

Filtragem Adaptativa por Quadrados Mínimos.

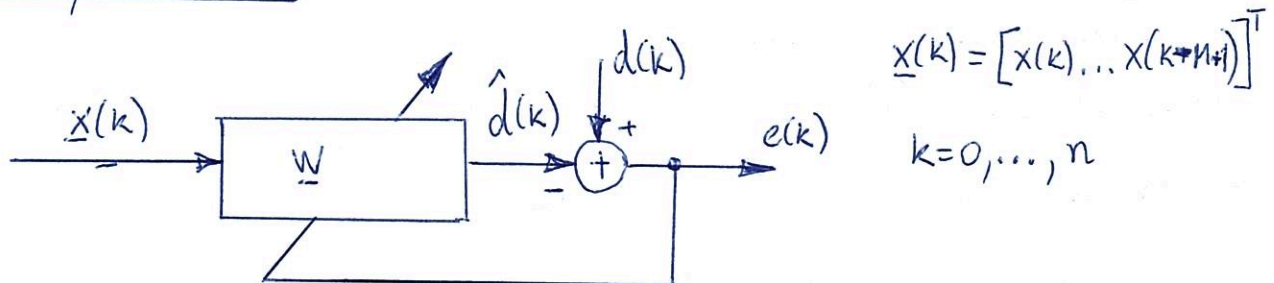
A teoria de filtragem de Wiener caracteriza-se pelo emprego de informações estatísticas a respeito dos sinais envolvidos: a entrada e o sinal desejado são considerados como sendo processos aleatórios; o critério MSE representa uma média estatística, assim como as medidas de correlação R_x e p_{xd} ; a aproximação estocástica substitui as médias estatísticas por estimativas instantâneas, tornando possível o desenvolvimento de técnicas adaptativas, como o LMS.

Em muitos cenários práticos, porém, o uso de informações estatísticas é utópico, particularmente quando o modelo de geração dos dados é desconhecido. É possível, então, considerar uma abordagem alternativa que é mais "orientada por dados", i.e., que se baseia em uma dada realização temporal dos sinais, de modo que o critério de filtragem poderia ser construído com base nas amostras disponíveis, por exemplo, explorando médias temporais, as quais podem ser prontamente calculadas.

O objetivo, portanto, não mais corresponde ao filtro ótimo com respeito a uma medida que leva em consideração as médias estatísticas (ensemble) subjacentes aos processos aleatórios envolvidos, mas sim ao filtro ótimo para os dados disponíveis em um certo período de observação.

⇒ O método dos quadrados mínimos (least squares, LS) surge como alternativa à estimação baseada no MSE quando se tem acesso apenas a um conjunto de medidas (uma única realização do processo).

* Solução LS



$\underline{w}^{(n)}$ representa o vetor (ótimo) de coeficientes para dados considerados até o instante n . O objetivo será minimizar a soma dos erros quadráticos associados a cada instante, i.e., $\sum e^2(i)$, onde $e(i) = d(i) - \underline{w}^T(i) \underline{x}(i)$.

A fim de levar em conta possíveis não-estacionariedades, aplica-se uma janela exponencial nos dados recebidos.

Em outras palavras, trabalhe - se com uma soma ponderada dos erros quadráticos:

$$J_{LS}(\underline{w}(n)) = \sum_{k=0}^n \lambda(k) e^2(k) = \sum_{k=0}^n \lambda(k) \{d(k) - \underline{w}^T(n) \underline{x}(k)\}^2$$

Usualmente, $\lambda(k)$ assume a forma de um fator de esquecimento, atenuando mais fortemente as amostras mais antigas: $\lambda(k) = \lambda^{n-k}$.

$$\text{Assim, } J_{LS}(\underline{w}(n)) = \sum_{k=0}^n \lambda(k) \{d(k) - \underline{w}^T(n) \underline{x}(k)\}^2$$

$$= \sum_{k=0}^n \lambda(k) \{d(k) - \underline{w}^T(n) \underline{x}(k)\} \{d(k) - \underline{w}^T(n) \underline{x}(k)\}^T$$

$$= \sum_{k=0}^n \lambda(k) \{d^2(k) - d(k) \underline{x}^T(k) \underline{w}(n) - d(k) \underline{w}^T(n) \underline{x}(k) + \underline{w}^T(n) \underline{x}(k) \underline{x}^T(k) \underline{w}(n)\}$$

Derivando com respeito a $\underline{w}^T(n)$:

$$\frac{\partial J_{LS}(\underline{w}(n))}{\partial \underline{w}^T(n)} = -2 \sum_{k=0}^n \lambda(k) d(k) \underline{x}(k) + 2 \sum_{k=0}^n \lambda(k) \underline{x}(k) \underline{x}^T(k) \underline{w}(n)$$

Igualando a zero, obtemos:

$$2 \sum_{k=0}^n \lambda(k) \underline{x}(k) d(k) = 2 \underline{w}(n) \sum_{k=0}^n \lambda(k) \underline{x}(k) \underline{x}^T(k)$$

$$\text{Seja } R_D(n) = \sum_{k=0}^n \lambda^{n-k} \underline{x}(k) \underline{x}^T(k) \quad \text{e} \quad p_D(n) = \sum_{k=0}^n \lambda^{n-k} \underline{x}(k) d(k)$$

Então; $\underline{w}(n) = R_D^{-1}(n) p_D(n)$

OBSERVAÇÕES :

* $R_D(n)$ e $p_D(n)$ são estimativas temporais da matriz de auto correlação e do vetor de correlação cruzada, respectivamente.

