AUTOMATIC SHRINKAGE TUNING ROBUST TO INPUT CORRELATION FOR SPARSITY-AWARE ADAPTIVE FILTERING

Kwangjin Jeong†, Masahiro Yukawa†,‡, Masao Yamagishi⋆, Isao Yamada⋆

† Dept. of Electronics and Electrical Engineering, Keio University, JAPAN
‡ Center for Advanced Intelligence Project, RIKEN, JAPAN
⋆ Dept. of Information and Communications Engineering, Tokyo Institute of Technology, JAPAN

ABSTRACT

We propose a novel automatic shrinkage tuning technique for the adaptive proximal forward-backward splitting (APFBS) algorithm. The shrinkage tuning aims to choose an appropriate value of the shrinkage parameter and achieve minimal system mismatch as possible. The system mismatch is approximated based on time-averaged second-order statistics. Numerical examples show that the proposed method achieves performance fairly close to that with a manually chosen shrinkage parameter for colored input signals at some signal to noise ratio (SNR).

Index Terms— Sparsity-aware adaptive filter, automatic parameter tuning, adaptive proximal forward-backward splitting algorithm

1. INTRODUCTION

In recent years, it has been known that exploiting sparsity of a target system leads to higher accuracy in adaptive filtering. The sparsity of the system to be estimated has been observed and exploited in many applications including echo cancellation, active noise control and network channel estimation [1–15]. For a linear model, the Normalized Least Mean Squares (NLMS) algorithm [16] is one of the major adaptive filtering algorithms and its many sparsity-aware versions have been proposed [1–8]. Among them, we focus on the Adaptive Proximal Forward-Backward Splitting (APFBS) algorithm [6] in this paper. APFBS is an adaptive scheme which minimizes the sum of a smooth convex function and a nonsmooth convex function. To implement it for a sparsity-aware adaptive filter, one can choose a quadratic error function and the ℓ_1-norm as a smooth term and a nonsmooth term, respectively, so that APFBS considers an ℓ_1-norm regularized least squares problem at each iteration. In this case, the APFBS algorithm consists of two steps: (i) the gradient descent step, which is equivalent to NLMS, and (ii) the shrinkage step based on the proximity operator of the ℓ_1-norm.

The second step of the APFBS algorithm involves a shrinkage parameter, which controls the magnitude of the update and the performance of the APFBS algorithm. If one choose an optimal value of the shrinkage parameter, the algorithm works much better than the NLMS algorithm. However, the optimal value of the shrinkage parameter does not appear extensively. It has been only known to be determined by many factors including statistical properties of the input and noise, which are often unknown in most situations. With a roughly chosen shrinkage parameter, the APFBS algorithm can yield a poor estimation of the target system.

This work was supported by JSPS Grants-in-Aid (15K06081, 15K13986, 15H02757).

To avoid an inappropriate setting of the shrinkage parameter and improve the performance of the algorithm, some automatic shrinkage tuning methods have been studied. Yamagishi *et al.* have proposed approaches to automatic shrinkage tuning based on an instantaneous approximation of a mean squared error [17], and a system mismatch [18]. To get a fine approximation, they have assumed that the noise is under an i.i.d. zero-mean normal distribution with known variance, which is utilized in their shrinkage tuning methods.

In this paper, we present a novel automatic shrinkage tuning technique under zero-mean additive noise with unknown variance. We use time-averaged second-order statistics to achieve minimal system mismatch as possible. Then, a piecewise quadratic cost function on the shrinkage parameter is obtained and we minimize it by comparing the piecewise optima. The proposed shrinkage tuning takes advantage especially under a colored input signal, as shown by numerical examples.

2. PRELIMINARIES

We consider the following linear adaptive filtering model:

$$d_n = \boldsymbol{u}_n^\mathsf{T} \boldsymbol{h}^* + \epsilon_n, \ n \in \mathbb{N}, \tag{1}$$

where $\boldsymbol{u}_n \in \mathbb{R}^m$ is the input, $d_n \in \mathbb{R}$ is the output, $\boldsymbol{h}^* \in \mathbb{R}^m$ is the target system to be estimated, and $\epsilon_n \in \mathbb{R}$ is zero-mean additive noise. Note that the zero-mean assumption is the only assumption that is made on the noise in this paper. The input $\boldsymbol{u}_n$ and output d_n are observable. It is widely accepted in signal processing community to assume that (i) the input $\boldsymbol{u}_n$ and the noise ϵ_n are independent to each other, and (ii) the norm of the input $\boldsymbol{u}_n$ does not change drastically at each iteration [19, 20]. Under these assumptions, we have

