A Hybrid Approach for Sparse Adaptive Filters Under Highly Colored Inputs

Osamu Toda, Masahiro Yukawa, Shigenobu Sasaki, and Hisakazu Kikuchi
Department of Electrical and Electronic Engineering, Niigata University, Japan

***Abstract*— We address an adaptive filtering problem for sparse linear systems excited by highly colored input signals. A proportionate approach is known to accelerate the convergence speed by exploiting the sparseness of the systems, while a transform-domain approach is known to alleviate the decay of the convergence rate for highly colored inputs. We highlight the improved proportionate NLMS (IPNLMS) and transform-domain NLMS (TD-NLMS) algorithms. The present experimental results show that the gain of IPNLMS against TD-NLMS changes from positive to negative as the input auto-correlation becomes strong. We propose a hybrid approach of IPNLMS and TD-NLMS, taking the advantages of both algorithms by means of a time-variant convex combination of the two matrices employed by those algorithms. Numerical examples show the efficacy of the proposed algorithm.**

I. Introduction

This paper addresses an adaptive filtering problem in which the linear unknown system tends to be *sparse* (i.e., many of the coefficients tend to be nearly zero) and it is excited by colored inputs. Indeed, adaptive filters exploiting the expected *sparseness* of the system have received considerable attention [1–10]. One of the major line of researches regarding this topic is *proportionate-type adaptive filtering* proposed originally by Duttweiler [1]. In particular, optimal diagonal matrices (or optimal individual step size for each filter tap) in certain senses have been presented in [4,5]. Both works however are based on the assumption of white input signals, thus the algorithms are no longer optimal when applied to colored inputs.

On the other hand, the transform-domain least mean square (TD-LMS) algorithm has been proposed to ameliorate the convergence behavior of the LMS algorithm for colored inputs [11–13]. It is known that the TD-LMS algorithm offers a constant rate of convergence even for highly correlated input signals by operating a predetermined orthogonal transformation followed by power normalization. It is of great interests to investigate which of the two families (proportionate and transform-domain) of algorithms performs better for sparse linear systems excited by highly colored inputs.

In this paper, we consider the following particular algorithms from each family: the improved proportionate normalized LMS (IPNLMS) algorithm [3] and the transform-domain normalized LMS (TD-NLMS) algorithm, which is a normalized version of TD-LMS. To present a fair comparison between the algorithms, we introduce a measure of performance gain based on the geometric mean of the ratio between the mean squared errors (MSEs) of the algorithms. Our simulations clarify the tendency that, whereas IPNLMS has a significant gain against TD-NLMS for uncorrelated or weakly-correlated input signals, the gain decreases as the correlation becomes strong and eventually TD-NLMS defeats IPNLMS when the correlation reaches a certain level. Focusing on a particular plot of MSE learning curves for a highly colored input, it is shown that, although IPNLMS exhibits faster initial convergence than TD-NLMS, its speed of convergence slows down before reaching the steady state and it is overtaken by TD-NLMS. Through an error surface analysis, the phenomenon of the performance degradation of IPNLMS is clearly explained. Based on the above observations, we propose a hybrid approach taking the advantages of both algorithms. The way of the hybrid follows the simple but effective idea of the IPNLMS algorithm; essentially IPNLMS is a hybrid approach of the standard NLMS algorithm and PNLMS. The proposed algorithm is a hybrid of IPNLMS and TD-NLMS (With an abuse of notation, PROPOSED = IPNLMS + TD-NLMS = NLMS + PNLMS + TD-NLMS). Three positive-definite matrices associated respectively with NLMS, PNLMS, and TD-NLMS are convexly combined with time-variant coefficients, thereby exploiting the positive features of IPNLMS and TD-NLMS. The coefficients are controlled based on detection of the performance degradation point of IPNLMS, which relies on a smoothed squared error of the proposed algorithm. The algorithm enjoys fast initial convergence and keeps a nearly-constant rate of convergence even though the input signal has high auto-correlation. Numerical examples clearly show the great advantages of the proposed algorithm.

