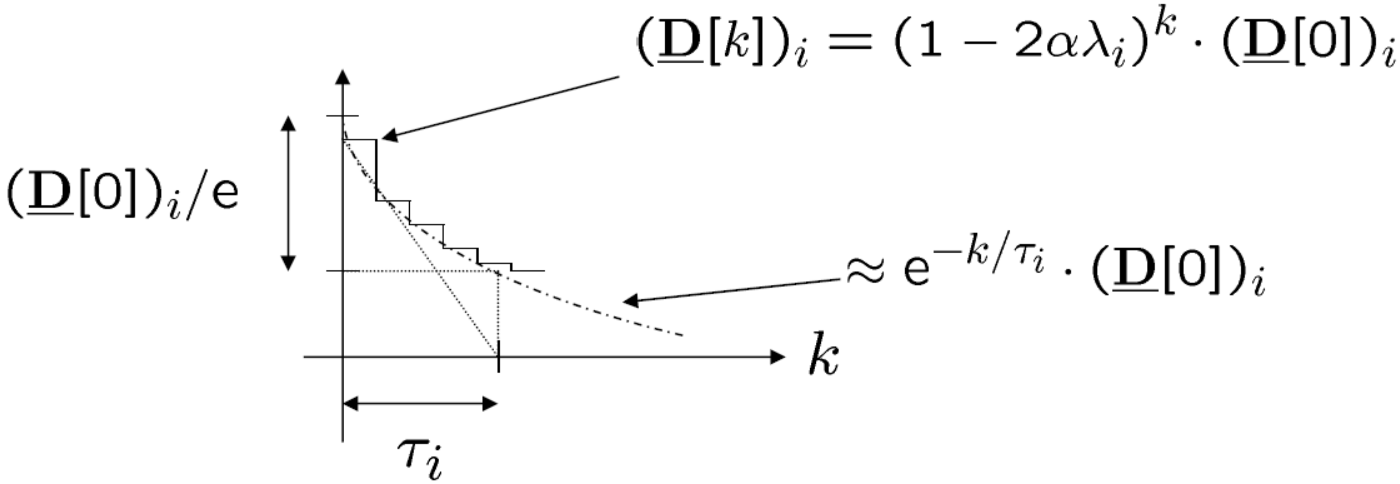
ADSP

Convergence rate SGD

Behaviour coefficient i of $\underline{\mathbf{D}}[k] = (\mathbf{I} - 2\alpha\mathbf{\Lambda})^k \, \underline{\mathbf{D}}[0]$:



Time constant follows from:

$$\mathrm{e}^{-k/\tau_i} \cdot (\underline{\mathbf{D}}[0])_i = (1 - 2\alpha\lambda_i)^k \cdot (\underline{\mathbf{D}}[0])_i$$

Convergence rate SGD

⇒ Time constant average weights behaviour:

$$\tau_{av,i} = \frac{-1}{\ln(1-2\alpha\lambda_i)} \qquad \text{for small } \alpha : \ \tau_{av,i} \approx \frac{1}{2\alpha\lambda_i}$$

Similar derivation for MMSE: $\tau_{mmse,i} \approx \dfrac{1}{4\alpha\lambda_i}$

Notes on overall time constant τ_{av}: $\frac{1}{2\lambda_{min}}$

- Depends on eigenvalue spread $\Gamma_x = \lambda_{max}/\lambda_{min}$

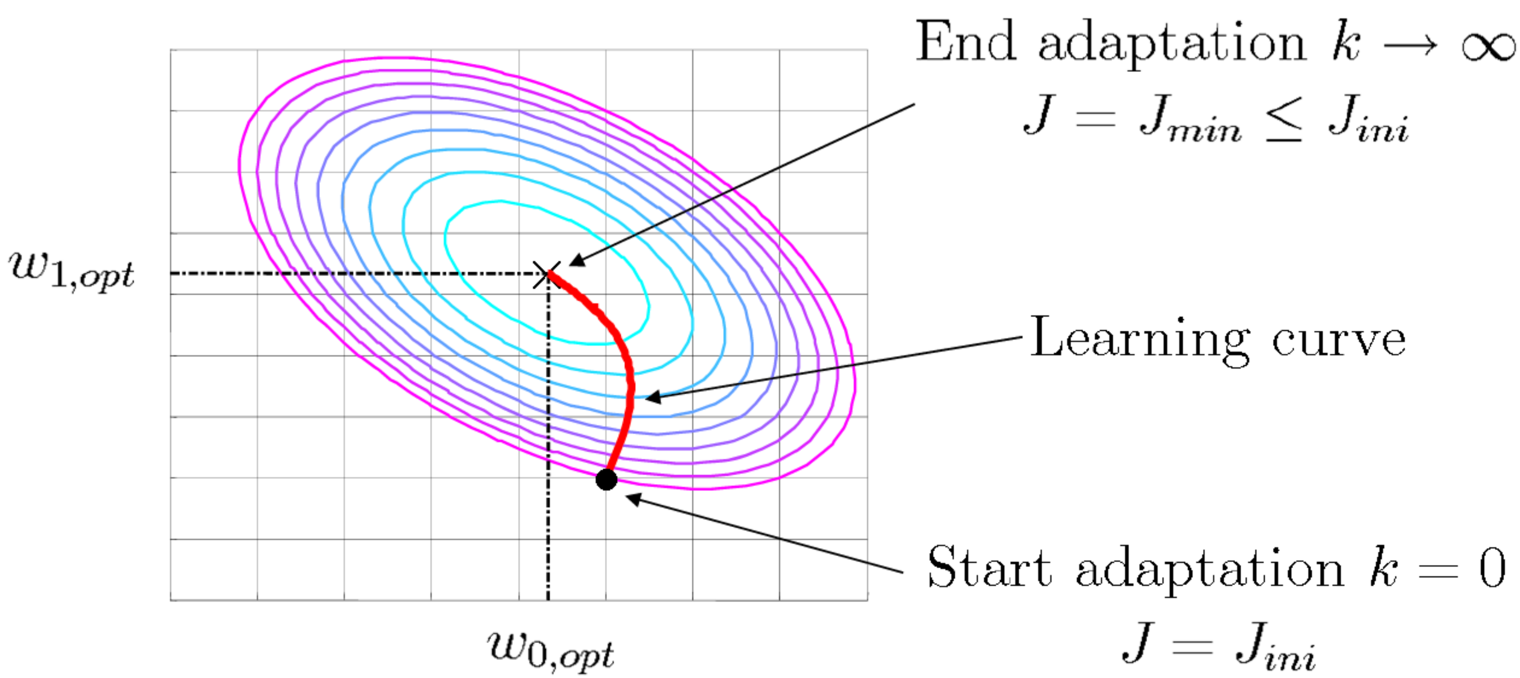
  Thus, the larger Γ_x the longer it takes for adaptation

