

スパース性に基づく RLS 型適応フィルタの追従特性改善法

Improvement of Tracking Performance of RLS-type Adaptive Filters based on Sparsity

野田祐一[†]

西川清史[†]

[†] 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

Yuichi NODA[†]

Kiyoshi NISHIKAWA[†]

[†]Department of Information and Communications Systems Engineering,
Tokyo Metropolitan Univesity

アブストラクト RLS (Recursive Least Squares) 型の適応アルゴリズムは、未知系の変化への追従のために過去の推定値に重みを与える忘却係数が使用されている。忘却係数は大きい値を取るほど、残留誤差は少なくなるが、追従特性が悪化する。一般に、適応アルゴリズムには少ない残留誤差と高速な追従の両方が求められる。本稿では、トレードオフの関係である残留誤差と追従特性の改善について検討する。提案法では忘却係数を大きい値で固定したまま、スパース性に基づき未知系の変動を推定し、フィルタ係数を初期化させることで、RLS 型適応アルゴリズムの追従特性が改善可能であることを示す。

1 はじめに

本稿では、スパース性に基づく RLS 型適応フィルタの追従特性改善法を提案する。

適応フィルタに求められる特性として、高速な収束特性、系の変化に対する追従特性、外乱などに対するロバスト性、少ない演算量などがあげられる。適応フィルタによるシステム同定は、入力信号が伝送経路を通った結果、受信側では遅延波などが足し合わされて受信され、その中から入力信号のみを取り出すため、どのような経路を辿ってきたかを推定するための手法である。その方法は、遅延など未知システムを通った信号と所望信号との誤差を最小化するようにフィルタの係数更新を行うもので、適応アルゴリズムと呼ばれている。

適応アルゴリズムは最小平均自乗 (Least Mean Square:LMS) アルゴリズムと再帰型最小自乗 (Recursive Least Square:RLS) アルゴリズムの 2 種類に大きく分けられる。LMS アルゴリズムは、収束速度は遅いが演算量が少なく、最も広く用いられているアルゴリズムである。LMS アルゴリズムの応用に、入力信号を正規化した正規化 LMS (Normalized LMS:NLMS) アルゴリズムがある [1]。一方、RLS アルゴリズムは演算量が多いが収束速度は

速いアルゴリズムである [2]。また、RLS アルゴリズムの演算量を削減した ERLS-DCD (Exponentially weighted RLS - Dichotomous Coordinate Descent) アルゴリズムが提案されている [3]。これらの RLS 型アルゴリズムは、伝搬路追従のために過去の信号に重み付けを行う忘却係数を用いている。

忘却係数は過去のデータを忘却することで、追従特性を調整するためのパラメータである。忘却係数 λ は $0 < \lambda \leq 1$ の値をとり、1 に近いほど残留誤差が小さくなるが追従特性は悪化するという問題がある。 λ の値は 1 に設定することで理論的には最小自乗解が得られるが、未知系の変化に追従できなくなるため、多くの場合 1 より小さい正の実数値に設定される。 λ の値を小さくすることで追従特性は改善するが、残留誤差が増加する [4]。

追従特性改善のために可変忘却係数を用いた、VFF-RLS (Variable Forgetting Factor - RLS) アルゴリズム [5]、GVFF-RLS (Gradient-based VFF-RLS) アルゴリズム [6]、SVFF-ERLS (Sparsity-based VFF-RLS) アルゴリズム [7] などの手法が提案されている。これらの可変忘却係数アルゴリズムは、それぞれ追従特性の改善を実現しているが、忘却係数を可変にするための演算量増加などの問題がある。

本稿では、忘却係数を固定したままで RLS 型適応アルゴリズムの追従特性を改善するための手法を提案する。忘却係数を大きい値で固定したまま、スパース性に基づき未知系の変動を推定し、フィルタ係数を初期化させることが可能であることを示す。また、計算機シミュレーションにより提案法の有効性を示す。