II. A Study of IPNLMS and TD-NLMS under Highly Colored Input

Throughout the paper, we let $\mathbb{R}$ and $\mathbb{N}$ denote the sets of all real numbers and non-negative integers, respectively. We denote by $(\cdot)^{\mathsf{T}}$ the transpose of a vector/matrix. Also we denote by $\|\cdot\|_1$ and $\|\cdot\|_2$ the ℓ_1 and ℓ_2 norms, respectively.

A. System Model and Conventional Algorithms

We consider the following linear system model (see Fig. 1):

$$d_k := z_k + n_k = \boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{h}^* + n_k,\ k \in \mathbb{N} \tag{1}$$

where $\boldsymbol{u}_k := [u_k, u_{k-1}, \cdots, u_{k-N+1}]^{\mathsf{T}} \in \mathbb{R}^N$ is the input vector of length N at time k with the input process $(u_k)_{k\in\mathbb{N}}$, $\boldsymbol{h}^* \in \mathbb{R}^N$ the unknown system (assumed *sparse*), $z_k := \boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{h}^* \in \mathbb{R}$, and $(n_k)_{k\in\mathbb{N}} \subset \mathbb{R}$ the noise process. The residual-error function for each sample data $(\boldsymbol{u}_k, d_k)$ is given

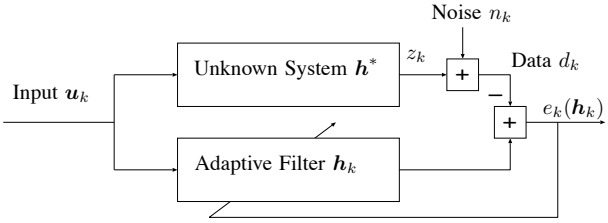


Fig. 1. A linear system model with an adaptive filter.

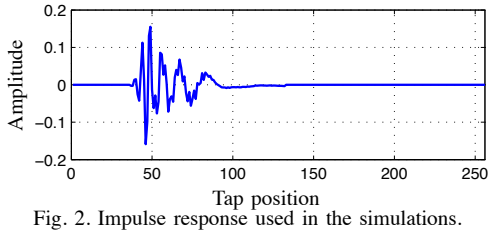


Fig. 2. Impulse response used in the simulations.

by $e_k(\boldsymbol{h}) := \boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{h} - d_k$, $\boldsymbol{h} \in \mathbb{R}^N$. The adaptive filter at time k is denoted by $\boldsymbol{h}_k := [h_k^{(1)}, h_k^{(2)}, \cdots, h_k^{(N)}]^{\mathsf{T}} \in \mathbb{R}^N$.

The update equations of the IPNLMS and TD-NLMS algorithms share the following form:

$$\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \mu \frac{e_k(\boldsymbol{h}_k)}{\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{A}_k\boldsymbol{u}_k}\boldsymbol{A}_k\boldsymbol{u}_k, \tag{2}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{N\times N}$ is a positive-definite matrix to be designed adequately according to some information on u_k and/or $\boldsymbol{h}^*$ for performance amelioration. The IPNLMS and TD-NLMS algorithms are obtained by designing $\boldsymbol{A}_k$ respectively as follows.

i) IPNLMS: $\boldsymbol{A}_k := (1-\eta)\boldsymbol{A}_k^{(1)} + \eta\boldsymbol{A}_k^{(2)}$, $\eta \in [0,1]$, with

$$\boldsymbol{A}_k^{(1)} = \frac{1}{N}\boldsymbol{I}, \tag{3}$$

$$\boldsymbol{A}_k^{(2)} = \frac{\boldsymbol{G}_k}{||\boldsymbol{h}_k||_1}, \tag{4}$$

where $\boldsymbol{I}$ denotes the identity matrix, and $\boldsymbol{G}_k := \mathrm{diag}(|h_k^{(1)}|, |h_k^{(2)}|, \cdots, |h_k^{(N)}|)$.