$$E[\epsilon_n \boldsymbol{u}_n] = E[\epsilon_n]E[\boldsymbol{u}_n] = \boldsymbol{0}, \tag{2}$$

and

$$E\left[\frac{\epsilon_n \boldsymbol{u}_n}{\|\boldsymbol{u}_n\|_2^2}\right] \approx \frac{E[\epsilon_n \boldsymbol{u}_n]}{E[\|\boldsymbol{u}_n\|_2^2]} = \boldsymbol{0}, \tag{3}$$

where $\|\boldsymbol{u}\|_2 := \sqrt{\sum_{i=1}^m u_i^2}$ is the ℓ_2-norm of a vector $\boldsymbol{u} := [u_1, \ u_2, \ \cdots, \ u_m]^\mathsf{T} \in \mathbb{R}^m$.

It has been known that the APFBS algorithm (See Algorithm 1) is known to estimate $\boldsymbol{h}^*$ with higher accuracy than the NLMS algorithm when the target system $\boldsymbol{h}^*$ is sparse [6]. Note here that $\mathrm{sgn}(\cdot)$ denotes the signum function, and each vector $\boldsymbol{e}_i$ $(i \in \{1, 2, \cdots, m\})$ stands for the unit vector whose i-th entry is one and the others are zero. The vector $\boldsymbol{w}_n := [w_{n,1}, \ w_{n,2}, \ \cdots, w_{n,m}]^\mathsf{T} \in \mathbb{R}^m$ is the weighting vector whose entries are given as follows:

$$w_{n,i} := \frac{1}{|g_{n,i}| + \nu}, \tag{4}$$

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

Algorithm 1. APFBS [6]
0. $\boldsymbol{h}_1 = \boldsymbol{0},\ \mu \in (0, 2)$: step size, $\lambda \geq 0$: shrinkage parameter
1. $\boldsymbol{g}_n = \boldsymbol{h}_n - \mu \dfrac{\boldsymbol{u}_n^\top \boldsymbol{h}_n - d_n}{\|\boldsymbol{u}_n\|_2^2} \boldsymbol{u}_n$
2. $\boldsymbol{h}_{n+1} = \sum_{i=1}^{m} \operatorname{sgn}(g_{n,i}) \max\{|g_{n,i}| - \mu\lambda w_{n,i}, 0\} \boldsymbol{e}_i$

where $\nu > 0$ is a sufficiently small constant which prevents the denominator from being exactly zero.

The first step of Algorithm 1

$$\boldsymbol{g}_n := [g_{n,1},\ g_{n,2},\ \cdots,\ g_{n,m}]^\top = \boldsymbol{h}_n - \mu \frac{\boldsymbol{u}_n^\top \boldsymbol{h}_n - d_n}{\|\boldsymbol{u}_n\|_2^2} \boldsymbol{u}_n \quad (5)$$

is the same with the update equation of the NLMS algorithm. On the other hand, the second step

$$\boldsymbol{h}_{n+1} = \sum_{i=1}^{m} \operatorname{sgn}(g_{n,i}) \max\{|g_{n,i}| - \mu\lambda w_{n,i}, 0\} \boldsymbol{e}_i \quad (6)$$

can be seen as the proximity operator of a weighted ℓ_1-norm

$$\|\boldsymbol{h}\|_{1,\boldsymbol{w}_n} := \sum_i w_{n,i} |h_i|. \quad (7)$$

Hence, Algorithm 1 can be regarded as a time-varying extension of the proximal forward-backward splitting method [21]. The nth iteration of Algorithm 1 is based on the following regularized least squares problem:

$$\min_{\boldsymbol{h} \in \mathbb{R}^m} \frac{1}{2} \frac{(\boldsymbol{u}_n^\top \boldsymbol{h} - d_n)^2}{\|\boldsymbol{u}_n\|_2^2} + \lambda \|\boldsymbol{h}\|_{1,\boldsymbol{w}_n}. \quad (8)$$