2 準備

ここでは、準備として、適応アルゴリズムおよびシステムのスパース性について簡単に説明する。

2.1 適応フィルタ

適応フィルタは、未知システムの入力信号や誤差信号からそのシステムの伝達関数を自律的に学習するフィルタである。図1に適応フィルタの構成を示す。

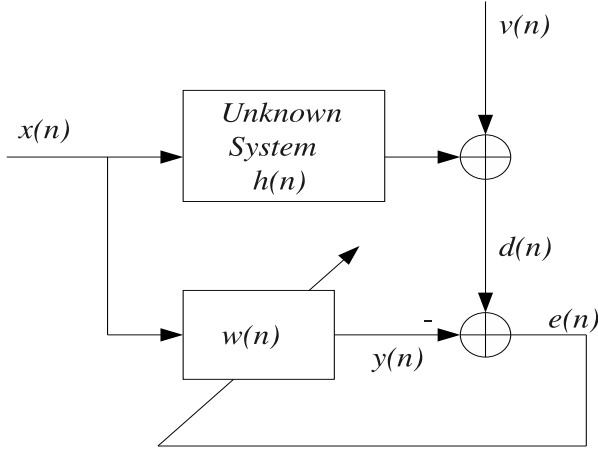


図 1: 適応フィルタの構成

ここで $x(n)$ は入力信号、 $d(n)$ は所望信号、 $v(n)$ は付加雑音、 $w(n)$ はフィルタ係数を表す。

時刻 n におけるフィルタ入力ベクトル $\mathbf{X}(n)$ を

$$\mathbf{X}(n) = [x(n), x(n-1), \dots, x(n-M+1)] \quad (1)$$

で表し、フィルタ係数ベクトル $\mathbf{W}(n)$ を

$$\mathbf{W}(n) = [w_0(n), w_1(n), \dots, w_M(n)]^T \quad (2)$$

とする。ここで、 T は転置を表す。入力信号 $x(n)$ に対するフィルタの出力信号は

$$y(n) = \mathbf{W}(n)^T \mathbf{X}(n) \quad (3)$$

となる。誤差信号 $e(n)$ は所望信号 $d(n)$ と出力信号 $y(n)$ の差

$$e(n) = d(n) - y(n) \quad (4)$$

と定義され、 $e(n)$ が小さくなるようにフィルタ係数 $w(n)$ が更新される。フィルタ係数の更新を行うためのアルゴリズムを、適応アルゴリズムと呼ぶ。

2.2 RLS アルゴリズム

最も基本的で、かつ広く用いられている適応アルゴリズムに、LMS アルゴリズムがある。LMS は平均自乗誤差 $E[e(n)^2]$ を最小にするアルゴリズムである。係数更新はフィルタ係数ベクトルの初期値を

$$\mathbf{W}(0) = [0, 0, \dots, 0]^T \quad (5)$$

として行う。LMS アルゴリズムを応用したものに、入力信号を正規化した、NLMS (Normalized LMS) アルゴリズム [1] がある。

一方、最小自乗法に基づく適応アルゴリズムとして RLS アルゴリズムが知られている。RLS アルゴリズムは時刻 n までの信号系列に基づく重みつき自乗誤差和

$$\varphi(n) = \sum_{i=1}^n \lambda^{n-i} e(i)^2 \quad (6)$$

を最小化するアルゴリズムである。ここで、過去のデータを忘却するための係数である λ は忘却係数と呼ばれ、 $0 < \lambda \leq 1$ の実数値をとる。誤差は

$$\eta(n) = d(n) - \mathbf{X}(n)^T \mathbf{W}(n-1) \quad (7)$$

で与えられる。

RLS アルゴリズムでは、フィルタ係数ベクトル、 $\mathbf{X}(n)$ の自己相関行列の逆行列 $\mathbf{P}(n)$ の初期値を

$$\mathbf{W}(0) = [0, 0, \dots, 0]^T \quad (8)$$

$$\mathbf{P}(0) = \frac{1}{\delta} \mathbf{I} \quad (9)$$

とする。ここで、 δ は $0 < \delta \ll 1$ の実数、 \mathbf{I} は $M \times M$ の単位行列を表す。また、 M は適応フィルタのタップ数である。係数更新式は