ii) TD-NLMS: $\boldsymbol{A}_k := \boldsymbol{A}_k^{(3)} := \boldsymbol{Q}^{\mathsf{T}}\boldsymbol{\Lambda}_k^{-1}\boldsymbol{Q}$. Here $\boldsymbol{Q} \in \mathbb{R}^{N\times N}$ is an orthogonal transformation matrix (e.g. the DCT matrix), and $\boldsymbol{\Lambda}_k := \mathrm{diag}\left(\lambda_k^{(1)}, \lambda_k^{(2)}, \cdots, \lambda_k^{(N)}\right)$, where $\lambda_k^{(i)}, (i = 1, 2, \cdots, N)$ is updated recursively for some $\lambda_0^{(i)} \geq 0$ and $\nu \in (0,1)$ as $\lambda_{k+1}^{(i)} := \nu\lambda_k^{(i)} + (1-\nu)|\check{u}_k^{(i)}|^2$ with $[\check{u}_k^{(1)}, \check{u}_k^{(2)}, \cdots, \check{u}_k^{(N)}]^{\mathsf{T}} := \boldsymbol{Q}\boldsymbol{u}_k$.

B. Comparison of IPNLMS and TD-NLMS

We compare the performance of the IPNLMS and TD-NLMS algorithms by simulations in an echo cancellation problem. We employ $\boldsymbol{h}^*$ shown in Fig. 2; the sparsity of the $\boldsymbol{h}^*$ is $\xi(\boldsymbol{h}^*) = 0.6796$, where we adopt the sparsity measure [14]: $\xi(\boldsymbol{h}) := \frac{N}{N-\sqrt{N}}\left(1 - \frac{\|\boldsymbol{h}\|_1}{\sqrt{N}\|\boldsymbol{h}\|_2}\right) \in [0,1]$, $\boldsymbol{h} \in \mathbb{R}^N$.

The input signals are generated by passing a white Gaussian noise through a filter whose transfer function is given as follows:

$$H_\alpha(z) := \frac{1 - \alpha z^{-2}}{1 - 1.70223\ \alpha z^{-1} + 0.71902\ \alpha z^{-2}},\ z \in \mathbb{C}$$

where $\alpha \geq 0$ controls the coloredness ($\alpha = 0$ gives the all pass filter, i.e., $H_0(z) = 1$, producing white inputs). At the moment, we set $\alpha = 1$ which gives the USASI signal known to be highly colored. The other conditions are as follows: $N = 256$, n_k is additive white Gaussian noise with SNR $= 30$ dB. An MSE value is computed by taking an arithmetic average over 200 independent trials. For both algorithms, we set $\mu = 0.1$. For IPNLMS, we set $\eta = 0.5$. For TD-NLMS, we set $\nu = 0.99$ and $\lambda_0^{(i)} = 0$. Figure 3 depicts the results. It is seen that IPNLMS exhibits faster initial convergence than TD-NLMS, but becomes slow down when it achieves the MSE value around -20 dB. On the other hand, TD-NLMS keeps a constant rate of convergence over a large number of iterations, reaching the steady state in a smaller number of iterations than IPNLMS.

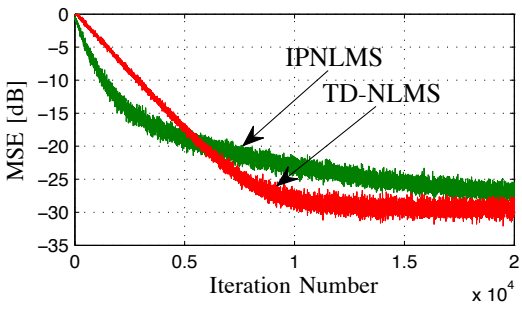


Fig. 3. MSE learning curves of IPNLMS and TD-NLMS under SNR $= 30$ dB for colored inputs.

