信号与图像处理基础

Adaptive Filter

中国科学技术大学 自动化系
曹 洋

主要内容

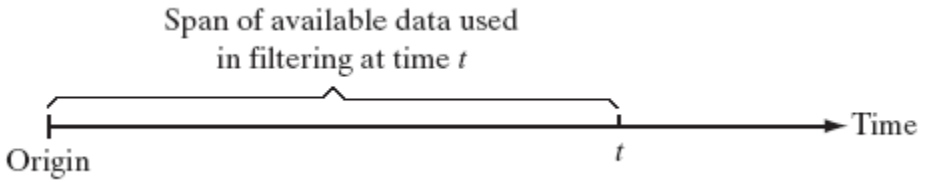
- 综述
- 维纳滤波
- 最小二乘滤波

4.1 自适应滤波综述

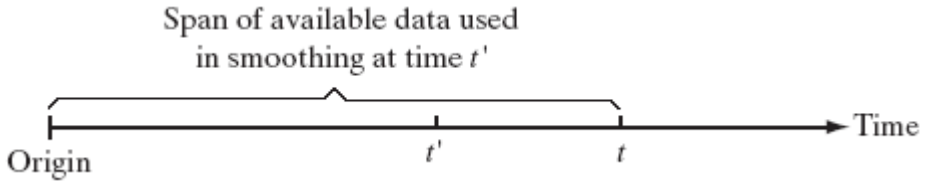
滤波与滤波器

在信号与图像处理领域，滤波器主要有三种应用

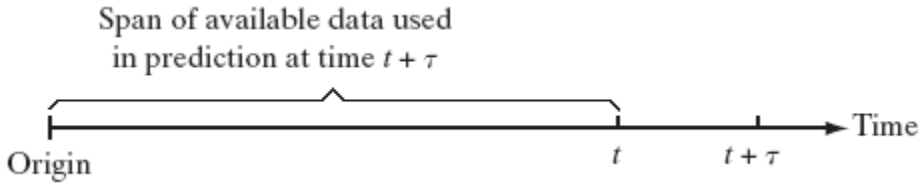
滤波去噪



平滑内插



预测估计



自适应滤波

- 信号或者噪声的特性并不是一成不变的，为了取得更好的滤波效果，滤波器的参数应该也随之调整。

- 自适应滤波器包含有一种参数自适应调整的机制，即能够对处理的信号进行监控，据此调整滤波器的参数。

- 自适应滤波器还包含有一种最优机制，即根据实际获取的信号，对滤波器结构和参数进行设计，以取得最优的滤波效果。

自适应滤波

- 需要给定一个优化目标
  - 需要关于信号或者噪声的先验信息
- 需要建立一种自调整机制
  - 通常是基于递归模型的，当缺少完整的先验信息时，递归模型是一种有效的手段。

自适应滤波

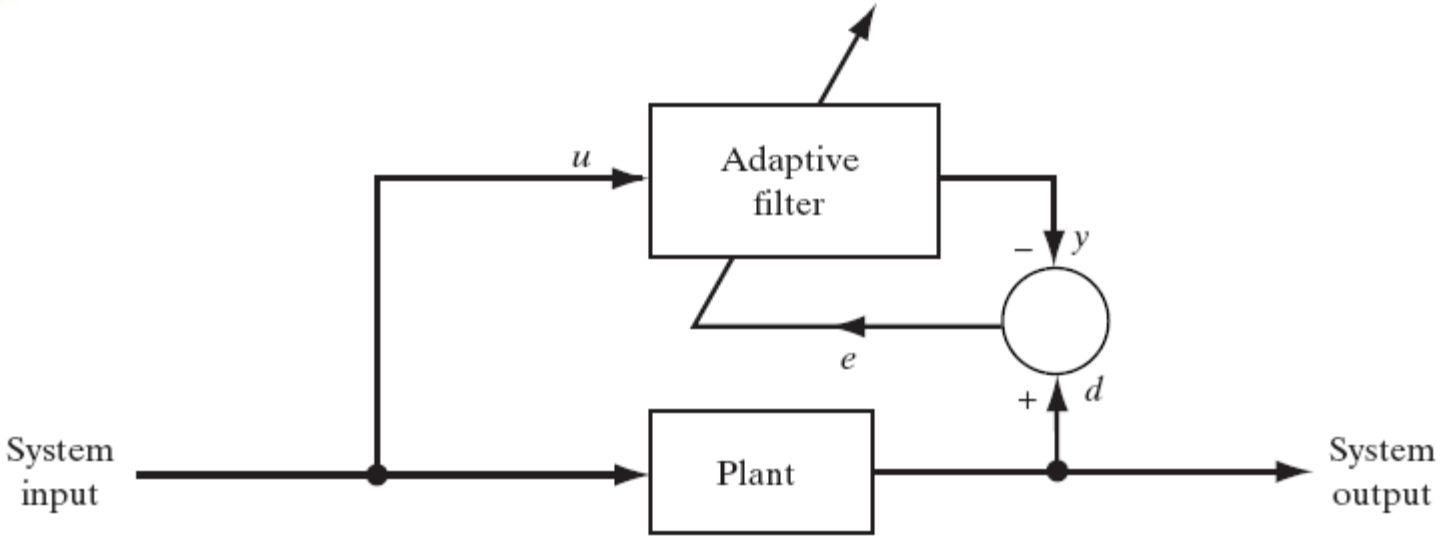
- 自适应滤波通常包含了两个阶段：

1. 滤波过程：根据输入信号产生预定的输出信号.
2. 自调整过程： 根据外界条件的变化调整滤波器的结构或者参数。

通常会采用信号的均方误差作为目标函数来设计滤波器

自适应滤波的应用：系统辨识

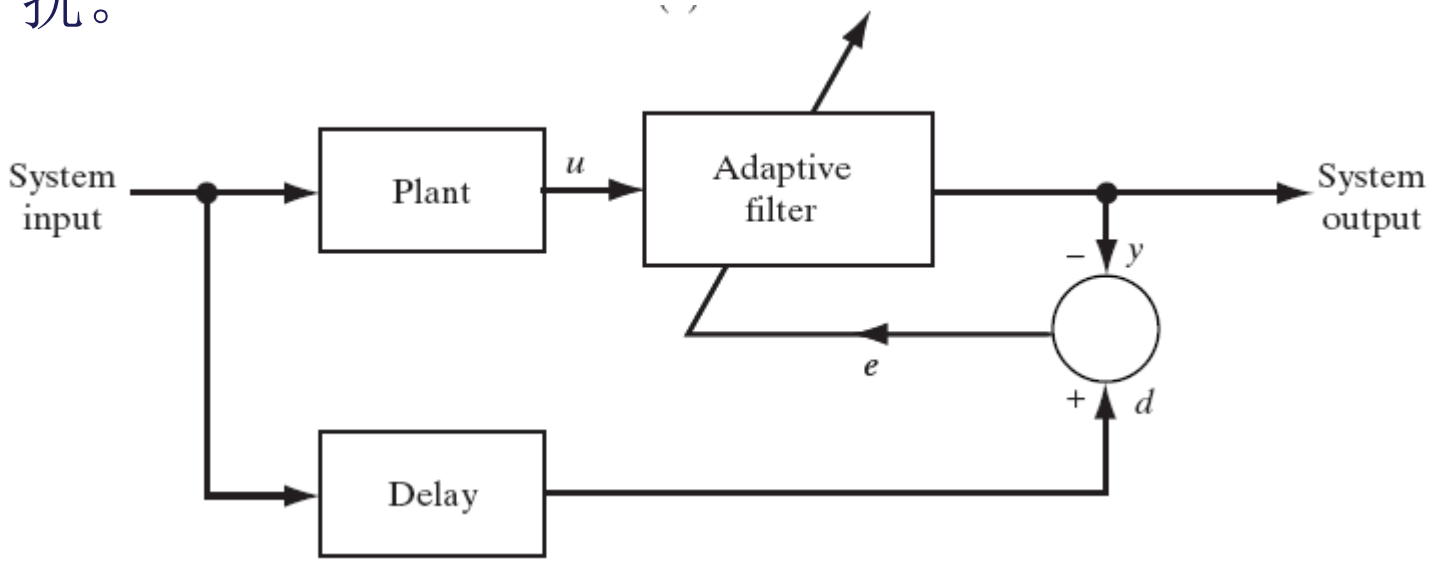
- 用于建立一个未知对象的系统模型



- Parameters
  - u=input of adaptive filter=input to plant
  - y=output of adaptive filter
  - d=desired response=output of plant
  - e=d-y=estimation error

