



1. Introdução

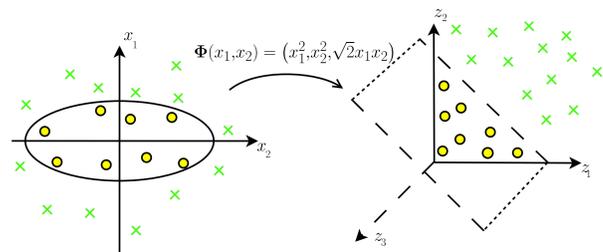
Filtros adaptativos podem ser empregados em diversas aplicações. Em muitas delas, soluções lineares podem apresentar um desempenho inferior em comparação com as não lineares, tais como os algoritmos baseados em núcleo. Este trabalho busca estudar duas versões de um desses algoritmos, denominado **KLMS (kernel least-mean-squares)**, expondo as diferenças entre elas e investigando algumas de suas propriedades.

2. Métodos

◊ Seja uma função de mapeamento não linear $\Phi : \mathbb{U} \rightarrow \mathbb{F}$. Define-se **núcleo** (ou *kernel*) como [1]

$$k(\mathbf{x}, \mathbf{x}') = \Phi^T(\mathbf{x})\Phi(\mathbf{x}'). \quad (\star)$$

◊ Motivação: utilizar $\Phi(\cdot)$ para realizar uma **filtragem não linear em \mathbb{U} adotando procedimentos lineares em \mathbb{F}** .



Exemplo de aplicação dos kernels.

◊ Escolhendo-se uma função de *kernel* adequada, o produto interno de (\star) pode ser calculado **sem mapear explicitamente os dados de entrada**;

◊ *Kernel* gaussiano:

$$k(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2h^2}}; \quad (\nabla)$$

◊ h : parâmetro que o projetista do filtro deve escolher, denominado *largura do kernel*;

◊ Um dos mais utilizados na literatura e adotado neste trabalho.

Algoritmos KLMS e KNLSM2

◊ Equações de atualização dos vetores de coeficientes do KLMS e do KNLSM2 dadas por [1,2]

$$\begin{aligned} \boldsymbol{\omega}(n) &= \boldsymbol{\omega}(n-1) + \mu e_\omega(n) \Phi(n), & (\bullet) \\ \boldsymbol{\alpha}(n) &= \boldsymbol{\alpha}(n-1) + \mu e_\alpha(n) \mathbf{k}(n), & (\circ) \end{aligned}$$

◊ **Dicionário** do algoritmo: $\{\mathbf{u}(c_i)\}$, $i = 1, \dots, L$;

◊ $\mathbf{k}(n) = [k(\mathbf{u}(n), \mathbf{u}(c_1)) \dots k(\mathbf{u}(n), \mathbf{u}(c_L))]$

◊ $e_\omega(n) = d(n) - \boldsymbol{\omega}^T(n-1)\Phi(n)$ e $e_\alpha(n) = d(n) - \boldsymbol{\alpha}^T(n-1)\mathbf{k}(n)$ são os erros de estimação, $d(n)$ é o sinal desejado e μ é o passo de adaptação;

Versões Normalizadas

◊ Algoritmo KNLSM

– Propõe-se normalizar o passo de adaptação de acordo com a norma de $\Phi(\mathbf{u}(n))$, i.e. [1]

$$\mu(n) = \frac{\tilde{\mu}}{\Phi^T(\mathbf{u}(n))\Phi(\mathbf{u}(n))}.$$

– Pela Equação (\star) , tem-se

$$\mu(n) = \frac{\tilde{\mu}}{k(\mathbf{u}(n), \mathbf{u}(n))}. \quad (\blacksquare)$$

– Para o *kernel* gaussiano $k(\mathbf{x}, \mathbf{x}) = 1$ para qualquer \mathbf{x} , o que implica que os algoritmos KLMS e KNLSM são **idênticos** quando se usa esse núcleo [1].

◊ Algoritmo KNLSM2

– Propõe-se normalizar o passo de acordo com a norma de $\mathbf{k}(n)$, i.e. [2]

$$\mu(n) = \frac{\tilde{\mu}}{\delta + \mathbf{k}^T(n)\mathbf{k}(n)} \quad (\blacksquare)$$

– δ : constante positiva utilizada para evitar divisão por zero.

Critério da Coerência

◊ Utilizado para **restringir** o crescimento do **dicionário**;

◊ O vetor de entrada atual $\mathbf{u}(n)$ é inserido no **dicionário** se

$$\max_j |k(\mathbf{u}(n), \mathbf{u}(c_j))| \leq \epsilon$$

◊ ϵ : parâmetro que o projetista do filtro deve escolher, denominado *nível de coerência* [2];

◊ Quanto **menor** ϵ , **menos** elementos serão incorporados ao **dicionário**;

◊ $k(\mathbf{x}, \mathbf{x}') \leq 1$ para o *kernel* gaussiano, o que implica que escolhendo-se $\epsilon \geq 1$ **todos** os elementos serão incorporados ao **dicionário**.

