

METHODOLOGY

Open Access



Improving sign-algorithm convergence rate using natural gradient for lossless audio compression

Taiyo Mineo*  and Hayaru Shouno

Abstract

In lossless audio compression, the predictive residuals must remain sparse when entropy coding is applied. The sign algorithm (SA) is a conventional method for minimizing the magnitudes of residuals; however, this approach yields poor convergence performance compared with the least mean square algorithm. To overcome this convergence performance degradation, we propose novel adaptive algorithms based on a natural gradient: the natural-gradient sign algorithm (NGSA) and normalized NGSA. We also propose an efficient natural-gradient update method based on the AR(p) model, which requires $\mathcal{O}(p)$ multiply-add operations at every adaptation step. In experiments conducted using toy and real music data, the proposed algorithms achieve superior convergence performance to the SA. Furthermore, we propose a novel lossless audio codec based on the NGSA, called the natural-gradient autoregressive unlossy audio compressor (NARU), which is open-source and implemented in C. In a comparative experiment with existing, well-known codecs, NARU exhibits superior compression performance. These results suggest that the proposed methods are appropriate for practical applications.

Keywords: Adaptive algorithm, Autoregressive model, Lossless audio codec, Natural gradient method, Sign algorithm

1 Introduction

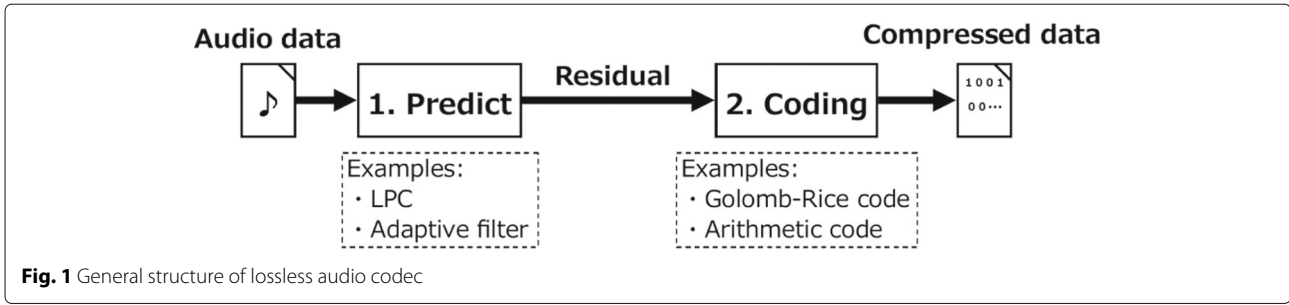
Greater storage capacity is required to further enrich digital audio content [1]. Therefore, lossless audio coding, which allows audio data compression without information loss, is vital for various applications, such as lossless music delivery, editing, and recording [2]. Figure 1 depicts the general structure of a lossless audio codec [3]. First, the codec converts the audio signal to a residual via prediction using a mathematical model. Second, it compresses the residual through entropy coding. If the model provides an accurate prediction, the residual signal is sparse and, thus, high compression performance is achieved. The Shorten lossless codec [4] was one of the first codecs with the structure shown in Fig. 1, and several codecs that follow the same structure have been implemented since. For

example, MPEG4-ALS [5], ALAC [6], and FLAC [7] use linear predictive coding (LPC) as the predictive model, whereas WavPack [8], TTA [9], and Monkey's Audio [10] use adaptive filters. LPC is generally formulated based on the assumption that the residual follows a Gaussian distribution; hence, FLAC and MPEG4-ALS are based on Gaussian distribution. In contrast, Wavpack, TTA, and Monkey's Audio are based on Laplacian distribution, using adaptive algorithms.

In entropy coding, the Golomb–Rice code [11] is generally employed, as this code is optimal when the residual follows a Laplace distribution. Therefore, a residual assumption for LPC is mismatched. To overcome this problem, Kameoka et al. [12] improved the compression rate by formulating an LPC under a Laplace distribution. The sign algorithm (SA) [13] is a practical choice for the adaptive algorithm when the residual follows a Laplace

*Correspondence: m2040009@edu.cc.uec.ac.jp

Graduate School of Informatics and Engineering, The University of Electro-Communications, 1–5–1 Chofugaoka, 182–8585 Chofu, Japan



distribution; however, the SA converges at a considerably slower rate than that of the least mean square (LMS) algorithm [14].

To overcome this performance gap, several SA variants such as the convex combination [15] and the logarithmic cost function [16] have been proposed. However, these attempts have not yielded superior convergence performance to the normalized LMS (NLMS). Notably, the algorithm proposed by Gay and Douglas [17] outperforms the NLMS through the use of a natural gradient [18].