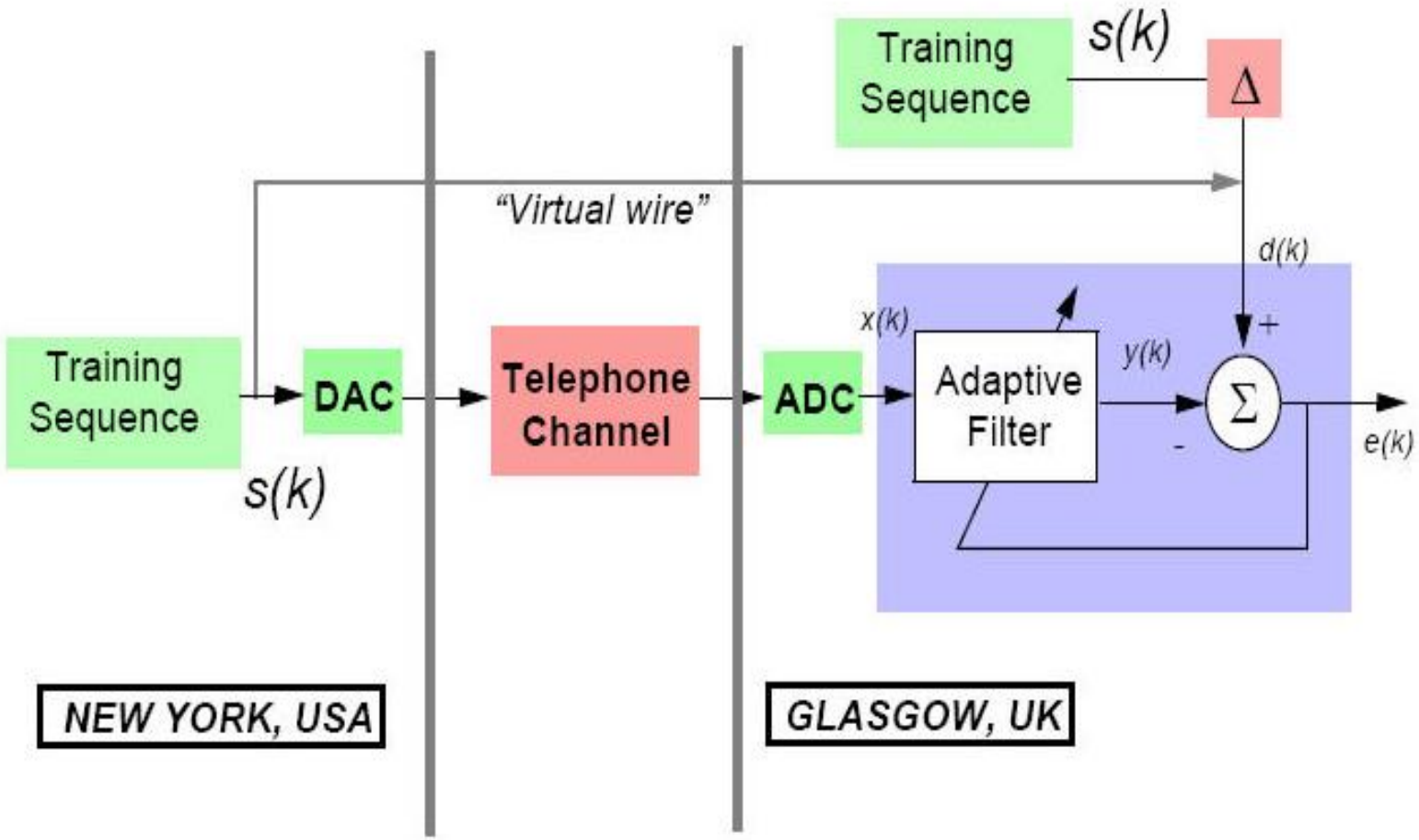
自适应滤波的应用：信道均衡

- 信道均衡是指在接收端的均衡器产生与信道特性相反的特性，用来减小或消除因信道的时变多径传播特性引起的码间干扰。



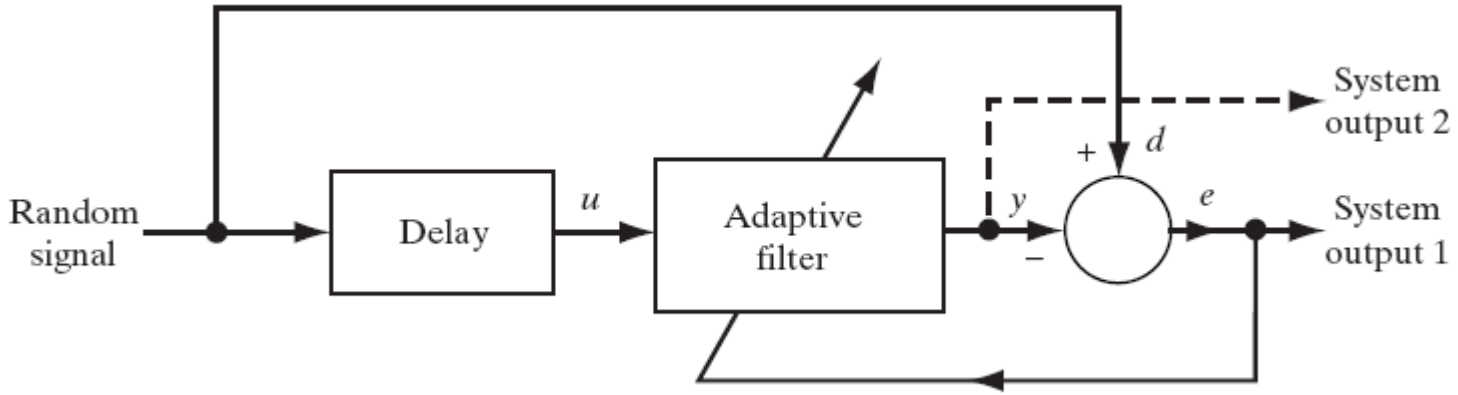
- Parameters
  - u=input of adaptive filter=output to plant； y=output of adaptive filter；d=desired response=delayed system input； e=d-y=estimation error