  Q: What happens for white noise input process?

Convergence rate SGD

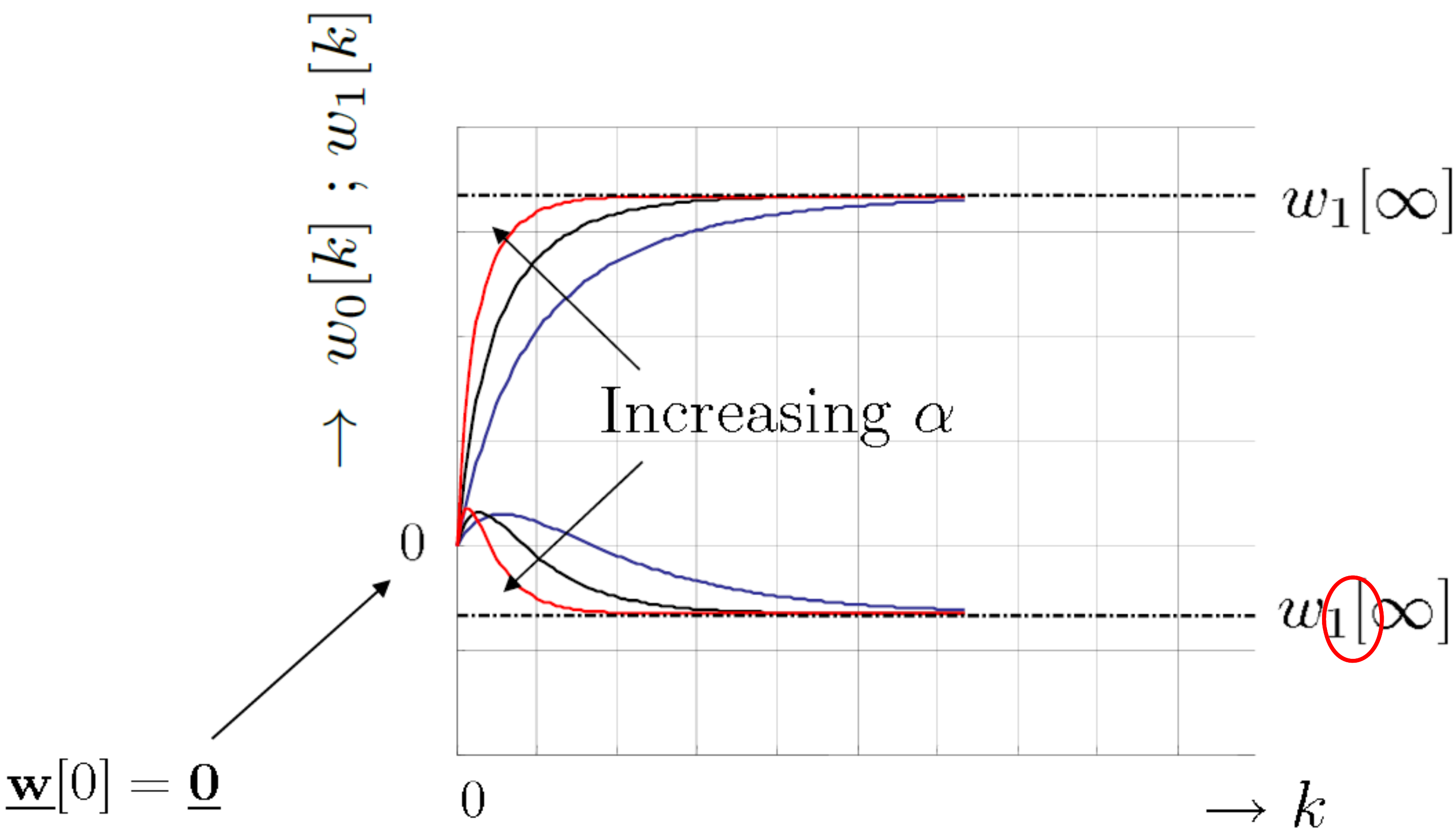
Learning curve in contour plot J

$$\Gamma_x = \frac{\lambda_{max}}{\lambda_{min}} = 3$$



Convergence rate SGD

Learning curves for different α



上海科技大学
ShanghaiTech University

ADSP Part B: Adaptive signal processing

Focus on single channel adaptive algorithms using FIR structure

- Applications Adaptive Algorithms
- Minimum Mean Square Error (MMSE)
- Constrained MMSE
- Least Square (LS)
- Steepest Gradient Descent (SGD)
- Three LMS variants: NLMS, Complex LMS, Constrained LMS
  基于梯度下降
- Newton
- Recursive Least Squares (RLS)
- Frequency Domain Adaptive Filter (FDAF)
- Summary

