

# Simulação do comportamento estocástico do algoritmo KLMS com diferentes kernels

Patrick Medeiros De Luca  
Universidade do Vale do Itajaí – UNIVALI  
patrick\_mdl@edu.univali.br

Wemerson Delcio Parreira  
Universidade do Vale do Itajaí – UNIVALI  
parreira@univali.br

## ABSTRACT

The kernel least-mean-square (KLMS) algorithm is a popular algorithm in nonlinear adaptive filtering due to its simplicity and robustness. In kernel adaptive filtering, the statistics of the input to the linear filter depends on the kernel and its parameters. Moreover, practical implementations on systems estimation require a finite non-linearity model order. In order to obtain finite order models, many kernelized adaptive filters use a dictionary of kernel functions. Dictionary size also depends on the kernel and its parameters. Therefore, KLMS may have different performances on the estimation of a nonlinear system, the time of convergence, and the accuracy using a different kernel. In order to analyze the performance of KLMS with different kernels, this paper proposes the use of the Monte Carlo simulation of both steady-state and the transient behavior of the KLMS algorithm using different types of kernel functions and Gaussian inputs.

## KEYWORDS

Adaptive filtering, KLSM algorithm, Kernel function, Monte Carlo simulation

## 1 INTRODUÇÃO

A filtragem adaptativa tem sido largamente utilizada nas últimas décadas em diversas áreas como comunicações, engenharia biomédica, entre outras. Sua principal característica é a possibilidade de ajuste ótimo dos parâmetros de filtragem na ausência de uma informação estatística prévia dos sinais envolvidos [Haykin 2002; Sayed 2003].

Modelos lineares são ainda rotineiramente usados devido à sua inerente simplicidade, do ponto de vista conceitual e de implementação. Em muitas situações práticas, entretanto, um processamento não-linear do sinal se faz necessário [Parreira 2012].

Um sistema não-linear (SNL) pode ser comumente encontrado em processos de identificação, predição e controle aplicados às áreas como comunicações e engenharia biomédica [Giannakis and Serpedin 2001]. Diferentemente do caso de sistemas lineares, os quais são completamente caracterizados por suas respostas ao impulso, não existe um contexto matemático simples para descrever um SNL. Técnicas tais como, os Filtros polinomiais, comumente chamados de Séries de Volterra [Mathews and Sicuranza 2000] e Redes Neurais [Haykin 1994] são os mais populares e estudados modelos não-lineares para filtragem adaptativa.

Os métodos baseados em kernels, em particular as Support Vector Machines (SVMs), são aplicados a muitos problemas práticos de interesse na área de engenharia, sendo considerados “estado da arte” em vários domínios [Schölkopf et al. 2004]. Apenas nas últimas décadas, entretanto, os métodos baseados em kernels têm-se tornado uma tendência em machine learning e inferência bayesiana.

Aplicações em estimação de SNL baseiam-se no mapeamento do espaço de entrada para um espaço de Hilbert definido por kernel reprodutivo (Reproducing Kernel Hilbert Space – RKHS). Esse mapeamento permite que uma combinação linear de kernels possa ser usada na estimação de um SNL. Muitos autores recorrem aos algoritmos adaptativos para estimar os coeficientes dessa combinação linear baseada em kernels. Porém, em aplicações práticas, torna-se inviável empregar um modelo no RKHS cujo número de coeficientes cresce com o número de observações realizadas [Parreira et al. 2011].

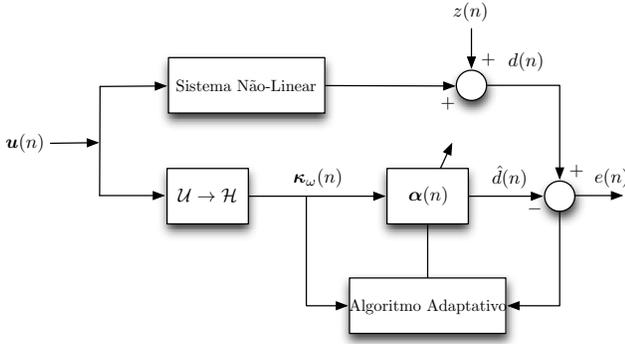
Diversos algoritmos foram propostos para limitar a dimensão dos modelos específicos no contexto de filtragem adaptativa baseada em kernel – KAF [Dodd et al. 2003; Engel et al. 2004; Wang et al. 2015]. Cada um desses consiste da adição de um procedimento extra de esparsificação a um processo de filtragem adaptativa. Em aplicações em tempo real, entretanto, esses algoritmos de duas etapas podem elevar os custos do processo. O modelo de ordem finita empregado em [Parreira et al. 2018], usando técnica de esparsificação de dicionário apresentado em [Honeine et al. 2007; Richard et al. 2009] permitiu a análise do comportamento estocástico do algoritmo adaptativo no espaço das características com custo inferior às técnicas citadas.