The performance deterioration of IPNLMS is caused by the high auto-correlation of the input signal. To verify this, we perform additional simulations under different conditions of auto-correlation. By increasing the α value from 0 to 1, the eigenvalue spread [*1] of $\boldsymbol{R}$ increases as shown in Fig. 4, where $\boldsymbol{R} := \boldsymbol{E}\{\boldsymbol{u}_k\boldsymbol{u}_k^{\mathsf{T}}\}$. We would like to define the gain of IPNLMS (against TD-NLMS) to illustrate its dependency on the eigenvalue spread. If we simply take an average difference over iterations between the MSEs of the two algorithms, the effect of the initial phase becomes dominant. For instance, let us take a fresh look at the curves of IPNLMS and TD-NLMS in Fig. 3. Although IPNLMS performs better in the initial phase, TD-NLMS performs even better after the middle phase. Nevertheless, an average difference between the MSEs would imply that IPNLMS would outperform TD-NLMS. This clearly shows that the average difference is an inappropriate indicator for a performance gain. Therefore, we consider the geometric mean of the ratios between the MSEs of the two algorithms. Specifically, the gain of Algorithm A against Algorithm B is defined as follows:

$$\gamma(\mathrm{A},\mathrm{B}) := \prod_{k=1}^{K}\left(\frac{\mathrm{MSE}_k^{\mathrm{B}}}{\mathrm{MSE}_k^{\mathrm{A}}}\right)^{1/K}$$

[*1] The eigenvalue spread is the condition number defined with the spectral norm.

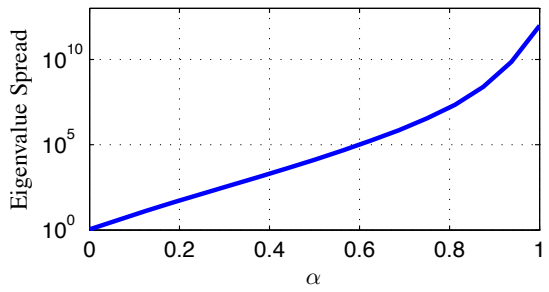


Fig. 4. Eigenvalue spread versus α.

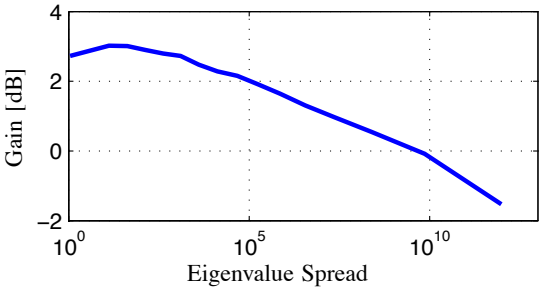


Fig. 5. Gain of IPNLMS against TD-NLMS.

where

$$\mathrm{MSE}_k := \frac{1}{T}\sum_{i=1}^{T} e_{k,i}^2$$

with $e_{k,i}$ denoting the output error at the kth iteration in the ith trial (T stands for the number of trials).

Figure 5 plots the gain of IPNLMS against the variation of the eigenvalue spread (which reflects the coloredness of the inputs). When the eigenvalue spread is close to unity (which means that an input signal is white), the gain of IPNLMS seems to be high. In contrast, when the eigenvalue spread becomes large, the gain drops down to minus values. We will present in the following subsection an intuitive explanation to this observed degradation phenomenon.

C. Error Surface Analysis

The cause of the phenomena can be understood through the error surface analysis presented below (cf. [15]). Consider the contours of the surface of the MSE function

$$f_u(\boldsymbol{h}) := \boldsymbol{h}^\mathsf{T}\boldsymbol{R}\boldsymbol{h} - 2\boldsymbol{h}^\mathsf{T}\boldsymbol{p} + E\{d_k^2\},$$

where $\boldsymbol{p} := E\{\boldsymbol{u_k}d_k\}$. The shape of the contours depends on the second-order term $\boldsymbol{h}^\mathsf{T}\boldsymbol{R}\boldsymbol{h}$.

