AUTOMATIC SHRINKAGE TUNING BASED ON A SYSTEM-MISMATCH ESTIMATE FOR SPARSITY-AWARE ADAPTIVE FILTERING

Masao Yamagishi† *Masahiro Yukawa*‡ *Isao Yamada*†

†Tokyo Institute of Technology, 2-12-1-S3-60 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
‡Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan
Email: {myamagi, isao}@sp.ce.titech.ac.jp, yukawa@elec.keio.ac.jp

ABSTRACT

Exploiting the sparsity in learning algorithms is a key to achieve excellent performances of adaptive filters. This can be realized by the adaptive proximal forward-backward splitting with carefully chosen parameters. In this paper, we propose an automatic parameter tuning based on a minimization principle of a stochastic approximation of the system-mismatch. The proposed approximation has a Tikhonov-type regularization term, which aims to minimize the disturbance by the update of the adaptive filter and mitigates over-fitting to an instantaneous observation. Thanks to these properties, the proposed method realizes adaptive parameter tuning without any user-defined parameters, unlike our previous method that utilizes the user-defined parameter to avoid over-fitting. A numerical example demonstrates the efficacy of the proposed parameter tuning.

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

1. INTRODUCTION

Exploiting the sparsity in learning algorithms is a key to achieve excellent performance of adaptive filters, where the sparsity implies that many coefficients of the system are zero. The sparsity of the system to be estimated has been observed and exploited in many applications including network/acoustic echo cancellation and active noise control (e.g. see [1–19] and references therein).

A typical way to exploit the sparsity is to utilize a sparsity promoting term with a regularization parameter in implicit/explicit optimization problems in adaptive learning algorithms. One of this kind of algorithms is the adaptive proximal forward-backward splitting (APFBS) scheme [8, 9], which is a principle to adaptively suppress the sum of a smooth convex function and a nonsmooth convex function. A typical choice of the nonsmooth convex term to promote the sparsity is a weighted ℓ_1 norm with a regularization parameter (which we refer to as *shrinkage parameter* in this paper). In this technique, careful adaptive tuning of the weight and the shrinkage parameter is required to achieve excellent performance.[1] Our previous method in [22] realizes an adaptive tuning by using an unbiased estimate of the mean squared error (MSE) which measures the difference of the outputs of the adaptive filter and the system to be estimated. More precisely, by adaptively determining the weight with conventional weight designs (e.g. [14]), the shrinkage parameter is chosen in a way to minimize an unbiased estimate of the MSE. Although this technique achieves excellent performance robustly against SNR environmental change, it requires to select a user-defined parameter to avoid selecting an excessively large shrinkage parameter due to over-fitting to the single observation. This is caused by its instantaneous nature, i.e., the unbiased estimate defined only with the instantaneous observation.

In this paper, to realize adaptive tuning of the shrinkage parameter without any user-defined parameters, we propose an automatic shrinkage parameter tuning based on a minimization principle of a stochastic approximation of the system-mismatch, where the system-mismatch measures the difference between the adaptive filter and the system to be estimated. Obviously, selecting the parameter minimizing the system-mismatch is a natural choice. However, the system-mismatch is unavailable in practice. To alleviate this difficulty, we focus on an identity of the system-mismatch where most terms of the identity are available in practical situations, directly or with the aid of stochastic approximation. By using this nature, we introduce a stochastic approximation of the identity as an approximation of the system-mismatch with renouncing a few terms that are unavailable in practice. This idea yields, in our stochastic approximation, the so-called Tikhonov regularization term which incorporates *the principle of minimal disturbance* [23], i.e., it attempts to minimally disturb the adaptive filter already trained, which mitigates the instantaneous nature as well as the over-fitting. Consequently, the proposed method does not require the user-defined parameter to avoid over-fitting, unlike our previous method [22]. In addition, the proposed approximation is piecewise quadratic, so that the global minimizer is computed efficiently.

A numerical example demonstrates the efficacy of the proposed parameter tuning by showing excellent performance robustly against environmental changes.

This work was supported in part by JSPS Grants-in-Aid (26730128).

[1] See [20, 21] for shrinkage tuning in non-adaptive settings.

2. PRELIMINARIES

2.1. Adaptive Filtering Problem

Let $\mathbb{R}, \mathbb{R}_+$, and $\mathbb{N}$ denote the sets of all real numbers, nonnegative real numbers, and nonnegative integers, respectively. Denote the set $\mathbb{N} \setminus \{0\}$ by $\mathbb{N}^*$ and transposition of a matrix or a vector by $(\cdot)^\top$.

Suppose that we observe an output sequence $(d_k)_{k\in\mathbb{N}} \subset \mathbb{R}$ (i.e., $d_k \in \mathbb{R}, \forall k \in \mathbb{N}$) that obeys the model,

$$d_k = \boldsymbol{u}_k^\top \boldsymbol{h}_\star + \epsilon_k, \tag{1}$$

where $k \in \mathbb{N}$ denotes the time index, $\boldsymbol{u}_k := [u_k, u_{k-1}, \ldots, u_{k-N+1}]^\top \in \mathbb{R}^N$ a known vector defined with the input sequence $(u_k)_{k\in\mathbb{N}} \subset \mathbb{R}$ (where $N \in \mathbb{N}^*$ is the tap length), $\boldsymbol{h}_\star \in \mathbb{R}^N$ the unknown system to be estimated (e.g., echo impulse response), and $\epsilon_k \in \mathbb{R}$ the noise process.