In this study, we improve the SA convergence performance using a natural gradient. We propose two novel adaptive algorithms: the natural-gradient sign algorithm (NGSA) and normalized NGSA (NNGSA) [19]. These algorithms employ $\mathcal{O}(p)$ multiply-add operations to calculate the natural gradient at every step based on the p -th order autoregressive model assumption for the input data. The proposed algorithms achieve superior convergence performance to the SA. Furthermore, we propose a novel lossless audio codec based on the NGSA, called the natural-gradient autoregressive unlossy audio compressor (NARU) (Taiyo Mineo, Hayaru Shouno: NARU: Natural-gradient AutoRegressive Unlossy Audio Compressor, submitted), which is implemented and published under the MIT license. NARU exhibits superior compression performance to existing codecs such as FLAC, WavPack, TTA, and MPEG4-ALS. Moreover, its decoding speed is faster than that of Monkey's Audio without strict optimization.

The remainder of this paper is organized as follows: Section 2 provides an overview of the relevant mathematical theories; Section 3 presents the proposed methods and the NARU codec structure; Section 4 reports computer-based experiments to demonstrate the performance of the proposed algorithms and codec; and Sections 5 and 6 present the discussion and conclusion, respectively.

2 Theoretical background

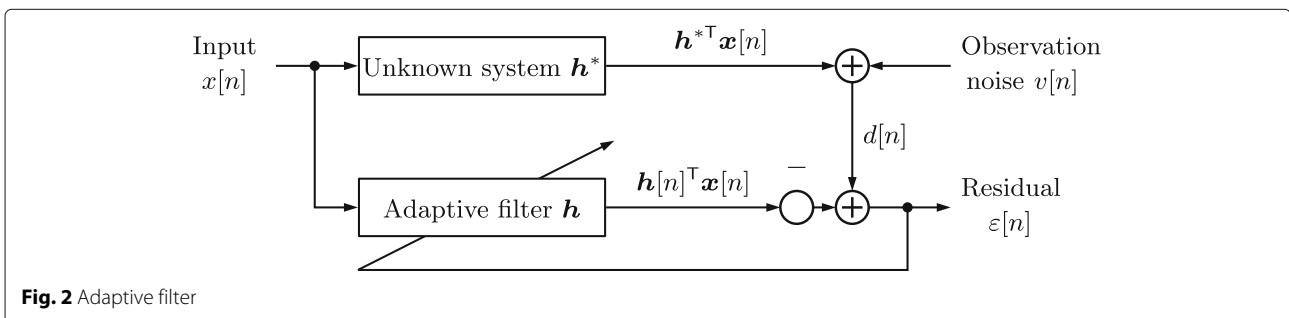
2.1 Adaptive filter

An overview of an adaptive filter is shown in Fig. 2. The input signal $x[n]$ and observation noise $v[n]$ are discrete-time signal sequences. $v[n]$ is a noise adding for unknown system outputs. In this study, $x[n]$ is assumed to have weak stationarity and to be an ergodic process. Let $\mathbf{h}[n] = [h_1[n], \dots, h_N[n]]^T$ be the adaptive filter coefficients, where T represents the matrix transposition. This study employs a finite impulse response (FIR) filter. Hence, the filter output is denoted as $\mathbf{h}[n]^T \mathbf{x}[n]$, where $\mathbf{x}[n] = [x[n-N+1], \dots, x[n]]^T$ represents the input vector. We denote the coefficient vector for an unknown system as \mathbf{h}^* . Filter adaptation is performed by updating the $\mathbf{h}[n]$ coefficients based on the observed signal

$$d[n] := \mathbf{h}^{*T} \mathbf{x}[n] + v[n], \quad (1)$$

and the residual

$$\varepsilon[n] := d[n] - \mathbf{h}[n]^T \mathbf{x}[n]. \quad (2)$$



2.2 Sign algorithm (SA)

The SA is derived using the maximum likelihood method under the assumption that $\varepsilon[n]$ follows a Laplace distribution. The probability density function of the Laplace distribution is

$$p(\varepsilon[n] | \mathbf{h}) = \frac{1}{2\sigma} \exp \left[-\frac{|\varepsilon[n]|}{\sigma} \right], \quad (3)$$

where $\sigma > 0$ represents the deviation. The likelihood $L(\mathbf{h})$ and log-likelihood $\log L(\mathbf{h})$ functions for M independent and identically distributed (i.i.d.) samples are expressed as

$$L(\mathbf{h}) = \frac{1}{(2\sigma)^M} \prod_{k=1}^M \exp \left[-\frac{|\varepsilon[k]|}{\sigma} \right], \quad (4)$$

$$\log L(\mathbf{h}) = -M \log(2\sigma) - \frac{1}{\sigma} \sum_{k=1}^M |\varepsilon[k]|. \quad (5)$$

