

Família PCM — Predição Linear

Comunicações I

Thadeu L. B. Dias

UFRJ

1. PCM Diferencial
2. Estimação Linear

Recap

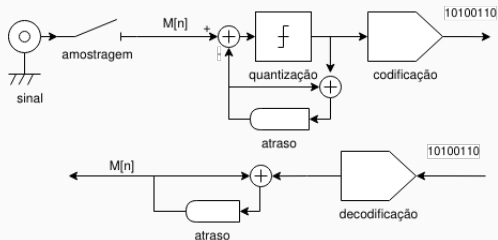


Figure 1: Codificador e decodificador do DPCM feedback.

Para sinais em que $M[n]$ e $M[n - 1]$ são correlacionados, o sinal diferença tem valor quadrático médio menor:

Distribuição mais justa \Rightarrow melhor taxa/distorção.

Na prática, tirar a diferença entre $M[n]$ e $M[n - 1]$ é similar a tirar a diferença entre o valor real e uma *predição*; para sinais ‘suaves’, o valor anterior é uma predição razoável:

- A diferença entre os valores não costuma ser alta.
- O valor anterior é causal, podendo ser usado no receptor com a estrutura feedback.

É possível melhorar essa estrutura, gerando um preditor melhor?

Estimação Linear

A idéia de um PCM com predição linear é estimar a próxima amostra com base apenas em amostras anteriores; Dessa forma, teremos uma predição $\hat{M}[n]$, para a amostra $M[n]$, e o que será transmitido é justamente a diferença $E[n] = M[n] - \hat{M}[n]$.

Se meu preditor for ‘bom’, então o erro/diferença será altamente concentrado em torno de zero, condição que nos permite alta eficiência em taxa/distorção!

Estimador linear causal

Nosso estimador deverá ser replicado no receptor, para que seja possível usar a estrutura feedback.

Ao invés de usar apenas a última amostra como ‘preditor’, agora usaremos uma estrutura que realiza uma *combinação linear* das ultimas N amostras para a estimativa:

$$\hat{M}[n] = w_1 M[n-1] + w_3 M[n-3] + \cdots + w_N M[n-N]. \quad (1)$$

Basta agora, encontrar os valores ótimos para os coeficientes w_i , $i \in \{1, \cdots N\}$.

Usaremos aqui a notação

$$\begin{aligned} \mathbf{w} &= [w_1 \ w_2 \ \cdots \ w_N]^T \\ \mathbf{m}[n] &= [M[n-1] \ M[n-2] \ \cdots \ M[n-N]]^T, \end{aligned}$$

de forma que $\hat{M}[n] = \mathbf{w}^T \mathbf{m}[n]$.

Vamos então equacionar o problema de otimização para minimizar a potência média do erro entre $M[n]$ e $\hat{M}[n]$:

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

Vamos então verificar a derivada do valor esperado em função do vetor \mathbf{w} :

$$\begin{aligned}\partial_{\mathbf{w}}\mathbb{E}[(M[n] - \hat{M}[n])^2] &= \mathbb{E}[\partial_{\mathbf{w}}(M[n] - \hat{M}[n])^2], \\ &= \mathbb{E}[2(M[n] - \hat{M}[n])\partial_{\mathbf{w}}\hat{M}[n]],\end{aligned}$$

como $\hat{M}[n] = \mathbf{w}^T \mathbf{m}[n]$,

$$\begin{aligned}&= 2\mathbb{E}[(M[n] - \mathbf{w}^T \mathbf{m}[n])\mathbf{m}[n]], \\ &= 2\mathbb{E}[M[n]\mathbf{m}[n] - \mathbf{m}[n]\mathbf{m}^T[n]\mathbf{w}].\end{aligned}$$

Vamos observar o primeiro termo da expectativa:

$$\mathbb{E}[M[n]\boldsymbol{m}[n]].$$

Expandindo o termo vetorial em suas componentes, temos o seguinte:

$$\mathbb{E}[M[n]\boldsymbol{m}[n]] = \mathbb{E} \left[\begin{bmatrix} M[n] \cdot M[n-1] \\ M[n] \cdot M[n-2] \\ M[n] \cdot M[n-3] \\ \vdots \\ M[n] \cdot M[n-N] \end{bmatrix} \right]$$

Para $M[n]$ WSS, os valores esperados dos produtos são diretamente os valores assumidos pela função de autocorrelação $C_{MM}[k]$ para os respectivos deslocamentos:

$$\mathbb{E}[M[n]\mathbf{m}[n]] = \mathbb{E} \left[\begin{bmatrix} M[n] \cdot M[n-1] \\ M[n] \cdot M[n-2] \\ M[n] \cdot M[n-3] \\ \vdots \\ M[n] \cdot M[n-N] \end{bmatrix} \right] = \begin{bmatrix} C_{MM}[1] \\ C_{MM}[2] \\ C_{MM}[3] \\ \vdots \\ C_{MM}[N] \end{bmatrix} = \mathbf{r}_M.$$

Para facilitar a nossa notação, vamos denotar esse vetor de correlação em \mathbb{R}^N por \mathbf{r}_M .

