(n)=\lambda\sigma_u^2(n-1)+e^2(n)$ 递推计算，这里 $\lambda\in(0,1]$ 为遗忘因子，由 $0<\alpha<\dfrac{2}{M}$ 确定，而 M 是滤波器的阶数。

注释 4: 当期望信号未知时，步骤 2 中的 $d(n)$ 可直接用滤波器的实际输出 $y(n)$ 代替。

4.5.2 解相关 LMS 算法

在 LMS 算法中，有一个独立性假设：假定横向滤波器的输入向量 $\boldsymbol{u}(1),\boldsymbol{u}(2),\cdots,\boldsymbol{u}(n)$ 是彼此统计独立的向量序列。当它们之间不满足统计独立的条件时，基本 LMS 算法的性能将下降，尤其是收敛速度会比较慢。因此，在这种情况下，就需要解除各时刻输入向量之间的相关 (这一操作称为 “解相关”)，使它们尽可能保持统计独立。大量的研究表明 ([78] 及其有关参考文献)，解相关能够有效加快 LMS 算法的收敛速率。

1. 时域解相关 LMS 算法

定义 $\boldsymbol{u}(n)$ 与 $\boldsymbol{u}(n-1)$ 在 n 时刻的相关系数为

$$a(n)\overset{\text{def}}{=}\frac{\boldsymbol{u}^{\mathrm{H}}(n-1)\boldsymbol{u}(n)}{\boldsymbol{u}^{\mathrm{H}}(n-1)\boldsymbol{u}(n-1)}\tag{4.5.16}$$

根据定义，若 $a(n)=1$, 则称 $\boldsymbol{u}(n)$ 是 $\boldsymbol{u}(n-1)$ 的相干信号；若 $a(n)=0$, 则 $\boldsymbol{u}(n)$ 与 $\boldsymbol{u}(n-1)$ 不相关；当 $0<a(n)<1$ 时，称 $\boldsymbol{u}(n)$ 与 $\boldsymbol{u}(n-1)$ 相关，并且 $a(n)$ 越大，它们之间的相关性越强。

显然，$a(n)\boldsymbol{u}(n-1)$ 代表了 $\boldsymbol{u}(n)$ 中与 $\boldsymbol{u}(n-1)$ 相关的部分。若从 $\boldsymbol{u}(n)$ 中减去该部分，则这一减法运算相当于 “解相关”。现在，用解相关的结果作为更新方向向

量 $v(n)$:

$$v(n) = u(n) - a(n)u(n-1) \tag{4.5.17}$$

从这一角度出发，把 $a(n)$ 称为解相关系数更为贴切。

另一方面，步长参数 $\mu(n)$ 应该是满足下列最小化问题的解：

$$\mu(n) = \arg\min_{\mu} J[w(n-1) + \mu v(n)] \tag{4.5.18}$$

由此得

$$\mu(n) = \frac{e(n)}{u^{\mathrm{H}}(n)v(n)} \tag{4.5.19}$$

综合以上结果，可以得到解相关 LMS 算法如下，它是 Doherty 与 Porayath 于 1997 年提出的 [63]。

算法 4.5.2 (解相关 LMS 算法)

步骤 1 初始化： $w(0) = 0$;

步骤 2 更新： $n = 1, 2, \cdots$

$$e(n) = d(n) - w^{\mathrm{H}}(n-1)u(n)$$

$$a(n) = \frac{u^{\mathrm{H}}(n)u(n-1)}{u^{\mathrm{H}}(n-1)u(n-1)}$$

$$v(n) = u(n) - a(n)u(n-1)$$

$$\mu(n) = \frac{\rho e(n)}{u^{\mathrm{H}}(n)v(n)}$$

$$w(n) = w(n-1) + \mu(n)v(n)$$

上述算法中，参数 ρ 称为修整因子 (trimming factor)。

解相关 LMS 算法可视为一自适应辅助变量法，其中辅助变量由 $v(n) = u(n) - a(n)u(n-1)$ 给出。粗略地讲，辅助变量的选择原则是：它应该与滞后的输入和输出强相关，而与干扰不相关。对辅助变量方法及其自适应算法感兴趣的读者可参考文献 [229]。

更进一步地，上述算法中的辅助变量可以使用一前向预测器的误差向量代替。令 $a(n)$ 为一 M 阶前向预测器的权向量，计算前向预测误差：

$$e^{\mathrm{f}}(n) = u(n) + \sum_{i=1}^{M} a_i(n)u(n-i) = u(n) + a^{\mathrm{H}}(n)u(n-1) \tag{4.5.20}$$

式中

$$\boldsymbol{u}(n-1)=[u(n-1),u(n-2),\cdots,u(n-M)]^{\mathrm{T}} \tag{4.5.21a}$$

$$\boldsymbol{a}(n)=[a_1(n),a_2(n),\cdots,a_M(n)]^{\mathrm{T}} \tag{4.5.21b}$$

用前向预测误差向量作辅助变量即更新方向向量：

$$\boldsymbol{v}(n)=\boldsymbol{e}^{\mathrm{f}}(n)=[e^{\mathrm{f}}(n),e^{\mathrm{f}}(n-1),\cdots,e^{\mathrm{f}}(n-M+1)]^{\mathrm{T}} \tag{4.5.22}$$

用前向预测器对瞬时估计误差 $e(n)=y(n)-\boldsymbol{w}^{\mathrm{H}}(n-1)\boldsymbol{u}(n)$ 滤波，则得到滤波型 LMS 算法如下 [129]。

算法 4.5.3 (滤波型 LMS 算法)

步骤 1 初始化： $\boldsymbol{w}(0)=\boldsymbol{0}$;

步骤 2 更新： $n=1,2,\cdots$

给定一前向预测器 $\boldsymbol{a}(n)$ 的估计

$e(n)=d(n)-\boldsymbol{w}^{\mathrm{H}}(n-1)\boldsymbol{u}(n)$

$\boldsymbol{e}(n)=[e(n),e(n-1),\cdots,e(n-M+1)]^{\mathrm{T}}$

$e^{\mathrm{f}}(n)=u(n)+\boldsymbol{a}^{\mathrm{T}}(n)\boldsymbol{u}(n-1)$

$\boldsymbol{e}^{\mathrm{f}}(n)=[e^{\mathrm{f}}(n),e^{\mathrm{f}}(n-1),\cdots,e^{\mathrm{f}}(n-M+1)]^{\mathrm{T}}$

$\tilde{e}(n)=e(n)+\boldsymbol{a}^{\mathrm{H}}(n)\boldsymbol{e}(n)$ (滤波)

$\boldsymbol{w}(n)=\boldsymbol{w}(n-1)+\mu\boldsymbol{e}^{\mathrm{f}}(n)\tilde{e}(n)$

2. 变换域解相关 LMS 算法

改进 LMS 算法性能的早期工作是对输入数据向量 $\boldsymbol{u}(n)$ 使用酉变换。对某些类型的输入信号，使用酉变换的算法可以提高收敛速率，而计算复杂度却与 LMS 算法类似。这类算法以及它们的变型统称变换域自适应滤波算法 [61],[137],[112],[134],[175],[16]。

酉变换可以使用离散 Fourier 变换 (DFT)、离散余弦变换 (DCT) 和离散 Hartley 变换 (DHT)，它们都可以有效地提高 LMS 算法的收敛速率。

令 $\boldsymbol{S}$ 是一 $M\times M$ 酉变换矩阵，即

$$\boldsymbol{S}\boldsymbol{S}^{\mathrm{H}}=\beta\boldsymbol{I} \tag{4.5.23}$$

式中 $\beta>0$ 为一固定标量。