We let $M = 1$ because the SA adapts at each step. To maximize the likelihood, we partially differentiate $\log L(\mathbf{h})$ with respect to \mathbf{h}

$$\frac{\partial \log L(\mathbf{h})}{\partial \mathbf{h}} = \frac{1}{\sigma} \text{sgn}(\varepsilon[n]) \mathbf{x}[n], \quad (6)$$

where $\text{sgn}(\cdot)$ denotes the sign function, which is defined as

$$\text{sgn}(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x = 0) \\ -1 & (x < 0) \end{cases}. \quad (7)$$

The SA adaptation rule is expressed as

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \mu \text{sgn}(\varepsilon[n]) \mathbf{x}[n], \quad (8)$$

where $\mu > 0$ denotes the step-size parameter.

2.3 Autoregressive model

To simplify the inverse calculation for an autocorrelation matrix for the input signal, we introduce an autoregressive model. Here, $\text{AR}(p)$ indicates the autoregressive model with order p that satisfies the following equation for signal s :

$$s[n] = \sum_{i=1}^p \psi_i s[n-i] + v[n], \quad \psi_i \in \mathbb{R} \quad (i = 1, \dots, p), \quad (9)$$

where $v[n]$ is a sample from an independent standard normal distribution. The i th row and j th column element of the inverse autocovariance matrix for the $\text{AR}(p)$ process \mathbf{K}_p^{-1} is calculated explicitly as [20]

$$\left(\mathbf{K}_p^{-1} \right)_{ij} = \begin{cases} \sum_{k=1}^j \psi_{i-k} \psi_{j-k}, & 1 \leq i \leq p+1 \\ \sum_{k=1}^{L-i+1} \psi_{L-i+1-k} \psi_{L-j+1-k}, & L-p \leq j \leq L \\ 0, & i \geq j+p+1 \\ \sum_{k=i-p}^j \psi_{i-k} \psi_{j-k}, & \text{otherwise} \end{cases}, \quad (10)$$

where $i \geq j$, $\psi_0 = 1$, and L is the matrix size satisfying $L > 2p$.

3 Proposed methods

3.1 Natural-gradient sign algorithm (NGSA)

The natural gradient is derived from the multiplication of the inverse of a Fisher information matrix \mathbf{F}^{-1} and a gradient of the cost function [18]. The matrix \mathbf{F} is calculated using the covariance of the gradient for the log-likelihood function (Eq. (6)), as follows:

$$\mathbf{F} := \mathbb{E} \left[\left\{ \frac{\partial \log L(\mathbf{h})}{\partial \mathbf{h}} \right\} \left\{ \frac{\partial \log L(\mathbf{h})}{\partial \mathbf{h}} \right\}^T \right] \quad (11)$$

$$= \mathbb{E} \left[\left\{ \frac{\text{sgn}(\varepsilon[n])}{\sigma} \right\}^2 \mathbf{x}[n] \mathbf{x}[n]^T \right] \quad (12)$$

$$= \frac{1}{\sigma^2} \mathbb{E} [\mathbf{x}[n] \mathbf{x}[n]^T] \quad (\text{a.s.}) \quad (13)$$

$$= \frac{1}{\sigma^2} \mathbf{R}, \quad (14)$$

where \mathbf{R} is the autocorrelation matrix of the input signal. Note that Eq. (13) holds because $\{\text{sgn}(x)\}^2 = 1$ is satisfied if $x \neq 0$. Using Eq. (14), we obtain the NGSA as follows:

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \mu_{\text{NGSA}} \text{sgn}(\varepsilon[n]) \mathbf{R}^{-1} \mathbf{x}[n], \quad (15)$$

where μ_{NGSA} denotes the step-size parameter and \mathbf{R} is assumed to be a regular matrix. In addition, the NGSA can be derived by replacing $\varepsilon[n]$ with $\text{sgn}(\varepsilon[n])$ in the LMS/Newton algorithm [21], which is an approximation of the Newton method for the LMS algorithm.

The NGSA adaptation rule (Eq. (15)) satisfies the following inequality:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \mathbb{E} [|\varepsilon[k]|] \leq \varepsilon_{\min} + \mu_{\text{NGSA}} \frac{h}{\lambda_{\min}}, \quad (16)$$

where $\varepsilon_{\min} = \mathbb{E} [|\varepsilon[n]|]$, $h = (1/2) \mathbb{E} [\|\mathbf{x}[n]\|_2^2]$, and λ_{\min} denotes the minimum eigenvalue of \mathbf{R} . The proof of Eq. (16) follows that provided in [14] (see Appendix 1: “NGSA inequality”).