Vamos observar agora o segundo termo da expectativa:

$$\mathbb{E}[\mathbf{m}[n]\mathbf{m}^T[n]]\mathbf{w}.$$

Expandindo o termo matricial em suas componentes, temos o seguinte:

$$\mathbb{E} \left[\begin{bmatrix} M[n-1]M[n-1] & M[n-1]M[n-2] & \cdots & M[n-1]M[n-N] \\ M[n-2]M[n-1] & M[n-2]M[n-2] & \cdots & M[n-2]M[n-N] \\ M[n-3]M[n-1] & M[n-3]M[n-2] & \cdots & M[n-3]M[n-N] \\ \vdots & \vdots & \ddots & \vdots \\ M[n-N]M[n-1] & M[n-N]M[n-2] & \cdots & M[n-N]M[n-N] \end{bmatrix} \right].$$

Novamente, identificaremos os termos de produto com a função de autocorrelação:

$$\mathbb{E}[M[n-i]M[n-j]] = C_{MM}[i-j].$$

O resultado é uma matriz Toeplitz simétrica, inteiramente descrita em termos da função de autocorrelação:

$$\mathbb{E} \left[\begin{bmatrix} C_{MM}[0] & C_{MM}[1] & \cdots & C_{MM}[N-1] \\ C_{MM}[1] & C_{MM}[0] & \cdots & C_{MM}[N-2] \\ C_{MM}[2] & C_{MM}[1] & \cdots & C_{MM}[N-3] \\ \vdots & \vdots & \ddots & \vdots \\ C_{MM}[N-1] & C_{MM}[N-2] & \cdots & C_{MM}[0] \end{bmatrix} \right] = \mathbf{R}_M.$$

Para facilitar a nossa notação, vamos denotar essa matriz de correlação em \mathbb{R}^N por \mathbf{R}_M .

Finalmente, podemos escrever o gradiente em função da matriz \mathbf{R}_M e os vetores \mathbf{w} e \mathbf{r}_M :

$$\partial_{\mathbf{w}} \mathbb{E}[(M[n] - \hat{M}[n])^2] = 2(\mathbf{r}_M - \mathbf{R}_M \mathbf{w}),$$

e naturalmente, para \mathbf{R}_M não-singular, o valor extremo da função é encontrado justamente com

$$\mathbf{w}^* = \mathbf{R}_M^{-1} \mathbf{r}_M. \quad (2)$$

O estimador assim derivado é chamado na literatura de *Filtro de Wiener*.

Exemplo de aplicação

Suponha que desejamos transmitir um trecho de um sinal conhecido a priori pelo codificador (mas não pelo decodificador).

Uma forma eficiente de se realizar essa tarefa é justamente usando o filtro de Wiener.

Podemos computar a autocorrelação do sinal no trecho correspondente, $C_M M[k]$, e resolver a equação de Wiener obtendo os coeficientes do preditor linear ótimo \mathbf{w}^* .

Para transmitir o sinal, primeiro quantizamos e codificamos os coeficientes w_i para que o receptor possa reproduzir as predições; em seguida, iterativamente computamos $\hat{M}[n] - M[n]$, codificamos e transmitimos a diferença.

Exemplo numérico

Vamos observar quão boa é a predição de um filtro de Wiener para um processo estocástico sintético.

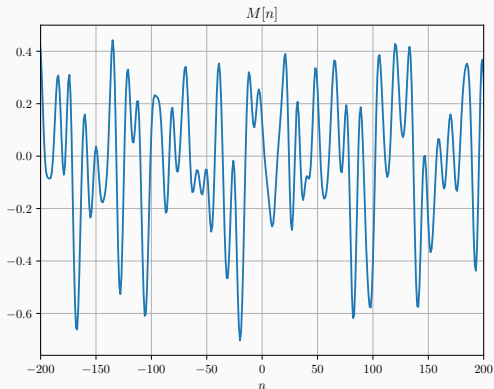


Figure 2: Trecho da amostra do processo aleatório.

Primeiramente, precisamos computar as estatísticas de autocorrelação do processo $M[n]$, contudo temos um problema:

Só temos *uma* realização do sinal.

Como podemos computar estatísticas com apenas uma amostra?

Ergodicidade

Para alguns sinais, podemos invocar a propriedade da *ergodicidade*. Mas o que é a ergodicidade?

Para um sinal ergódico (no sentido da estatística f), nós podemos estimar expectativas sobre o *ensemble* por médias no *tempo*. Por exemplo, considere a média *temporal* de um processo aleatório estacionário:

$$\hat{\mu}_M(T) = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} M(t) dt.$$

$\hat{\mu}_M(T)$ é uma variável aleatória, pois não computamos a expectativa. Para um processo ergódico para a média, contudo, temos a seguinte propriedade:

$$\begin{aligned}\lim_{T \rightarrow \infty} \mathbb{E}[\hat{\mu}_M(T)] &= \mu_M \\ \lim_{T \rightarrow \infty} \mathbb{E}[\text{Var}(\hat{\mu}_M(T))] &= 0,\end{aligned}$$

ou seja, a variável *converge* para a média verdadeira.