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mathbf{K}(n)\eta(n) \quad (10)$$

$$\mathbf{K}(n) = \frac{\mathbf{P}(n-1)\mathbf{X}(n)}{\lambda + \mathbf{X}^T(n)\mathbf{P}(n-1)\mathbf{X}(n)} \quad (11)$$

$$\mathbf{P}(n) = \lambda^{-1}[\mathbf{P}(n-1) - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{P}(n-1)] \quad (12)$$

として表される。ここで、 $\mathbf{K}(n)$ はゲインベクトルと呼ばれる。RLS アルゴリズムは、フィルタ長を M としたとき、一回の更新に $O(M^2)$ の演算量が必要である。RLS アルゴリズムは、必要となる演算量が多いため、演算量の低減方法が提案されている [3], [8]。

2.3 ERLS-DCD アルゴリズム

ERLS-DCD アルゴリズムは RLS アルゴリズムよりも少ない演算量で、LMS、NLMS アルゴリズムよりも優れた収束特性を有するアルゴリズムである [3]。RLS アルゴリズムでは、フィルタの更新に正規方程式

$$\mathbf{R}(n)\mathbf{h}(n) = \beta(n) \quad (13)$$

を解く必要がある。ここで、 \mathbf{R} は入力信号の自己相関行列、 β は相互相関ベクトルを表す。ERLS-DCD アルゴリズムは式 (13) を直接計算するのではなく、 \mathbf{h} の近似解 $\hat{\mathbf{h}}$ 、差分 $\Delta\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1)$ を用いた以下の近似式、

$$\mathbf{R}(n)\Delta\mathbf{h}(n) = \beta_0(n) \quad (14)$$

$$\beta_0(n) = \lambda\mathbf{r}(n-1) + e(n)\mathbf{x}(n) \quad (15)$$

$$\mathbf{r}(n-1) = \beta_0(n-1) - \mathbf{R}(n-1)\Delta\hat{\mathbf{h}}(n-1) \quad (16)$$

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n) \quad (17)$$

$$\Delta\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1) \quad (18)$$

によりフィルタ係数を更新する．ここで， λ は忘却係数， $e(n)$ はフィルタの出力と所望信号との誤差， $\mathbf{r}(n)$ は時刻 $n-1$ における正規方程式の剰余ベクトルを表す．また，対称行列となる \mathbf{R} は，その対称性を利用することで更新する際の演算量を低減している．式 (19) の近似方程式の解法には DCD アルゴリズムを用いることができる [3]．DCD アルゴリズムは，毎時刻の更新回数 N_u を設定することが可能であり，更新回数を増やすと収束速度は速くなるが，演算量が増大する．また，DCD アルゴリズムは乗算がないため，ERLS-DCD アルゴリズムは， $O(M)$ の演算量での更新が可能である．表 1 に各アルゴリズムの演算量を示す． M_u は DCD アルゴリズムの更新回数を表す．

表 1: 演算量比較

	乗算	加算	除算
NLMS	$2M + 3$	$2M + 3$	1
RLS	$M^2 + 5M + 1$	$M^2 + 3M$	1
ERLS-DCD	$3M$	$2MM_u + 6M$	-

2.4 従来の可変忘却係数アルゴリズム

追従特性改善のために忘却係数を可変にした，可変忘却係数アルゴリズムが提案されている．

VFF-RLS アルゴリズム [5] は，誤差信号および雑音信号の分散を利用して忘却係数の値を変動させる．GVFF-RLS アルゴリズム [6] は，平均自乗誤差 (Mean Square Error) 解析を応用した勾配型の可変忘却係数アルゴリズムである．また，SVFF-RLS アルゴリズム [7] は，スパース性 (Sparsity) に基づく可変忘却係数アルゴリズムで，非定常環境での追従特性を改善したアルゴリズムである．