The major goal of adaptive system identification is to approximate the unknown system $\boldsymbol{h}_\star$ by the adaptive filter $\boldsymbol{h}_k := [h_k^{(1)}, h_k^{(2)}, \ldots, h_k^{(N)}]^\top \in \mathbb{R}^N$ with $(\boldsymbol{u}_i, d_i)_{i=0}^k$ together with a prior knowledge on $\boldsymbol{h}_\star$, e.g., the sparsity.

2.2. Adaptive proximal forward-backward splitting (APFBS)

The APFBS [8, 9] provides a systematic design of the update of adaptive learning algorithm. Define a time-varying cost function[2]

[2] $\Gamma_0(\mathbb{R}^N)$ is the class of all lower semicontinuous convex functions from $\mathbb{R}^N$ to $(-\infty, +\infty]$ that are not identically $+\infty$ [24].

$\Theta_k \in \Gamma_0(\mathbb{R}^N)$ for $k \in \mathbb{N}$ by

$$\Theta_k(\boldsymbol{h}) := \varphi_k(\boldsymbol{h}) + \psi_k(\boldsymbol{h}), \tag{2}$$

where $\psi_k \in \Gamma_0(\mathbb{R}^N)$ and $\varphi_k \colon \mathbb{R}^N \to \mathbb{R}$ is a smooth convex function with its gradient $\nabla\varphi_k$ Lipschitz continuous, i.e., there exists a some $L_k > 0$ (which is called a Lipschitz constant) s.t.

$$\|\nabla\varphi_k(\boldsymbol{h}) - \nabla\varphi_k(\boldsymbol{g})\| \leq L_k\|\boldsymbol{h} - \boldsymbol{g}\| \tag{3}$$

for all $\boldsymbol{h}, \boldsymbol{g} \in \mathbb{R}^N$, where $||\cdot||$ stands for the standard Euclidean norm. Typically, φ_k plays the role of a data fidelity term and ψ_k plays the role of a penalty term that exploits the sparsity of $\boldsymbol{h}_\star$ in the learning process (e.g. weighted ℓ_1 norms are adopted as ψ_k). Then, the APFBS is summarized as follows.

Algorithm 1 (APFBS)**.** *For an arbitrarily chosen $\boldsymbol{h}_0 \in \mathbb{R}^N$, generate a sequence $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^N$ by*

$$\boldsymbol{h}_{k+1} := \mathrm{prox}_{\frac{\mu_k}{L_k}\psi_k}\left(\boldsymbol{h}_k - \frac{\mu_k}{L_k}\nabla\varphi_k(\boldsymbol{h}_k)\right), \tag{4}$$

where $\mu_k \in (0, 2)$ is the step-size and $\mathrm{prox}_{\frac{\mu_k}{L_k}\psi_k} : \mathbb{R}^N \to \mathbb{R}^N$:

$$\mathrm{prox}_{\frac{\mu_k}{L_k}\psi_k}(\boldsymbol{h}) := \underset{\boldsymbol{g}\in\mathbb{R}^N}{\arg\min}\left(\psi_k(\boldsymbol{g}) + \frac{L_k}{2\mu_k}||\boldsymbol{h} - \boldsymbol{g}||^2\right)$$

is called the proximity operator of ψ_k of index $\frac{\mu_k}{L_k} > 0$ [25].

Note that Algorithm 1 is a time-varying extension of the proximal forward-backward splitting method [26, 27] (see also [28, 29]) and satisfies the (strictly) monotone approximation property [30]:

$$\left\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*_{\Theta_k}\right\| < \left\|\boldsymbol{h}_k - \boldsymbol{h}^*_{\Theta_k}\right\| \tag{5}$$

for every $\boldsymbol{h}^*_{\Theta_k} \in \Omega_k := \underset{\boldsymbol{h}\in\mathbb{R}^N}{\arg\min}\, \Theta_k(\boldsymbol{h})$ if $\boldsymbol{h}_k \notin \Omega_k \neq \varnothing$. An acceleration of the APFBS has been proposed in [11].

Here we show a simple sparsity-aware adaptive filtering algorithm in the frame of the APFBS.

Example 1. *Let the smooth term φ_k be the squared distance*[3]

$$\varphi_k(\boldsymbol{h}) := \frac{1}{2}d^2(\boldsymbol{h}, S_k) \tag{6}$$

to a closed convex set $S_k := \arg\min_{\boldsymbol{h}\in\mathbb{R}^N} |d_k - \boldsymbol{u}_k^\top\boldsymbol{h}|$, of which the elements are consistent with the data available at time k (note: $L_k = 1$ in this case). Moreover, we adopt a weighted ℓ_1-norm as the sparsity promoting nonsmooth term, i.e., $\psi_k = \lambda_k\|\cdot\|_1^{\boldsymbol{\omega}_k}$ with

$$\|\boldsymbol{h}\|_1^{\boldsymbol{\omega}_k} := \sum_{j=1}^N \omega_k^{(j)}|h_j|,\ \boldsymbol{h} := [h_1, h_2, \cdots, h_N]^\top \in \mathbb{R}^N,$$