Invocando a ergodicidade no nosso processo, temos

$$\mu_M \approx \frac{1}{K} \sum_k m[k]$$
$$R_{MM}[l] \approx \frac{1}{K} \sum_k m[k]m[k-l],$$

onde $m[k]$ é a *realização* do processo, e o somatório é realizado sobre um intervalo adequado.

Resultado da estimativa para o nosso processo

Realizando o procedimento acima, para a realização que temos do nosso processo, obtemos a seguinte função de autocorrelação:

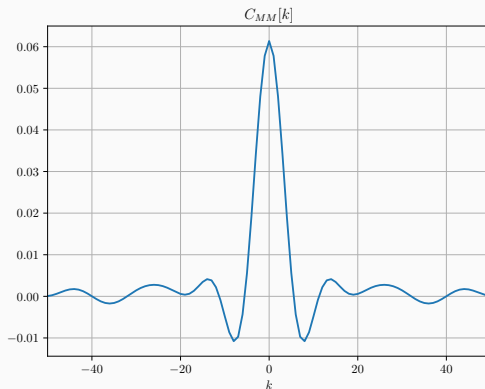


Figure 3: Estimativa da autocorrelação de $M[n]$.

Computando coeficientes do estimador

Com base nas estatísticas computadas, usamos a equação de Wiener-Hopf para encontrar os coeficientes:

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

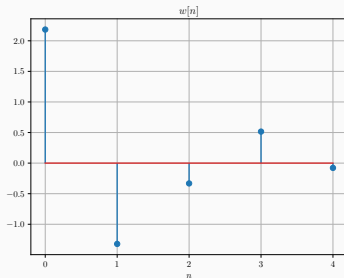


Figure 4: Coeficientes do filtro de Wiener de ordem 5 para o processo $M[n]$.

Sinal real vs. sinal estimado

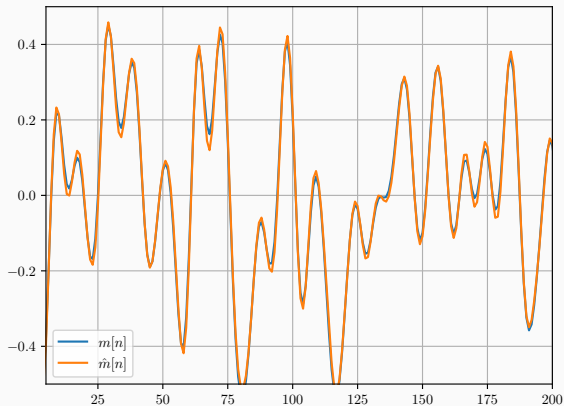


Figure 5: Sinal real e estimativa para o filtro projetado.

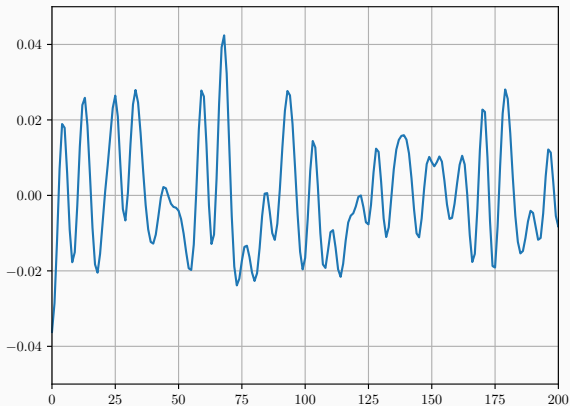


Figure 6: Diferença entre sinal real e estimativa para o filtro projetado.

Note a escala do gráfico!!!

Distribuição dos sinais

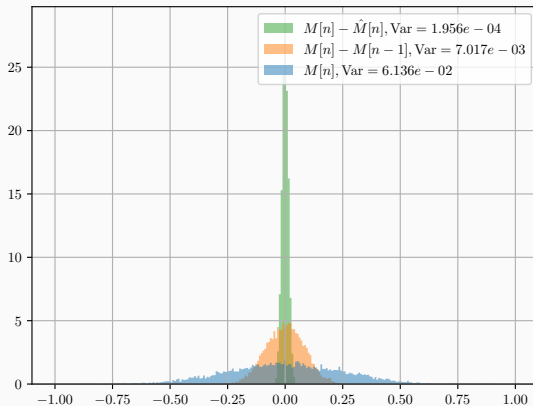


Figure 7: Distribuição das variáveis relevantes.

Vimos como a predição linear pode gerar um resíduo a ser transmitido com variância muito menor, sendo mais ‘fácil’ de ser quantizado com poucos bits, mas para isso, precisamos saber *a priori* a estatística do sinal a ser transmitido.

Na próxima aula, veremos como fazer algo parecido sem saber *a priori* a estatística do sinal transmitido.