Uma das principais dificuldades existentes no projeto de filtros adaptativos não-lineares baseados em kernels é a escolha do kernel mais adequado à aplicação, bem como o projeto otimizado dos parâmetros do kernel escolhido. Contribuições para novas discussões sobre filtragem adaptativa envolvendo algoritmos baseados em kernel foram apresentadas em [Chen et al. 2013; Parreira et al. 2012, 2018] para ambientes estacionários para o kernel Gaussiano. Assim, uma importante contribuição para essa área é a avaliação do comportamento estocástico dos algoritmos KLMS em diferentes cenários de aplicação com diferentes kernels.

## 2 ALGORITMO KLMS EM ESTIMAÇÃO DE SISTEMAS NÃO-LINEARES

Uma revisão dos algoritmos KAF foi apresentada em [Liu et al. 2010]. A caracterização de uma estimação de um SNL pode ser representada pela Figura 1. Nesta figura tem-se que o espaço  $U \subseteq \mathbb{R}^q$  é o espaço dos vetores de entrada  $\mathbf{u}(n)$ ,  $\kappa : U \times U \rightarrow \mathbb{R}$  é o kernel que define o RKHS,  $\mathcal{H}$ , induzido pelo produto interno  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ ,  $d(n)$  é a soma do sinal a ser estimado com um ruído aditivo  $z(n)$  de média zero e decorrelacionado de qualquer outro sinal.

Esse problema pode ser visto de maneira reduzida com a determinação do vetor  $\alpha = [\alpha_1, \dots, \alpha_N]^T$  que minimiza  $\|\mathbf{d} - \mathbf{K}\alpha\|^2$ , em que  $\mathbf{K}_{N \times N}$  é a matriz Gram com  $(n, \ell)$ -ésima entrada dada por  $\kappa(\mathbf{u}(n), \mathbf{u}(\ell))$  e  $\mathbf{d} = [d(1), \dots, d(N)]^T$ . Neste caso, a ordem do problema, e consequentemente a ordem do filtro adaptativo, cresce com o número  $N$  de observações do sinal de entrada  $\mathbf{u}(\cdot)$ . Tal situação



**Figura 1: Estimação de sistemas utilizando algoritmos KAF [Parreira et al. 2018].**

inviabiliza a aplicação desse algoritmo na resolução de problemas em tempo real. Uma forma de contornar essa limitação é utilizar modelos de ordem limitada (ou ordem finita) da forma

$$\hat{d}(\cdot) = \sum_{m=1}^M \alpha_m \kappa(\cdot, \mathbf{u}(\omega_m)) \quad (1)$$

em que  $M$  kernels  $\{\kappa(\cdot, \mathbf{u}(\omega_m))\}_{m=1}^M$  compõem o dicionário  $\mathcal{D}$  de dimensão  $M$ , em que  $\omega_j, j = 1, \dots, M$ , formam um subconjunto de  $\{1, 2, \dots, N\}$  correspondendo aos índices temporais dos  $M < N$  vetores de entrada escolhidos para compor o modelo definido em (1).

O vetor de pesos  $\alpha$  é calculado iterativamente para o modelo (1) usando o KLMS com ordem finita [Parreira et al. 2012] dado por:

$$\alpha(n+1) = \alpha(n) + \eta e(n) \kappa_\omega(n) \quad (2)$$

em que  $\eta$  é o passo de adaptação do algoritmo KLMS,  $\kappa_\omega(n) = [\kappa(u(n), u(\omega_1)), \dots, \kappa(u(n), u(\omega_M))]^\top$  e o erro na  $n$ -ésima iteração é calculado usando:

$$e(n) = d(n) - \hat{d}(n). \quad (3)$$

Além da escolha da ordem  $M$  do modelo, o dicionário pode ser estático, se um conjunto fixo de vetores for escolhido, ou dinâmico, se o dicionário for atualizado com intervalos temporais pré-definidos. O segundo caso é especialmente importante nos casos de dicionários pequenos ou de sinais não estacionários [Parreira et al. 2018], comuns em problemas envolvendo aplicações reais.