where $\lambda_k > 0$ is the regularization parameter, and $\omega_k^{(j)} > 0$, $j \in \{1, 2, \ldots, N\}$, the weights of the ℓ_1 norm defined with available knowledge. Then the update equation (4) *becomes*

$$\boldsymbol{h}_{k+1} := \mathrm{prox}_{\mu_k\lambda_k\|\cdot\|_1^{\boldsymbol{\omega}_k}}\left(\boldsymbol{h}_k + \mu_k\frac{d_k - \boldsymbol{u}_k^\top\boldsymbol{h}_k}{\|\boldsymbol{u}_k\|^2}\boldsymbol{u}_k\right), \tag{7}$$

$$\mathrm{prox}_{\mu_k\lambda_k\|\cdot\|_1^{\boldsymbol{\omega}_k}}(\boldsymbol{h}) = \sum_{j=1}^N \mathrm{sgn}\big(h_j\big)\max\left\{|h_j| - \mu_k\lambda_k\omega_k^{(j)},\ 0\right\}\boldsymbol{e}_i,$$

where the signum function $\mathrm{sgn}\colon \mathbb{R} \to \{-1, 0, 1\}$ is defined as $\mathrm{sgn}(x) := x/|x|$ if $x \neq 0$, $\mathrm{sgn}(x) := 0$ otherwise, and the canonical basis of $\mathbb{R}^N$ is denoted by $\{\boldsymbol{e}_j := [0, \ldots, 0, 1, 0, \ldots, 0]^\top\}_{j=1}^N$ (the value 1 assigned to the j-th position).

[3]The *distance* between an arbitrary point $\boldsymbol{x} \in \mathbb{R}^N$ and a closed convex set $C \subset \mathbb{R}^N$ is defined by $d(\boldsymbol{x}, C) := \min_{\boldsymbol{y}\in C}\|\boldsymbol{x} - \boldsymbol{y}\|$.

2.3. Previous tuning based on unbiased MSE estimate

Our previous parameter tuning in [22] was derived from minimization of an unbiased MSE estimate: select a candidate of the adaptive filter at time $k + 1$ parameterized by $\lambda \in \mathbb{R}_+$, i.e.,

$$\hat{\boldsymbol{h}}_{k+1}(\lambda) = \mathrm{prox}_{\lambda\|\cdot\|_1^{\boldsymbol{\omega}_k}}\left(\boldsymbol{h}_k + \mu_k\frac{d_k - \boldsymbol{u}_k^\top\boldsymbol{h}_k}{\|\boldsymbol{u}_k\|^2}\boldsymbol{u}_k\right) \tag{8}$$

by minimizing an unbiased MSE estimate $\widetilde{J}$ (defined in (11) below). Note that $\widetilde{J}$ can be minimized efficiently with $\mathcal{O}(N)$ multiplications because it is a piecewise quadratic function (see (11) and Fact 2). Detail steps of our previous parameter tuning is described in Algorithm 3 below for a comparison with the proposed parameter tuning.

Fact 1 ([22])**.** *Assume that the additive noise is according to the zero mean Gaussian distribution with variance σ^2, i.e.,*

$$p(\epsilon_k) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\epsilon_k^2}{2\sigma^2}\right).$$

Then[4] *(i)*

$$E\left[\epsilon_k\hat{\boldsymbol{h}}_{k+1}(\lambda)\right] = E\left[\sigma^2\mu_k\boldsymbol{A}_k(\lambda)\boldsymbol{u}_k\right] \tag{9}$$

with a diagonal matrix $\boldsymbol{A}_k(\lambda)$ whose diagonal entries indicate the support of the adaptive filter $\hat{\boldsymbol{h}}_{k+1}(\lambda)$, i.e.,

$$\boldsymbol{A}_k(\lambda) := \mathrm{diag}(a_k^{(1)}(\lambda), a_k^{(2)}(\lambda), \ldots, a_k^{(N)}(\lambda)) \in \mathbb{R}^{N\times N}, \tag{10}$$

$$a_k^{(j)}(\lambda) := \begin{cases} 1, & \text{if } \left|g_k^{(j)}\right| > \lambda\omega_k^{(j)}; \\ 0, & \text{otherwise,} \end{cases}$$

$$\boldsymbol{g}_k := (g_k^{(1)}, g_k^{(2)}, \ldots, g_k^{(N)})^\top := \boldsymbol{h}_k + \mu_k\frac{d_k - \boldsymbol{u}_k^\top\boldsymbol{h}_k}{\|\boldsymbol{u}_k\|^2}\boldsymbol{u}_k.$$

(ii) Define $\widetilde{J}\colon \mathbb{R}_+ \to \mathbb{R}$:

$$\widetilde{J}(\lambda) := \frac{(\boldsymbol{u}_k^\top\hat{\boldsymbol{h}}_{k+1}(\lambda) - d_k)^2}{\|\boldsymbol{u}_k\|^2} - \sigma^2 + 2\sigma^2\mu_k\frac{\boldsymbol{u}_k^\top\boldsymbol{A}_k(\lambda)\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|^2}. \tag{11}$$

