

#### Moderní metody zpracování signálů

Optimální filtry ve smyslu kvadratické vzdálenosti

Zbyněk Koldovský



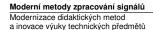
















# Minimalizace kvadratické vzdálenosti







# Vstup, výstup, chyba

- x[n] je vstupní, pro jednoduchost reálný, signál
- Signál zpracováváme FIR filtrem w délky L, výstup je

$$y[n] = w[0]x[n] + w[1]x[n-1] + \dots + w[L-1]x[n-L]$$
$$= \sum_{k=0}^{L-1} w[k]x[n-k].$$

Vektorový zápis

$$y[n] = \mathbf{w}^T \mathbf{x}_n$$

kde

$$\mathbf{x}_n = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[L-1] \end{bmatrix},$$







### Vstup, výstup, chyba

- Nechť d[n] je jiný signál, který chceme získat filtrováním signálu x[n].
- Chyba rozdílu

$$e[n] = d[n] - y[n]$$
  
=  $d[n] - \mathbf{w}^T \mathbf{x}_n$ .

 Filtr w můžeme hledat optimalizací nějakého kritéria, např. kvadratického

$$J_n(\mathbf{w}) = e[n]^2$$

$$= (d[n] - \mathbf{w}^T \mathbf{x}_n)^2 = d[n]^2 - 2d[n]\mathbf{w}^T \mathbf{x}_n + \mathbf{w}^T \mathbf{x}_n \mathbf{x}_n^T \mathbf{w}.$$







#### Gradient $J_n(\mathbf{w})$

Gradient  $J_n(\mathbf{w})$ , tj. vektor parciálních derivací podle jednotlivých složek  $\mathbf{w}$ , lze zapsat vektorově

$$\triangle J_n(\mathbf{w}) = -2\mathbf{x}_n d(n) + 2\mathbf{x}_n \mathbf{x}_n^T \mathbf{w}.$$

Zavedeme označení

$$\mathbf{R}_n = \mathbf{x}_n \mathbf{x}_n^T$$
  
 $\mathbf{p}_n = \mathbf{x}_n d[n],$ 

Gradient můžeme zapsat jako

$$\triangle J_n(\mathbf{w}) = -2\mathbf{p}_n + 2\mathbf{R}_n\mathbf{w}.$$







#### **Least Mean Square**

Položíme-li  $\triangle J_n(\mathbf{w})$  roven nule, dostáváme rovnici

$$\mathbf{R}_n\mathbf{w}=\mathbf{p}_n$$

- Matice  $\mathbf{R}_n$  má hodnost 1, takže nemá inverzi.
- Definujeme nové kritérium zprůměrováním  $J_n(\mathbf{w})$  přes interval n = 1, ..., N

$$J_{\text{LMS}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} J_n(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} e[n]^2.$$

Gradient  $J_{LMS}(\mathbf{w})$  je průměr gradientů  $J_n(\mathbf{w})$ 

$$\triangle J_{\text{LMS}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \triangle J_n(\mathbf{w})$$





Modernizace didaktických metod a inovace výuky technických předmětů

#### Least Mean Square

Takže

$$\triangle J_{\mathrm{LMS}}(\mathbf{w}) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}$$

kde

$$\mathbf{R} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T = \frac{1}{N} \sum_{n=1}^{N} \mathbf{R}_n$$
$$\mathbf{p} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n d[n] = \frac{1}{N} \sum_{n=1}^{N} \mathbf{p}_n.$$

Matice **R** již může mít inverzi (je-li interval n = 1, ..., Ndostatečně dlouhý). Proto

$$\mathbf{w}_{\mathrm{LMS}} = \mathbf{R}^{-1} \mathbf{p}$$
.