For a while, let us consider the case of white inputs. In this case, $\boldsymbol{R} = \sigma_u^2\boldsymbol{I}$ with $\sigma_u^2 := E\{u_k^2\}$, yielding spherical contours (see Fig. 6(a)). We now make a coordinate transformation and see how the shape of the contours is changed. For simplicity we drop the time index k from $\boldsymbol{A}_k$ (one can regard $\boldsymbol{A}$ as an 'ideal' matrix that each of TD-NLMS/IPNLMS tries to approximate). Left-multiplying both-sides of (2) by $\boldsymbol{A}^{-1/2}$ yields

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \mu\frac{\widetilde{e}_k(\boldsymbol{w}_k)}{\boldsymbol{v}_k^\mathsf{T}\boldsymbol{v}_k}\boldsymbol{v}_k \tag{5}$$

where $\boldsymbol{w}_k := \boldsymbol{A}^{-1/2}\boldsymbol{h}_k$, $\boldsymbol{v}_k := \boldsymbol{A}^{1/2}\boldsymbol{u}_k$, and $\widetilde{e}_k(\boldsymbol{w}) := \boldsymbol{v}_k^\mathsf{T}\boldsymbol{w} - d_k$, $\boldsymbol{w} \in \mathbb{R}^N$. Note that (5) expresses the same update equation as (2) in the $\boldsymbol{w}$-coordinate, and it is the standard NLMS algorithm for the input-output pair $(\boldsymbol{v}_k, d_k)$. The MSE function in the $\boldsymbol{w}$-coordinate is defined as

$$f_v(\boldsymbol{w}) := \boldsymbol{w}^\mathsf{T}\boldsymbol{R}_v\boldsymbol{w} - 2\boldsymbol{w}^\mathsf{T}\boldsymbol{p}_v + E\{d_k^2\},$$

where $\boldsymbol{R}_v := E\{\boldsymbol{v}_k\boldsymbol{v}_k^\mathsf{T}\} = \boldsymbol{A}^{1/2}\boldsymbol{R}\boldsymbol{A}^{1/2}$ and $\boldsymbol{p}_v := E\{\boldsymbol{v_k}d_k\}$. The contours in the $\boldsymbol{w}$-coordinate are therefore determined by $\boldsymbol{w}^\mathsf{T}\boldsymbol{R}_v\boldsymbol{w}$.

Let us consider the specific case of IPNLMS for $N = 2$ and $|h_1^*| \ll |h_2^*|$. We assume that $\boldsymbol{A} = \frac{1}{||\boldsymbol{h}^*||_1}\mathrm{diag}(|h_1^*|, |h_2^*|)$, which is obtained when $\eta = 1$, $h_k^{(1)} = h_1^*$, and $h_k^{(2)} = h_2^*$ (the ideal case). In the $\boldsymbol{w}$-coordinate, the minimizer of f_v is given by $\boldsymbol{w}^* := \boldsymbol{A}^{-1/2}\boldsymbol{h}^* = ||\boldsymbol{h}^*||_1^{1/2}[\mathrm{sgn}(h_1^*)|h_1^*|^{1/2}, \mathrm{sgn}(h_2^*)|h_2^*|^{1/2}]^\mathsf{T}$. Meanwhile, the contours of f_v in the $\boldsymbol{w}$-coordinate are ellipsoids and the ratio of the larger to shorter radii is given by $|h_2^*|^{1/2}/|h_1^*|^{1/2}$. This implies that the contours are compressed in the direction from the origin toward $\boldsymbol{h}^*$ due to the $\boldsymbol{A}$ matrix, which makes the slope from the origin (which is typically the initial filter $\boldsymbol{h}_0$) steep. This is an interpretation of the mechanism of the faster convergence of IPNLMS compared to the standard NLMS algorithm.

Now we turn our attention to the case of colored inputs. In this case, the shape of the contours is distorted due to the large eigenvalue spread of $\boldsymbol{R}$ (see Fig. 6(b)). Thus, there is no guarantee that IPNLMS reshapes the contours desirably. This clearly explains the performance degradation phenomenon of IPNLMS observed in Section II-B. On the other hand, it is known that, for some types of inputs, the transform-domain algorithm with an adequate orthogonal transform has the property of whitening [12,13]. This implies that the $\boldsymbol{A}_k$ matrix employed by TD-NLMS reshapes ellipsoidal contours into nearly-spherical ones (see Fig. 6(d)) [12]. This agrees with the constant rate of convergence of TD-NLMS. It should be mentioned that the transformed $\boldsymbol{w}^* := (\boldsymbol{A}_k^{(3)})^{-1/2}\boldsymbol{h}^*$ is no longer sparse in general.