自适应滤波的应用：信道均衡



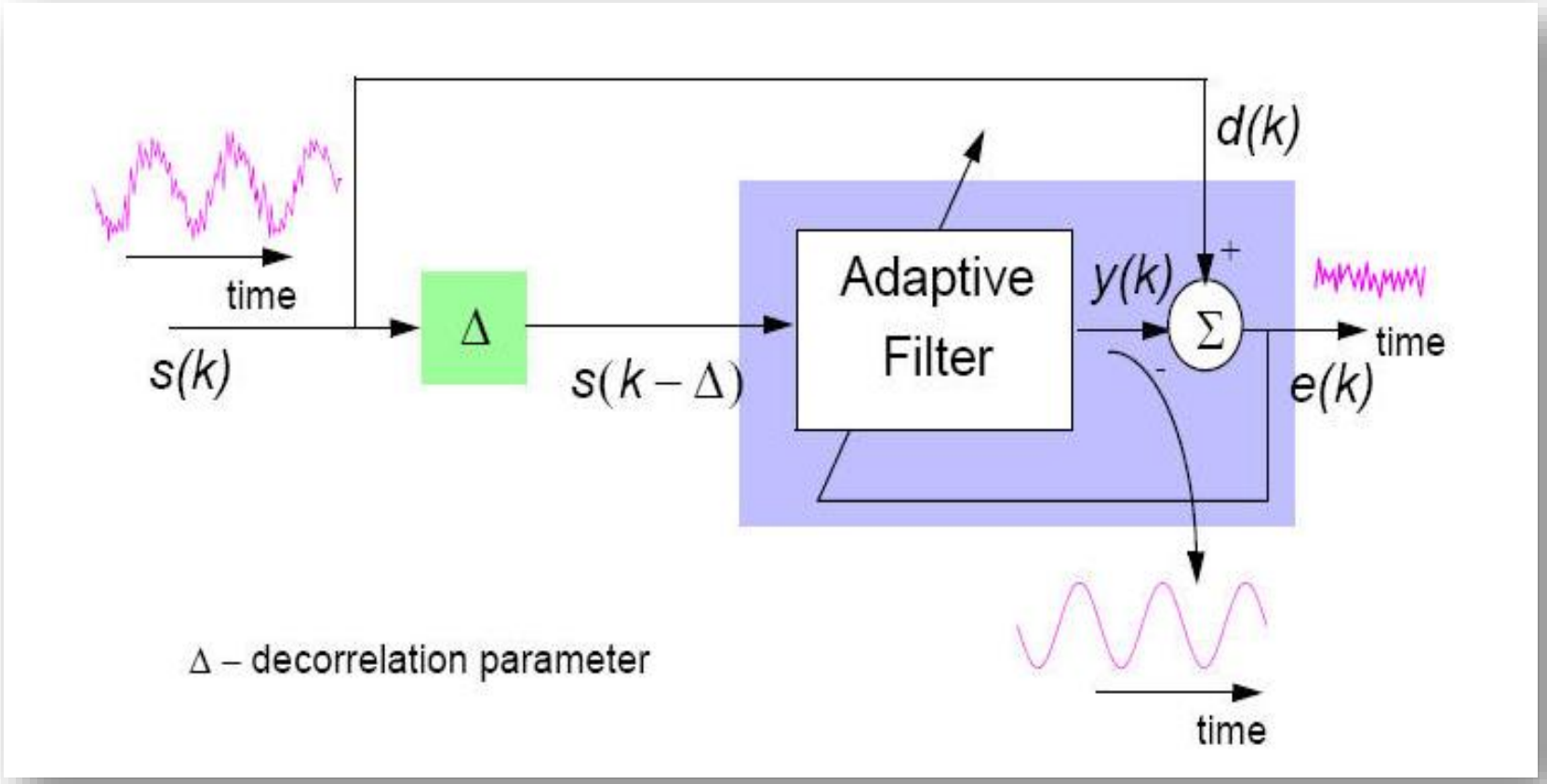
自适应滤波的应用：预测估计

- 基于当前随机信号预测估计未来的信号输出



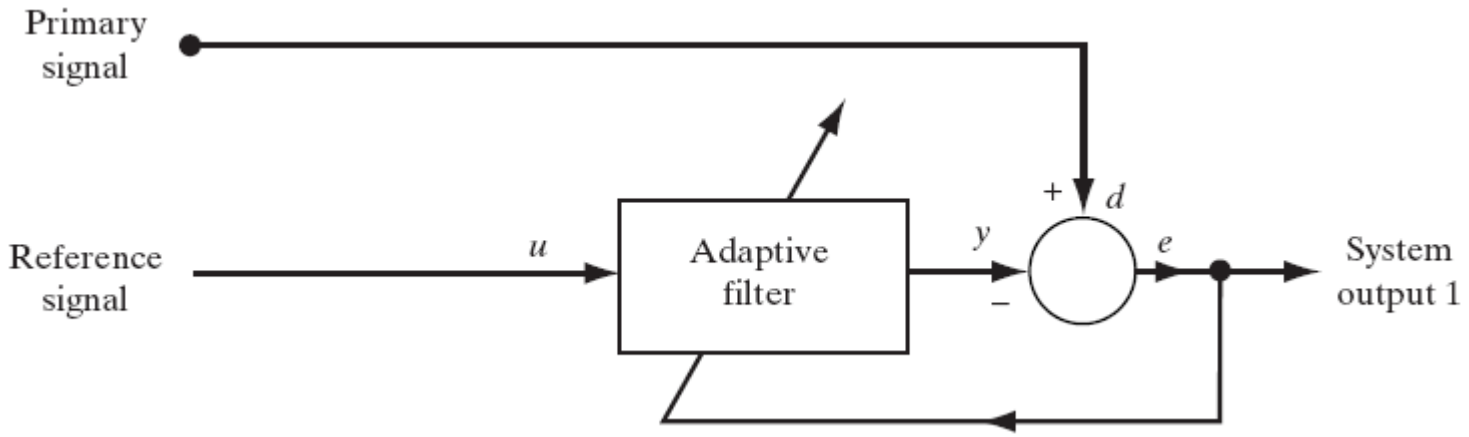
- Parameters
  - u=input of adaptive filter=delayed version of random signal
  - y=output of adaptive filter
  - d=desired response=random signal
  - e=d-y=estimation error=system output