3.2 Normalized natural-gradient sign algorithm (NNGSA)

The NNGSA encounters difficulties in determining μ_{NNGSA} because its optimal settings vary according to the input signal. To overcome this difficulty, we introduce a variable step-size adaptation method that minimizes the posterior residual criterion; this approach is identical to that of the NLMS [22].

Let $\mu[n]$ and $\varepsilon^+[n]$ be the adaptive step size and the posterior residual at time n , respectively. Then, $\varepsilon^+[n]$ is calculated as

$$\varepsilon^+[n] := d[n] - \mathbf{h}[n+1]^T \mathbf{x}[n] \quad (17)$$

$$= d[n] - \{\mathbf{h}[n] + \mu[n] \text{sgn}(\varepsilon[n]) \mathbf{R}^{-1} \mathbf{x}[n]\}^T \mathbf{x}[n] \quad (18)$$

$$= \varepsilon[n] - \mu[n] \text{sgn}(\varepsilon[n]) \mathbf{x}[n]^T \mathbf{R}^{-1} \mathbf{x}[n]. \quad (19)$$

We let $\varepsilon^+[n] = 0$; then, solving Eq. (19) for $\mu[n]$, we obtain

$$\mu[n] = \frac{|\varepsilon[n]|}{\mathbf{x}[n]^T \mathbf{R}^{-1} \mathbf{x}[n]}. \quad (20)$$

Substituting Eq. (20) into Eq. (15), we obtain the NNGSA as follows:

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \frac{\mu_{\text{NNGSA}} \varepsilon[n]}{\mathbf{x}[n]^T \mathbf{R}^{-1} \mathbf{x}[n]} \mathbf{R}^{-1} \mathbf{x}[n], \quad (21)$$

where $\mu_{\text{NNGSA}} > 0$ denotes the scale parameter. If $\mu_{\text{NNGSA}} < 2$ holds and $\mathbf{h}[n]$ and $\mathbf{x}[n]$ are statistically independent, this adaptation rule achieves a first-order convergence rate. The proof of this proposition follows that of the NLMS provided in [22] (see Appendix 1: “NNGSA convergence condition”).

The NNGSA can be interpreted as a variable step-size modification of the LMS/Newton algorithm [23]. In [24], the authors state that [23] is a generalization of the recursive least squares (RLS) algorithm. Furthermore, it is

evident that Eq. (21) is identical to the NLMS if $\mathbf{R} = \mathbf{I}$, where \mathbf{I} denotes the identity matrix.

3.3 Geometric interpretation of NNGSA

The adaptation rule in Eq. (21) is used to solve the following optimization problem:

$$\begin{aligned} \underset{\mathbf{h}}{\text{argmin}} \quad & (\mathbf{h} - \mathbf{h}[n])^T \mathbf{R} (\mathbf{h} - \mathbf{h}[n]), \\ \text{subject to} \quad & d[n] = \mathbf{h}^T \mathbf{x}[n]. \end{aligned} \quad (22)$$

The Lagrange multiplier can be used to solve the aforementioned problem. Therefore, Eq. (21) projects $\mathbf{h}[n]$ onto the hyperplane $W = \{\mathbf{h} \mid d[n] = \mathbf{h}^T \mathbf{x}[n]\}$, the metric of which is defined as \mathbf{R} (see Fig. 3). Moreover, according to information geometry [25], the Kullback–Leibler divergence $\text{KL}[\cdot \parallel \cdot]$ for models associated with the neighborhoods of parameter $\mathbf{h}[n]$ can be calculated as

$$\begin{aligned} \text{KL}[p(\varepsilon[n] \mid \mathbf{h}[n]) \parallel p(\varepsilon[n] \mid \mathbf{h})] \\ \approx \frac{1}{2} (\mathbf{h} - \mathbf{h}[n])^T \mathbf{F} (\mathbf{h} - \mathbf{h}[n]) \end{aligned} \quad (23)$$

$$= \frac{1}{2\sigma^2} (\mathbf{h} - \mathbf{h}[n])^T \mathbf{R} (\mathbf{h} - \mathbf{h}[n]). \quad (24)$$

Thus, Eq. (21) can be considered the m-projection from model $p(\varepsilon[n] \mid \mathbf{h}[n])$ to the statistical manifold $S = \{p(\varepsilon[n] \mid \mathbf{h}) \mid d[n] = \mathbf{h}^T \mathbf{x}[n]\}$, the elements of which have the minimum posterior residual.