可変忘却係数を用いたこれらのアルゴリズムにおいて，フィルタ係数の更新は RLS アルゴリズムと同様の手順で行われる．また，ERLS アルゴリズムに対してもそれぞれ同様に応用できる．これらの従来法は，それぞれ追従特性の改善を実現しているが，忘却係数を可変にするための演算量増加が問題となる．

2.5 スパース性

提案法では，システム変動の指標として，スパース性を用いる [9]．システムのスパース性とは対象とするシステムのスパースの度合いを数値化したものと見なすことができる．システム \mathbf{h} のスパース性を評価する指標 $\xi(\mathbf{h})$ として

$$\xi(\mathbf{h}) = \frac{L}{L - \sqrt{L}} \left(1 - \frac{\|\mathbf{h}\|_1}{\sqrt{L}\|\mathbf{h}\|_2} \right) \quad (19)$$

が提案されている [9]．ここで， L はシステムのインパルス応答の長さである． $\|\mathbf{h}\|_1$ は

$$\|\mathbf{h}\|_1 = \sum_{n=0}^{L-1} |h_n| \quad (20)$$

で計算される L_1 ノルムを， $\|\mathbf{h}\|_2$ は

$$\|\mathbf{h}\|_2 = \sqrt{\sum_{n=0}^{L-1} h_n^2} \quad (21)$$

で計算される L_2 ノルムを表す． $\xi(\mathbf{h})$ の値は，ディラックのデルタ関数のような完全にスパースなシステムでは 1 となり，スパース性が低下するごとに 0 へと近づく．図 2 にスパースな系の例を示す．図 2 の系に対するスパース性 $\xi(\mathbf{h})$ は，0.9153 である．

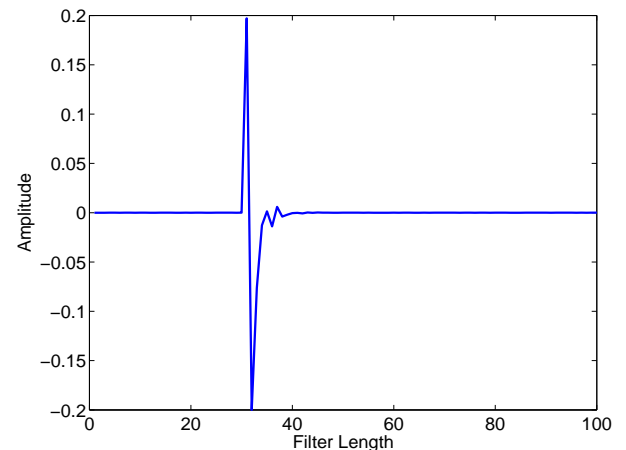


図 2: スパースなシステム

3 提案法

ここでは，提案法である RLS 型適応アルゴリズムの学習特性改善手法について説明する．提案法は RLS および ERLS-DCD アルゴリズムに適用することを想定する．

3.1 RLS アルゴリズムにおける提案法

図 3 に RLS アルゴリズムの忘却係数の値と収束特性，追従特性の関係の一例を示す．図 3 は，未知系のインパ

ルス応答の長さ 100, 適応フィルタのタップ数 100, 係数更新 4000 回目の時点で未知系が変化した場合を想定し, 忘却係数が $\lambda_{RLS} = 0.9999$ の場合と $\lambda_{RLS} = 0.9990$ の場合を比較したシミュレーション結果である.

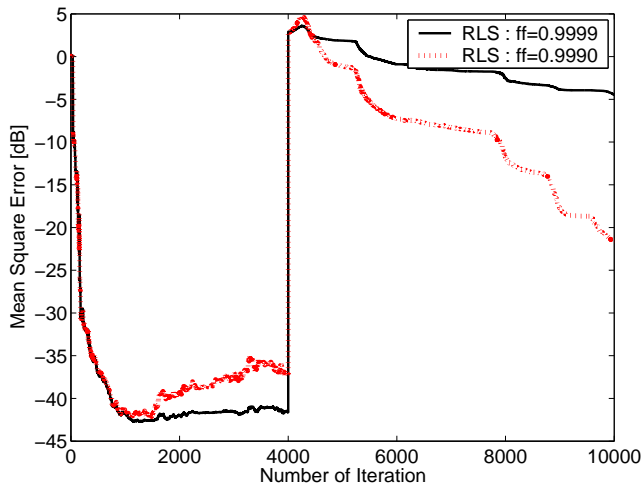


図 3: 忘却係数と収束特性, 追従特性の関係 (RLS)