自适应滤波的应用：预测估计



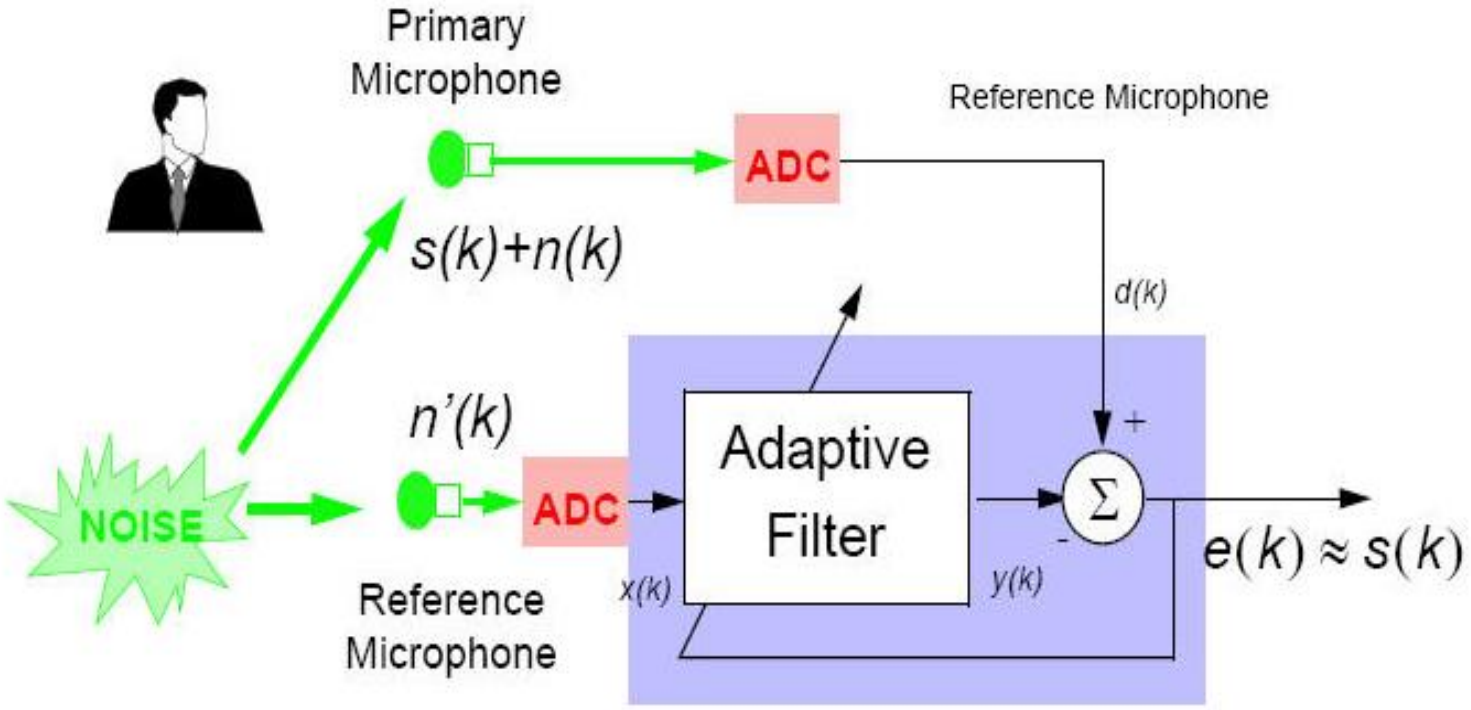
自适应滤波的应用：消除干扰

- 用于消除未知的干扰信号

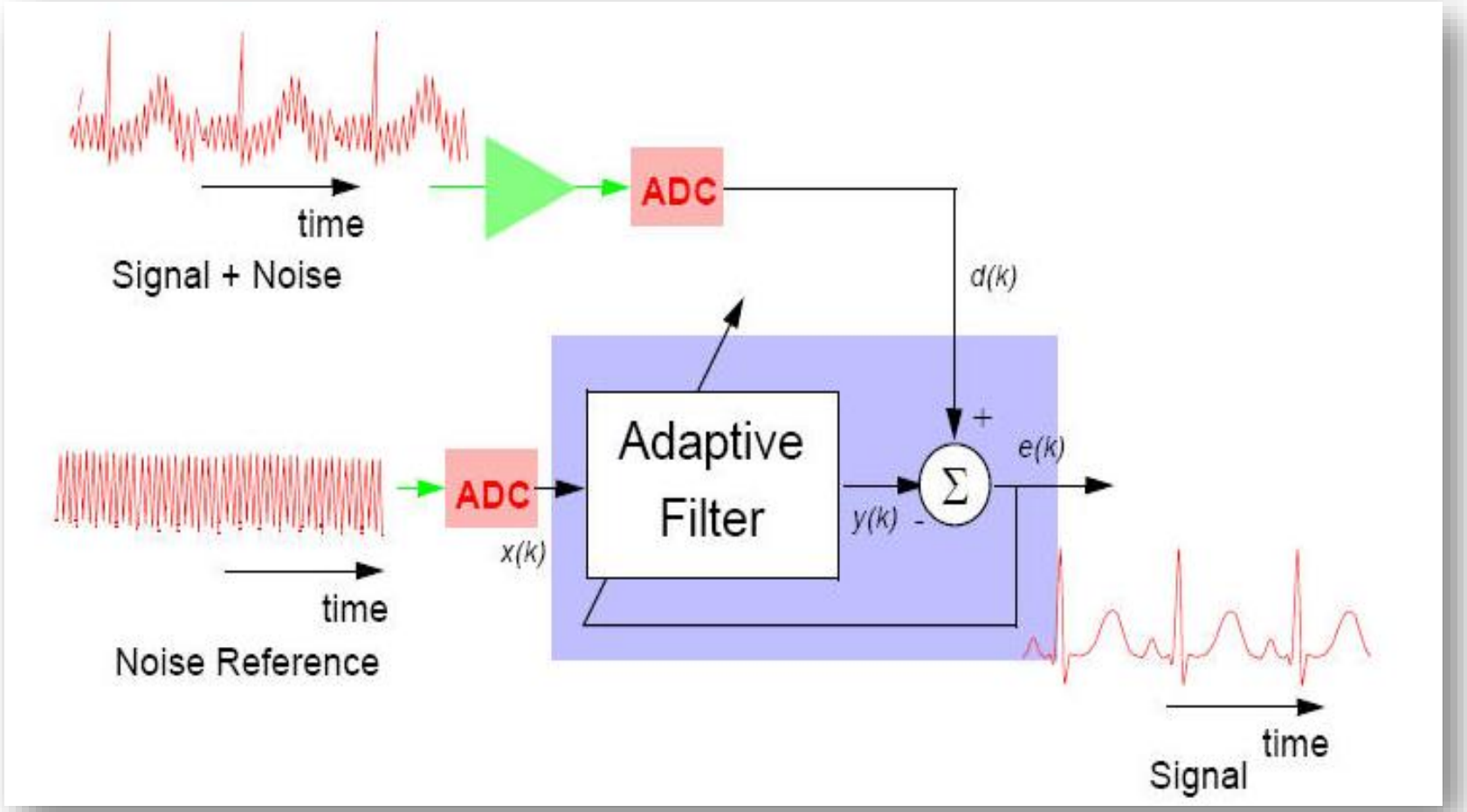


- u=input of adaptive filter=reference signal
- y=output of adaptive filter
- d=desired response=primary signal
- e=d-y=estimation error=system output