III. Proposed Adaptive Algorithm

Below are observations obtained from the simulation results presented in the previous section.

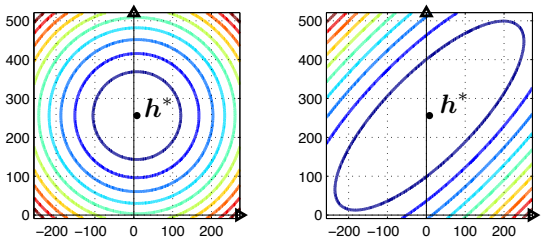
- IPNLMS is more effective for fast initial convergence.
- TD-NLMS converges faster from the middle phase.

This leads to the idea of a hybrid approach for taking the advantages of the two algorithms. To construct such a hybrid algorithm, we exploit the simple but effective idea of using a convex combination of positive definite matrices, as used in IPNLMS.

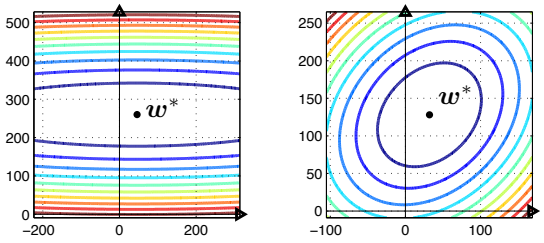
The proposed algorithm employs the following matrix:

$$\boldsymbol{A}_k := \sum_{i=1}^{3}\omega_k^{(i)}\boldsymbol{A}_k^{(i)} \tag{6}$$

where $\omega_k^{(i)} \geq 0$, $i = 1, 2, 3$, satisfies $\sum_{i=1}^{3}\omega_k^{(i)} = 1$.



(a) white input case: $\boldsymbol{h}$-coordinate (b) colored input case: $\boldsymbol{h}$-coordinate



(c) white input case: $\boldsymbol{w}$-coordinate for IPNLMS (d) colored input case: $\boldsymbol{w}$-coordinate for TD-NLMS

Fig. 6. The contours of MSE in the $\boldsymbol{h}$- and $\boldsymbol{w}$- coordinates.

Specifically, the proposed algorithm controls $\omega_k^{(i)}$ from the following aspects:

i) According to the aforementioned observation, large weights should be assigned to $\boldsymbol{A}_k^{(1)}$ and $\boldsymbol{A}_k^{(2)}$ in the initial phase, while the large weights should be shifting to $\boldsymbol{A}_k^{(3)}$ when the convergence speed slows down.

ii) The analysis in [16] tells us that the fluctuations of the metric $\boldsymbol{A}_k$ should be sufficiently small for the algorithm to converge.

The second aspect suggests that the $\omega_k^{(i)}$ should be changing slowly. To detect the timing of the convergence deterioration, an average amount of the squared output-error is exploited. The criterion for the detection is given as follows:

$$q_k := \frac{1}{m}\sum_{i=1}^{m} \tilde{p}_{k-i+1} \tag{7}$$

where

$$\tilde{p}_k := \frac{p_k}{\max\{p_1, p_2, \cdots, p_k\}} \tag{8}$$

$$p_{k+1} := (1-\tau)p_k + \tau e_k^2 \tag{9}$$

with $p_0 = 0$ and $0 < \tau \leq 1$. The denominator in (8) is introduced for the sake of normalization. Since the graphs of $\tilde{p}_k$ still have many spikes, we further take a time average of $\tilde{p}_k$. The $\omega_k^{(i)}$s are controlled as follows:

$$\omega_k^{(1)} := \begin{cases} \omega^{(1)} & \text{if } q_k \geq \delta \\ \max\{\omega_{k-1}^{(1)} - \omega_0^{(1)}/n_1, 0\}, & \text{otherwise} \end{cases}$$