3. Resultados

Desenvolvimento Teórico

Hipóteses

H1 As saídas dos algoritmos são iguais, i.e.

$$\boldsymbol{\alpha}^T(n-1)\mathbf{k}(n) = \boldsymbol{\omega}^T(n-1)\Phi(n),$$

consequentemente os erros de estimação também são os mesmos:

$$e_\omega(n) = e_\alpha(n) = e(n);$$

H2 Pode-se fazer a aproximação

$$\boldsymbol{\alpha}^T(n)\mathbf{k}(n) \approx \boldsymbol{\omega}^T(n)\Phi(n)$$

Análise das Equações

◊ Manipulações algébricas em (\bullet) levam a

$$\boldsymbol{\omega}^T(n)\Phi(n) = \boldsymbol{\omega}^T(n-1)\Phi(n) + \mu e_\omega(n)k(\mathbf{u}(n), \mathbf{u}(n)) \quad (\blacklozenge)$$

◊ Usando **H1** e **H2**, a Equação (\blacklozenge) pode ser reescrita como

$$\boldsymbol{\alpha}^T(n)\mathbf{k}(n) = \boldsymbol{\alpha}^T(n-1)\mathbf{k}(n) + \mu e(n)k(\mathbf{u}(n), \mathbf{u}(n)). \quad (\blacktriangle)$$

◊ Manipulações algébricas em (\circ) levam a

$$\boldsymbol{\alpha}^T(n)\mathbf{k}(n) = \boldsymbol{\alpha}^T(n-1)\mathbf{k}(n) + \mu e(n)\mathbf{k}^T(n)\mathbf{k}(n). \quad (\blacktriangle)$$

◊ $k(\mathbf{u}(n), \mathbf{u}(n)) \neq \mathbf{k}^T(n)\mathbf{k}(n)$, de onde se conclui que os algoritmos são diferentes (consequentemente, suas versões normalizadas também).

Simulações

◊ Aplicação: equalização de um canal de comunicações com **distorção não linear**;

◊ $a(n) \in \{-1, 1\}$: mensagem binária;

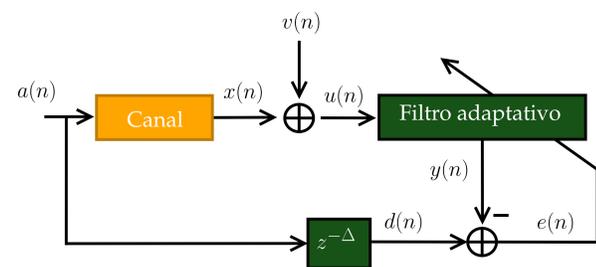
◊ $z(n)$: ruído gaussiano branco de média nula e desvio padrão σ_z escolhido de modo a se obter uma relação sinal-ruído de 30 dB;

◊ Média de 100 realizações.

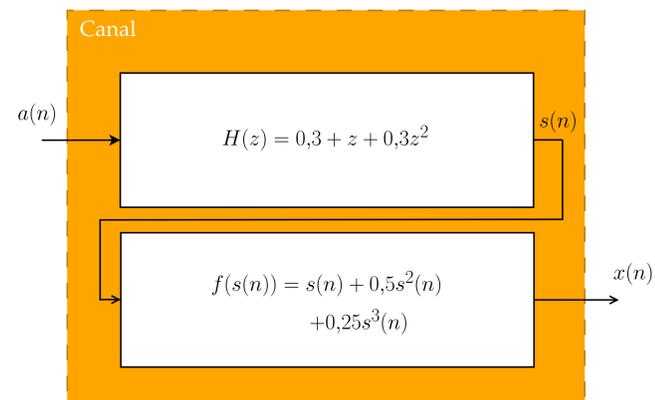
◊ Para o algoritmo KLMS (KNLSM) escolheu-se $\mu = 0,5$ e $h = 1,0$;

◊ Para o algoritmo KNLSM2 escolheu-se $\tilde{\mu} = 0,05$ e $h = 1,0$;

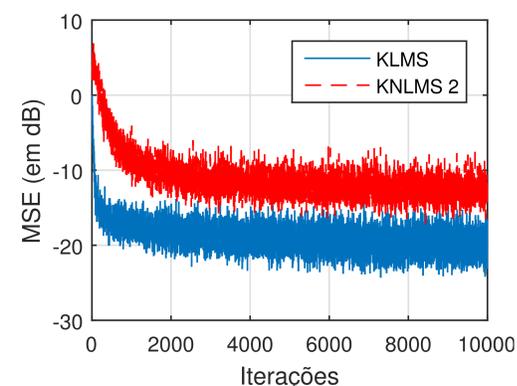
◊ Valores escolhidos de modo a se obter o **melhor desempenho** em regime sem restrições ao **dicionário**.



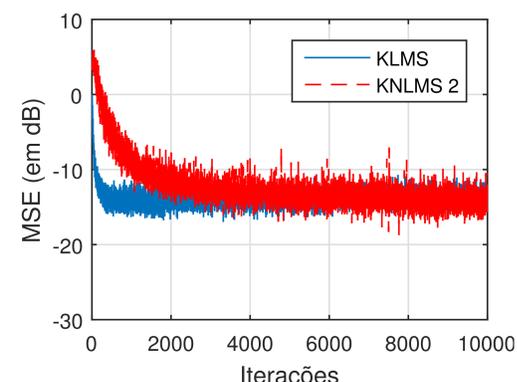
Equalização adaptativa de um canal de comunicação.



Estrutura interna do canal não linear.



MSE ao longo das iterações sem restrição de **dicionário**



MSE ao longo das iterações com o critério da coerência e $\epsilon = 0,9$.

4. Conclusões

◊ A partir das equações (\bullet) e (\circ) , verifica-se que os algoritmos estudados são **diferentes** entre si;

◊ Analisando-se os gráficos, pode-se constatar que o algoritmo KNLSM (resp., KLMS) apresenta **maior sensibilidade à restrição do crescimento de seu dicionário** que o KNLSM2 (resp., KLMS2).

[1] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering*, Wiley, 2010.

[2] W.D. Parreira *et al*, "Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm," *IEEE Trans. on Signal Process.*, vol. 60, pp. 2208-2222, 2012.