RLS アルゴリズムに用いられる忘却係数は, 値が大きければ定常環境での残留誤差などの収束特性が向上するが, 未知系が変化した場合の追従特性は劣化する. そこで本稿では, 固定忘却係数により RLS アルゴリズムの追従特性改善法を提案する.

未知系がスパースな系であれば, 推定が進むと適応フィルタの係数もスパースとなり, $\xi(\mathbf{w})$, $\xi(\hat{\mathbf{h}})$ の値は増加する. 一方, 未知系が変化すると推定誤差が増加し, $\xi(\mathbf{w})$, $\xi(\hat{\mathbf{h}})$ の値は減少する [7]. このスパース性の減少をとらえ,

$$\mathbf{X}(n) = \mathbf{0} \quad (22)$$

$$\mathbf{W}(n) = \mathbf{0} \quad (23)$$

$$d(n) = 0 \quad (24)$$

$$\mathbf{P}(n) = \frac{1}{\delta} \mathbf{I} \quad (25)$$

と RLS アルゴリズムの各パラメータを初期化することによって, システム変化後も初期収束と同等の速度での追従が可能となる. 提案法における RLS アルゴリズムを使用した際のスパース性の変化を図 4 に示す.

3.2 ERLS-DCD アルゴリズムにおける提案法

図 5 に RLS アルゴリズムの忘却係数の値と収束特性, 追従特性の関係の一例を示す. 図 5 は, 未知系のインパルス応答の長さ 100, フィルタタップ数 100, 係数更新 4000 回目の時点で未知系が変化した場合を想定し, 忘却係数が $\lambda_{RLS} = 0.9999$ の場合と $\lambda_{RLS} = 0.9990$ の場合を比較したシミュレーション結果である.

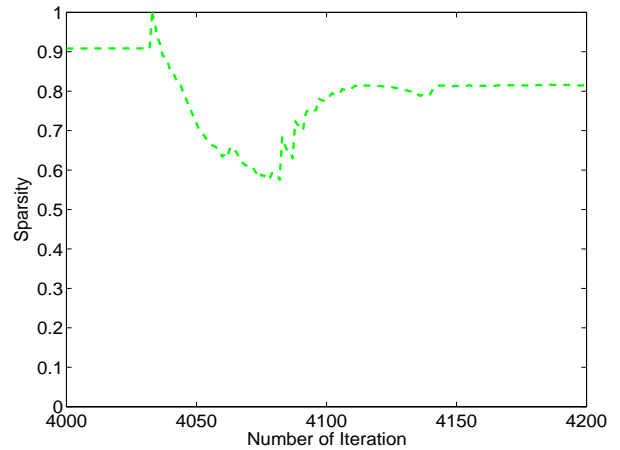


図 4: スパース性の変化 (RLS)

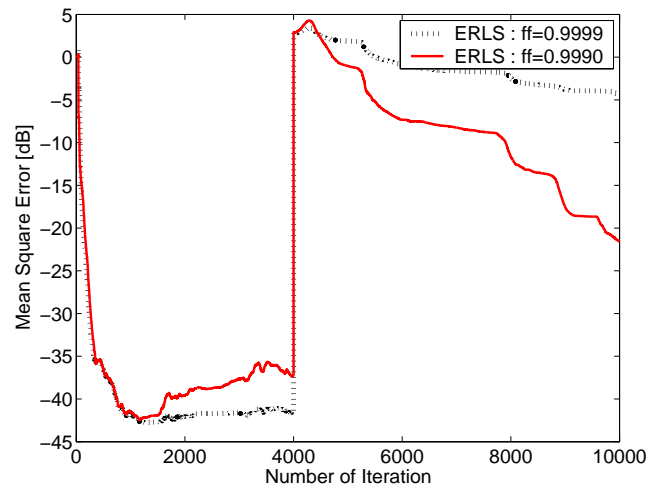


図 5: 忘却係数と収束特性, 追従特性の関係 (ERLS)