3.4 Efficient natural-gradient update method

The natural gradient $\mathbf{R}^{-1} \mathbf{x}[n]$ must be calculated at every step. The Sherman–Morrison formula is typically used to reduce RLS complexity; however, this algorithm involves $\mathcal{O}(N^2)$ operations, which generate high cost in practical applications [26]. Therefore, we propose an efficient method to solve this problem.

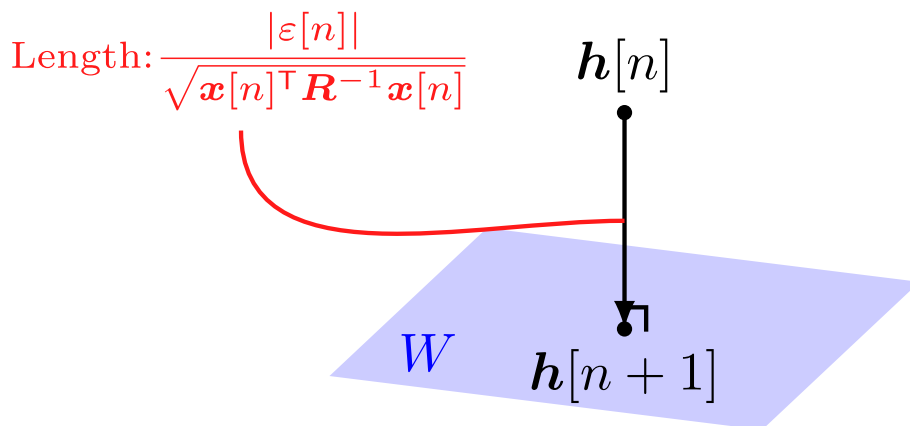


Fig. 3 Geometric interpretation of NNGSA. The NNGSA update procedure (Eq. (21)) projects $\mathbf{h}[n]$ onto hyperplane W , having the metric \mathbf{R}

We assume that the input signals follow the AR(p) process. The natural gradient at time n , i.e., $\mathbf{m}[n] = [m_1[n], \dots, m_N[n]]^T := \mathbf{K}_p^{-1} \mathbf{x}[n]$, can be updated as

$$\begin{aligned} \mathbf{K}_p^{-1} \mathbf{x}[n+1] = & \begin{bmatrix} m_2[n] \\ m_3[n] \\ \vdots \\ m_N[n] \\ 0 \end{bmatrix} + m_1[n] \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_p \\ \mathbf{0}_{N-p} \end{bmatrix} \\ & - m_N[n+1] \begin{bmatrix} \mathbf{0}_{N-p-1} \\ \psi_p \\ \vdots \\ \psi_1 \\ -1 \end{bmatrix}, \quad (25) \\ m_N[n+1] = & x[n+1] - \sum_{i=1}^p \psi_i x[n+1-i], \end{aligned}$$

where $\mathbf{0}_N$ is an $N \times 1$ zero vector. Equation (25) is followed by a direct calculation (see Appendix 1: “Derivation of efficient natural-gradient update method”). Furthermore, the Mahalanobis norm $\mathbf{x}[n]^T \mathbf{K}_p^{-1} \mathbf{x}[n]$ can be updated as follows:

$$\begin{aligned} \mathbf{x}[n+1]^T \mathbf{K}_p^{-1} \mathbf{x}[n+1] &= \mathbf{x}[n]^T \mathbf{K}_p^{-1} \mathbf{x}[n] - m_1[n]^2 + m_N[n+1]^2. \quad (26) \end{aligned}$$

Equation (25) requires $3p$ multiply-add (subtract) calculations, and Eq. (26) requires 2. Hence, we can update the natural gradient in $\mathcal{O}(p)$ operations. Besides, Eq. (25) requires $\mathcal{O}(N)$ space complexity since its referring to previous step gradient $\mathbf{m}[n]$. Equation (25) is essentially the same as that of [27], in which a lattice filter (with partial autocorrelation coefficients) is used for gradient updating. The present method is suitable for norm updating.

Algorithm 1 describes the NNGSA coding procedure under the AR(p) assumption.