$$\omega_k^{(2)} := \begin{cases} \omega^{(2)} & \text{if } q_k \geq \delta \\ \max\{\omega_{k-1}^{(2)} - \omega_0^{(2)}/n_2, 0\}, & \text{otherwise} \end{cases}$$

$$\omega_k^{(3)} := 1 - \omega_k^{(1)} - \omega_k^{(2)}, \tag{10}$$

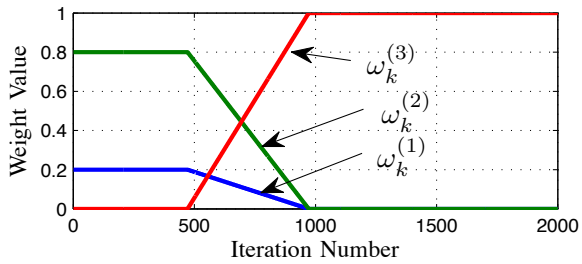


Fig. 7. The assignments of $\omega_k^{(i)}$s.

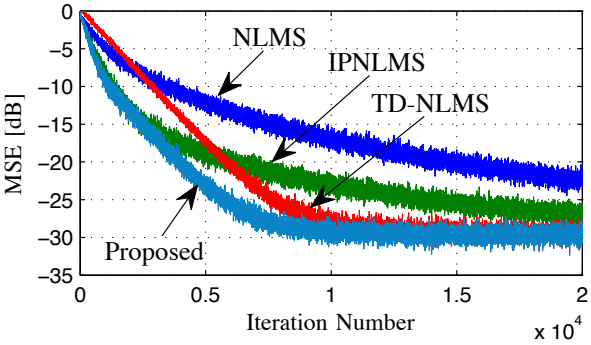


Fig. 8. MSE learning curves of the NLMS, IPNLMS, TD-NLMS, and proposed algorithms under SNR = 30 dB.

where $\omega^{(1)}, \omega^{(2)} \in (0,1)$, and $n_1, n_2 > 0$ determines the period of weight transition.

IV. Numerical Examples

This section shows the advantages of the proposed algorithm over NLMS, IPNLMS, and TD-NLMS. The simulation settings are exactly the same as in Fig. 3 (SNR = 30 dB). For the proposed algorithm, the step size is set to $\mu = 0.1$. Figure 7 plots the change of each $\omega_k^{(i)}$, where we set $\tau = 0.01$, $m = 100$, $\omega^{(1)} = 0.2$, $\omega^{(2)} = 0.8$, $\omega_0^{(3)} = 0$, $n_1 = n_2 = 500$ and $\delta = 0.5$ (This gave the best performance in our extensive experiments).

The transition behavior of the $\omega_k^{(i)}$s is depicted in Fig. 7. Figure 8 illustrates the simulation results under $\alpha = 1$ and SNR = 30 dB. It seen that the proposed algorithm achieves the initial convergence as fast as IPNLMS and its rate of convergence does not severely degraded unlike IPNLMS due to the successful control of $\omega_k^{(i)}$. We plot the gains of the proposed algorithm in Fig. 9. We can see that the proposed algorithm attains positive gains against IPNLMS for the eigenvalue spread larger than $10^{3.32}$. The gain grows as the eigenvalue spread increases. On the other hand, the gain against TD-NLMS is approximately 2 dB over the whole range of the eigenvalue spread.

V. Conclusion

This paper has shown that the gain of IPNLMS against TD-NLMS changes from positive to negative as the eigenvalue spread of input covariance matrix increases. We have proposed a hybrid approach of IPNLMS and TD-NLMS, taking the advantages of both algorithms by means of a time-variant convex combination of the two matrices. We have presented a