ERLS アルゴリズムに用いられる忘却係数も RLS アルゴリズムと同様に, 値が大きければ定常環境での残留誤差などの収束特性が向上するが, 非常環境での追従特性は劣化する. そこで本稿では, 固定忘却係数により ERLS アルゴリズムの追従特性改善法を提案する. スパース性の変化を検出した後, ERLS アルゴリズムの各パラメータを,

$$\mathbf{X}(n) = \mathbf{0} \quad (26)$$

$$\mathbf{W}(n) = \mathbf{0} \quad (27)$$

$$d(n) = 0 \quad (28)$$

$$\mathbf{R}(n) = \frac{1}{\delta} \mathbf{I} \quad (29)$$

$$\beta_o(n) = 0 \quad (30)$$

$$\Delta \hat{\mathbf{h}}(n) = \mathbf{0} \quad (31)$$

と初期化することによって、システム変化後も初期収束と同等の速度での追従が可能となる。提案法における ERLS アルゴリズムを使用した際のスパース性の変化を図 6 に示す。

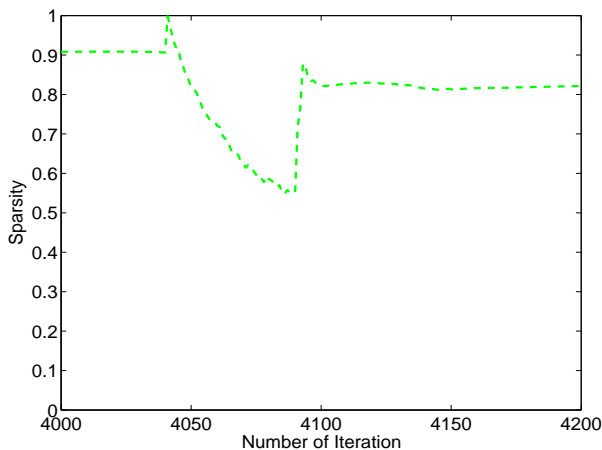


図 6: スパース性の変化 (ERLS)

4 シミュレーション

提案法の有効性を評価するため、システム同定のシミュレーションを行った。

4.1 シミュレーション条件

入力信号には図 7 で表されるサンプリング周波数 $8kHz$ の音声信号、付加雑音に $SNR = 40[db]$ で白色ガウス雑音を用い、未知システム、適応フィルタのタップ数は共に 100 とした。また、提案法の忘却係数は RLS, ERLS それぞれ、 $\lambda_{RLS} = 0.9999$, $\lambda_{ERLS} = 0.9990$ とした。評価は平均自乗誤差 (MSE) により行う。

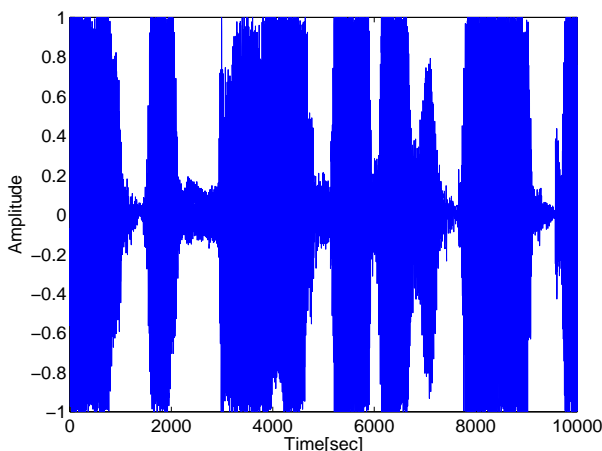


図 7: 入力に用いる音声信号

4.2 RLS アルゴリズムとの収束特性比較

まず、RLS アルゴリズムにおいて忘却係数 $\lambda_{RLS} = 0.9999$, $\lambda_{RLS} = 0.9990$ の時の追従特性と RLS アルゴリズムにおける提案法との追従特性を比較する。図 8 にシミュレーション結果を示す。