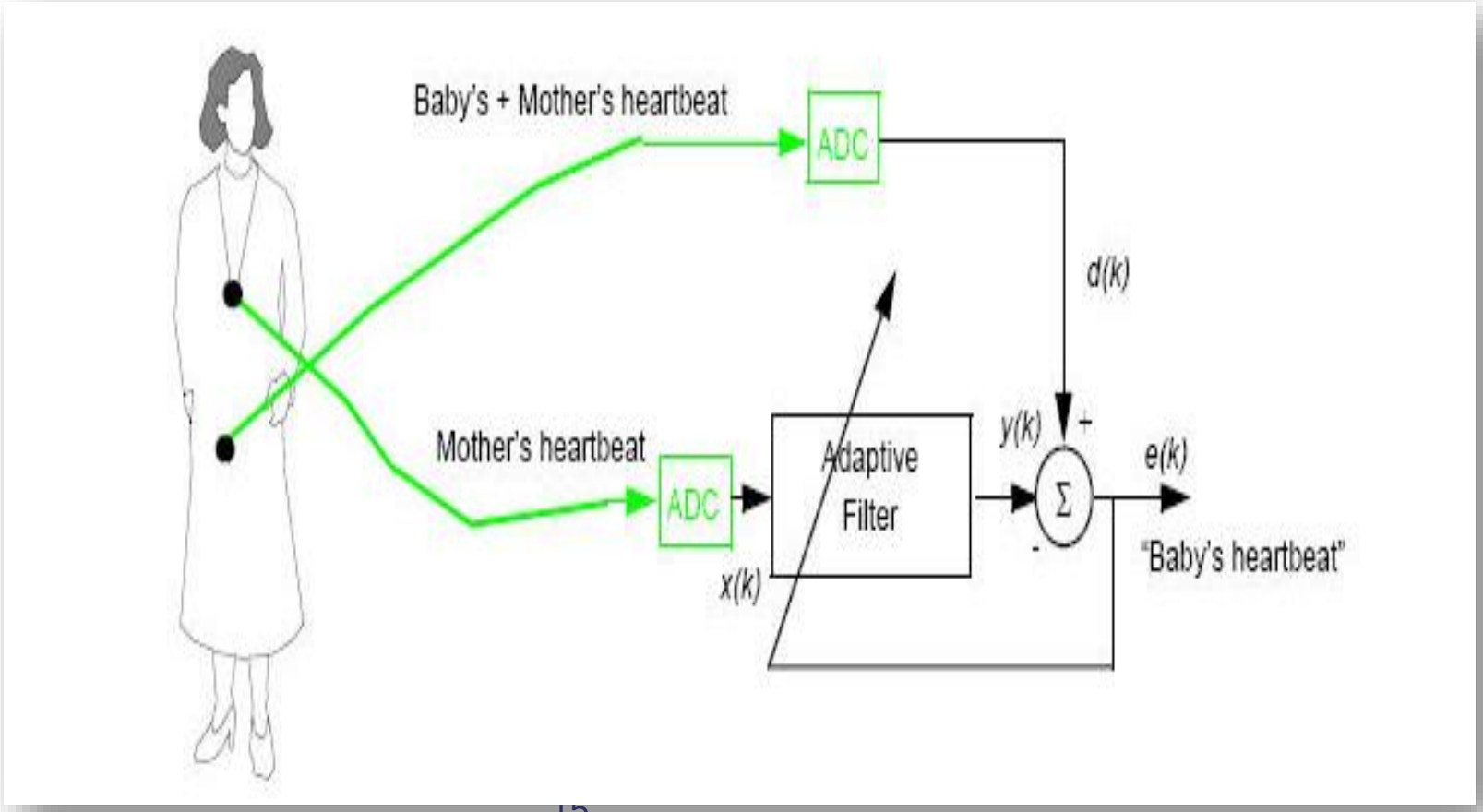
自适应滤波的应用：消除噪声



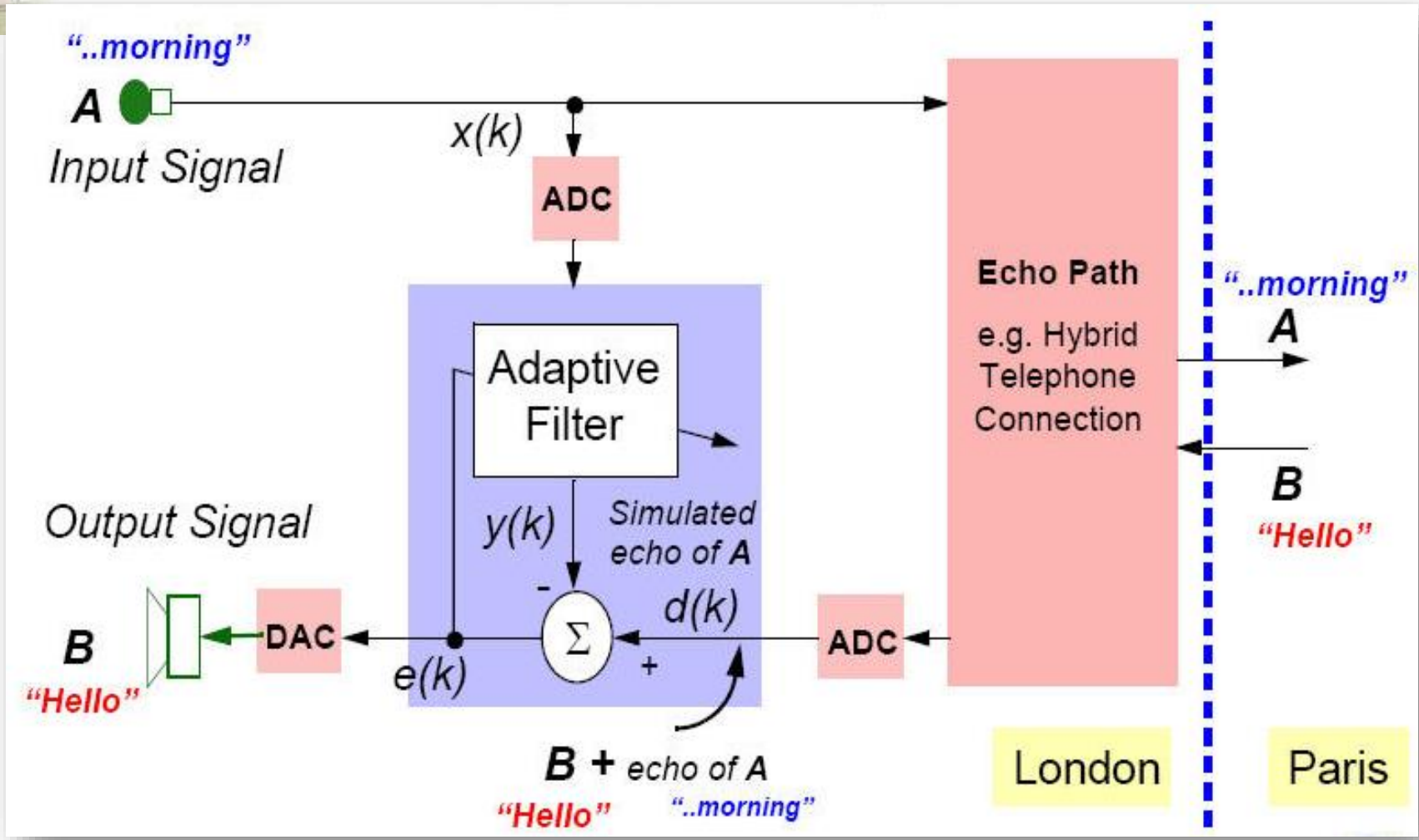
自适应滤波的应用：消除噪声



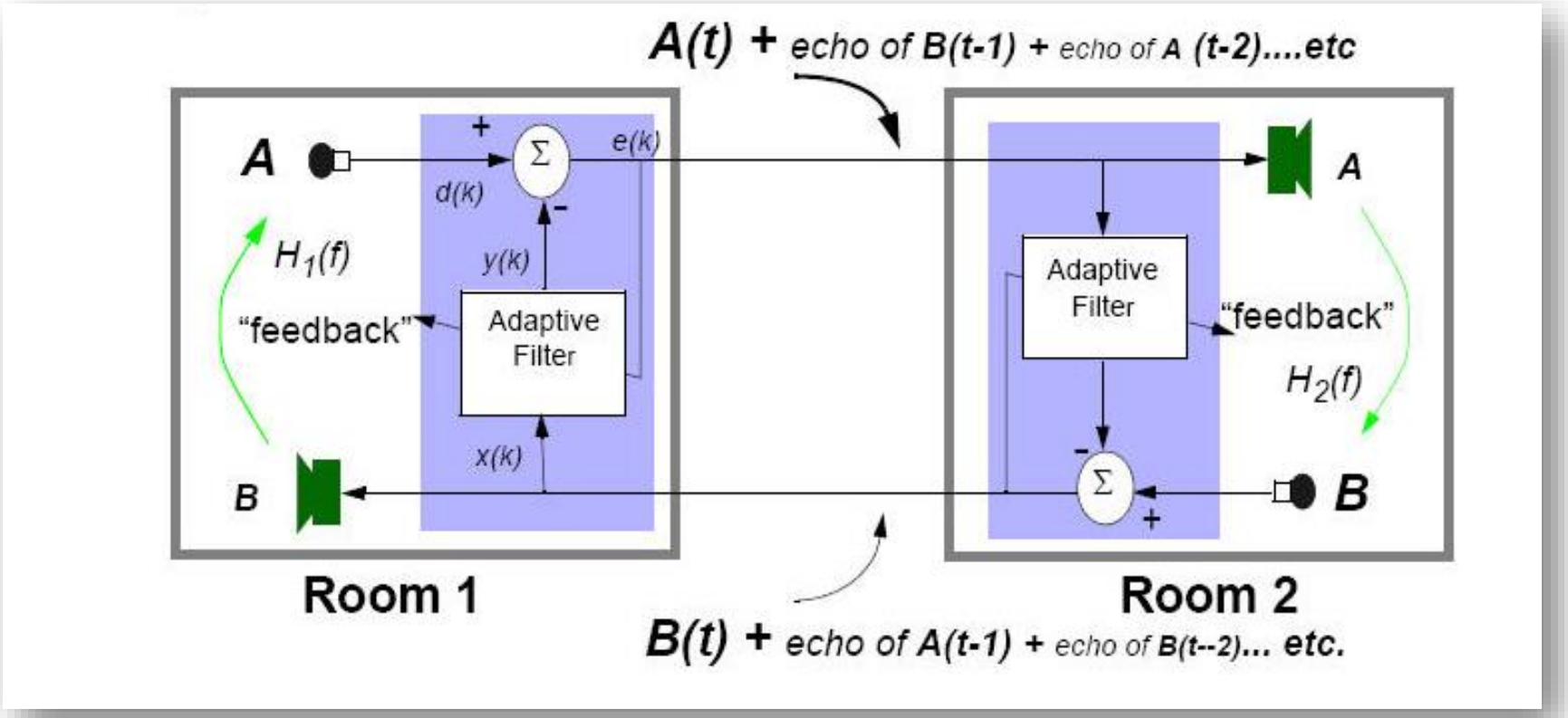
自适应滤波的应用：消除干扰



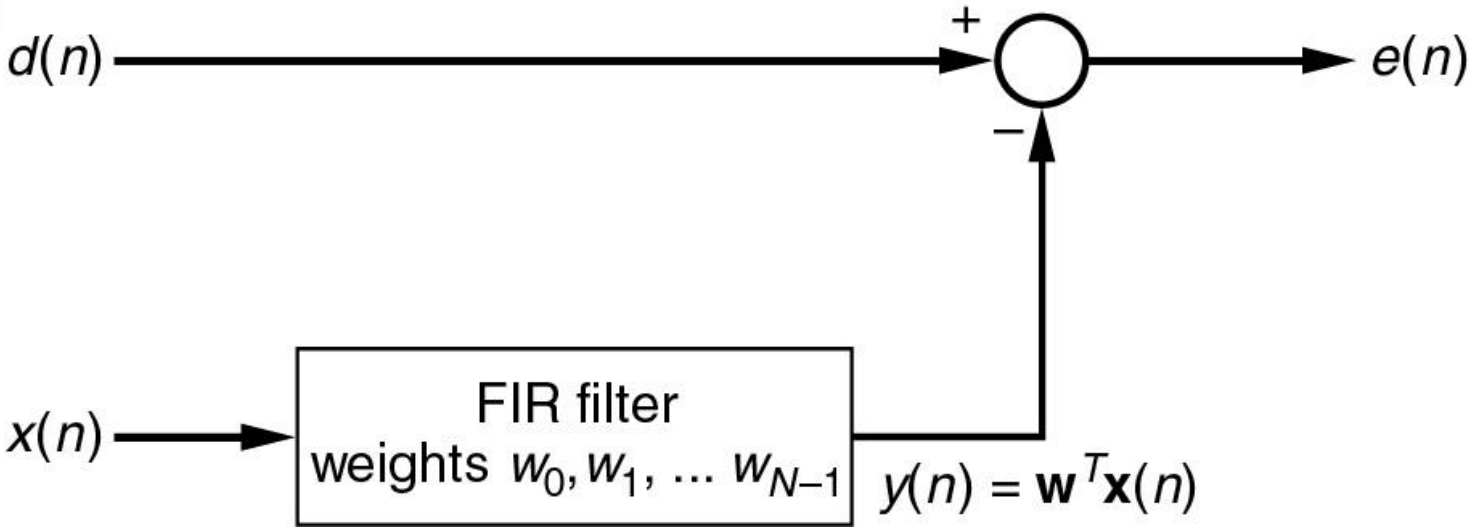
自适应滤波的应用：消除回声



自适应滤波的应用：消除回声



4.2 维纳滤波



Parameters

x(n) = input of FIR filter
y(n) = output of FIR filter
d(n) = desired response
e(n)= d(n)-y(n) = estimation error

FIR filter:

$y(n) = w_0x(n)+w_1x(n-1)+\ldots+w_{N-1}x(n-N+1)$
$= \boldsymbol{W}^T\boldsymbol{X}(n)$

$\boldsymbol{W}^T=[\, w_0\ w_1\ \ldots\ w_{N-1}\,]$

$\boldsymbol{X}(n)=[x(n)\ x(n-1)\ldots,x(n-N+1)]$

维纳滤波

目标：找到最佳的滤波器参数使得均方误差 $E\{e^2(n)\}$最小

定义目标函数

$$\begin{aligned} J(\mathbf{w}) &= E\{e^2(n)\} \\ &= E\{[d(n) - y(n)]^2\} \\ &= E\{d^2(n) - 2d(n)y(n) + y^2(n)\}. \end{aligned}$$