3.5 Application to LMS/Newton algorithm

We can apply the proposed procedure to the LMS/Newton algorithm:

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \mu_{\text{LMSN}} \mathbf{R}_p^{-1} \mathbf{x}[n], \quad (27)$$

$$\mathbf{R}_p^{-1} := \sigma_p^{-1} \mathbf{K}_p^{-1}, \quad (28)$$

Algorithm 1 NNGSA with AR(p) assumption

Require: Desired output $d[n]$, Input $x[n]$, Number of inputs M , Filter order N , Scale size μ , Autoregressive order p , Small constant $c > 0$

Ensure: Residual $\varepsilon[n]$

Calculate ψ_i ($i = 1, \dots, p$) using Levinson–Durbin algorithm [28]

$\mathbf{m} \leftarrow \mathbf{0}$, $\mathbf{h} \leftarrow \mathbf{0}$, $w \leftarrow 0$

for $n = 1, \dots, M$ **do**

$\mathbf{x}[n] \leftarrow [x[n-N+1], \dots, x[n]]^T$ $x[n] = 0$ s.t. $n \leq 0$

$\varepsilon[n] \leftarrow d[n] - \mathbf{h}^T \mathbf{x}[n]$

$w \leftarrow w - (\mathbf{m})_1^2$

for $i = 1, \dots, p$ **do**

$(\mathbf{m})_{i+1} \leftarrow (\mathbf{m})_{i+1} + \psi_i (\mathbf{m})_1$

end for

for $i = 1, \dots, N-1$ **do**

$(\mathbf{m})_i \leftarrow (\mathbf{m})_{i+1}$

end for

$(\mathbf{m})_N \leftarrow x[n] - \sum_{i=1}^p \psi_i x[n-i]$

for $i = 1, \dots, p$ **do**

$(\mathbf{m})_{N-i} \leftarrow (\mathbf{m})_{N-i} - \psi_i (\mathbf{m})_N$

end for

$w \leftarrow w + (\mathbf{m})_N^2$

$\mathbf{h} \leftarrow \mathbf{h} + (\mu \varepsilon[n] / \max\{w, c\}) \mathbf{m}$

end for

where $\mu_{\text{LMSN}} > 0$ denotes the step-size parameter and σ_p is a constant that depends on p . For $p = 1$, Eq. (27) achieves first-order convergence if

$$\mu_{\text{LMSN}} < \frac{2(1 - \psi_1)}{N(1 + \psi_1)}. \quad (29)$$

The proof of this proposition follows that for the LMS provided in [21], and employs the eigenvalue range of \mathbf{R}_1 [29] (see Appendix 1: “Convergence condition for LMS/Newton algorithm”).

3.6 Codec structure

This section describes the NARU encoder and decoder.

3.6.1 Encoder

The NARU encoding procedure is illustrated in Fig. 4. Below, we describe each component of the NARU encoding procedure.

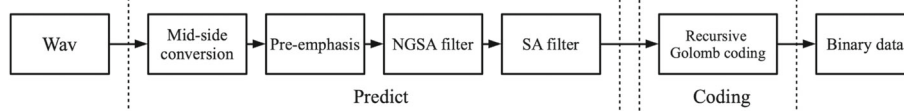


Fig. 4 NARU encoder structure

Mid-side conversion

The mid-side conversion eliminates the inter-channel correlation from the stereo signal. This conversion is expressed as follows:

$$M = \frac{L + R}{2}, \quad (30)$$

$$S = L - R, \quad (31)$$

where L, R, M , and S are the signals of the left, right, mid, and side channels, respectively.

Pre-emphasis

The pre-emphasis is the first-order FIR filter with a fixed coefficient, which is expressed as follows:

$$y[n] = x[n] - \eta x[n-1], \quad (32)$$

where η denotes a constant that satisfies $\eta \approx 1$, and $x[n]$ and $y[n]$ are the filter input and output at time n , respectively. This filter reduces the static offset of the input signal. Hence, we can prevent R from being ill-conditioned [28]. Here, we choose $\eta = 31/32 = 0.96875$ because its division is implemented by a 5-bit arithmetic right shift.

NGSA filter

The NGSA filter is the core predictive model of this codec and is the highest-order ($N \leq 64$) FIR filter. Here, we adopt a rule that follows Algorithm 1 and set $d[n] := x[n+1]$ in Eq. (2) so that the filter equalizes to become the input signal.

SA filter

We cascade the SA filter after the NGSA filter, as this cascaded filter scheme [30] exhibits superior compression performance. This filter has a lower filter order than the NGSA ($N \leq 8$) and follows the same rule as the SA (Eq. (8)).

Recursive Golomb coding

This stage converts the residual signal to a compressed bitstream. We employ recursive Golomb coding [31] as the entropy coder; this is a refinement of the Golomb–Rice code and has exhibited acceptable performance in WavPack and TTA.