LMS 最小均方

LMS: Least Mean Square algorithm

Motivation: SGD not practical since gradient

assumes **known** autocorrelation $\mathbf{R}_x = E\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\}$

and cross correlation $\underline{\mathbf{r}}_{ex} = E\{e[k]\underline{\mathbf{x}}[k]\}$

LMS principle: Use *instantaneous* estimate gradient

$\hat{r}_{ex}$ $N\times 1$ $N\times N$ $\to \hat{R}_x$ 使用估计值

$\underline{\nabla} = -2(\underline{r}_{ex} - R_x \underline{w})$

$$
\begin{aligned}
\hat{\underline{\nabla}}[k] &= -2\left(e[k]\underline{\mathbf{x}}[k] - \underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\underline{\mathbf{w}}[k]\right) \\
&= -2\underline{\mathbf{x}}[k]\left(e[k] - \underline{\mathbf{x}}^t[k]\underline{\mathbf{w}}[k]\right) \\
&= -2\underline{\mathbf{x}}[k]\left(e[k] - \hat{e}[k]\right) \\
&= -2\underline{\mathbf{x}}[k]r[k]
\end{aligned}
$$

LMS

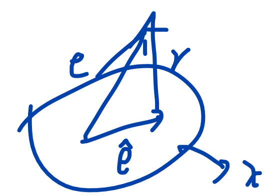
$$\underline{\mathbf{w}} := \underline{\mathbf{w}} - \alpha \underline{\widehat{\bigtriangledown}} \quad \Rightarrow$$

Least Mean Square (LMS) algorithm (Widrow, 1975):

$$k = 0: \quad \underline{\mathbf{w}}[0] = \underline{\mathbf{0}} \text{ (usually)}$$

"convolution"

$$k > 0: \quad \hat{e}[k] = \underline{\mathbf{w}}^t[k] \cdot \underline{\mathbf{x}}[k]_{N\times 1}$$



$$r[k] = e[k] - \hat{e}[k]$$

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha\underline{\mathbf{x}}[k]r[k]$$

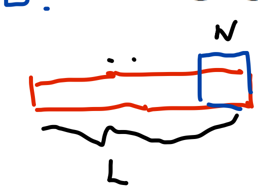
当迭代时收敛

$\hat{e}_j[k] = \underline{w}_j^t[k] \cdot \underline{x}_j[k]$

"cross correlation"

$j+1 \Rightarrow$ $\underline{w}$更新. $\underline{x}$稍往后看 L.

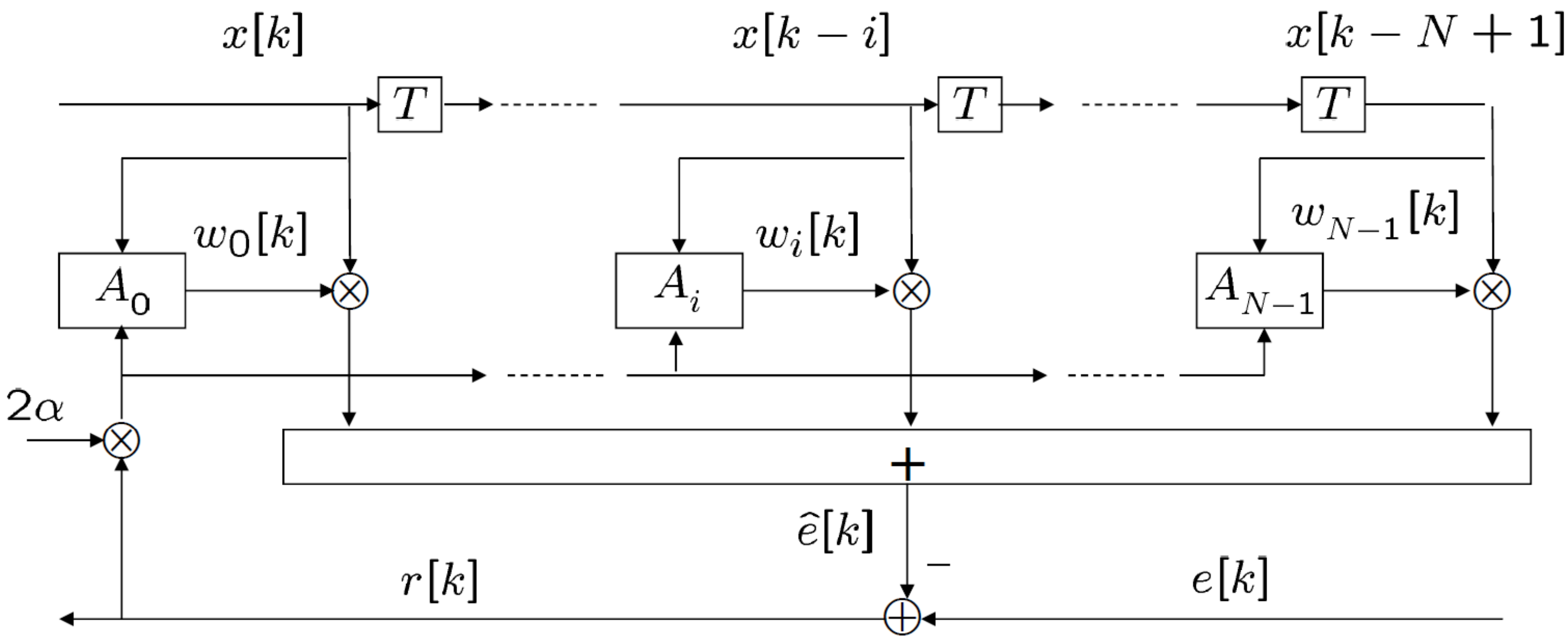
$\underline{w}$多次更新后. $\underline{x} \Rightarrow$