将滤波器定义代入，可得

$$\begin{aligned} J(\mathbf{w}) &= E\{d^2(n) - 2d(n)y(n) + y^2(n)\} \\ &= E\{d^2(n)\} - 2E\{d(n)\mathbf{w}^T\mathbf{x}(n)\} + E\{\mathbf{w}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}\}. \end{aligned}$$

维纳滤波

$$J(\mathbf{w}) = E\{d^2(n) - 2d(n)y(n) + y^2(n)\}$$
$$= E\{d^2(n)\} - 2E\{d(n)\mathbf{w}^T\mathbf{x}(n)\} + E\{\mathbf{w}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}\}.$$

因为W而是滤波器参数而不是随机变量，所以上式可以简化为：

$$J(\mathbf{w}) = E\{d^2(n)\} - 2\mathbf{w}^T E\{d(n)\mathbf{x}(n)\} + \mathbf{w}^T E\{\mathbf{x}(n)\mathbf{x}^T(n)\}\mathbf{w}.$$

定义 $P = E\{d(n)\,X(n)\}$ 为互相关矩阵， $R = E\{X(n)\,X^T(n)$ 为自相关矩阵，则有

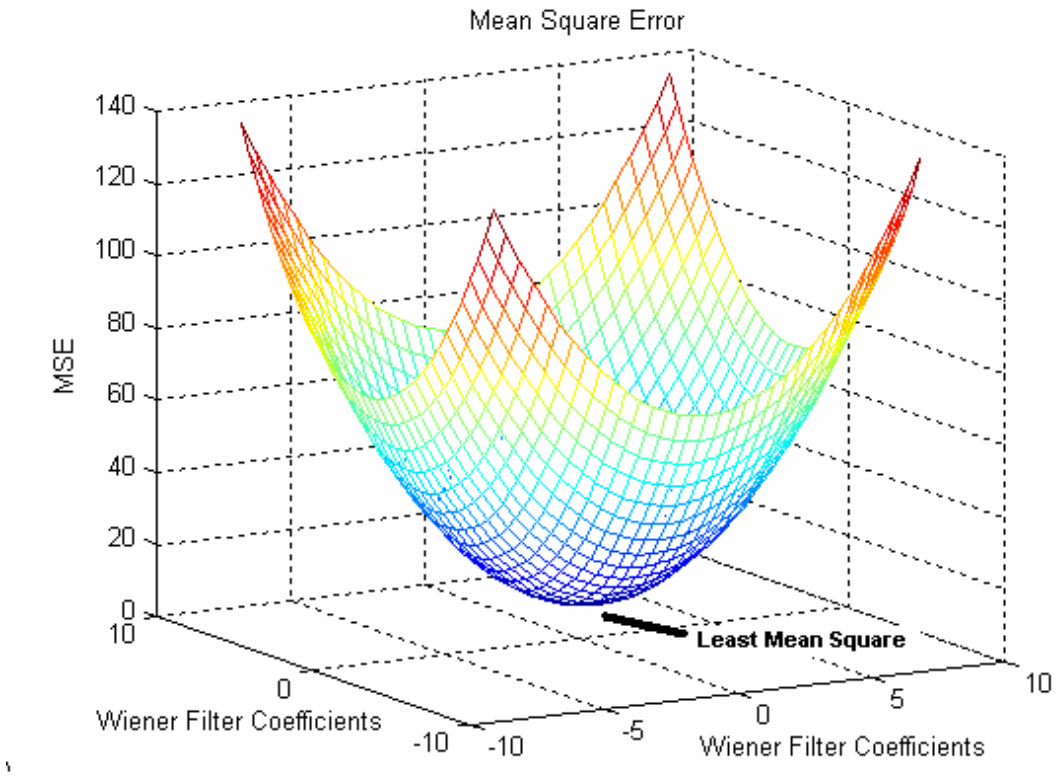
$$J(\mathbf{w}) = \sigma_d^2 - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w}.$$

维纳滤波

以二元维纳滤波为例，代价函数为：

$$
\begin{aligned}
J(w_0, w_1) &= E\left\{d^2(n)\right\} - 2E\left\{d(n)y(n)\right\} + E\left\{y^2(n)\right\} \\
&= \sigma_d^2 - 2E\left\{d(n)[w_0 x(n) + w_1 x(n-1)]\right\} + E\left\{[w_0 x(n) + w_1 x(n-1)]^2\right\} \\
&= \sigma_d^2 - 2w_0 E\left\{d(n)x(n)\right\} - 2w_1 E\left\{d(n)x(n-1)\right\} \\
&\quad + w_0^2 E\left\{x^2(n)\right\} + 2w_0 w_1 E\left\{x(n)x(n-1)\right\} + w_1^2 E\left\{x^2(n-1)\right\} \\
&= \sigma_d^2 - 2w_0 p(0) - 2w_1 p(1) + w_0^2 r(0) + 2w_0 w_1 r(1) + w_1^2 r(0).
\end{aligned}
$$

二次型代价函数


Mean Square Error
MSE
140
120
100
80
60
40
20
0
10
0
-10
Wiener Filter Coefficients
-10
-5
0
5
10
Wiener Filter Coefficients
Least Mean Square

维纳滤波

对代价函数分别求两个权重系数的梯度

$$\frac{\partial J}{\partial w_0} = -2p(0) + 2w_0 r(0) + 2w_1 r(1)$$

$$\frac{\partial J}{\partial w_1} = -2p(1) + 2w_0 r(1) + 2w_1 r(0).$$

将梯度写成矩阵形式：

$$\nabla \mathbf{J}(w_0, w_1) = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \frac{\partial J}{\partial w_1} \end{bmatrix} = -2 \begin{bmatrix} p(0) \\ p(1) \end{bmatrix} + 2 \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}.$$

维纳滤波

对于通用的N-1元维纳滤波， 可以采用类似策略来计算梯度，

$$\nabla \mathbf{J}(\mathbf{w}) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}.$$

通过计算上式等于0，可以得到维纳滤波的表达式：

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}$$

其中，$P = E\{d(n)\, X(n)\}$ 为互相关矩阵， $R = E\{X(n)\, X^T(n)\}$ 为自相关矩阵

维纳滤波

维纳滤波的表达式：

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$$

其中，$P = E\{d(n)\,X(n)\}$ 为互相关矩阵，$R = E\{X(n)\,X^T(n)\}$ 为自相关矩阵

自适应滤波？

性能分析：

- 利用平稳随机过程的相关特性和频谱特性，对混有噪声的信号进行滤波，奠定了最优滤波理论的基础。
- 当输入过程是广义平稳的，且统计特性已知时，能够取得较好的结果。
- 但是，输入过程取决于外界环境的信号和干扰，其统计特性常常是未知的、变化的。

4.3 最小二乘滤波（LMS Filter）

- 梯度下降算法（**Steepest Descent Algorithm**）
- 最小二乘算法（**Least-mean-square algorithm**）
- 性能分析

最小二乘滤波（LMS Filter）

- 由Widrow & Hoff 在1959年提出
- 位列自适应算法的Top10
  - ✔ 算法简单，自适应调整过程速度快
    - ✔ 属于梯度下降算法，可以保证收敛性

最小二乘滤波（LMS Filter）

- **LMS filtering**分为两个步骤：
  - ✔ Filtering, producing
    1) output signal
    2) estimation error
  - ✔ Adaptive process, automatic adjustment of filter tap weights

最小二乘滤波（LMS Filter）

- **LMS filtering分为两个步骤：**
  - ✔ Filtering, producing
    1) output signal
    2) estimation error
  - ✔ Adaptive process, automatic adjustment of filter tap weights

最小二乘滤波（LMS Filter）

- Input signal (vector): $\mathbf{u}(n)$
  - ✓ Autocorrelation matrix of input signal: $\mathbf{R}=\mathrm{E}\left[\mathbf{u}(n)\mathbf{u}^{\mathrm{H}}(n)\right]$
- Desired response: $d(n)$
  - ✓ Crosscorrelation vector between the input vector $\mathbf{u}(n)$ and the desired response $d(n)$: $\mathbf{p}=\mathrm{E}\left[\mathbf{u}(n)d^{*}(n)\right]$
- Filter tap weights: $\mathbf{w}(n)$
- Filter output: $y(n)=\sum_{k=0}^{M-1}w_k^{*}u(n-k)=\mathbf{w}^{\mathrm{H}}(n)\mathbf{u}(n)$
- Estimation error: $e(n)=d(n)-y(n)$
- Mean squared error: $J=\mathrm{E}\left[\left|e(n)\right|^{2}\right]=\mathrm{E}\left[e(n)e^{*}(n)\right]$

最小二乘滤波（LMS Filter）

基于递归机制的参数自适应

设W(n)为 第n次迭代后得到的滤波器参数，则第n+1次迭代的参数估计可表示为：

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{g}(n)$$

其中，g(n) 为方向向量，其作用是最小化目标函数J(W(n+1)) =J(W(n)+g(n)).