3.6.2 Decoder

The decoder structure is shown in Fig. 5. As apparent from the figure, the decoding procedure is simply the inverse of

the encoding procedure: the SA filter and NGSA filter produce the same predictions as for encoding at each instance and, hence, the input signal is perfectly reconstructed. Additionally, the de-emphasis follows

$$x[n] = y[n] + \eta x[n-1], \quad (33)$$

and the left–right conversion is expressed as

$$L = M + \frac{S}{2}, \quad (34)$$

$$R = M - \frac{S}{2}. \quad (35)$$

3.7 Codec implementation

As part of this study, the developed codec was implemented. To ensure speed and portability, we implemented the codec in the C programming language [32]. All encoding/decoding procedures were implemented via fixed-point operations so that the decoder reconstructed the input signal perfectly. We published this implementation under the MIT license.

The fixed-point numbers were represented by 32-bit signed integers with 15 fractional bits. Note that, at present, the codec supports 16-bit linear pulse code modulation (PCM) Wav (Waveform Audio File Format) files only, to prevent multiplication overflow and to maintain implementation simplicity. We assume that the appropriate bit-width rounding is available for 24-bit Wav.

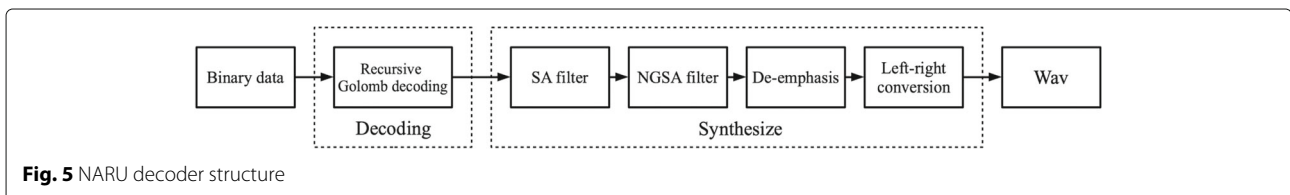
4 Experiment results

This section reports the evaluation results for the proposed algorithms and codec.

4.1 Adaptive algorithm comparison

4.1.1 Toy-data experiments

We observed the convergence performance under the following artificial settings. The elements of the unknown parameter \mathbf{h}^* were randomly chosen with a uniform distribution of $[-1, 1]$, the filter order N was set to 5, and the observation noise $v[n]$ was white Gaussian noise with -20 , -40 , and -60 dB variances. These settings were adopted from [16]. We calculated the mean square deviation (MSD) criteria $\|\mathbf{h}^* - \mathbf{h}\|_2$ from 200 independent trials. In addition, we set $p = 1$ and the following step sizes for the proposed algorithms: $\mu_{\text{NGSA}} = 0.01$, $\mu_{\text{NNGSA}} = 0.1$, and $\mu_{\text{LMSN}} = 0.01$. We implemented the algorithms



in Python 3.8.1 and performed simulations using an Intel® Core-i7 2.8 GHz Dual Core CPU with 16 GB RAM.

First, we tracked the MSD learning curves for $x[n]$ with a variance of 0 dB. Figure 6 shows a comparison between the results obtained for the proposed algorithms and the SA, NLMS, and RLS (see Fig. 9 in Appendix 2 for -20 and -60 dB results). We set various step sizes for the SA and NLMS and employed various forgetting factors λ for the RLS. Figure 6 shows that the NNGSA and NNGSA achieved almost the same performance as the SA and NLMS, respectively. This is because $R_1^{-1} \approx I$ holds for i.i.d. noise input.

Second, we observed the case in which the Gaussian noise is correlated with $x[n] \leftarrow x[n] + x[n-1] \times 0.8$. Figure 7 shows the correlation results (see Fig. 10 in Appendix 2 for -20 and -60 dB results). The SA and NLMS exhibited poorer convergence performance than for the non-correlated noise input (Fig. 6). Moreover, the steady-state errors for the proposed algorithms also deteriorated. This is because R was close to being ill-conditioned, and the right-hand side of Eq. (16) was large.

4.1.2 Real-data experiments

We observed the absolute error (AE) for filter prediction using real music data from the Real World Computing (RWC) music dataset [33]. In this experiment, we assumed that the input data was composed of an audio

data signal only and that the reference output and observation noise was zero (silence). We set the same configurations for the proposed algorithms as in the toy-data experiments. Figure 8 shows the AE curves obtained for the first second (at a 44100 Hz sampling rate) for the left channel of the tune “When the Saints Go Marching In.” From Fig. 8, the NNGSA and LMS/Newton exhibited superior performance to the NLMS and approximately the same performance as the RLS. However, the NNGSA with AR(1) exhibited considerably poorer performance. We assume that this poor performance stemmed from a greater steady-state error for the NNGSA, which arose from long-term (≈ 10000 samples) signal stationarity.