Remark 1. *The second step of Algorithm 1 is based on the proximity operator [21]:*

$$\begin{aligned} \boldsymbol{h}_{n+1} &= \operatorname*{argmin}_{\boldsymbol{h}} \left[\frac{1}{2} \|\boldsymbol{g}_n - \boldsymbol{h}\|_2^2 + \mu\lambda \|\boldsymbol{h}\|_{1,\boldsymbol{w}_n} \right] \\ &= \operatorname{prox}_{\mu\lambda\|\cdot\|_{1,\boldsymbol{w}_n}}(\boldsymbol{g}_n) \\ &= \operatorname{prox}_{\mu\lambda\|\cdot\|_{1,\boldsymbol{w}_n}}\big(\boldsymbol{h}_n - \mu \nabla \varphi_n(\boldsymbol{h}_n)\big), \end{aligned} \quad (9)$$

where $\varphi_n(\cdot) := (\boldsymbol{u}_n^\top(\cdot) - d_n)^2 / \|\boldsymbol{u}_n\|_2^2$. Denoting by I the identity operator, the composite operator $T_n := \operatorname{prox}_{\mu\lambda\|\cdot\|_{1,\boldsymbol{w}_n}} \circ (I - \mu\nabla\varphi_n)$ satisfies

$$\|T_n(\boldsymbol{a}) - T_n(\boldsymbol{b})\|_2^2 \leq \|\boldsymbol{a} - \boldsymbol{b}\|_2^2, \forall \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^m. \quad (10)$$

In addition, any vector

$$\boldsymbol{z} \in \Omega_n := \operatorname*{argmin}_{\boldsymbol{h} \in \mathbb{R}^m} \left[\frac{1}{2} \frac{(\boldsymbol{u}_n^\top \boldsymbol{h} - d_n)^2}{\|\boldsymbol{u}_n\|_2^2} + \lambda \|\boldsymbol{h}\|_{1,\boldsymbol{w}_n} \right] \quad (11)$$

satisfies $T_n(\boldsymbol{z}) = \boldsymbol{z}$, and thus the inequality in (10) *verifies that*

$$\|\boldsymbol{h}_{n+1} - \boldsymbol{z}\|_2 \leq \|T_n(\boldsymbol{h}_n) - T_n(\boldsymbol{z})\|_2 \leq \|\boldsymbol{h}_n - \boldsymbol{z}\|_2. \quad (12)$$

The shrinkage parameter $\lambda \geq 0$ governs the performance of Algorithm 1. There exists an optimal value of λ in the sense of minimizing the system mismatch $\|\boldsymbol{h} - \boldsymbol{h}^*\|_2^2$. The optimal λ depends on statistical properties of the input and noise, which are often unknown in practice.

3. A TIME-AVERAGING APPROACH

3.1. Approximating the System Mismatch

The main purpose of the proposed automatic shrinkage tuning method is to minimize the system mismatch

$$\begin{aligned} \|\boldsymbol{h}_{n+1} - \boldsymbol{h}^*\|_2^2 = &\|\boldsymbol{h}_{n+1} - \boldsymbol{g}_n\|_2^2 + \|\boldsymbol{g}_n - \boldsymbol{h}^*\|_2^2 \\ &+ 2(\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top (\boldsymbol{g}_n - \boldsymbol{h}^*). \end{aligned} \quad (13)$$

According to the second step of Algorithm 1, the vector $\boldsymbol{h}_{n+1}$ can be rewritten as follows:

$$\boldsymbol{h}_{n+1} = \boldsymbol{A}_n(\lambda)(\boldsymbol{g}_n - \mu\lambda \boldsymbol{v}_n), \quad (14)$$

where $\boldsymbol{A}_n(\lambda)$ is a diagonal matrix with

$$[\boldsymbol{A}_n(\lambda)]_{ii} = \begin{cases} 1, & |g_{n,i}| > \mu\lambda v_{n,i}, \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

and $\boldsymbol{v}_n = \sum_{i=1}^m \operatorname{sgn}(g_{n,i}) w_{n,i} \boldsymbol{e}_i$. Then, the first term of the right-hand side (RHS) in (13) can be expanded as

$$\begin{aligned} \|\boldsymbol{h}_{n+1} - \boldsymbol{g}_n\|_2^2 &= \|(\boldsymbol{I} - \boldsymbol{A}_n(\lambda))\boldsymbol{g}_n + \mu\lambda \boldsymbol{A}_n(\lambda)\boldsymbol{v}_n\|_2^2 \\ &= \|\boldsymbol{g}_n\|_2^2 - \boldsymbol{g}_n^\top \boldsymbol{A}_n(\lambda) \boldsymbol{g}_n + (\mu\lambda)^2 \boldsymbol{v}_n^\top \boldsymbol{A}_n(\lambda) \boldsymbol{v}_n. \end{aligned} \quad (16)$$