#### Modernizace didaktických metod a inovace výuky technických předmětů

#### Co isou prvky R?

Auto-kovariance x

$$\mathbf{R}_{ij} = \frac{1}{N} \sum_{n=1}^{N} x[n-i+1]x[n-j+1] = \widehat{C}_{xx}[i-j]$$

Platí tedy (až na koncové efekty)

$$\mathbf{R} = \begin{pmatrix} \widehat{C}_{xx}[0] & \widehat{C}_{xx}[1] & \widehat{C}_{xx}[2] & \dots & \widehat{C}_{xx}[L-1] \\ \widehat{C}_{xx}[1] & \widehat{C}_{xx}[0] & \widehat{C}_{xx}[1] & \dots & \widehat{C}_{xx}[L-2] \\ \widehat{C}_{xx}[2] & \widehat{C}_{xx}[1] & \widehat{C}_{xx}[0] & \dots & \widehat{C}_{xx}[L-3] \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \widehat{C}_{xx}[L-1] & \widehat{C}_{xx}[L-2] & \widehat{C}_{xx}[L-3] & \dots & \widehat{C}_{xx}[0] \end{pmatrix}$$







#### Co jsou prvky R?

Matici R lze též zapsat jako

$$\mathbf{R} = \frac{1}{N} \mathbf{X} \mathbf{X}^T$$

kde

$$\mathbf{X} = \begin{bmatrix} x[1] & x[2] & \dots & x[N] & 0 & 0 & \dots & 0 \\ 0 & x[1] & \dots & x[N-1] & x[N] & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & x[1] & \dots & x[N-L] & \dots & \dots & x[N] \end{bmatrix}$$

R je toeplitzovská (diagonálně konstatní), symetrická a positivně semidefinitní (má reálná nezáporná vlastní čísla).







### Co isou prvky p?

Cross-kovariance

$$\mathbf{p}_{j} = \frac{1}{N} \sum_{n=1}^{N} x[n-j+1]d[n]$$

Lze zapsat

$$\mathbf{p} = \frac{1}{N} \mathbf{Xd}$$

kde

$$\mathbf{d} = [d[1], d[2], \dots, d[N]]^T$$







### Řešení soustavy $Rw_{LMS} = p$

- Řešení soustavy  $\mathbf{Rw}_{LMS} = \mathbf{p}$  vyžaduje  $\mathcal{O}(L^3)$  operací
- Levinson-Durbinův algoritmus využívá toeplitzovskou strukturu **R** a počítá pomocí rekurze řešení pro všechny délky filtru 1,..., L přičemž složitost je  $\mathcal{O}(L^2)$ .
- Rovněž paměťové nároky jsou řádově nižší, protože neukládáme do paměti celou matici R, pouze jeden její řádek (sloupec).





#### Příklad: vzájemné zpoždění signálů

 Předpoklad: d je zpožděný signál x plus neznámá změna (šum, filtr,...)

$$d[n] = x[n-D] + \nu[n]$$

- Hledáme filtr takový, aby h \* x byl co nejvíce podobný d[n].
- Ideální výsledek:

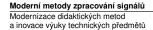
$$h[n] = \delta[n - D]$$

Jinými slovy h zpozdí x o D vzorků.











Část II

# Wienerův filtr







#### Wienerův filtr

- Předpoklad: x a d jsou slabě stacionární (jejich auto-kovarianční funkce (a tedy spektrum) se v čase nemění).
- Kritérium definujeme jako

$$J(\mathbf{w})=E[e[n]^2].$$

Díky stacionaritě je toto kritérium nezávislé na *n* (na čase).

Gradient  $J(\mathbf{w})$  položíme roven nule, takže

$$\mathbf{w}_{\text{wiener}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) = \mathbf{R}^{-1}\mathbf{p},$$

kde

$$\mathbf{R} = E\left[\mathbf{x}_{n}\mathbf{x}_{n}^{T}\right]$$
$$\mathbf{p} = E\left[\mathbf{x}_{n}d[n]\right].$$



Porovnejte definice R a p pro LMS a Wienerův filtr

$$\mathbf{R} = E\left[\mathbf{x}_n\mathbf{x}_n^T\right] \qquad \mathbf{R} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$$

Jsou-li x a d stacionární, pak

$$\mathbf{w}_{\mathrm{LMS}} \overset{N \to +\infty}{\longrightarrow} \mathbf{w}_{\mathrm{wiener}}$$

Pokud signály nejsou stacionární, nelze nic obecně usoudit.