提案法ではシステム変化後の追従速度が初期収束と同等に改善していることがわかる。

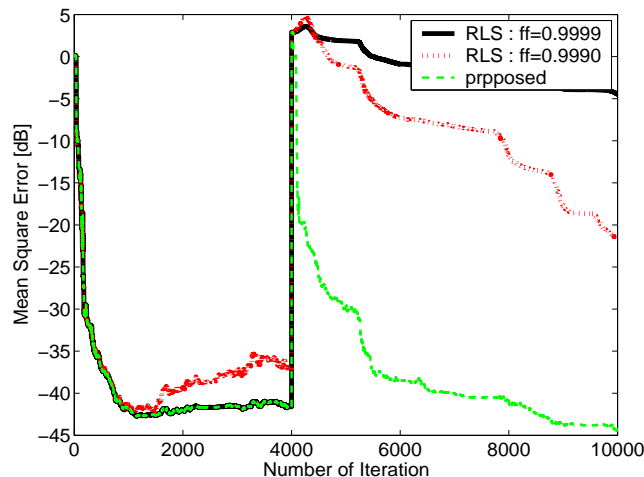


図 8: シミュレーション結果 (RLS)

4.3 ERLS アルゴリズムとの収束特性比較

次に、ERLS アルゴリズムにおいて忘却係数 $\lambda_{ERLS} = 0.9999$, $\lambda_{ERLS} = 0.9990$ の時の追従特性と ERLS アルゴリズムにおける提案法との追従特性を比較する。図 9 にシミュレーション結果を示す。

提案法ではシステム変化後の追従速度が初期収束と同等に改善していることがわかる。

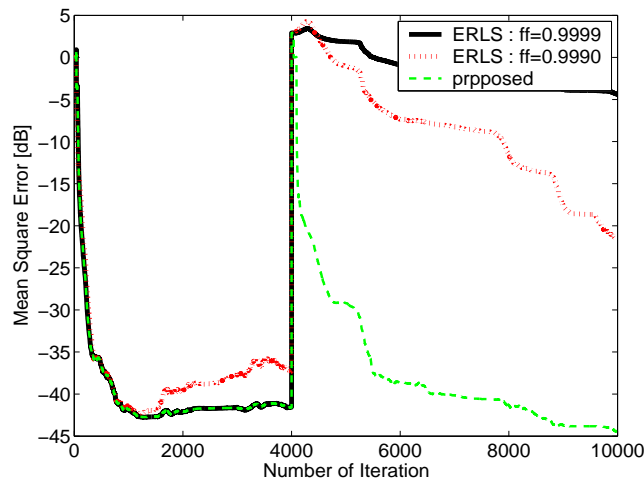


図 9: シミュレーション結果 (ERLS)

4.4 従来の可変忘却係数の手法との収束特性比較

最後に、NLMS アルゴリズム、VFF-ERLS アルゴリズム、GVFF-ERLS アルゴリズム、SVFF-ERLS アルゴリズムとの収束特性を比較する。図 10 にシミュレーション結果を示す。また、図 11 にスパース性の変化を示す。提案法は従来法に比べ、システム変化後の追従速度が最も速いことがわかる。