Note here that $\|\boldsymbol{g}_n\|_2^2$, as well as the second term of RHS in (13), is a constant in λ. The third term of RHS in (13) can be rewritten as

$$\begin{aligned} (\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top (\boldsymbol{g}_n - \boldsymbol{h}^*) = &(\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top (c_n \hat{\boldsymbol{R}}_n)(\boldsymbol{g}_n - \boldsymbol{h}^*) \\ &+ (\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top (\boldsymbol{I} - c_n \hat{\boldsymbol{R}}_n)(\boldsymbol{g}_n - \boldsymbol{h}^*), \end{aligned} \quad (17)$$

where $\hat{\boldsymbol{R}}_n = \frac{1}{n}\sum_{k=1}^n \frac{\boldsymbol{u}_k \boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|_2^2}$, and $c_n > 0$ (a design example of c_n is given later) is a constant which makes the first term of the RHS in (17) close to the left-hand side in (17), which is unavailable because of the unknown vector $\boldsymbol{h}^*$. The first term of the RHS in (17) can be expanded and approximated as follows:

$$\begin{aligned} &(\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top (c_n \hat{\boldsymbol{R}}_n)(\boldsymbol{g}_n - \boldsymbol{h}^*) \\ &= c_n (\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top \left(\hat{\boldsymbol{R}}_n \boldsymbol{g}_n - \frac{1}{n} \sum_{k=1}^n \frac{\boldsymbol{u}_k^\top \boldsymbol{h}^*}{\|\boldsymbol{u}_k\|_2^2} \boldsymbol{u}_k \right) \\ &= c_n (\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top \left(\hat{\boldsymbol{R}}_n \boldsymbol{g}_n - \frac{1}{n} \sum_{k=1}^n \frac{d_k - \epsilon_k}{\|\boldsymbol{u}_k\|_2^2} \boldsymbol{u}_k \right) \\ &= c_n (\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top \left(\hat{\boldsymbol{R}}_n \boldsymbol{g}_n - \hat{\boldsymbol{p}}_n + \frac{1}{n} \sum_{k=1}^n \frac{\epsilon_k \boldsymbol{u}_k}{\|\boldsymbol{u}_k\|_2^2} \right) \\ &\approx c_n (\boldsymbol{h}_{n+1} - \boldsymbol{g}_n)^\top (\hat{\boldsymbol{R}}_n \boldsymbol{g}_n - \hat{\boldsymbol{p}}_n), \end{aligned} \quad (18)$$

where $\hat{\boldsymbol{p}}_n = \frac{1}{n}\sum_{k=1}^n \frac{d_k}{\|\boldsymbol{u}_k\|_2^2} \boldsymbol{u}_k$. The approximation in (18) is followed by the assumption that $E\big[\frac{\epsilon_k \boldsymbol{u}_k}{\|\boldsymbol{u}_k\|_2^2}\big] \approx \boldsymbol{0}$, which leads to $\frac{1}{n}\sum_{k=1}^n \frac{\epsilon_k \boldsymbol{u}_k}{\|\boldsymbol{u}_k\|_2^2} \approx \boldsymbol{0}$. The last term in (17) is unavailable in practice, and therefore we neglect it. Then, substituting (16) and (18) into (13) and excluding the constant terms, we finally reach the following cost function:

$$\begin{aligned} J_n(\lambda) := &(\mu\lambda)^2 \boldsymbol{v}_n^\top \boldsymbol{A}_n(\lambda) \boldsymbol{v}_n - \boldsymbol{g}_n^\top \boldsymbol{A}_n(\lambda) \boldsymbol{g}_n \\ &+ 2c_n (\boldsymbol{g}_n - \mu\lambda \boldsymbol{v}_n)^\top \boldsymbol{A}_n(\lambda)(\hat{\boldsymbol{R}}_n \boldsymbol{g}_n - \hat{\boldsymbol{p}}_n). \end{aligned} \quad (19)$$

The coefficient c_n should be determined to make (18) an appropriate approximation. Consider an extreme case that $\hat{\boldsymbol{R}}_n = \frac{1}{m}\boldsymbol{I}$. Note here that $\mathrm{tr}(\hat{\boldsymbol{R}}_n) = 1$ by the definition of $\hat{\boldsymbol{R}}_n$. In this case, choosing $c_n = m$ leads to the following equality:

$$(\boldsymbol{h}_{n+1}-\boldsymbol{g}_n)^\mathsf{T}(\boldsymbol{g}_n-\boldsymbol{h}^*) = (\boldsymbol{h}_{n+1}-\boldsymbol{g}_n)^\mathsf{T}(c_n\hat{\boldsymbol{R}}_n)(\boldsymbol{g}_n-\boldsymbol{h}^*). \quad (20)$$

