

# 繰り返し法によるカーネル NLMS アルゴリズムの収束特性改善

Repeating Method for Kernel NLMS Algorithm for Improving Convergence Characteristics

牧崎 幸司  
Koji MAKIZAKI

西川 清史  
Kiyoshi NISHIKAWA

首都大学東京大学院システムデザイン研究科情報通信システム学域  
Department of Information and Communications Systems, Tokyo Metropolitan University

## 1 まえがき

本報告では、Kernel Normalized Least Mean Square (KNLMS) アルゴリズム [1, 2] に対する繰り返し法 [3, 4] を提案する。提案手法を用いることで、学習特性が改善可能である事を示す。KNLMS は、非線形システムの学習が可能な適応アルゴリズムであり、非線形チャネル等化、非線形能動的ノイズ制御などに有用である [1, 2]。

## 2 準備

### 2.1 KNLMS アルゴリズム [2]

KNLMS アルゴリズムの学習の手順を示す。

Initialization

$$\mathbf{D}_1 = \mathbf{x}_1, \quad \mathbf{h}_1 = \kappa(\mathbf{x}_1, \mathbf{D}_1(1)), \quad m = 1$$

for  $n = 2, 3, \dots$

$$\text{if } \max_{j=1, \dots, m} |\kappa(\mathbf{x}_n, \mathbf{D}_n(j))| > \mu_0$$

$$\mathbf{D}_n = \mathbf{D}_{n-1}$$

$$\mathbf{h}_n = [\kappa(\mathbf{x}_n, \mathbf{D}_n(1)) \quad \dots \quad \kappa(\mathbf{x}_n, \mathbf{D}_n(m))]^T$$

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \frac{\eta}{\epsilon + \|\mathbf{h}_n\|^2} (d_n - \mathbf{h}_n^T \hat{\alpha}_{n-1}) \mathbf{h}_n \quad (1)$$

else

$$m = m + 1, \quad \mathbf{D}_n = \mathbf{D}_{n-1} \cup \{\mathbf{x}_n\}$$

$$\mathbf{h}_n = [\kappa(\mathbf{x}_n, \mathbf{D}_n(1)) \quad \dots \quad \kappa(\mathbf{x}_n, \mathbf{D}_n(m))]^T$$

$$\hat{\alpha}_n = \begin{bmatrix} \hat{\alpha}_{n-1} \\ 0 \end{bmatrix} + \frac{\eta}{\epsilon + \|\mathbf{h}_n\|^2} (d_n - \mathbf{h}_n^T \begin{bmatrix} \hat{\alpha}_{n-1} \\ 0 \end{bmatrix}) \mathbf{h}_n \quad (2)$$

end if

end for

ここで、 $\mathbf{x}_n$  は入力ベクトル、 $d_n$  は所望信号、 $m$  はフィルタ次数、 $\kappa(\cdot, \cdot)$  はカーネル関数、 $\mu_0$  は閾値、 $\eta$  はステップサイズ、 $\epsilon$  は安定化パラメータである。KNLMS アルゴリズムは閾値  $\mu_0$  により決定される条件に基づいて、 $M$  次入力ベクトル  $\mathbf{x}_n$  を  $M \times m$  次の辞書  $\mathbf{D}_n$  に保持する。入力ベクトル  $\mathbf{x}_n$  と辞書  $\mathbf{D}_n$  より、カーネルトリックによって得られる  $m$  次の  $\mathbf{h}_n$  を新たな入力ベクトルと見なし、NLMS アルゴリズムにより処理する。KNLMS は NLMS に基づいているため、収束が遅く定常誤差が大きいという問題がある。

### 2.2 繰り返し法 (repeating method) [3, 4]

線形な LMS 型の適応フィルタの収束特性を改善する手法として、繰り返し法が知られている [3, 4]。更新に過去  $L$  点のデータを保存し再利用することで、収束特性の改善を行う。更新式は以下で与えられる。