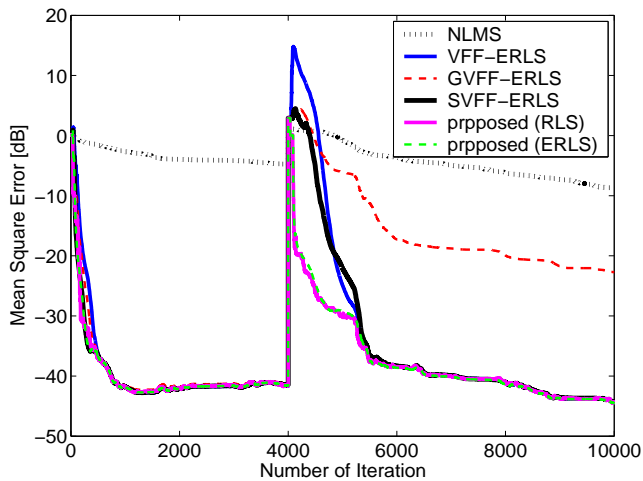


図 10: 従来法との比較

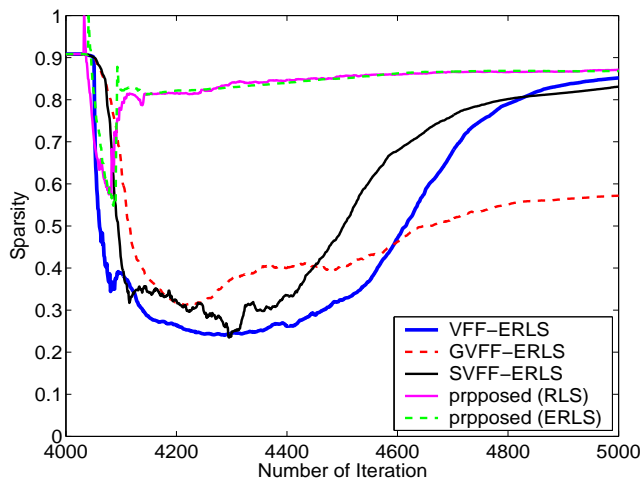


図 11: スパース性の変化の比較

5 おわりに

本稿では、忘却係数を大きい値で固定したまま、スパース性に基づき未知系の変動を推定し、フィルタ係数を初期化させることで、RLS 型適応アルゴリズムの追従特性が改善することを提案した。また、計算機シミュレーションによりその有効性を確認した。

参考文献

- [1] Dolald L.Duttweiler et al, "Proportionate Normalized Least-Mean-Squares Adaption in Echo Cancellers," IEEE Trans. Speech Audio Processing, vol.8, pp.508-518, September 2000.
- [2] A. H. Sayed, Fundamentals of Adaptive Filtering. Hoboken, NJ: Wiley, 2003.
- [3] Zakharov et al, "Low-Complexity RLS Algorithms Using Dichotomous Coordinate Descent Iterations" IEEE Trans. Signal Process.,VOL. 56, NO.7, July 2008.
- [4] Simon Haykin "Adaptive Filter Theory", Prentice Hall, 2001.
- [5] Constantin Paleologu, Jacob Benesty, and Silviu Ciochina, "A Robust Variable Forgetting Factor Recursive Least-Squares Algorithm for System Identification" IEEE Signal Process. Letters, VOL.15, pp.597-600, 2008.
- [6] Shu-Hung Leung and C. F. So, "Gradient-Based Variable Forgetting Factor RLS Algorithm in Time-Varying Environments" IEEE Trans. Signal Process., vol.53, no.8, pp.3141-3150, Aug 2005.
- [7] Kiyoshi NISHIKAWA and Junya HASEGAWA, "Variable Forgetting Factor Algorithm Based on Sparsity for RLS-type Adaptive Algorithms," Proc.APSIPA Annual Summit and Conference, pp.173-176, Biopolos, Singapore, December 2010.
- [8] Seokjin Lee, Jun-seok Lim and Koeng-Mo Sung, "A low-complexity AFF-RLS algorithm using a normalization technique", IEICE Electron. Express, Vol. 6, No. 24, pp.1774-1780, (2009) .
- [9] Y.Huang, J.Benesty, and J.Chen "Acoustic MIMO Signal Processing (Signal and Communication Technology)." Secaucus, NJ, USA:Springer-Verlag New York, Inc., 2006.