Then, it is an unbiased estimate of the MSE, i.e.,

$$E[\widetilde{J}(\lambda)] = E[(\boldsymbol{u}_k^\top\hat{\boldsymbol{h}}_{k+1}(\lambda) - \boldsymbol{u}_k^\top\boldsymbol{h}_*)^2]. \tag{12}$$

Fact 2 ([22])**.** *Let $(\hat{\lambda}_j)_{j=0}^N$ be a sorted sequence of all entries of*

$$\left(0, \frac{|g_k^{(1)}|}{\omega_k^{(1)}}, \frac{|g_k^{(2)}|}{\omega_k^{(2)}}, \ldots, \frac{|g_k^{(N)}|}{\omega_k^{(N)}}\right) \tag{13}$$

in nondecreasing order.[5] *Then (i) $\boldsymbol{A}_k(\lambda)$ is invariant over $[\hat{\lambda}_j, \hat{\lambda}_{j+1})$:*

$$\boldsymbol{A}_k(\lambda) = \boldsymbol{A}_k(\hat{\lambda}_j),\ \forall\lambda \in [\hat{\lambda}_j, \hat{\lambda}_{j+1}),$$

(ii) $\hat{\boldsymbol{h}}_{k+1}$ is linear over $[\hat{\lambda}_j, \hat{\lambda}_{j+1})$, and
(iii) $\widetilde{J}$ is quadratic over $[\hat{\lambda}_j, \hat{\lambda}_{j+1})$.

[4]For a given square matrix $\boldsymbol{A}$, $\mathrm{tr}[\boldsymbol{A}]$ indicates its trace. For a vector $\boldsymbol{x} \in \mathbb{R}^N$, $\mathrm{diag}(\boldsymbol{x}) \in \mathbb{R}^{N\times N}$ denotes the diagonal matrix whose diagonal entries are given by $\boldsymbol{x}$.

[5]In this paper, for simplicity, we assume that the vector (13) has no overlapping entries. This assumption can be relaxed easily.

3. PROPOSED PARAMETER TUNING

We propose an automatic shrinkage parameter tuning based on minimization of an estimation of the system-mismatch: select a candidate of the adaptive filter at time $k+1$, i.e., (8) by minimizing the estimate J (defined in (16) below) of the system-mismatch $\|\hat{\boldsymbol{h}}_{k+1}(\lambda)-\boldsymbol{h}_\star\|^2$. Fortunately, our design of J preserves piecewise quadratic nature. Hence its minimization is efficiently implemented (see Algorithm 2 for the detail of the proposed parameter tuning).

We shall derive the estimate J. Construction of J is motivated by the identity[6]

$$\begin{aligned}
&\|\boldsymbol{h}-\boldsymbol{h}_\star\|^2\\
&=\frac{(\boldsymbol{u}_k^\top\boldsymbol{h}-d_k)^2}{\|\boldsymbol{u}_k\|^2}+\frac{\epsilon_k^2}{\|\boldsymbol{u}_k\|^2}+\frac{2\epsilon_k(\boldsymbol{u}_k^\top\boldsymbol{h}-d_k)}{\|\boldsymbol{u}_k\|^2}\\
&\quad+\left\|\left(\boldsymbol{I}-\frac{\boldsymbol{u}_k\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|^2}\right)(\boldsymbol{h}-\boldsymbol{h}_k)\right\|^2+\left\|\left(\boldsymbol{I}-\frac{\boldsymbol{u}_k\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|^2}\right)(\boldsymbol{h}_k-\boldsymbol{h}_\star)\right\|^2\\
&\quad+2\left\langle\left(\boldsymbol{I}-\frac{\boldsymbol{u}_k\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|^2}\right)(\boldsymbol{h}-\boldsymbol{h}_k),\boldsymbol{h}_k-\boldsymbol{h}_\star\right\rangle \qquad (15)
\end{aligned}$$

for any $\boldsymbol{h}\in\mathbb{R}^N$. Benefits of the identity (15) are three-fold: the 1st and the 4th terms of the RHS are available in practice; the 2nd and the 3rd terms of the RHS have affordable unbiased estimates in our setting because of $E[\epsilon_k^2-2\epsilon_k d_k]=-\sigma^2$ and (9); the 5th term of the RHS is irrelevant to select parameters since it is constant.

Observing these benefits, we introduce an approximation of the system-mismatch by replacing the 2nd and the 3rd terms of the identity (15) by their unbiased estimates, i.e., $J\colon\mathbb{R}_+\to\mathbb{R}$:

$$\begin{aligned}
J(\lambda)&:=\frac{(\boldsymbol{u}_k^\top\hat{\boldsymbol{h}}_{k+1}(\lambda)-d_k)^2}{\|\boldsymbol{u}_k\|^2}-\sigma^2+2\sigma^2\mu_k\frac{\boldsymbol{u}_k^\top\boldsymbol{A}_k(\lambda)\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|^2}\\
&\quad+\left\|\left(\boldsymbol{I}-\frac{\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|}\frac{\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|}\right)(\hat{\boldsymbol{h}}_{k+1}(\lambda)-\boldsymbol{h}_k)\right\|^2\\
&=\widetilde{J}(\lambda)+\left\|\left(\boldsymbol{I}-\frac{\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|}\frac{\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|}\right)(\hat{\boldsymbol{h}}_{k+1}(\lambda)-\boldsymbol{h}_k)\right\|^2, \qquad (16)
\end{aligned}$$

where we eliminate constants irrelevant to select the shrinkage parameter, and where we renounce the last term of the RHS of (15) because it is unavailable in practice.