* A solução exposta na equação acima é exata e ótima (no sentido de quadrados mínimos) para o conjunto de dados que se encontra disponível.

* Ao mesmo tempo, esta solução pode ser vista como uma tentativa de estimar a própria solução de Wiener, que pertence ao "mundo ideal" das médias estatísticas.

Agora, passaremos à derivação de um procedimento que compute de maneira recursiva (no tempo) a solução de quadrados mínimos.

* Algoritmo dos Quadrados Mínimos Recursivo (RLS)

• Objetivo: encontrar uma solução recursiva na forma

$$\underline{w}(n) = \underline{w}(n+1) + \Delta \underline{w}(n).$$

Inicialmente, vamos expressar $p_D(n)$ e $R_D(n)$ em função de $p_D(n-1)$ e $R_D(n-1)$, respectivamente.

$$p_D(n) = \sum_{k=0}^n \lambda^{n-k} \underline{x}(k) d(k)$$

$$= \sum_{k=0}^{n-1} \lambda^{n-k} \underline{x}(k) d(k) + \underline{x}(n) d(n) = \lambda \sum_{k=0}^{n-1} \lambda^{(n-1)-k} \underline{x}(k) d(k) + \underline{x}(n) d(n)$$

Como $p_D(n-1) = \sum_{k=0}^{n-1} \lambda^{(n-1)-k} \underline{x}(k) d(k)$, então

$$p_D(n) = \lambda p_D(n-1) + d(n) \underline{x}(n)$$

Semelhantemente, vamos expressar $R_D(n)$ em função de $R_D(n-1)$:

$$\begin{aligned} R_D(n) &= \sum_{k=0}^n \lambda^{n-k} \underline{x}(k) \underline{x}^T(k) \\ &= \sum_{k=0}^{n-1} \lambda^{n-k} \underline{x}(k) \underline{x}^T(k) + \underline{x}(n) \underline{x}^T(n) \\ &= \lambda \sum_{k=0}^{n-1} \lambda^{(n-1)-k} \underline{x}(k) \underline{x}^T(k) + \underline{x}(n) \underline{x}^T(n) \\ &= \lambda R_D(n-1) + \underline{x}(n) \underline{x}^T(n) \end{aligned}$$

Seria mais interessante, porém, obter uma fórmula que compute recursivamente a inversa da matriz R_D , uma vez que a solução LS depende de R_D^{-1} . Isto é possível se recorrermos ao lema de inversão de matrizes:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Fazendo a seguinte associação:

$A = \lambda R_D(n-1)$, $B = \underline{x}(n)$, $C = I_1$ (identidade 1×1) e $D = \underline{x}^T(n)$, obtemos:

$$\left(\lambda R_D(n-1) + \underline{x}(n) \cdot I_1 \cdot \underline{x}^T(n) \right)^{-1} = R_D^{-1}(n)$$

$$R_D^{-1}(n) = \frac{1}{\lambda} R_D^{-1}(n-1) - \frac{1}{\lambda} R_D^{-1}(n-1) \underline{x}(n) \left(1 + \underline{x}^T(n) \frac{1}{\lambda} R_D^{-1}(n-1) \underline{x}(n) \right)^{-1} \underline{x}^T(n) \frac{1}{\lambda} R_D^{-1}(n-1)$$

Seja $P(n) = R_D^{-1}(n)$, então, podemos escrever:

$$P(n) = \lambda^{-1} P(n-1) - g(n) \underline{x}^T(n) \lambda^{-1} P(n-1),$$

onde $g(n) = \lambda^{-1} P(n-1) \underline{x}(n) [1 + \underline{x}^T(n) \lambda^{-1} P(n-1) \underline{x}(n)]^{-1}$ é o vetor de ganho.

Vamos, agora, colocar $g(n)$ numa forma mais conveniente.