On the other hand, if $\hat{\boldsymbol{R}}_n$ has nonzero off-diagonal entries, the equality in (20) can be violated with $c_n = m$, since the largest eigenvalue of $\hat{\boldsymbol{R}}_n$ is larger than $1/m$. The largest eigenvalue of $\hat{\boldsymbol{R}}_n$ tends to be larger as the off-diagonal entries increase in magnitude. *e.g.*, when all the entries of $\hat{\boldsymbol{R}}_n$ are $1/m$, the largest eigenvalue of $\hat{\boldsymbol{R}}_n$ is 1. To negate such an effect, c_n must be set small when the off-diagonals have large magnitudes. Based on these observations, c_n must satisfy the following properties:

property 1) If $\hat{\boldsymbol{R}}_n = \frac{1}{m}\boldsymbol{I}$, then $c_n = m$,

property 2) when the off-diagonal entries of $\hat{\boldsymbol{R}}_n$ are large, c_n must be smaller than m.

As a design example of c_n that satisfies properties 1) and 2), we use the following design in the current work.

Example 1. *At each iteration n, one can set c_n as follows:*

$$c_n = \frac{m}{\sum_{i=1}^{m}\sum_{j=1}^{m}|\hat{r}_{n,ij}|}, \quad (21)$$

where $\tilde{r}_{n,ij}$ is the (i,j) entry of $\hat{\boldsymbol{R}}_n$. With the above setting,

1. *$c_n = \frac{m}{m/m} = m$ when $\hat{\boldsymbol{R}}_n = \frac{1}{m}\boldsymbol{I}$; and*
2. *large off-diagonal entries make c_n small, since c_n is inversely proportional to $\sum_{i=1}^{m}\sum_{j=1}^{m}|\hat{r}_{n,ij}| = 1+\sum_{i=1}^{m}\sum_{j\neq i}|\hat{r}_{n,ij}|$.*

3.2. Derivation of Proposed Shrinkage Tuning

Since the matrix $\boldsymbol{A}_n(\lambda)$ takes a discrete (binary) value by its definition (22), the cost function $J_n(\lambda)$ is piecewise quadratic. If we focus on the interval $\rho_j \leq \lambda \leq \rho_{j+1}$ ($j \in \{0, 1, \cdots, m\}$), where we obtain $\{\rho_0, \rho_1, \cdots, \rho_m\}$ by sorting $\left\{0, \frac{|g_{n,1}|}{\mu w_{n,1}}, \frac{|g_{n,2}|}{\mu w_{n,2}}, \cdots, \frac{|g_{n,m}|}{\mu w_{n,m}}\right\}$ in nondecreasing order, then $\boldsymbol{A}_n(\lambda)$ is constant at each time n, and we thus let $\boldsymbol{A}_{n,j} = \boldsymbol{A}_n(\lambda)$. With the fixed $\boldsymbol{A}_{n,j}$, we define

$$\begin{aligned} J_{\mathrm{fix},n}(\lambda) :=& (\mu\lambda)^2\boldsymbol{v}_n^\mathsf{T}\boldsymbol{A}_{n,j}\boldsymbol{v}_n - \boldsymbol{g}_n^\mathsf{T}\boldsymbol{A}_{n,j}\boldsymbol{g}_n \\ &+ 2c_n(\boldsymbol{g}_n - \mu\lambda\boldsymbol{v}_n)^\mathsf{T}\boldsymbol{A}_{n,j}(\hat{\boldsymbol{R}}_n\boldsymbol{g}_n - \hat{\boldsymbol{p}}_n). \end{aligned} \quad (22)$$

Differentiating (22) in terms of $\lambda \in \mathbb{R}$ and equating the derivative to zero, we have the unique stationary point:

$$\lambda_{n,j} = \frac{c_n\boldsymbol{v}_n^\mathsf{T}\boldsymbol{A}_{n,j}(\hat{\boldsymbol{R}}_n\boldsymbol{g}_n - \hat{\boldsymbol{p}}_n)}{\mu\boldsymbol{v}_n^\mathsf{T}\boldsymbol{A}_{n,j}\boldsymbol{v}_n}. \quad (23)$$

Since $J_{\mathrm{fix},n}(\lambda)$ is a quadratic function defined on the interval $\rho_j \leq \lambda \leq \rho_{j+1}$ ($j \in \{1, 2, \cdots, m-1\}$), we calculate $\lambda^*_{n,j}$ minimizing $J_{\mathrm{fix},n}(\lambda)$ as follows:

$$\lambda^*_{n,j} = P_{[\rho_j,\rho_{j+1}]}(\lambda_{n,j}) = \begin{cases} \rho_j, & (\lambda_{n,j} < \rho_j), \\ \lambda_{n,j}, & (\rho_i \leq \lambda_{n,j} \leq \rho_{j+1}), \\ \rho_{j+1}, & (\lambda_{n,j} > \rho_{j+1}). \end{cases} \quad (24)$$

We show the APFBS algorithm with the proposed shrinkage tuning in Algorithm 2.