The global minimizer of J is obtained efficiently. Similar to $\widetilde{J}$, the function J is piecewise quadratic because of Fact 2(i)(ii). Hence J is a quadratic function over the interval where $\boldsymbol{A}_k(\lambda)$ is invariant, the local minimizer of J in each invariant interval can be computed easily. In addition, by evaluating the value of J at all local minima, we can find the global minimizer efficiently.

Proposition 1. *(i) J is quadratic over $[\hat{\lambda}_j,\hat{\lambda}_{j+1})$.*

(ii) $\min_{\lambda\in\mathbb{R}_+}J(\lambda)=\min_{\lambda\in\{\hat{\lambda}_j^*\}_{j=0}^N}J(\lambda)$, *where*[7]

$$\hat{\lambda}_j^*:=P_{[\hat{\lambda}_j,\hat{\lambda}_{j+1}]}\left[(\mu_k-1)\frac{d_k-\boldsymbol{u}_k^\top\boldsymbol{h}_k}{\|\boldsymbol{u}_k\|^2}\frac{\mathrm{tr}\left[\boldsymbol{A}_k(\hat{\lambda}_j)\,\mathrm{diag}(\boldsymbol{\xi}_k)\right]}{\mathrm{tr}\left[\boldsymbol{A}_k(\hat{\lambda}_j)\,\mathrm{diag}(\boldsymbol{\zeta}_k)\right]}\right]$$

is a minimizer of the local quadratic function $J_j:[\hat{\lambda}_j,\hat{\lambda}_{j+1}]\to\mathbb{R}$:

$$\lambda\mapsto\frac{(\boldsymbol{u}_k^\top\hat{\boldsymbol{h}}_{k+1}(\lambda)-d_k)^2}{\|\boldsymbol{u}_k\|^2}+\left\|\left(\boldsymbol{I}-\frac{\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|}\frac{\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|}\right)(\hat{\boldsymbol{h}}_{k+1}(\lambda)-\boldsymbol{h}_k)\right\|^2$$

of J except that $\hat{\lambda}_N^*:=\hat{\lambda}_N$, and where $\boldsymbol{\xi}_k:=\boldsymbol{\omega}_k\odot\boldsymbol{g}_k\odot\boldsymbol{u}_k$, $\boldsymbol{\zeta}_k:=\boldsymbol{\omega}_k\odot\boldsymbol{\omega}_k$, and $\odot$ represents Hadamard product (or entry-wise multiplication).

[6]For completeness, we describe the derivation of the identity (15). First, we decompose $\boldsymbol{h}-\boldsymbol{h}_\star$ orthogonally in the span of $\boldsymbol{u}_k$ and in its complement

$$\|\boldsymbol{h}-\boldsymbol{h}_\star\|^2=\left\|\left(\frac{\boldsymbol{u}_k\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|^2}\right)(\boldsymbol{h}-\boldsymbol{h}_\star)\right\|^2+\left\|\left(\boldsymbol{I}-\frac{\boldsymbol{u}_k\boldsymbol{u}_k^\top}{\|\boldsymbol{u}_k\|^2}\right)(\boldsymbol{h}-\boldsymbol{h}_\star)\right\|^2. \quad (14)$$

Then we eliminate $\boldsymbol{h}_\star$ in the first term of the RHS of (14) by substituting (1)

(1st term of the RHS of (14))

$$=\frac{(\boldsymbol{u}_k^\top\boldsymbol{h}-d_k+\epsilon_k)^2}{\|\boldsymbol{u}_k\|^2}=\frac{(\boldsymbol{u}_k^\top\boldsymbol{h}-d_k)^2}{\|\boldsymbol{u}_k\|^2}+\frac{\epsilon_k^2}{\|\boldsymbol{u}_k\|^2}+\frac{2\epsilon_k(\boldsymbol{u}_k^\top\boldsymbol{h}-d_k)}{\|\boldsymbol{u}_k\|^2}.$$

Finally, we expand the second term of the RHS of (14) as the second order Taylor series at $\boldsymbol{h}_k$, which completes the proof.

[7]For $a,b\in\mathbb{R}\colon a<b$, the projection onto $[a,b]$ is given as

$$P_{[a,b]}\colon\mathbb{R}\to\mathbb{R}\colon P_{[a,b]}(r)=\begin{cases}a, & \text{if } r<a;\\ r, & \text{if } r\in[a,b];\\ b, & \text{if } r>b.\end{cases}$$

Algorithm 2: APFBS (7) with the proposed parameter tuning

Repeat the following step:

1. Compute $\lambda^*\in\underset{\lambda\in\mathbb{R}_+}{\arg\min}\,J(\lambda)$ in (16) by Steps (1a)–(1e).
2. Update $\boldsymbol{h}_{k+1}=\mathrm{prox}_{\lambda^*\|\cdot\|_1^{\boldsymbol{\omega}_k}}\left(\boldsymbol{h}_k+\mu_k\frac{d_k-\boldsymbol{u}_k^\top\boldsymbol{h}_k}{\|\boldsymbol{u}_k\|^2}\boldsymbol{u}_k\right)$.