为实现目标函数的最小化，可以对J(W(n+1))进行泰勒级数展开。

最小二乘滤波（LMS Filter）

J(W(n+1)) 的泰勒级数展开为：

$$J(\mathbf{w}+\mathbf{g}) = J(\mathbf{w}) + \sum_i g_i \frac{\partial J(\mathbf{w})}{\partial w_i} + \frac{1}{2}\sum_i \sum_j g_i g_j \frac{\partial^2 J(\mathbf{w})}{\partial w_i \partial w_j} + \cdots$$

以二元滤波器为例，

$$J(\mathbf{w}+\mathbf{g}) = J(\mathbf{w}) + g_0 \frac{\partial J(\mathbf{w})}{\partial w_0} + g_1 \frac{\partial J(\mathbf{w})}{\partial w_1} + \frac{1}{2}[g_0^2 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_0} + g_0 g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} +$$

$$g_0 g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_0} + g_1^2 \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_1}].$$

最小二乘滤波（LMS Filter）

对上式分别对g_0 和 g_1 求梯度，可得

$$\frac{\partial J(\mathbf{w}+\mathbf{g})}{\partial g_0} = \frac{\partial J(\mathbf{w})}{\partial w_0} + g_0 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_0} + g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1}$$

$$\frac{\partial J(\mathbf{w}+\mathbf{g})}{\partial g_1} = \frac{\partial J(\mathbf{w})}{\partial w_1} + g_0 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} + g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_1}.$$

其矩阵形式为：

$$\frac{\partial J(\mathbf{w}+\mathbf{g})}{\partial \mathbf{g}} = \nabla \mathbf{J} + \begin{bmatrix} \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_0} & \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} \\ \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} & \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_1} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \end{bmatrix}.$$

Hessian Matrix

最小二乘滤波（LMS Filter）

上式中的矩阵叫做**Hessian Matrix**，表示为 ***H*.**

为实现**J(W+g)**的最小化，令上式等于 **0**，可以得到：

$$\mathbf{g} = -\mathbf{H}^{-1}\nabla\mathbf{J}.$$

可得到牛顿迭代法（一种梯度下降策略）的表达式：

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mathbf{H}^{-1}\nabla\mathbf{J}.$$

最小二乘滤波（LMS Filter）

将上式中的Hessian矩阵近似为 H=2I ，则可以得到最速下降法的通用表达式：

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2}\nabla \mathbf{J}.$$

这里, 常数μ 用于控制收敛速度。

The a marized in the

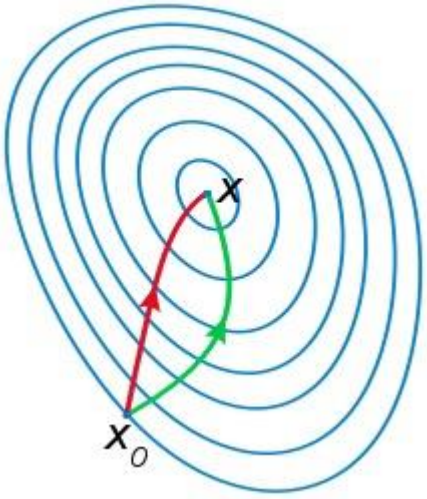


TABLE 7.1 Steepest Descent Algorithm

Initialization

$$\mathbf{w}(0) = \mathbf{0}$$

Algorithm

For $n = 0, 1, 2, \ldots$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$$

最小二乘滤波（LMS Filter）

- 将维纳滤波器的表达式代入到最速下降滤波中，可得:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla J(n)]$$
$$= \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$$

- 其中，R和P为瞬态估计，

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^{\mathrm{H}}(n), \qquad \hat{\mathbf{p}}(n) = \mathbf{u}(n)d^{*}(n)$$

- 上式也有其它的表现形式，比如使用瞬态误差 $|e(n)|^2$

最小二乘滤波（LMS Filter）

- 将瞬态误差 $|e(n)|^2$ 入到最速下降滤波中，

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \mu\left[\mathbf{p} - \mathbf{R}\mathbf{w}(n)\right] \\ &\approx \mathbf{w}(n) + \mu\left[\mathbf{u}(n)d^*(n) - \mathbf{u}(n)\underbrace{\mathbf{u}^{\mathrm{H}}(n)\mathbf{w}(n)}_{y^*(n)}\right] \\ &= \mathbf{w}(n) + \mu\mathbf{u}(n)\left[d^*(n) - y^*(n)\right] \\ &= \mathbf{w}(n) + \mu\mathbf{u}(n)e^*(n)\end{aligned}$$

滤波器输出：$y(n) = W^{T}u(n)$

最小二乘滤波（LMS Filter）

- 由于只能得到关于***R***和***p***的粗略估计，每一步的自适应步长是随机的。
- 算法中所采用的递归机制，相当于在参数的自适应调整过程中对于每一次的粗略估计进行了有效的平均处理，提高了整体算法的有效性。

最小二乘滤波（LMS Filter）

收敛性分析

- Notations:

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}, \quad \text{optimum Wiener solution}$$

$$\boldsymbol{\varepsilon}(n) = \mathbf{w}(n) - \mathbf{w}_o, \quad \text{tap-weight-error vector}$$

$$e(n) = d(n) - \mathbf{w}^{\mathrm{H}}(n)\mathbf{u}(n), \quad \text{estimation error}$$

$$e_o(n) = d(n) - \mathbf{w}_o^{\mathrm{H}}\mathbf{u}(n), \quad \text{estimation error for Wiener solution}$$

$$J_{\min} = \mathrm{E}\left[\left|e_o(n)\right|^2\right], \quad \text{minimum MSE for Wiener solution}$$

$$\lambda_{\min}, \lambda_{\max}, \quad \text{min. and max. eigenvalues of } \mathbf{R}$$

- Convergence criteria:

a) Convergence in the mean: $E\left[\left\|\boldsymbol{\varepsilon}(n)\right\|\right] \to 0 \quad \text{as } n \to \infty$

b) Convergence in the mean square: $D(n) = E\left[\left\|\boldsymbol{\varepsilon}(n)\right\|^2\right] \to 0 \quad \text{as } n \to \infty$

c) $J(n) = E\left[\left|e(n)\right|^2\right] \to \text{constant} \quad \text{as } n \to \infty$

最小二乘滤波（LMS Filter）

收敛性分析

由上述定义可知，tap-weight-error的变化率为

$$\begin{aligned}\boldsymbol{\varepsilon}(n+1) &= \left[\mathbf{I}-\mu\mathbf{u}(n)\mathbf{u}^{\mathrm{H}}(n)\right]\boldsymbol{\varepsilon}(n)+\mu\mathbf{u}(n)e_o^*(n)\\ &\approx \left[\mathbf{I}-\mu\mathbf{R}\right]\boldsymbol{\varepsilon}(n)+\mu\mathbf{u}(n)e_o^*(n)\end{aligned}$$

计算tap-weight-error的自相关矩阵K，

$$\begin{aligned}\mathbf{K}(n) &= E\left[\boldsymbol{\varepsilon}(n)\boldsymbol{\varepsilon}^{\mathrm{H}}(n)\right]\\ &= \left(\mathbf{I}-\mu\mathbf{R}\right)\mathbf{K}(n-1)\left(\mathbf{I}-\mu\mathbf{R}\right)+\mu^2 J_{\min}\mathbf{R}\end{aligned}$$

K(n) does not go to zero

为此， LMS算法的收敛性与（I-μR）的平方有着极大关系。

最小二乘滤波（LMS Filter）

收敛性分析

对于回归式 $x_{ii}(n+1)=(1-\mu\lambda_i)^2 x_{ii}(n)+\mu^2\lambda_i J_{\min}$ ，其中λ是R的特征值，