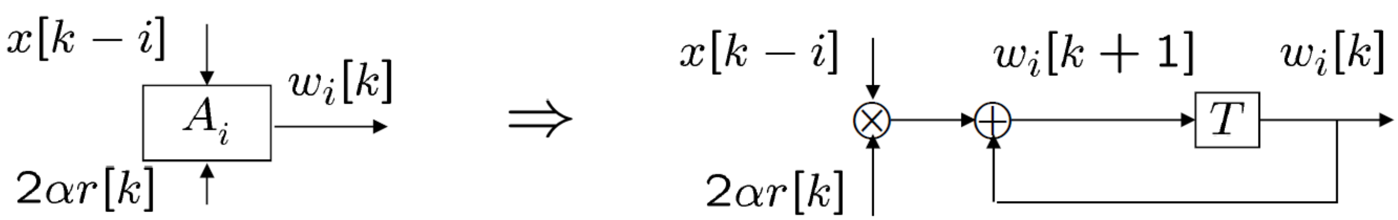


若 $\underline{w}$ 不收敛 则 $\underline{x}$ 可循环

LMS

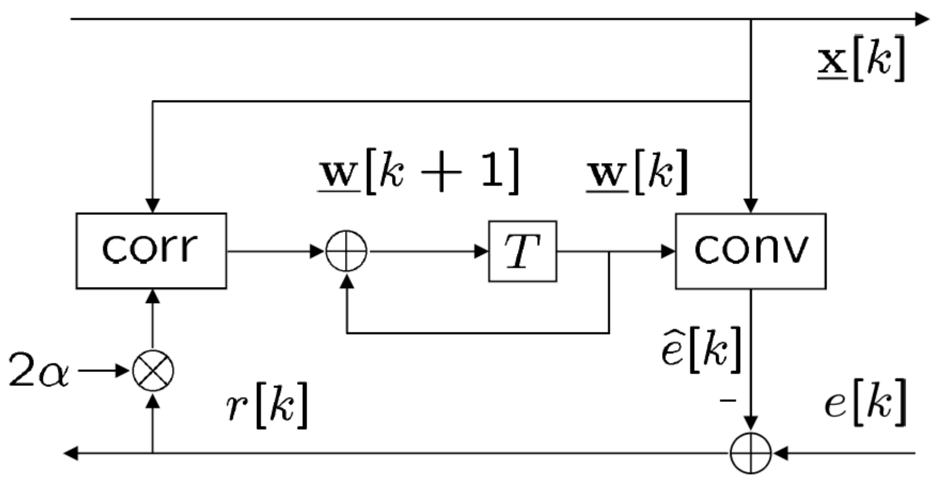
Realization scheme:





LMS

Simplified realization scheme:



Notes LMS:

Simple and robust algorithm (Complexity $O(2N)$)

LMS tries to "decorrelate" signals x and r 目的：去相关

In contrast to SGD:

Weights fluctuate around optimal values

自动使用 $\hat{R}_x$

There LMS variants

- **NLMS:**

  LMS with normalization of input signal variance

- **Complex-LMS:**

  LMS for complex signals and weights

  → Similar as before with MMSE

- **Constrained-LMS:**

  LMS with constrained updating

NLMS

分 = ··· $\underline{x}^t\underline{x}$

In LMS convergence properties depends on σ_x^2

$\Rightarrow$ May cause problems with non-stationary input

归一化

Solution: Normalized LMS (NLMS)

变步长

$$\boxed{\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + \frac{2\alpha}{\sigma_x^2}\underline{\mathbf{x}}[k]r[k]}$$

In practice: $\widehat{\sigma}_x^2[k]$ $\Rightarrow$ time-varying step size

可以用迭代

e.g. 1: $\widehat{\sigma}_x^2[k] = \beta\widehat{\sigma}_x^2[k-1] + (1-\beta)\left(\frac{\underline{\mathbf{x}}^t[k]\underline{\mathbf{x}}[k]}{N}\right)$ $\quad 0 < \beta < 1$

e.g. 2: $\widehat{\sigma}_x^2 = \frac{\underline{\mathbf{x}}^t[k]\underline{\mathbf{x}}[k]}{N} + \varepsilon$ with ε small constant

减去掉 mean 避免 0 的 出现

Complex LMS

LMS update rule: $\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] - \alpha \hat{\underline{\nabla}}$

with $\hat{\underline{\nabla}}$ estimate of $\underline{\nabla}$ 'without' $E\{\cdot\}$ $\Rightarrow$