### 3 SOLUÇÃO PROPOSTA

Estudos recentes [Chen et al. 2013; Parreira et al. 2012, 2018] conduziram a análise do comportamento estocástico do algoritmo KLMS usando o kernel Gaussiano. Esses trabalhos visam auxiliar o projetista na implementação desses algoritmos em cenários de estimação de SNL. Porém, nenhum deles analisou o comportamento com relação a acurácia e estimação considerando diferentes kernels.

A proposta é avaliar o comportamento estocástico do algoritmo KLMS para diferentes aplicações vinculadas a estimação de SNL, baseado em diferentes kernels. Para isso, deverá ser implementado um banco com vários kernels para ser usado no algoritmo KLMS. A avaliação do desempenho na estimação é baseada na simulação de

Monte Carlo do comportamento do erro quadrático médio (MSE), definido por,

$$J_{ms}(n) := E\{e^2(n)\}. \quad (4)$$

Além disso, uma análise da velocidade de convergência e do esforço computacional serão realizadas. As aplicações em engenharia biomédica que serão consideradas futuramente envolvem sistemas correlacionados no tempo com não linearidade do tipo sigmoide presentes em aparelhos auditivos.

### 4 RESULTADOS INICIAS

Para avaliar o comportamento do algoritmo KLMS baseado em diferentes kernels para o problema de estimação, Figura 1, foi considerado um sinal de entrada  $\mathbf{u}(\cdot)$  Gaussiano, independente e identicamente distribuído (i.i.d) com média nula e variância  $\sigma_u^2 = 10^{-4}$ . A saída  $\phi(n)$  é dada por:

$$\phi(n) = \omega_0^\top \text{sen}(\mathbf{u}(n)) \quad (5)$$

em que  $\mathbf{u}(n) = [u(n) u(n-1), \dots, u(n-M+1)]^\top$  e o vetor de pesos  $\omega_0 = [-0, 0098; -0, 0143; -0, 0251; -0, 0399; -0, 0518; -0, 0619; -0, 0709; -0, 0781; -0, 0834; -0, 0869; -0, 0886; -0, 0886; -0, 0869; -0, 0836; -0, 0789; -0, 0729; -0, 0657; -0, 0576; -0, 0488; -0, 0394; -0, 0297; -0, 0198; -0, 0100; -0, 0004; 0, 0088; 0, 0175; 0, 0254; 0, 0325; 0, 0387; 0, 0439; 0, 0480; 0, 0510; 0, 0530; 0, 0538; 0, 0536; 0, 0524; 0, 0503; 0, 0473; 0, 0435; 0, 0391; 0, 0341; 0, 0287; 0, 0230; 0, 0172; 0, 0112; 0, 0053; -0, 0004; -0, 0059; -0, 0110; -0, 0157; -0, 0199; -0, 0236; -0, 0266; -0, 0290; -0, 0307; -0, 0318; -0, 0322; -0, 0320; -0, 0312; -0, 0299; -0, 0280; -0, 0257; -0, 0231; -0, 0201; -0, 0168; -0, 0134; -0, 0099; -0, 0064; -0, 0029; 0, 0005; 0, 0037; 0, 0067; 0, 0095; 0, 0119; 0, 0140; 0, 0158; 0, 0171; 0, 0181; 0, 0187; 0, 0190; 0, 0188; 0, 0183; 0, 0175; 0, 0164; 0, 0150; 0, 0135; 0, 0117; 0, 0098; 0, 0078; 0, 0058; 0, 0037; 0, 0017; -0, 0003; -0, 0022; -0, 0039; -0, 0055; -0, 0069; -0, 0081; -0, 0091; -0, 0099; -0, 0105; -0, 0108; -0, 0109; -0, 0109; -0, 0106; -0, 0101; -0, 0095; -0, 0087; -0, 0078; -0, 0068; -0, 0057; -0, 0045; 0, 0034; -0, 0022; -0, 0010; 0, 0001; 0, 0012; 0, 0022; 0, 0031; 0, 0039; 0, 0046; 0, 0051; 0, 0056; 0, 0059; 0, 0061; 0, 0062; 0, 0062; 0, 0060; 0, 0057; 0, 0054; 0, 0049; 0, 0044; 0, 0039; 0, 0033; 0, 0026; 0, 0020; 0, 0013; 0, 0007; 0, -0, 0006; -0, 0011; -0, 0016, -0, 0021; -0, 0025; -0, 0028; -0, 0031; -0, 0033; -0, 0034; -0, 0034; -0, 0034]^\top$ <sup>1</sup>. A saída do sistema  $\phi(n)$ , Equação (5), é corrompida por um ruído Gaussiano branco, com  $\sigma^2 = 10^{-6}$  decorrelacionado de qualquer outro sinal. O sinal  $d(n)$  foi estimado do modelo finito, Equação (1), usando o algoritmo KLMS com  $M = 150$  coeficientes e os kernels Gaussiano, Polinomial e de Cauchy, definidas por:

A. Kernel Gaussiano: Para dois padrões  $\mathbf{u}, \mathbf{u}' \in \mathcal{U}$  o kernel Gaussiano é dado por,

$$\kappa_G(\mathbf{u}, \mathbf{u}') = \exp\left(-\frac{\|\mathbf{u} - \mathbf{u}'\|_2^2}{2\xi^2}\right) \quad (6)$$

em que  $\|\cdot\|_2$  representa a norma Euclidiana e  $\xi \in \mathbb{R}_{\neq 0}$ .

B. Kernel polinomial: Para dois padrões  $\mathbf{u}, \mathbf{u}' \in \mathcal{U}$  o kernel polinomial é dado por,

$$\kappa_P(\mathbf{u}, \mathbf{u}') = (\alpha + \mathbf{u}^\top \mathbf{u}')^\beta \quad (7)$$

<sup>1</sup>Esse vetor tem uma importância prática na simulação do efeito de oclusão em aparelhos auditivos [Borges and Costa 2016].

em que  $(.)^\top$  representa a operação de transposição e  $\alpha, \beta \in \mathbb{R}_{>0}$ .

C. Kernel de Cauchy: Para dois padrões  $\mathbf{u}, \mathbf{u}' \in \mathcal{U}$  o kernel de Cauchy é dado por,

$$\kappa_C(\mathbf{u}, \mathbf{u}') = \frac{\gamma^2}{\gamma^2 + \|\mathbf{u} - \mathbf{u}'\|_2^2} \quad (8)$$

em que  $\gamma \in \mathbb{R}_{\neq 0}$ .

Essas funções foram escolhidas pelas suas características matemáticas e o frequente uso em processamento de sinais. Após alguns testes foram selecionados o seguinte conjunto de parâmetros para os kernels para ilustrar o comportamento estocástico do MSE:  $\{\xi = 0, 5; \alpha = 5, \beta = 1; \gamma = 0, 5\}$ . O passo de adaptação usado nas simulações apresentadas foi  $\eta = 10^{-4}$ .

A Figura 2 apresenta o comportamento estocástico do algoritmo KLMS. Os resultados iniciais foram sumarizados na Tabela 1. Nesta comparação o KLMS Polinomial apresenta uma convergência em aproximadamente 160 iterações, KLMS Gaussiano em 500 iterações e o KLMS de Cauchy em 750 iterações. O número de iterações não considera o tempo e o custo computacional associado aos kernels, apenas o tempo de aprendizado.

Com o objetivo melhorar a apresentação dos resultados do MSE em regime permanente foi calculado a partir da média das últimas 100 observações da simulação de Monte Carlo. Neste exemplo para se obter um desempenho inferior a -40 dB em regime permanente o kernel Polinomial apresentou o melhor desempenho em tempo para convergência. Porém, neste caso não foi explorado as diferentes dimensões do filtro, pois para os casos simulados foi usado apenas  $M = 150$ .

Os resultados iniciais apenas fornecem indícios de que os kernels podem ser exploradas no sentido de minimizar o custo e maximizar o desempenho. Novas simulações devem conduzir a uma modelagem que permita refletir sobre o compromisso tempo para convergência e erro em ambos os regimes, transitório e permanente, do algoritmo KLMS operando com diferentes kernels.

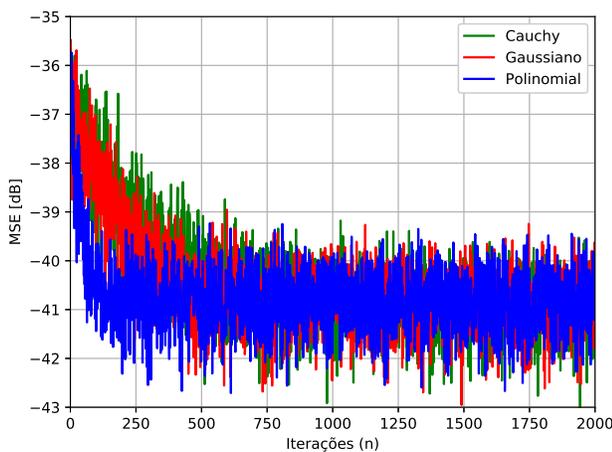


Figura 2: MSE para algoritmo KLMS: kernel Polinomial em azul, kernel Gaussiano em vermelho e kernel de Cauchy em verde.