Algorithm 2. APFBS with the proposed shrinkage tuning

---

0. $\boldsymbol{h}_1 = \boldsymbol{0},\ \mu \in (0, 2)$
1. $\boldsymbol{g}_n = \boldsymbol{h}_n - \mu\frac{\boldsymbol{u}_n^\mathsf{T}\boldsymbol{h}_n - d_n}{\|\boldsymbol{u}_n\|_2^2}\boldsymbol{u}_n$
2. $w_{n,i} = \frac{1}{|g_{n,i}| + \nu}$
3. $\boldsymbol{v}_n = \sum_{i=1}^{m}\mathrm{sgn}(g_{n,i})w_{n,i}\boldsymbol{e}_i$
4. $\hat{\boldsymbol{R}}_n = \frac{1}{n}\sum_{k=1}^{n}\frac{\boldsymbol{u}_k\boldsymbol{u}_k^\mathsf{T}}{\|\boldsymbol{u}_k\|_2^2},\ \hat{\boldsymbol{p}}_n = \frac{1}{n}\sum_{k=1}^{n}\frac{d_k}{\|\boldsymbol{u}_k\|_2^2}\boldsymbol{u}_k.$
5. Calculate c_n by (21)
6. Sort $\left\{0, \frac{|g_{n,1}|}{\mu w_{n,1}}, \frac{|g_{n,2}|}{\mu w_{n,2}}, \cdots, \frac{|g_{n,m}|}{\mu w_{n,m}}\right\}$ into $\{\rho_0, \rho_1, \cdots, \rho_m\}$ in nondecreasing order.
7. $\lambda^*_{n,j} = P_{[\rho_j,\rho_{j+1}]}\left(\frac{c_n\boldsymbol{v}_n^\mathsf{T}\boldsymbol{A}_n(\rho_j)(\hat{\boldsymbol{R}}_n\boldsymbol{g}_n - \hat{\boldsymbol{p}}_n)}{\mu\boldsymbol{v}_n^\mathsf{T}\boldsymbol{A}_n(\rho_j)\boldsymbol{v}_n}\right)$ for $j \in \{0, 1, \cdots, m-1\}$, and $\hat{\lambda}_m = \rho_m$.
8. Choose $\lambda^*_n = \operatorname*{argmin}_{\lambda\in\{\lambda^*_{n,j}\}_{j=0}^m} J_n(\lambda)$, according to (19)
9. $\boldsymbol{h}_{n+1} = \sum_{i=1}^{m}\mathrm{sgn}(g_{n,i})\max\{|g_{n,i}| - \mu\lambda^*_n w_{n,i}, 0\}\boldsymbol{e}_i$

The proposed shrinkage tuning technique involves matrix calculations for $\hat{\boldsymbol{R}}_n$, which causes more expensive computational cost than the shrinkage tuning in [18]. However, one can also see that the proposed tuning is more robust to a colored input, as shown in the following section.

4. NUMERICAL EXAMPLES

We compare the APFBS algorithm with the proposed tuning (APFBS-Proposed), with (i) the NLMS algorithm; (ii) the APFBS algorithm with a carefully tuned shrinkage parameter (APFBS-Fixed); and (iii) the APFBS algorithm with the shrinkage tuning in [18] (APFBS-ST). The weighting technique proposed in [22] is employed for the later two algorithms.

4.1. Initial Setting

We use the echo path model #4 for testing of speech echo cancelers defined in the ITU-T Recommendation G.168 [23] as the target system. We put zeros as inactive regions before and after echo path model, so that the entire length of the system is $m = 512$ (See Figure 1). As input signals, we have (i) a white Gaussian signal and (ii) a colored signal generated by an autoregressive (AR) process as follows:

$$u_1 = 0,\ u_n = 0.9u_{n-1} + 0.1\zeta_n\ (n \geq 2), \quad (25)$$

where $\zeta_n \in \mathbb{R}$ is a white Gaussian noise. We employ an additive white Gaussian noise, and utilize its variance in the shrinkage tuning of [18]. The step size is set as $\mu = 0.5$ for all the algorithms.