Minimization of J with efficient $\mathcal{O}(N)$ multiplications

1(a). Compute $\boldsymbol{g}_k=\boldsymbol{h}_k+\mu_k\frac{d_k-\boldsymbol{u}_k^\top\boldsymbol{h}_k}{\|\boldsymbol{u}_k\|^2}\boldsymbol{u}_k$.

1(b). Calculate the weight $\boldsymbol{\omega}_k$ (e.g. (18)).

1(c). Sort $\left(0,\frac{|g_k^{(1)}|}{\omega_k^{(1)}},\frac{|g_k^{(2)}|}{\omega_k^{(2)}},\dots,\frac{|g_k^{(N)}|}{\omega_k^{(N)}}\right)$ into $(\hat{\lambda}_j)_{j=0}^N$ by nondecreasing order.

1(d). Compute $(\hat{\lambda}_j^*)_{j=0}^N$ by

$$\hat{\lambda}_j^*:=P_{[\hat{\lambda}_j,\hat{\lambda}_{j+1}]}\left[(\mu_k-1)\frac{d_k-\boldsymbol{u}_k^\top\boldsymbol{h}_k}{\|\boldsymbol{u}_k\|^2}\frac{\mathrm{tr}\left[\boldsymbol{A}_k(\hat{\lambda}_j)\,\mathrm{diag}(\boldsymbol{\xi}_k)\right]}{\mathrm{tr}\left[\boldsymbol{A}_k(\hat{\lambda}_j)\,\mathrm{diag}(\boldsymbol{\zeta}_k)\right]}\right],$$

where $\boldsymbol{A}_k(\hat{\lambda}_j)=\mathrm{diag}(a_k^{(1)}(\hat{\lambda}_j),\dots,a_k^{(N)}(\hat{\lambda}_j))$

$$a_k^{(i)}(\hat{\lambda}_j)=\begin{cases}1, & \left|g_k^{(i)}\right|>\hat{\lambda}_j\omega_k^{(i)};\\ 0, & \text{otherwise},\end{cases}$$

$\boldsymbol{\xi}_k:=\boldsymbol{\omega}_k\odot\boldsymbol{g}_k\odot\boldsymbol{u}_k$, and $\boldsymbol{\zeta}_k:=\boldsymbol{\omega}_k\odot\boldsymbol{\omega}_k$.

1(e). Find $\lambda^*\in\ \arg\min\{J(\lambda)\mid\lambda\in\{\hat{\lambda}_j^*\}_{j=0}^N\}$.

Algorithm 3: APFBS (7) with our previous parameter tuning [22]

Repeat the following step:

1. Compute $\lambda^*\in\underset{\lambda\in\mathbb{R}_+}{\arg\min}\,\widetilde{J}(\lambda)$ in (11) by Steps (1a)–(1e).
2. Same as Step 2 of Algorithm 2.

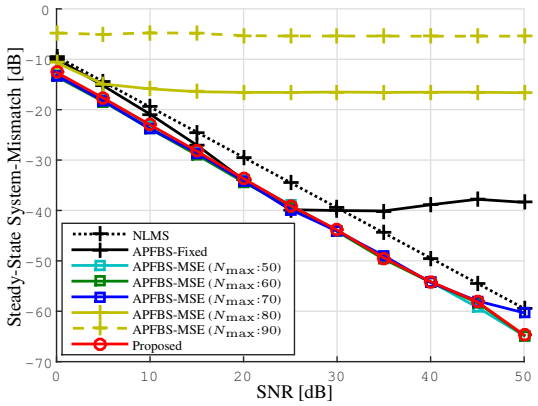
Minimization of $\widetilde{J}$ with efficient $\mathcal{O}(N)$ multiplications

1(a)(b)(c). Same as Steps 1(a)(b)(c) in Algorithm 2.

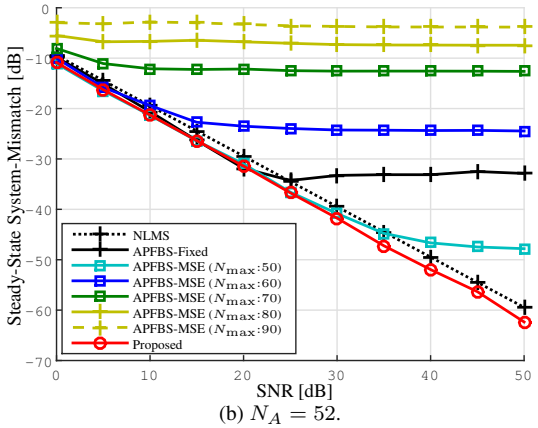
1(d). Compute $(\hat{\lambda}_j^*)_{j=0}^{N_{\max}}$ by

$$\hat{\lambda}_j^*:=P_{[\hat{\lambda}_j,\hat{\lambda}_{j+1}]}\left[\frac{\mathrm{tr}\left[\boldsymbol{A}_k(\hat{\lambda}_j)\,\mathrm{diag}(\boldsymbol{\xi}_k)\right]-d_k}{\mathrm{tr}\left[\boldsymbol{A}_k(\hat{\lambda}_j)\,\mathrm{diag}(\boldsymbol{\zeta}_k)\right]}\right],$$

1(e). Find $\lambda^*\in\ \arg\min\{J(\lambda)\mid\lambda\in\{\hat{\lambda}_j^*\}_{j=0}^{N_{\max}}\}$.