Tabela 1: Sumário comparativo do algoritmo KLMS.

Kernel	Tempo de Convergência	MSE Reg. Perm.
Polinomial	160 iterações	40,8730 dB
Gaussiano	500 iterações	41,7419 dB
Cauchy	750 iterações	40,1157 dB

## 5 CONSIDERAÇÕES FINAIS

Até o presente, três kernels foram implementados em linguagem de programação Python. Apesar do kernel Gaussiano ser o comumente usado em virtude de sua robustez, o kernel Polinomial tem apresentado resultados relevantes tais como os encontrados em [Parreira 2012]. Para dar continuidade a pesquisa serão simulados o comportamento estocástico do algoritmo KLMS baseado em outros kernels em diferentes cenários.

## AGRADECIMENTOS

Os autores agradecem à Universidade do Vale do Itajaí e ao Governo do Estado de Santa Catarina pelo incentivo à pesquisa.

## REFERÊNCIAS

- R. C. Borges and M. H. Costa. 2016. A feed forward adaptive canceller to reduce the occlusion effect in hearing aids. *Computers in Biology and Medicine* 75 (2016), 266 – 275.
- J. Chen, W. Gao, C. Richard, and J. C. M. Bermudez. 2013. Convergence Analysis of Kernel LMS Algorithm with Pre-tuned Dictionary. In *Proc. IEEE ICASSP'13*. Vancouver, Canada.
- T. J. Dodd, B. Mitchinson, and R. F. Harrison. 2003. Sparse stochastic gradient descent learning in kernel models. In *Proceedings of the Second International Conference on Computational Intelligence, Robotics and Autonomous Systems (Singapore)*.
- Y. Engel, S. Mannor, and R. Meir. 2004. Kernel recursive least squares. *IEEE Trans. Signal Process.* 52, 8 (2004), 2275–2285.
- G. B. Giannakis and E. Serpedin. 2001. A bibliography on nonlinear system identification. *Signal Processing* 81, 3 (2001), 533–580.
- S. Haykin. 1994. *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
- S. Haykin. 2002. *Adaptive filter theory*.
- P. Honeine, C. Richard, and J. C. M. Bermudez. 2007. On-line nonlinear sparse approximation of functions. In *Proc. IEEE ISIT'07*. Nice, France, 956–960.
- W. Liu, J. C. Principe, and S. Haykin. 2010. *Kernel Adaptive Filtering*. John Wiley & Sons, Inc.
- V. J. Mathews and G. L. Sicuranza. 2000. *Polynomial Signal Processing*.
- W. D. Parreira. 2012. *Comportamento estocástico do algoritmo kernel least-mean-square*. Ph.D. Dissertation. Universidade Federal de Santa Catarina.
- W. D. Parreira, J.C.M. Bermudez, C. Richard, and J.-I. Tournier. 2012. Stochastic Behavior Analysis of the Gaussian Kernel Least-Mean-Square Algorithm. *IEEE Trans. Signal Process.* 60, 5 (March 2012), 2208 – 2222.
- W. D. Parreira, J. C. M. Bermudez, C. Richard, and J.-Y. Tournier. 2011. Steady-State Behavior and Design of the Gaussian KLMS Algorithm. In *Proc. EUSIPCO'11*. Barcelona, Spain.
- W. D. Parreira, M. C. Costa, and J. C. M. Bermudez. 2018. Stochastic behavior analysis of the Gaussian KLMS algorithm for a correlated input signal. *Signal Processing* 152 (2018), 286–291.
- C. Richard, J. C. M. Bermudez, and P. Honeine. 2009. Online Prediction of Time Series Data With Kernels. *IEEE Trans. Signal Process.* 57, 3 (March 2009), 1058 –1067. <https://doi.org/10.1109/TSP.2008.2009895>
- A. H. Sayed. 2003. *Fundamentals of adaptive filtering*. John Wiley & Sons, Hoboken, NJ.
- B. Schölkopf, K. Tsuda, and J.-P. Vert. 2004. *Support vector machine applications in computational biology*. MIT press.
- S. Wang, Y. Zheng, and C. Ling. 2015. Regularized kernel least mean square algorithm with multiple-delay feedback. *IEEE Signal Processing Letters* 23, 1 (2015), 98–101.