(see slides Complex MMSE)

$$\hat{\underline{\nabla}} = 2\left(\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^h[k] \cdot \underline{\mathbf{w}} - \underline{\mathbf{x}}[k]e^*[k]\right)$$

? 咳?

$$= 2\underline{\mathbf{x}}[k] \cdot \left(\underline{\mathbf{x}}^h[k] \cdot \underline{\mathbf{w}} - e^*[k]\right) = 2\underline{\mathbf{x}}[k]r^*[k]$$

$$\boxed{\Rightarrow \text{ LMS: } \underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha \underline{\mathbf{x}}[k]r^*[k]}$$

Rule of thump C-(N)LMS:

Complex $\underline{\mathbf{w}}$, $\underline{\mathbf{x}}$ and replace r by r^* in (N)LMS

Constrained LMS

Derivation LMS algorithm with **constraints**:

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] - \alpha \underline{\hat{\nabla}}[k]$$

From constrained MMSE slides it follows:

$$\begin{aligned} \underline{\nabla}[k] &= -2\underline{\mathbf{r}}_{ex} + 2\mathbf{R}_x \underline{\mathbf{w}}[k] + \mathbf{C}\underline{\lambda} \\ &= -2E\{\underline{\mathbf{x}}[k]e[k]\} + 2E\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\}\underline{\mathbf{w}}[k] + \mathbf{C}\underline{\lambda} \end{aligned}$$

LMS estimate of gradient (no $E\{\ \}$):

$$\begin{aligned} \underline{\hat{\nabla}} &= -2\underline{\mathbf{x}}[k]e[k] + 2\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\underline{\mathbf{w}}[k] + \mathbf{C}\underline{\lambda} \\ &= -2\underline{\mathbf{x}}[k]\left(e[k] - \underline{\mathbf{x}}^t[k]\underline{\mathbf{w}}[k]\right) + \mathbf{C}\underline{\lambda} \\ &= -2\underline{\mathbf{x}}[k]r[k] + \mathbf{C}\underline{\lambda} \end{aligned}$$

Constrained LMS

$$\Rightarrow \quad \underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha \underline{\mathbf{x}}[k] r[k] - \alpha \mathbf{C} \underline{\lambda}$$

Furthermore: $\mathbf{C}^t \underline{\mathbf{w}}[k+1] = \underline{\mathbf{f}}$

$$\Rightarrow \quad \alpha \underline{\lambda} = \left(\mathbf{C}^t \mathbf{C}\right)^{-1} \left(\mathbf{C}^t \underline{\mathbf{w}}[k] + 2\alpha \mathbf{C}^t \underline{\mathbf{x}}[k] r[k] - \underline{\mathbf{f}}\right)$$

Final result:

$$\boxed{\underline{\mathbf{w}}[k+1] = \tilde{\mathbf{P}} \cdot \{\underline{\mathbf{w}}[k] + 2\alpha \underline{\mathbf{x}}[k] r[k]\} + \mathbf{C}\left(\mathbf{C}^t \mathbf{C}\right)^{-1} \underline{\mathbf{f}}}$$

TP_tmp

with projection matrix $\tilde{\mathbf{P}}$ (see Appendix):

$$\tilde{\mathbf{P}} = \underline{\mathbf{I} - \mathbf{C}\left(\mathbf{C}^t \mathbf{C}\right)^{-1} \mathbf{C}^t}$$

$\tilde{P}$ 的操作含义是垂直于向C矩阵投影

因为已知，所以

Note: Initial guess $\underline{\mathbf{w}}[0] = \underline{\mathbf{C}\left(\mathbf{C}^t \mathbf{C}\right)^{-1} \underline{\mathbf{f}}}$