(a) $N_A = 30$.



(b) $N_A = 52$.

Fig. 1. Steady-state performance averaged over 100 trials.

Remark 1: (Comparison with the previous method [22]) *The so-called Tikhonov regularization is incorporated as the last term of J in* (16) *in contrast with $\widetilde{J}$ in* (11)*. It realizes to select a filter consistent with the previous adaptive filter $\boldsymbol{h}_k$, which robustifies the resulting parameter tuning.*

Paradoxically, $\widetilde{J}$ in (11) *does not care for the consistency with $\boldsymbol{h}_k$ as it is defined by instantaneous observations. For example, if the observation vanishes (i.e. $d_k = 0$), any $\overline{\lambda}$ satisfying $\hat{\boldsymbol{h}}_{k+1}(\overline{\lambda}) = \boldsymbol{0}$ (i.e. $\boldsymbol{A}_k(\overline{\lambda}) = \boldsymbol{O}$) is a minimizer of $\widetilde{J}(\lambda)$. In other words, at every time the observation vanishes, the filter coefficient may be initialized to $\boldsymbol{0}$ through the direct minimization of $\widetilde{J}$, which discards coefficients already trained. This demonstrates incorrect tuning based on the direct minimization of $\widetilde{J}$. To avoid this, we employed in [22] a heuristic to limit the choice of λ by introducing a user-defined parameter $N_{\max}$ (see Step 1(e) in Algorithm 3).*

Meanwhile, Algorithm 2 does not require $N_{\max}$ because the above situations are avoided by the Tikhonov regularization term.

Remark 2: (Computational Cost for Parameter Tuning) *The computational cost of Step 1 in Algorithm 2 is $\mathcal{O}(N \log N)$ comparisons and $\mathcal{O}(N)$ multiplications.* Comparison is required in Steps 1(c)(e). In Step 1(c), the sorting requires $\mathcal{O}(N \log N)$ comparisons. In Step 1(e), the $N+1$ elements in $\{J(\hat{\lambda}_j^*)\}_{j=0}^N$ are compared hence $\mathcal{O}(N)$. Multiplication is required in every step. Steps 1(a)(c)(d) require $\mathcal{O}(N)$ multiplications. Step 1(b) depends on the weight design but typically has $\mathcal{O}(N)$. Step 1(e) can be implemented with $\mathcal{O}(N)$ multiplications because the difference $\Delta J_j := J(\hat{\lambda}_{j+1}^*) - J(\hat{\lambda}_j^*)$ can be computed with $\mathcal{O}(1)$ multiplications.

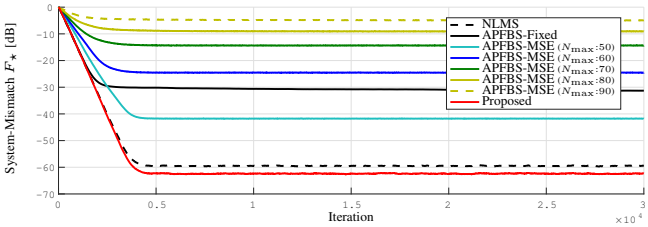


Fig. 2. Learning curves averaged over 100 trials ($N_A = 52$ and SNR $= 50$).

4. NUMERICAL EXAMPLE

We examine the efficacy of the proposed parameter tuning technique. The unknown system $\boldsymbol{h}_\star \in \mathbb{R}^N$ ($N = 100$) is generated artificially to be sparse, where we consider two scenarios: $\boldsymbol{h}_\star$ has $N_A = 30$ or $N_A = 52$ active coefficients. The additive noise $(\epsilon_k)_{k\geq 0}$ is drawn from the zero mean Gaussian noise with unit variance. The input signal $(u_k)_{k\geq 0}$ is also generated from the zero mean Gaussian noise, and the SNR is varied in 5dB increments from 0dB to 50dB. We adopt as a performance measure the system-mismatch

$$F_\star(\boldsymbol{h}_k) = 10 \log_{10} \left(\|\boldsymbol{h}_k - \boldsymbol{h}_\star\|^2 / \|\boldsymbol{h}_\star\|^2 \right) \tag{17}$$

of $\boldsymbol{h}_k$ normalized by $\|\boldsymbol{h}_\star\|^2$.

Four adaptive filtering algorithms are examined: the normalized least mean square (NLMS) [31]; the APFBS (7) with adaptively weighted soft-thresholding of fixed parameter (labeled as APFBS-Fixed) [8]; Algorithm 3 (referred to as APFBS-MSE) [22]; Algorithm 2 (referred to as Proposed). The step-size of the algorithms is chosen as $\mu_k = 0.2$. We adopt as the weight design in [14]

$$\omega_k^{(j)} := \left(\left| h_k^{(j)} \right|^{1-p} + \nu \right)^{-1}, \tag{18}$$

where $\nu > 0$ is a small positive constant. In this experiment, we set $p = 0$ and $\nu = 10^{-5}$ (see [14] for a superior performance of the choice $p = 0$ compared with different choices). The parameter $\lambda_k = 4.5 \times 10^{-2}$ of APFBS-Fixed is chosen to minimize the system-mismatch at 25dB for $N_A = 30$. The interest region of λ of the APFBS-MSE is limited as $N_{\max} = 50, 60, 70, 80$, or 90. All the algorithms are terminated at 30000 iterations. The steady-state system-mismatch is evaluated by the average of $F_\star(\boldsymbol{h}_k)$ over the last 10000 iterations.