Note que o termo $[1 + \underline{x}^T(n) \lambda^{-1} P(n-1) \underline{x}(n)]^{-1}$ é um escalar:

$$g(n) \cdot (1 + \underline{x}^T(n) \lambda^{-1} P(n-1) \underline{x}(n)) = \lambda^{-1} P(n-1) \underline{x}(n)$$

$$g(n) + g(n) \underline{x}^T(n) \lambda^{-1} P(n-1) \underline{x}(n) = \lambda^{-1} P(n-1) \underline{x}(n)$$

$$g(n) = \underbrace{\lambda^{-1} [P(n-1) - g(n) \underline{x}^T(n) P(n-1)]}_{P(n)} \underline{x}(n)$$

Então, $g(n) = P(n) \underline{x}(n)$

Esteamos, agora, prontos para obter a recursão para $\underline{w}(n)$

$$\begin{aligned} \underline{w}(n) &= R_D^{-1}(n) \underline{p}_D(n) \\ &= P(n) \underline{p}_D(n) \end{aligned}$$

$$= P(n) (\lambda \underline{p}_D(n-1) + d(n) \underline{x}(n))$$

Substituindo $P(n)$:

$$\underline{w}(n) = \lambda [\lambda^{-1} P(n-1) - g(n) \underline{x}^T(n) \lambda^{-1} P(n-1)] \underline{p}_D(n-1) + d(n) \underbrace{P(n) \underline{x}(n)}_{g(n)}$$

$$= P(n-1) \underline{p}_D(n-1) - g(n) \underline{x}^T(n) P(n-1) \underline{p}_D(n-1) + d(n) g(n)$$

$$= P(n-1) \underline{p}_D(n-1) + g(n) [d(n) - \underline{x}^T(n) P(n-1) \underline{p}_D(n-1)]$$

Como $\underline{w}(n-1) = P(n-1)p_d(n-1)$,

$$\underline{w}(n) = \underline{w}(n-1) + g(n) [d(n) - \underline{x}^T(n) \underline{w}(n-1)]$$

Definindo $E(n) = d(n) - \underline{x}^T(n) \underline{w}(n-1)$ como sendo o erro a priori (antes da atualização do vetor de coeficientes), obtemos:

$$\underline{w}(n) = \underline{w}(n-1) + \underbrace{g(n) E(n)}_{\Delta \underline{w}(n)}$$

o fator de correção é diretamente proporcional ao erro (a priori) e ao vetor de ganho, o qual controla a sensibilidade através do parâmetro λ .

Em certo sentido, a equação de atualização do vetor de coeficientes lembra aquela associada ao método de Newton, pois aparecem tanto o gradiente (na forma do produto $\underline{x}(n)E(n)$) quanto a inversa da matriz de autocorrelação (uma estimativa temporal e recursiva).

O algoritmo RLS pode ser resumido da seguinte forma:

Inicialização: $\underline{w}(-1) = [0 \dots 0]^T$

$$\underline{x}(-1) = [0 \dots 0]^T$$

$$P(-1) = \delta^{-1} \mathbf{I} \quad \longrightarrow \quad \delta = \begin{cases} \text{constante positiva pequena} \\ \text{para alta SNR} \\ \text{constante positiva grande} \\ \text{para SNR baixa} \end{cases}$$

Para $n \geq 0$, faça

$$E(n) = d(n) - \underline{x}^T(n) \underline{w}(n-1)$$

$$g(n) = P(n-1) \underline{x}(n) \cdot (\lambda + \underline{x}^T(n) P(n-1) \underline{x}(n))^{-1}$$

$$P(n) = \lambda^{-1} P(n-1) - g(n) \underline{x}^T(n) \lambda^{-1} P(n-1)$$

$$\underline{w}(n) = \underline{w}(n-1) + g(n) E(n)$$

OBS: • a recursão p/ $P(n)$ segue uma equação algébrica de Riccati.

• a otimização de mínimos quadrados que nos leva ao algoritmo recursivo acima visa minimizar uma função custo baseada em $e(n) = d(n) - \underline{w}^T(n) \underline{x}(n)$, que representa o erro a posteriori, e não baseada em $E(n)$ (erro a priori).

Algumas propriedades do RLS:

* o vetor de erro ponderado $E(n) = [e(n) \lambda^{1/2} e(n-1) \dots \lambda^{n/2} e(0)]^T$ é ortogonal a todas as linhas da matriz $X(n) = [\underline{x}(n) \lambda^{1/2} \underline{x}(n-1) \dots \lambda^{n/2} \underline{x}(0)]$

Em outras palavras, a saída $y(n)$ corresponde à projeção do vetor de sinal desejado ponderado no subespaço gerado pelas linhas de $X(n)$.

* A solução de mínimos quadrados tende à solução de Wiener se os sinais envolvidos forem estacionários e ergódicos.

* Convergência (na média) dos coeficientes $\underline{w}(n)$ para o vetor ótimo \underline{w}_0 .

* Análises de erro mínimo, erro em excesso (desajuste) e taxa de convergência consideram sempre $\lambda \rightarrow 1$.

* Insensibilidade com respeito à variação no espalhamento dos autovalores da matriz de autocorrelação dos dados de entrada.