ADSP

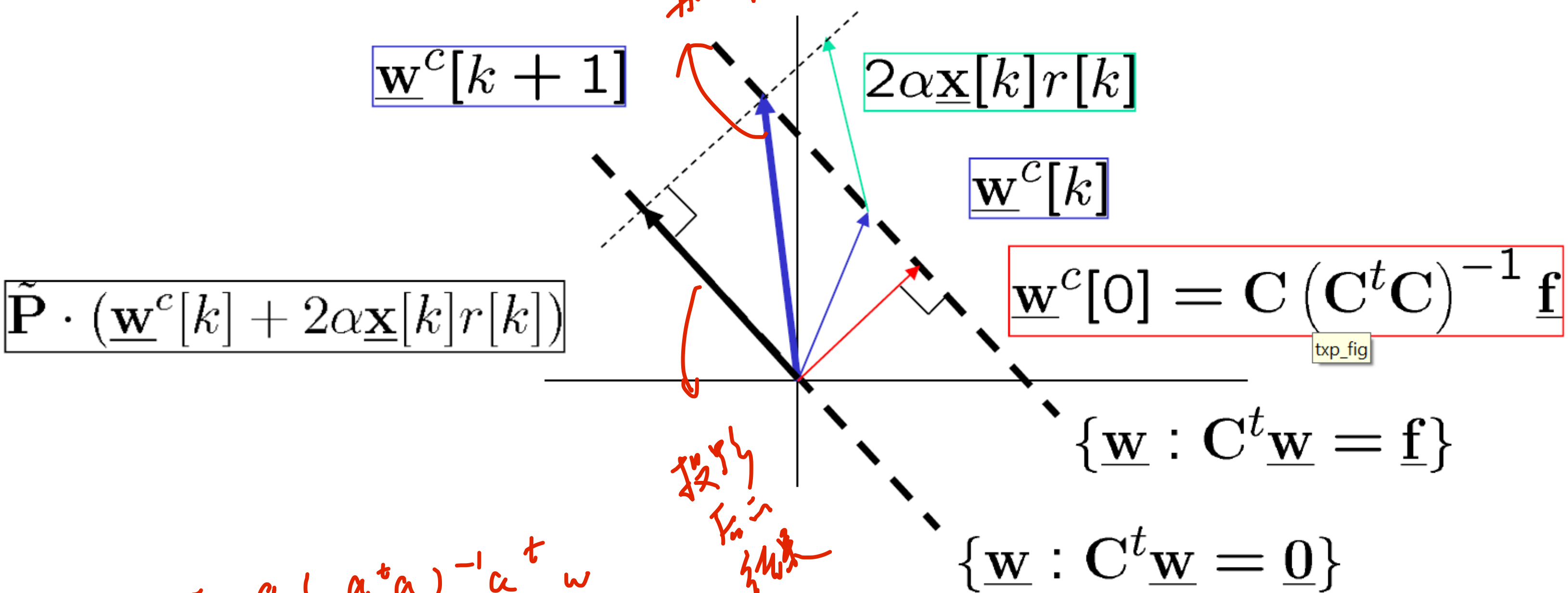
Constrained LMS

Geometrical explanation constrained LMS

Rewrite update equation:

$$\underline{\mathbf{w}}^c[k+1] \quad = \quad \tilde{\mathbf{P}} \cdot (\underline{\mathbf{w}}^c[k] + 2\alpha \underline{\mathbf{x}}[k] r[k]) + \underline{\mathbf{w}}^c[0]$$