其收敛的必要条件是 $|1-\mu\lambda_i|<1,$

也就是说LMS算法收敛的必要条件是

$$0<\mu<\frac{2}{\lambda_{\max}}$$

最小二乘滤波（LMS Filter）

收敛性分析

步长的选择

$$0 < \mu < \frac{2}{\lambda_{\max}}, \quad \lambda_{\max}?$$

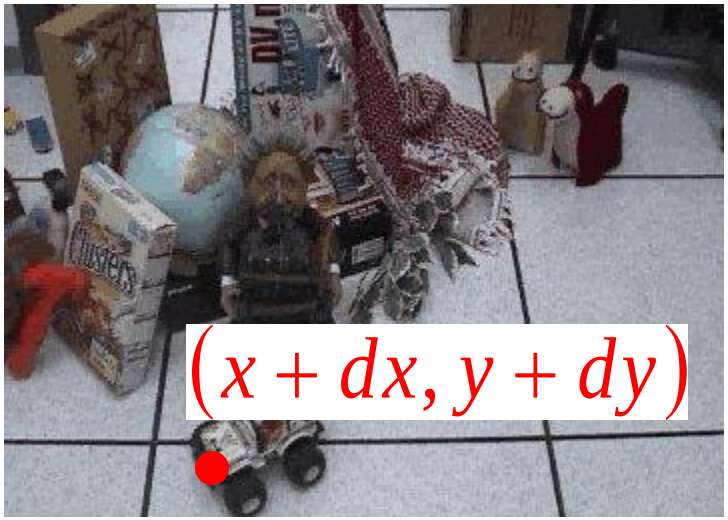
✓ Conservative estimate: $\mathrm{tr}[\mathbf{R}] = \sum_{i=1}^{M} \lambda_i \Rightarrow \mathrm{tr}[\mathbf{R}] > \lambda_{\max}$

$$\mathrm{tr}[\mathbf{R}] = Mr(0) = E[\mathbf{u}^{\mathrm{H}}(n)\mathbf{u}(n)]$$

Thus $0 < \mu < \dfrac{2}{\text{power of input vector}}$ should be a safe choice.

二维运动估计

- 关键性假设
  - 假设各幅图像的亮度具有一致性
  - 运动较小



二维运动估计

- **基础性假设**

点的亮度变化仅由运动引起

$$I(\mathbf{x}, t) \approx I(\mathbf{x} + \delta\mathbf{x}, t + \delta t)$$

↓ **Taylor展开**

$$I(\mathbf{x}, t) = I(\mathbf{x}, t) + \nabla I \cdot \delta\mathbf{x} + \delta t I_t + O^2$$

⇓

$$\nabla I \cdot \mathbf{v} + I_t = 0$$ **光流约束方程**

物理意义:如果一个固定的观察者观察一幅活动的场景，那么所得图象上某点灰度的(一阶)时间变化率是场景亮度变化率与该点运动速度的乘积。

Application of Covariance Matrix

- **Prob: we have more equations than unknowns**

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } ||Ad - b||^2$$

- Solution: solve least squares problem
  - minimum least squares solution given by solution (in d) of:

$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$A^T A$ $\qquad\qquad$ $A^T b$

  - The summations are over all pixels in the K x K window

Application of Covariance Matrix

– **Optimal (u, v) satisfies Lucas-Kanade equation**

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$A^T A$ $\qquad\qquad$ $A^T b$

When is This Solvable?

- $\mathbf{A^T A}$ should be invertible
- $\mathbf{A^T A}$ should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of $\mathbf{A^T A}$ should not be too small
- $\mathbf{A^T A}$ should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Application of Covariance Matrix

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

- Suppose (x,y) is on an edge. What is A^TA?
  - gradients along edge all point the same direction
  - gradients away from edge have small magnitude

$$\left(\sum \nabla I (\nabla I)^T\right) \approx k \nabla I \nabla I^T$$

$$\left(\sum \nabla I (\nabla I)^T\right) \nabla I = k \|\nabla I\| \nabla I$$

  - ∇I is an eigenvector with eigenvalue $k\|\nabla I\|$

Application of Covariance Matrix

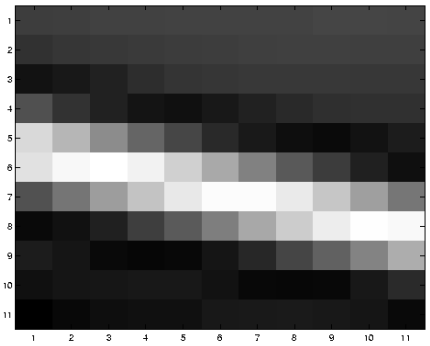
$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

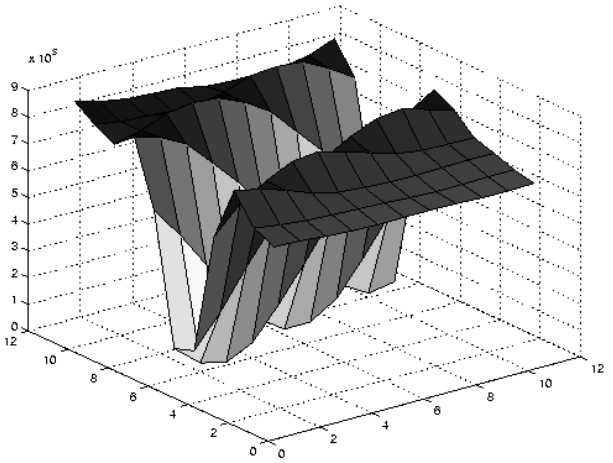
- Suppose (x,y) is on an edge. What is A^TA?
  - What's the other eigenvector of A^TA?
    - let N be perpendicular to ∇I

$$\left(\sum \nabla I (\nabla I)^T\right) N = 0$$

    - N is the second eigenvector with eigenvalue 0
- The eigenvectors of A^TA relate to edge direction and magnitude

Covariance Matrix of Edge



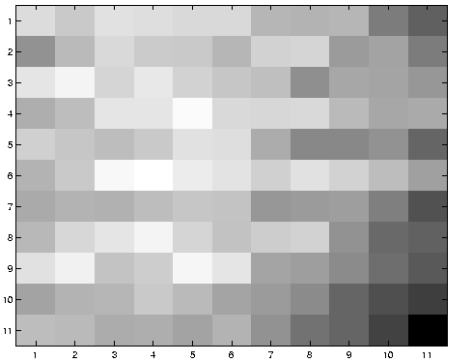


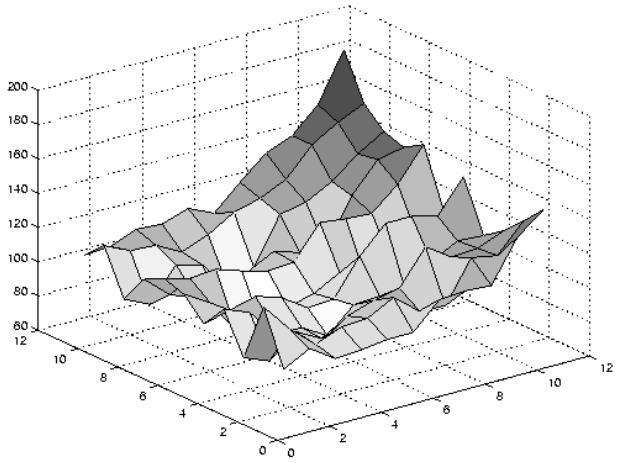
$$\sum \nabla I (\nabla I)^T$$

– large gradients, all the same

– large λ_1, small λ_2

Covariance Matrix of Low Texture

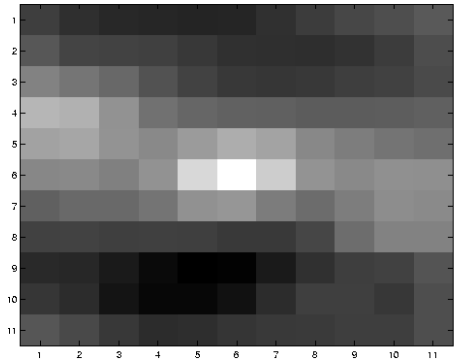


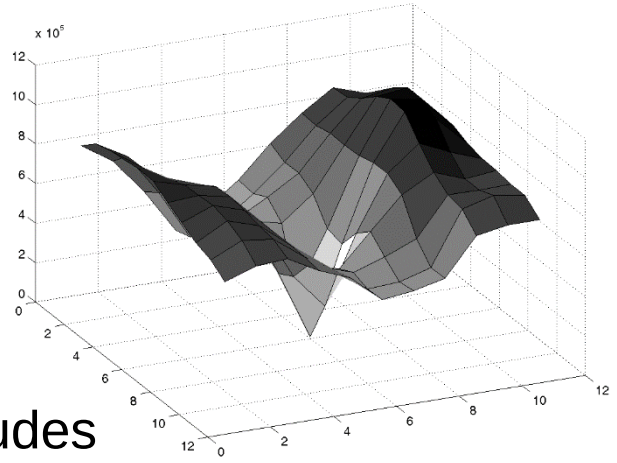


$$\sum \nabla I (\nabla I)^T$$

- gradients have small magnitude
- small λ_1, small λ_2

Covariance Matrix of High Texture





$$\sum \nabla I (\nabla I)^T$$

- gradients are different, large magnitudes
- large λ_1, large λ_2