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{2}{L} \sum_{j=0}^{L-1} \frac{e_{n-j}}{\|\mathbf{x}_{n-j}\|^2} \mathbf{x}_{n-j} \quad (3)$$

ここで、 $e_{n-j} = d_{n-j} - \mathbf{x}_{n-j}^T \mathbf{w}_{n-1}$  である。

### 3 提案法

KNLMS アルゴリズムでは、フィルタ  $\hat{\alpha}_n$  の次数  $m$  が時変となるため、繰り返し法を直接適用することはでき

ない。本報告では、フィルタ次数  $m$  が変化する時刻  $n$  では、式 (2) によってフィルタ係数を更新し、 $m$  が変化する時刻  $n$  でのみ、繰り返し処理を行う手法を提案する。提案するフィルタの更新式を以下に示す。

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \frac{\eta}{L'} \sum_{j=0}^{L'-1} \frac{(d_{n-j} - \mathbf{h}_{n-j}^T \hat{\alpha}_{n-1})}{\epsilon + \|\mathbf{h}_{n-j}\|^2} \mathbf{h}_{n-j} \quad (4)$$

$$L' = \min(L_c, L) \quad (5)$$

ここで、 $L_c$  は次数  $m$  が不変な期間とする。また、 $\min(A, B)$  は、 $A, B$  の最小値を表す。

### 4 シミュレーション

提案法の有効性を示すために、以下の式で与えられる非線形システム [2] のシステム同定を行った。

$$d_n = (0.8 - 0.5 \exp(-d_{n-1}^2)) d_{n-1} - (0.3 + 0.9 \exp(-d_{n-1}^2)) d_{n-2} + 0.1 \sin(d_{n-1} \pi) \quad (6)$$

初期状態  $(d_{-1}, d_{-2})$  は区間  $[0, 1]$  の一様乱数により与え、 $d_n = \psi(d_{n-1}, d_{n-2})$  で表される非線形モデルを推定し、MSE により評価した。 $M = 2$ 、 $\mu_0 = 0.5$ 、 $\eta = 1.0$ 、 $\epsilon = 2^{-16}$  とし、カーネルにはガウシアンカーネル  $\kappa(\mathbf{a}, \mathbf{b}) = \exp(-3.73 \|\mathbf{a} - \mathbf{b}\|^2)$  を用いた。フィルタ更新に利用したデータ数  $L$  は 2, 32 とし、提案法と KNLMS を比較した。シミュレーション結果を図 1 に示す。図より、提案法により収束特性が改善されていることが分かる。

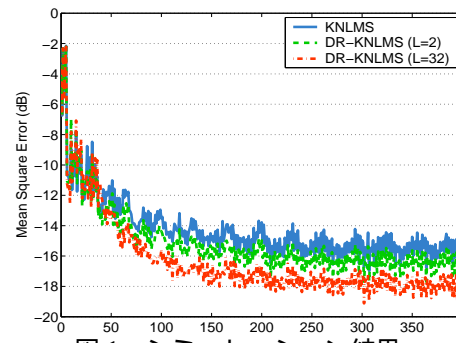


図 1 シミュレーション結果

### 参考文献

- [1] Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe "The Kernel Least-Mean-Square Algorithm" IEEE Trans. Signal Processing, VOL. 56, NO. 2, February 2008.
- [2] Cédric Richard, José Carlos M. Bermudez, and Paul Honeine "Online Prediction of Time Series Data With Kernels" IEEE Trans. Signal Processing, VOL. 57, NO. 3, March 2009.
- [3] J. Nagumo and A. Noda, "A Learning Method for System Identification," IEEE Trans. Automatic Control, Vol. AC-12, No.3, June 1967.
- [4] Robert A. Soni, Kyle A. Gallivan, and W. Kenneth Jenkins "Low-Complexity Data Reusing Methods in Adaptive Filtering" IEEE Trans. Signal Processing, VOL. 52, NO. 2, February 2004.