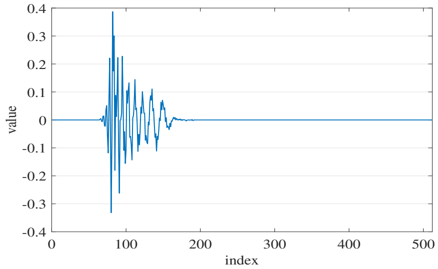

Fig. 1. The target system based on the echo path model #4 [23]

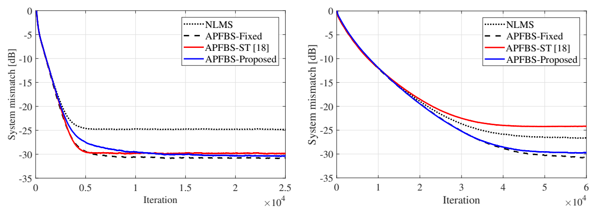

Fig. 2. The learning curves under the white input (left) and the colored input (right) (SNR = 20[dB]).

4.2. Numerical Results

We first show learning curves of the algorithms for SNR = 20[dB] by Figure 2. One can readily see that the proposed shrinkage tuning does not slow the initial convergence rate of the APFBS algorithm. In addition, the proposed shrinkage tuning works better than the NLMS algorithm under both white and colored inputs.

Figures 3, 4 show the system mismatches in the steady state, under various SNR conditions. Figure 3 is the case of the white Gaussian input, and Figure 4 is the case of the colored (AR) input. Under the white Gaussian input signal, the proposed shrinkage tuning is better than the one proposed in [18]. The performance of the proposed shrinkage tuning method is closer to that of the manual tuning. When the input is colored, the shrinkage tuning based on instantaneous approximation fails and leads to the larger system mismatch than that of the NLMS algorithm. On the other hand, although not so fine as the manual tuning, the proposed shrinkage tuning shows a more robust performance.

4.3. Discussion

The main differences of the proposed shrinkage tuning from the shrinkage tuning method of [18] can be presented as follows:

- The proposed shrinkage tuning utilizes time-averaged terms;
- it does not need the constraint of white Gaussian noise.

In the approximation of the system mismatch, the proposed method utilizes the time-averaged second-order statistics, while the shrinkage tuning in [18] uses instantaneous observations at each iteration. The time-averaging operation involves a term on $\boldsymbol{u}_n\boldsymbol{u}_n^{\mathsf{T}} \in \mathbb{R}^{m\times m}$, and thus the proposed shrinkage tuning is more expensive than the one proposed in [18]. Despite such computational cost, the proposed shrinkage tuning takes advantage in a sense of minimizing the system mismatch. Especially, the proposed shrinkage tuning is robust to the colored input signal, while the shrinkage tuning based on an instantaneous approximation can lead to a poorer performance of the APFBS algorithm even than that of the NLMS algorithm, which has no shrinkage parameter (See Figure 4).

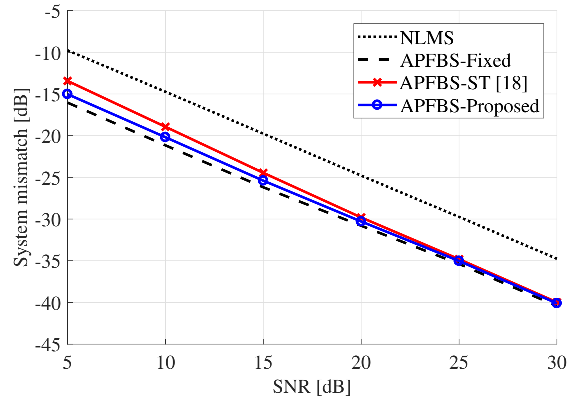

Fig. 3. System mismatch vs. SNR (white input).

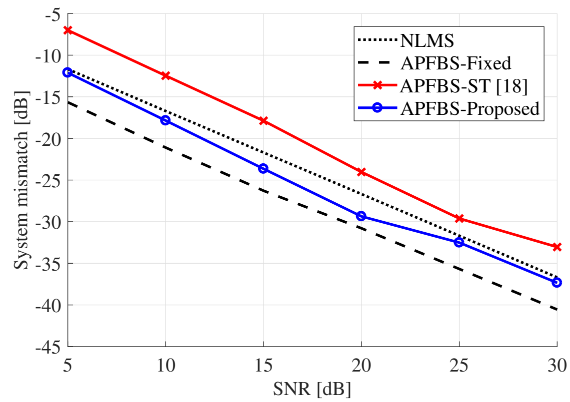

Fig. 4. System mismatch vs. SNR (colored input).