Figure 1 depicts the steady-state performance averaged over 100 trials. Figure 1(a) shows that the proposed method achieves excellent steady-state performance over observed SNR situations, while APFBS-Fixed deteriorates the performance in high SNR. In this case, APFBS-MSE appears good performance if the parameter $N_{\max}$ is chosen appropriately ($N_{\max} = 50, 60$). Meanwhile, Figure 1(b) illustrates that the suitable parameter $N_{\max}$ is directly affected by the number N_A of active coefficients of $\boldsymbol{h}_\star$. In this case, all the choice of $N_{\max}$ fail to select a suitable shrinkage parameter in a certain SNR range. Even for this situation, the proposed tuning achieves robustness against environmental changes. Finally, Figure 2 illustrates learning curves averaged over 100 trials, which shows that the proposed parameter tuning does not affect convergence speed of adaptive learning.

5. CONCLUDING REMARKS

We have proposed an automatic shrinkage parameter tuning for a sparsity-aware variant of the APFBS based on the minimization of an estimation of the system-mismatch. In addition, we have introduced its efficient implementation with $\mathcal{O}(N)$ multiplications. A numerical example demonstrated that the proposed method have succeeded in selecting suitable shrinkage parameter robustly against environmental changes. Our future work includes extensions of the proposed parameter tuning strategy to various sparsity-aware adaptive filtering algorithms, e.g., [1–7, 10, 17–19].

6. REFERENCES

[1] S. Makino and Y. Kaneda, "Exponentially weighted 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-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[3] A. W. H. Khong and P. A. Naylor, "Efficient use of sparse adaptive filters," in *Proc. IEEE Asilomar Conference on Signals, Systems and Computers (ACSSC)*, 2006, pp. 1375–1379.

[4] 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.

[5] Y. Chen, Y. Gu, and A. O. Hero III, "Sparse LMS for system identification," in *Proc. IEEE ICASSP*, 2009, pp. 3125–3128.

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

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

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

[9] M. Yamagishi, M. Yukawa, and I. Yamada, "Sparse system identification by exponentially weighted adaptive parallel projection and generalized soft-thresholding," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2010, pp. 367–370.

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

[11] M. Yamagishi, M. Yukawa, and I. Yamada, "Acceleration of adaptive proximal forward-backward splitting method and its application to sparse system identification," in *Proc. IEEE ICASSP*, 2011, pp. 4296–4299.

[12] I. Yamada, S. Gandy, and M. Yamagishi, "Sparsity-aware adaptive filtering based on a Douglas-Rachford splitting," in *Proc. EUSIPCO*, 2011, pp. 1929–1933.

[13] A. Asaei, M. J. Taghizadeh, H. Bourlard, and V. Cevher, "Multi-party speech recovery exploiting structured sparsity models," in *Proc. INTERSPEECH*, 2011, pp. 185–188.

[14] M. Yukawa, Y. Tawara, M. Yamagishi, and I. Yamada, "Sparsity-aware adaptive filters based on ℓ_p-norm inspired soft-thresholding technique," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2012, pp. 2749–2752.

[15] M. Yamagishi and I. Yamada, "Exploiting sparsity in feed-forward active noise control with adaptive Douglas-Rachford splitting," in *Proc. APSIPA ASC*, 2013, p. 6pp.

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

[17] S. Ciochina, C. Paleologu, J. Benesty, and S. L. Grant, "An optimized proportionate adaptive algorithm for sparse system identification," in *Proc. IEEE Asilomar Conference on Signals, Systems and Computers*, 2015, pp. 1546–1550.

[18] Y. Kopsinis, S. Chouvardas, and S. Theodoridis, "Sparse models in echo cancellation: When the old meets the new," *Trends in Digital Signal Processing: A Festschrift in Honour of AG Constantinides*, p. 175, 2015.

[19] 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.

[20] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[21] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[22] M. Yamagishi, M. Yukawa, and I. Yamada, "Shrinkage tuning based on an unbiased MSE estimate for sparsity-aware adaptive filtering," in *Proc. IEEE ICASSP*, 2014, pp. 5477–5481.

[23] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," *Computer*, vol. 21, no. 3, pp. 25–39, 1988.

[24] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms*, vol. 1–2, Springer-Verlag, 1993.

[25] J.-J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *CR Acad. Sci. Paris Sér. A Math*, vol. 255, pp. 2897–2899, 1962.

[26] 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.

[27] I. Yamada, M. Yukawa, and M. Yamagishi, "Minimizing the Moreau envelope of nonsmooth convex functions over the fixed point set of certain quasi-nonexpansive mappings," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 345–390. Springer, 2011.

[28] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[29] M. Yamagishi and I. Yamada, "Over-relaxation of the fast iterative shrinkage-thresholding algorithm with variable stepsize," *Inverse Problems*, vol. 27, no. 10, pp. 105008, 2011.

[30] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numerical Functional Analysis and Optimization*, pp. 593–617, 2004.

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