Família PCM — Predição Linear Adaptativa

Comunicações I

Thadeu L. B. Dias

UFRJ

ToC

Recap

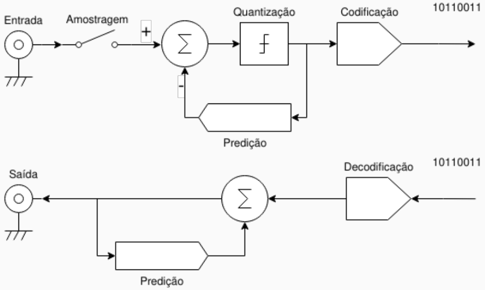

Figure 1: Codificador e decodificador do DPCM com predição linear.

$$\boldsymbol{w} = \boldsymbol{R}_M^{-1}\boldsymbol{r}_M.$$

Predição ‘Fixa’

Aqui, o preditor ótimo foi projetado já sabendo a função de autocorrelação $R_{MM}[k]$, de $M[n]$.

Saber a estatística de um sinal geralmente requer que nós já conheçamos o próprio sinal a ser transmitido e que *a estatística não varie com o tempo*. Esse segundo é possivelmente o mais grave de se assumir.

Como lidar com sinais com estatísticas

1. Desconhecidas a priori.
2. Variantes no tempo.

Estimação Linear Adaptativa

Estimador Linear Adaptativo

A idéia da predição linear adaptativa é basicamente estimar as estatísticas *a medida* que o sinal é medido.
Isso significa que os coeficientes do filtro, anteriormente $\boldsymbol{w}^*$, variam com o tempo, $\boldsymbol{w}[n]$, a medida que novas amostras do sinal são geradas e transmitidas.

Como fazer?

Estimador linear causal

Naturalmente, ainda teremos um estimador linear causal, similar ao filtro de Wiener, porém, os coeficientes serão variantes no tempo:

$$\hat{M}[n] = w_1[n]M[n-1] + w_2[n]M[n-2] + \cdots + w_N[n]M[n-N]. \quad (1)$$

Como 'atualizar' os coeficientes $w_i[n], \ i \in \{1, \cdots N\}$ de forma que:

1. Os coeficientes convirjam para o estimador ótimo $\boldsymbol{w}^*[n]$.
2. O processo lide com variações da estatística do sinal.
3. O processo seja estável.

Solução ótima instantânea

Podemos equacionar o problema de otimização para minimizar a potência média do erro entre $M[n]$ e $\hat{M}[n]$ da mesma forma da estatística fixa, porém, com o filtro variável:

$$\boldsymbol{w}^*[n+1] = \arg\min_{\boldsymbol{w}[n]} \mathbb{E}[(M[n] - \hat{M}[n])^2]$$

A solução para esse caso (não-estácionário) ainda envolve a função de autocorrelação $R_{MM}[n,k]$, que não possuímos.

Note que o valor ótimo é para o valor do filtro no passo *seguinte*, pois devemos preservar a causalidade!

Se pudermos assumir que a estatística de $M[n]$ varia *suavemente*, então sem muitos problemas, $R_{MM}[n-1,k]$ será uma aproximação razoável para $R_{MM}[n,k]$, ou ainda, $\boldsymbol{w}^*[n-1]$ será uma boa aproximação para $\boldsymbol{w}^*[n]$.

A estimativa será continuamente atualizada, mas não precisamos de uma representação explícita de $R_{MM}[n,k]$.
Como vamos assumir que a autocorrelação do sinal varia lentamente, faz sentido que $\boldsymbol{w}[n]$ varie lentamente também.
Isso significa que teremos uma regra de atualização na forma

$$\boldsymbol{w}[n] = \boldsymbol{w}[n-1] + \Delta\boldsymbol{w}[n], \tag{2}$$

bastando agora que

$$\lim_{n\to\infty} \boldsymbol{w}[n] = \boldsymbol{w}^*[n].$$

Descida pelo gradiente

Como não possuímos o *ensemble* do sinal, nosso problema de otimização será posto exclusivamente nas amostras da realização que possuímos, $m[n]$, de forma que teremos algo da forma

$$\boldsymbol{w}^*[n+1] = \arg\min_{\boldsymbol{w}[n]} \left(m[n] - \hat{m}[n]\right)^2.$$

Derivando e equacionando para $\boldsymbol{w}^*$, teremos

$$\boldsymbol{m}[n]\boldsymbol{m}^{\mathrm{T}}[n]\boldsymbol{w}^*[n+1] - m[n]\boldsymbol{m}[n] = \boldsymbol{0}.$$

Qual a diferença desse problema para o anterior (com a expectativa)?