The shrinkage tuning in [18] has been developed under that assumption that the additive noise is Gaussian with known variance. The assumption is needed in order to utilize Stein's lemma [24] and achieve an approximation of the system mismatch well enough. On the other hand, the proposed shrinkage tuning is developed without such an assumption. Nevertheless, it results a better performance even under the white Gaussian noise (See Figure 3).

5. CONCLUDING REMARKS

We have proposed an automatic shrinkage tuning for the APFBS algorithm. The APFBS algorithm considers an ℓ_1-norm regularized least square problem at each iteration, and involves a shrinkage parameter to exploit the sparsity. Choosing the shrinkage parameter carefully, one can achieve an excellent performance of the APFBS algorithm.

The proposed shrinkage tuning has been derived by approximating the system mismatch using time-averaged terms. The cost function has been piecewise quadratic, and it is thus minimized by comparing the piecewise optima. In the numerical examples, we have shown that the proposed shrinkage tuning method suppresses the system mismatch better than the shrinkage tuning with an instantaneous approximation, under both of white and colored input signals.

6. REFERENCES

[1] S. Makino and Y. Kaneda, "Exponentially weight step-size projection algorithm for acoustic echo cancellers," *IEICE transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 75, no. 11, pp. 1500–1508, 1992.

[2] D. L. Duttweiler, "Proportionate normalized least squares adaptation in echo cancelers," *IEEE Trans. Speech and Audio Process.*, vol. 8, no. 5, pp. 508–518, 2000.

[3] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. 25nd Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2002, pp. II–1881–II–1884.

[4] C. Paleologu, J. Benesty and S. Ciochina, "Sparse adaptive filters for echo cancelation," *Synthesis Lectures on Speech and Audio Processing*, vol. 6, no. 1, pp. 1–124, 2010.

[5] C. Paleologu, J. Benesty and S. Ciochina, "An improved proportionate NLMS algorithm based on the ℓ_0-norm," in *Proc. 33nd Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2010, pp. 309–312.

[6] Y. Murakami, M. Yamagishi, M. Yukawa and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. 33rd Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2010, pp. 3734–3737.

[7] F. C. de Souza, O. Tobias, R. Seara and D. Morgan, "A PNLMS algorithm with individual activation factors," *IEEE trans. on Signal Process.*, vol. 58, no. 4, pp. 2036–2047, 2010.

[8] G. Gui and F. Adachi, "Improved adaptive sparse channel estimation based on the least mean square algorithm," *EURASIP J. Wir. Comm. Netw.*, pp. 1–18, 2013.

[9] Y. Chen, Y. Gu and A. O. Hero, "Sparse LMS for system identification," in *Proc. 32nd Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2009, pp. 3125–3128.

[10] Y. Gu, J. Jin and S. Mei, "ℓ_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, 2009.

[11] D. Angelosante, J. A. Bazerque and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the norm," *IEEE trans. on Signal Process.*, vol. 58, no. 7, pp. 3436–3447, 2010.

[12] Y. Kopsinis, K. Slavakis and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted balls," *IEEE trans. on Signal Process.*, vol. 59, no. 3, pp. 936–952, 2011.

[13] J. Benesty, C. Paleologu, C. Gansler and S. Ciochina, *A Perspective on Stereophonic Acoustic Echo Cancellation*, Springer, 2011.

[14] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE trans. on Signal Process.*, vol. 61, no. 6, pp. 1419–1433, 2013.

[15] J. Liu and S. L. Grant, "Proportionate adaptive filtering for block-sparse system identification," *IEEE/ACM transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 623–630, 2016.

[16] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automatic Control*, vol. 12, no. 3, pp. 282–287, 1967.

[17] M. Yamagishi, M. Yukawa and I. Yamada, "Shrinkage tuning based on an unbiased MSE estimate for sparsity-aware adaptive filtering," in *Proc. 37th Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 5477–5481.

[18] M. Yamagishi, M. Yukawa and I. Yamada, "Automatic shrinkage tuning based on a system-mismatch estimate for sparsity-aware adaptive filtering," in *Proc. 40th Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2017, pp. 4800–4804.

[19] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 2002.

[20] A. Sayed, *Adaptive Filters*, Wiley-IEEE Press, 2008.

[21] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[22] M. Yukawa, Y. Tawara and I. Yamada, "Sparsity-aware adaptive filters based on ℓ_p-norm inspired soft-thresholding technique," in *Proc. Int. Symp. Circuit, Sys. (ISCAS)*, 2012, pp. 2749–2752.

[23] ITU-T Recommendation G. 168, *Digital Network Echo Cancellers*, Int. Telecomm. Union, 2015.

[24] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, pp. 1135–1151, 1981.