加上偏移



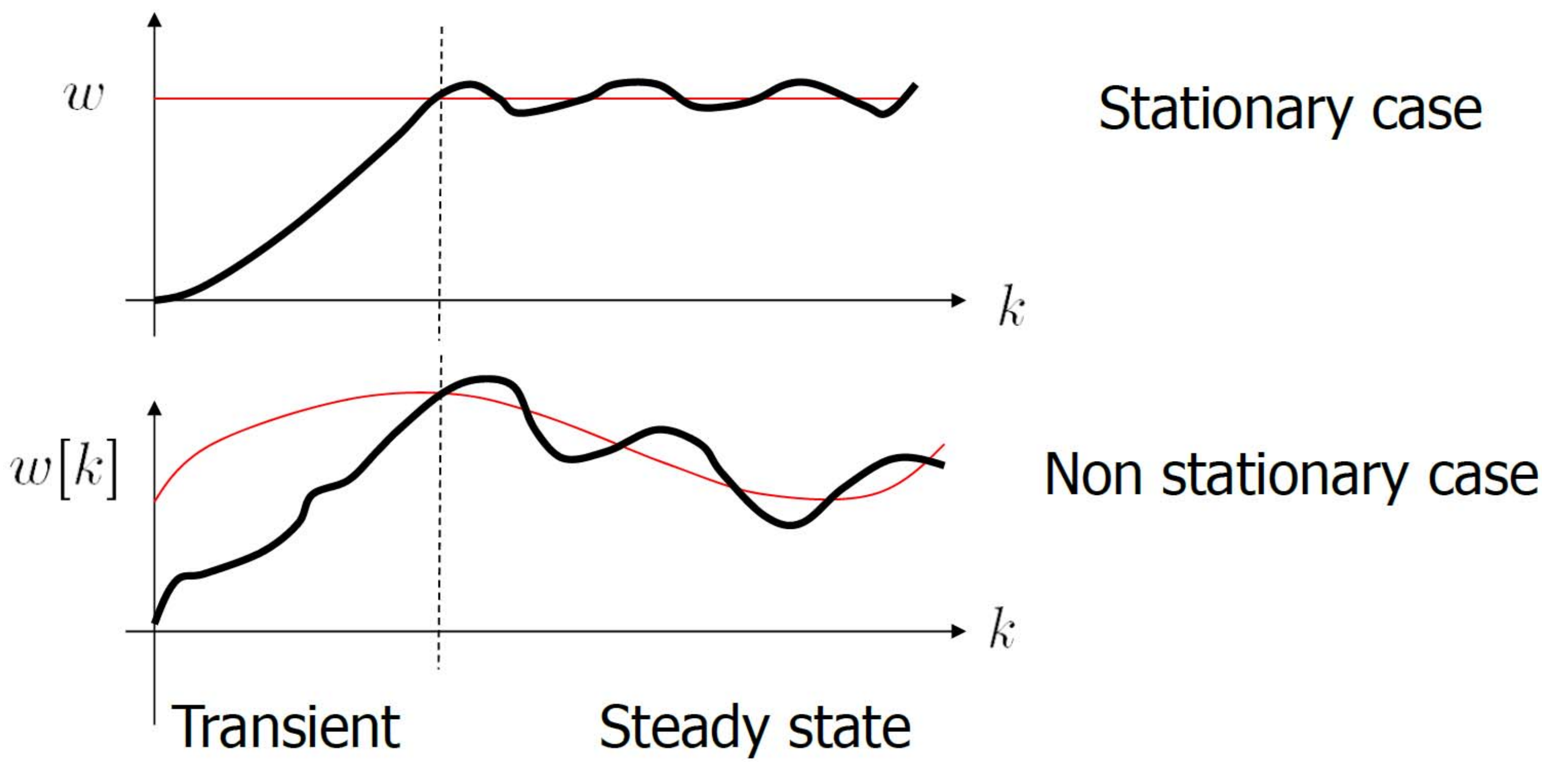
$w_p = a(a^t a)^{-1} a^t w$

w在a之投影

ADSP

Convergence LMS

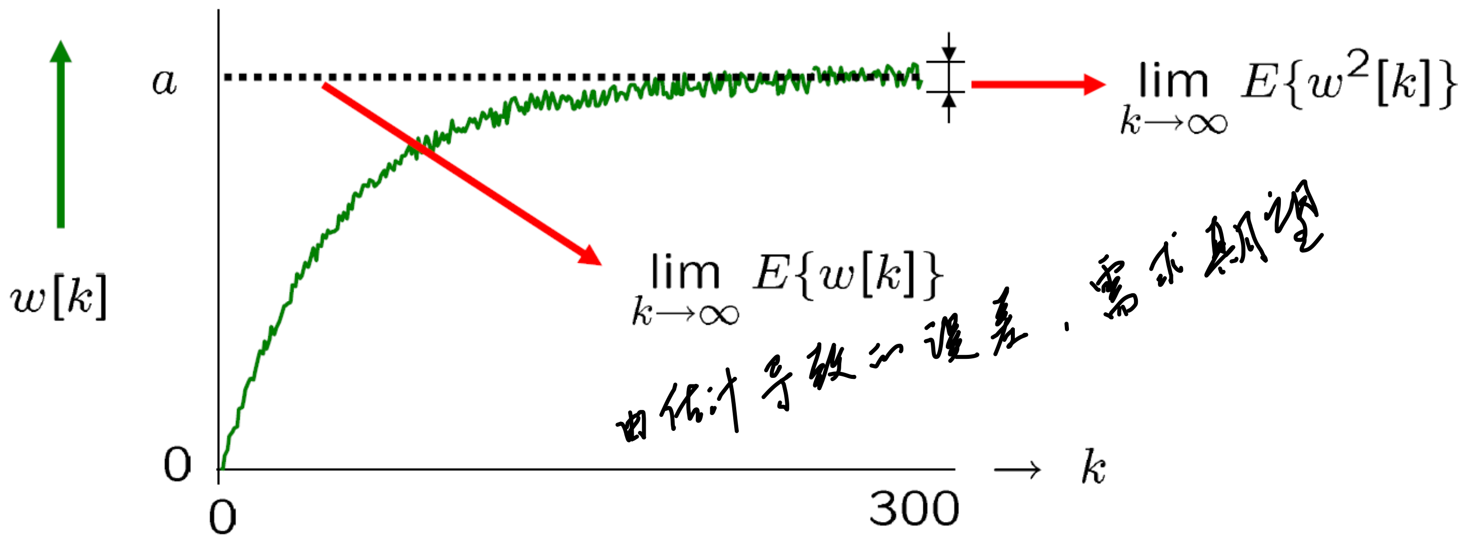
Acquisition and tracking:



Convergence LMS

Consequence of not using $E\{\cdot\}$?

Example: LMS, $N = 1$, $w[0] = 0$ $w_o = a$



Questions about convergence:

$$\lim_{k\rightarrow\infty} E\{w[k]\} = w_o = a \text{ and } \lim_{k\rightarrow\infty} E\{w^2[k]\} < \infty?$$

ADSP

Convergence LMS

First compare difference with (optimal) Wiener weights:

$$\underline{\mathbf{d}}[k] = \underline{\mathbf{w}}[k] - \underline{\mathbf{w}}_o \quad \text{with} \quad \underline{\mathbf{w}}_o = \mathbf{R}_x^{-1} \cdot \underline{\mathbf{r}}_{ex}$$

：减化前的负梯度

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha \left(\underline{\mathbf{x}}[k] e[k] - \underline{\mathbf{x}}[k] \underline{\mathbf{x}}^t[k] \underline{\mathbf{w}}[k] \right)$$

$$\underline{\mathbf{w}}[k+1] - \underline{\mathbf{w}}_o = \left(\mathbf{I} - 2\alpha \underline{\mathbf{x}}[k] \underline{\mathbf{x}}^t[k] \right) \underline{\mathbf{w}}[k] - \underline{\mathbf{w}}_o + 2\alpha \underline{\mathbf{x}}[k] e[k]$$

$$\underline{\mathbf{d}}[k+1] = \left(\mathbf{I} - 2\alpha \underline{\mathbf{x}}[k] \underline{\mathbf{x}}^t[k] \right) \underline{\mathbf{d}}[k] + 2\alpha \underline{\mathbf{x}}[k] r_{min}[k]$$

with $r_{min}[k] = e[k] - \underline{\mathbf{x}}^t[k] \underline{\mathbf{w}}_o$

Convergence LMS

Convergence in the mean:

$$E\{\underline{\mathbf{d}}[k+1]\} \;=\; E\left\{\left(\mathbf{I} - 2\alpha\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\right)\underline{\mathbf{d}}[k]\right\}$$
$$+2\alpha\underbrace{\left(E\{\underline{\mathbf{x}}[k]e[k]\} - E\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\}\underline{\mathbf{w}}_o\right)}_{0}$$

With independence assumption: $\underline{x}[k]$ 独立于 $\underline{d}[k]$

$$E\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\underline{\mathbf{d}}[k]\} \approx E\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^t[k]\}\cdot E\{\underline{\mathbf{d}}[k]\}$$

$$\Rightarrow \quad E\{\underline{\mathbf{d}}[k+1]\} \;=\; (\mathbf{I} - 2\alpha\mathbf{R}_x)\,E\{\underline{\mathbf{d}}[k]\}$$

Average convergence behaviour LMS same as SGD

$$0 < \alpha < 1/\lambda_{max} : \quad \lim_{k\to\infty} E\{\underline{\mathbf{w}}[k]\} = \underline{\mathbf{w}}_o \;;\; \tau_{av,i} \approx 1/2\alpha\lambda_i$$

$\Rightarrow$ Depends on coloration input process!