Expansão linear

Se uma função $f : \mathbb{R}^N \to R$ é (infinitamente) continuamente diferenciável, então ela admite uma série de Taylor na seguinte forma

$$f(\boldsymbol{x} + \Delta \boldsymbol{x}) = f(\boldsymbol{x}) + Df(\boldsymbol{x})(\Delta \boldsymbol{x}) + \ldots + \frac{1}{i!} D^i f(\boldsymbol{x})(\Delta \boldsymbol{x}, \ldots, \Delta \boldsymbol{x}) + \ldots,$$

onde $Df(\boldsymbol{x})$ é o *gradiente* avaliado em $\boldsymbol{x}$, $D^2 f(\boldsymbol{x})$ é a *Hessiana* avaliada em $\boldsymbol{x}$, e $D^i f(\boldsymbol{x})$ é o tensor de derivadas de i-ésima ordem, e assim por diante.

O interessante é que como as derivadas sempre são funções multilineares avaliadas em $(\Delta \boldsymbol{x}, \dots, \Delta \boldsymbol{x})$, os termos são do tipo

$$D^n f(\boldsymbol{x})(\Delta \boldsymbol{x}, \dots, \Delta \boldsymbol{x}) = \sum_{i_1} \cdots \sum_{i_n} d_{i_1 \dots i_n} \Delta x_{i_1} \dots \Delta x_{i_n},$$

de forma que se fizermos $\|\Delta \boldsymbol{x}\| \to 0$, podemos ignorar os termos de ordem superior, resultando em

$$f(\boldsymbol{x} + \Delta \boldsymbol{x}) \approx f(\boldsymbol{x}) + Df(\boldsymbol{x})(\Delta \boldsymbol{x}).$$

Direção de descida máxima

Qual a direção que minimiza *maximamente* a expansão linear? Se

$$f(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx f(\boldsymbol{x}) + \boldsymbol{g}^{\mathrm{T}}\Delta\boldsymbol{x},$$

uma forma de garantir que o segundo termo seja menor ou igual a zero é fazer $\Delta\boldsymbol{x} = -\boldsymbol{g}$:

$$\boldsymbol{g}^{\mathrm{T}}(-\boldsymbol{g}) = -\|\boldsymbol{g}\|^2 \leq 0.$$

Naturalmente, a aproximação só vale para $\|\Delta\boldsymbol{x}\| \to 0$. Isso significa, na prática, que para minimizar f, podemos caminhar com passos 'pequenos', sempre na direção oposta ao gradiente avaliado no ponto atual, ou seja,

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} - \eta\boldsymbol{g} \quad \eta \approx 0^+ \tag{3}$$

Experimento numérico: GD

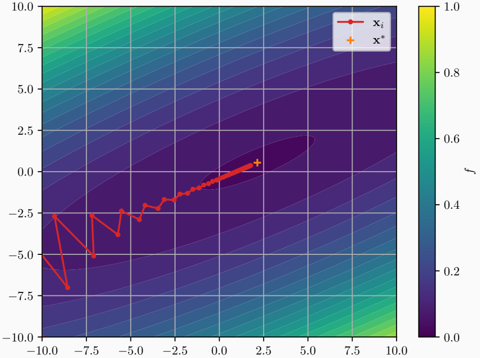

Figure 2: Evolução da otimização por descida do gradiente.

Observe como $\boldsymbol{x}_i$ se aproxima sucessivamente do valor ótimo.
Para se pensar: e se f for multimodal (mínimos locais)?

Algoritmo LMS

O processo de descida pelo gradiente pode ser generalizado para o nosso problema.

$$\boldsymbol{w}^*[n+1] = \arg\min_{\boldsymbol{w}[n]} \left(m[n] - \hat{m}[n]\right)^2.$$

Se $e[n] = m[n] - \hat{m}[n]$, então o gradiente da função é justamente

$$\partial_{\boldsymbol{w}[n]}\left(m[n] - \hat{m}[n]\right)^2 = -2e[n]\boldsymbol{m}[n]! \tag{4}$$

Aplicando o esquema de minimização pelo gradiente na atualização do filtro gera a equação do algoritmo LMS (Least Mean Squares):

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] + 2\eta e[n]\boldsymbol{m}[n]. \tag{5}$$

O algoritmo LMS é amplamente utilizado, talvez sendo o mais comum dos algoritmos de filtragem adaptativa.

Sem dúvida, sua maior vantagem é a simplicidade de implementação (literalmente $N+1$ multiplicações por etapa de ajuste), enquanto os algoritmos que o superam em algum critério geralmente tem equações de estado significativamente mais complexas.

O LMS não é usado apenas como preditor: basicamente, com um sinal de entrada qualquer $x[n]$ e um sinal 'desejado' $d[n]$ (com algum grau de correlação cruzada), é possível realizar diferentes tarefas:

- Identificação de sistemas (controle, estimador de estados)
- Inversão de sistemas (equalizador de canal)
- Cancelamento de ruído
- Cancelamento de eco
- many more...