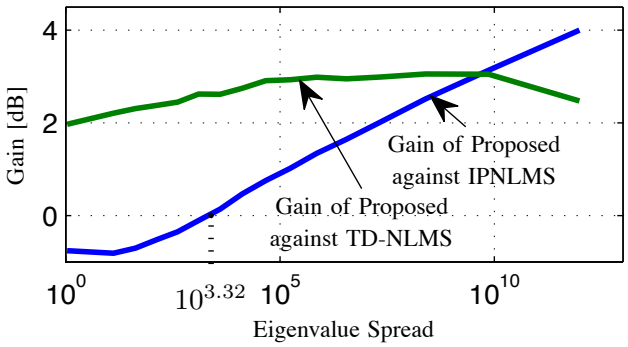


Fig. 9. Gains of the proposed algorithm.

technique to control the coefficients of the convex combination for exploiting the positive feature of IPNLMS in the initial phase and, at the same time, the positive feature of TD-NLMS from the middle phase. Numerical examples have shown that the proposed algorithm attains the initial convergence as fast as IPNLMS and keeps the same speed until it reaches the steady state even for highly colored input signals. We finally stress that the present work differs from the previous studies on sparse adaptive filters, as coloredness of input signals is taken into account in the algorithm construction.

ACKNOWLEDGMENT

This work was conducted under a contract of research and development for radio resource enhancement, organized by the Ministry of Internal Affairs and Communications, Japan.

REFERENCES

[1] D. L. Duttweiler. Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Trans. Speech Audio Processing*, 8(5):508–518, Sept. 2000.

[2] S. L. Gay. An efficient fast converging adaptive filter for network echo cancellation. In *Proc. Asilomar Conf. Signals, Syst., Comput.*, pages 394–398, 1998.

[3] J. Benesty and S. L. Gay. An improved PNLMS algorithm. In *Proc. IEEE ICASSP*, pages 1881–1884, 2002.

[4] H. Deng and M. Doroslovački. Proportionate adaptive algorithms for network echo cancellation. *IEEE Trans. Signal Processing*, 54(5):1794–1803, May 2006.

[5] K. T. Wagner and M. Doroslovački. Proportionate-type NLMS algorithms based on maximization of the joint conditional PDF for the weight deviation vector. In *Proc. IEEE ICASSP*, pages 3738–3741, 2010.

[6] Y. Gu, J. Jin, and S. Mei. ℓ_0 norm constraint LMS algorithm for sparse system identification. *IEEE Signal Processing Lett.*, 16(9):774–777, Sep. 2009.

[7] Y. Chen, Y. Gu, and A. O. Hero. Sparse LMS for system identification. In *Proc. IEEE ICASSP*, pages 3125–3128, 2009.

[8] D. Angelosante, J. A. Bazerque, and G. B. Giannakis. Online adaptive estimation of sparse signals: where RLS meets the ℓ_1-norm. *IEEE Trans. Signal Processing*, 58(7):3436–3447, Jul. 2010.

[9] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada. A sparse adaptive filtering using time-varying soft-thresholding techniques. In *Proc. IEEE ICASSP*, pages 3734–3737, 2010.

[10] K. Slavakis, Y. Kopsinis, and S. Theodoridis. Adaptive algorithm for sparse system identification using projections onto weighted ℓ_1 balls. In *Proc. IEEE ICASSP*, pages 3742–3745, 2010.

[11] S. S. Narayan, A. M. Peterson, and M. J. Narasimha. Transform domain LMS algorithm. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-31(3):609–615, Jun. 1983.

[12] D. F. Marshall, W. K. Jenkins, and J. J. Murphy. The use of orthogonal transforms for improving performance of adaptive filters. *IEEE Trans. Circuits and Systems*, 36(4):474–484, Apr. 1989.

[13] F. Beaufays. Transform-domain adaptive filters: an analytical approach. *IEEE Trans. Signal Processing*, 43(2):422–431, Feb. 1995.

[14] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Machine Learning Research*, 5:1457–1469, 2004.

[15] M. Yukawa. Krylov-proportionate adaptive filtering techniques not limited to sparse systems. *IEEE Trans. Signal Processing*, 57(3):927–943, Mar. 2009.

[16] M. Yukawa and I. Yamada. A unified view of adaptive variable-metric projection algorithms. *EURASIP J. Advances in Signal Processing*, 2009, Article ID 589260, 13 pages, 2009.