Convergence LMS

Mean-square convergence:

$$J_{LMS} \;=\; E\{r^2\} = E\{(e - \underline{\mathbf{w}}^t\underline{\mathbf{x}})(e - \underline{\mathbf{x}}^t\underline{\mathbf{w}})\}$$

With $\underline{\mathbf{d}} = \underline{\mathbf{w}} - \underline{\mathbf{w}}_o$ or $\underline{\mathbf{w}} = w_o + \underline{\mathbf{d}} \;\; \Rightarrow$

$$J_{LMS} = E\Big\{\Big((e - \underline{\mathbf{w}}_o^t\underline{\mathbf{x}}) - \underline{\mathbf{d}}^t\underline{\mathbf{x}}\Big)\Big((e - \underline{\mathbf{x}}^t\underline{\mathbf{w}}_o) - \underline{\mathbf{x}}^t\underline{\mathbf{d}}\Big)\Big\}$$

$$J_{LMS} = E\{(e - \underline{\mathbf{w}}_o^t\underline{\mathbf{x}})(e - \underline{\mathbf{x}}^t\underline{\mathbf{w}}_o)\} + E\{\underline{\mathbf{d}}^t\underline{\mathbf{x}}\underline{\mathbf{x}}^t\underline{\mathbf{d}}\}$$

$$-\underbrace{E\{\underline{\mathbf{d}}^t\underline{\mathbf{x}}(e - \underline{\mathbf{x}}^t\underline{\mathbf{w}}_o)\}}_{\to 0} - \underbrace{E\{(e - \underline{\mathbf{w}}_o^t\underline{\mathbf{x}})\underline{\mathbf{x}}^t\underline{\mathbf{d}}\}}_{\to 0}$$

Independence assumption $\;\Rightarrow$

$$E\{\underline{\mathbf{d}}^t\underline{\mathbf{x}}(e - \underline{\mathbf{x}}^t\underline{\mathbf{w}}_o)\} \;\approx\; E\{\underline{\mathbf{d}}^t\}\cdot\Big(E\{\underline{\mathbf{x}}e\} - E\{\underline{\mathbf{x}}\underline{\mathbf{x}}^t\}\underline{\mathbf{w}}_o\Big)$$

$$=\; E\{\underline{\mathbf{d}}^t\}\cdot(\underline{\mathbf{r}}_{ex} - \mathbf{R}_x\underline{\mathbf{w}}_o) = 0$$

Similar $E\{(e - \underline{\mathbf{w}}_o^t\underline{\mathbf{x}})\underline{\mathbf{x}}^t\underline{\mathbf{d}}\} \to 0$

上海科技大学
ShanghaiTech University

ADSP

Convergence LMS

Compare with MMSE expression

$$\Rightarrow \; J_{LMS} \;\approx\; \underbrace{E\{(e - \underline{\mathbf{w}}_o^t\underline{\mathbf{x}})(e - \underline{\mathbf{x}}^t\underline{\mathbf{w}}_o)\}}_{\text{Fixed}} + \underbrace{E\{\underline{\mathbf{d}}^t\underline{\mathbf{x}}\underline{\mathbf{x}}^t\underline{\mathbf{d}}\}}_{\text{Adaptive}}$$

Wiener error: $J_{min} = E\{(e-\underline{\mathbf{w}}_o^t\underline{\mathbf{x}})(e-\underline{\mathbf{x}}^t\underline{\mathbf{w}}_o)\} = E\{r_{min}^2\}$

Excess error:
$$\begin{aligned} J_{ex} &= E\{\underline{\mathbf{d}}^t\underline{\mathbf{x}}\underline{\mathbf{x}}^t\underline{\mathbf{d}}\} \approx E\{\underline{\mathbf{d}}^t E\{\underline{\mathbf{x}}\underline{\mathbf{x}}^t\}\underline{\mathbf{d}}\} \\ &= E\{\underline{\mathbf{d}}^t \mathbf{R}_x \underline{\mathbf{d}}\} \end{aligned}$$

Dynamic behaviour adaptive filter: $\tilde{J}[k] = \dfrac{J_{ex}[k]}{J_{min}[k]}$

Depends on coloration input process!

Convergence LMS

Conclusion convergence LMS:

Convergence gradient based algorithms, like SGD heavily relates on corolation input process and initiation of adaptive weights!

This also follows from rewriting gradient as:

$$\underline{\nabla} = -2\mathbf{R}_x \cdot \left(\mathbf{R}_x^{-1}\underline{\mathbf{r}}_{ex} - \underline{\mathbf{w}}[k]\right)$$

$\Rightarrow$ Gradient (=update) depends on coloration input

收敛过程由Rx决定

Solution:

Alternative that decorrelates input $\rightarrow$ Newton