Parâmetros para o designer

O filtro LMS como preditor (ou em outras aplicações) tem apenas 2 parâmetros de escolha:

- Ordem do filtro (número de coeficientes)
- Passo de aprendizado (η)

Os valores ótimos naturalmente dependem do sinal e da aplicação, porém de forma geral, o comportamento do algoritmo com relação aos parâmetros costuma ser

- N maior $\Rightarrow$: menor erro, mais complexo (multiplicações);
- η maior: convergência mais rápida, porém maior erro em fase permanente e chance de instabilizar;

Experimento numérico: LMS

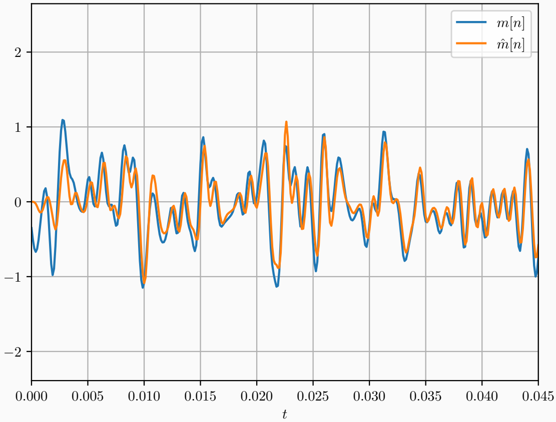

Figure 3: Exemplo do ‘rastreio’ de um filtro LMS em operação.

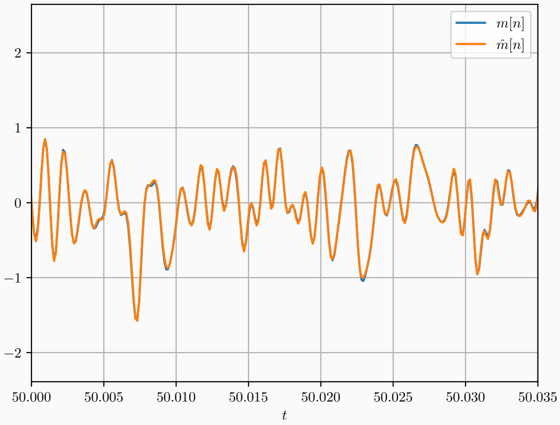

Figure 4: Exemplo do filtro LMS em já convergido (estacionário).

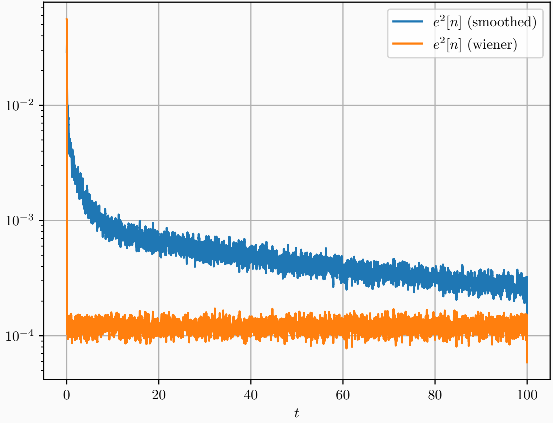

Figure 5: Exemplo da convergência de um filtro LMS ao longo do tempo.

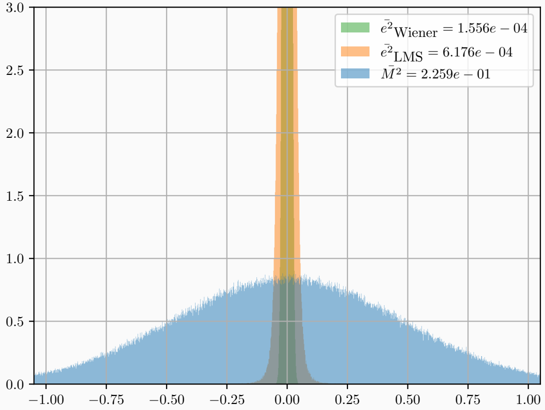

Figure 6: Distribuições do sinal/erro para a simulação.

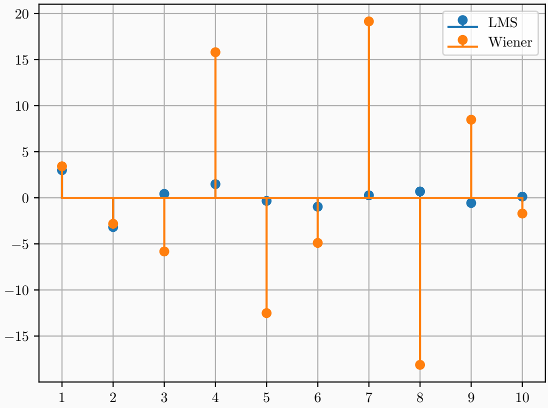

Figure 7: Comparação dos coeficientes entre o LMS e filtro de Wiener.