4.2 Codec evaluation

We observed the compression performance under the following settings, treating the following existing codecs as competitors:

FLAC version 1.3.2

with “highest compression” option (-8).

WavPack version 5.4.0

with “very high quality” option ($-hh$).

TTA version 2.3

with default setting.

Monkey’s Audio version 6.14

with “extra high” option ($-c4000$).

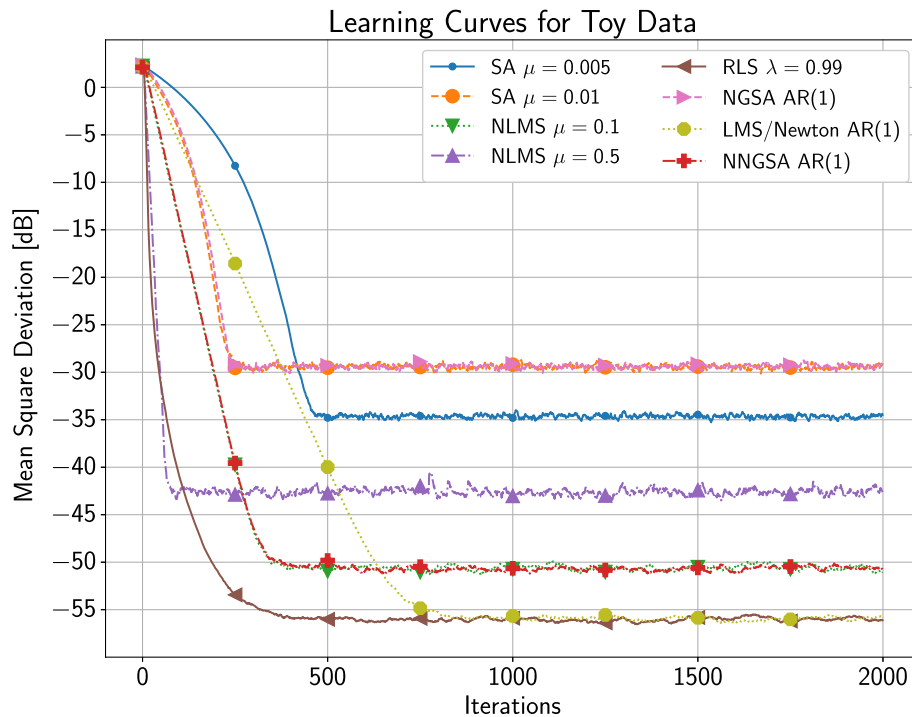
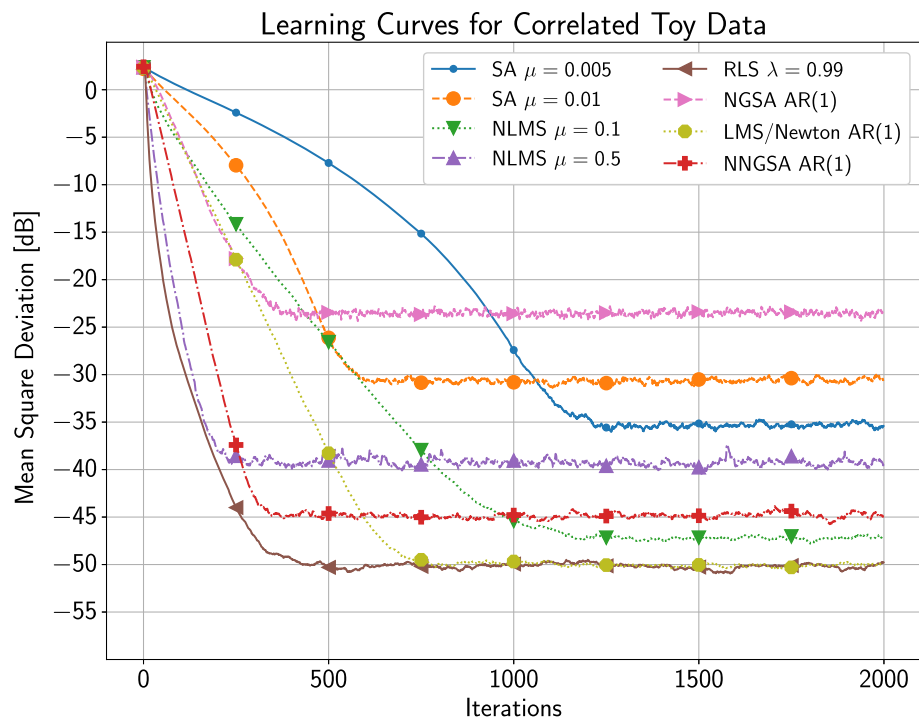
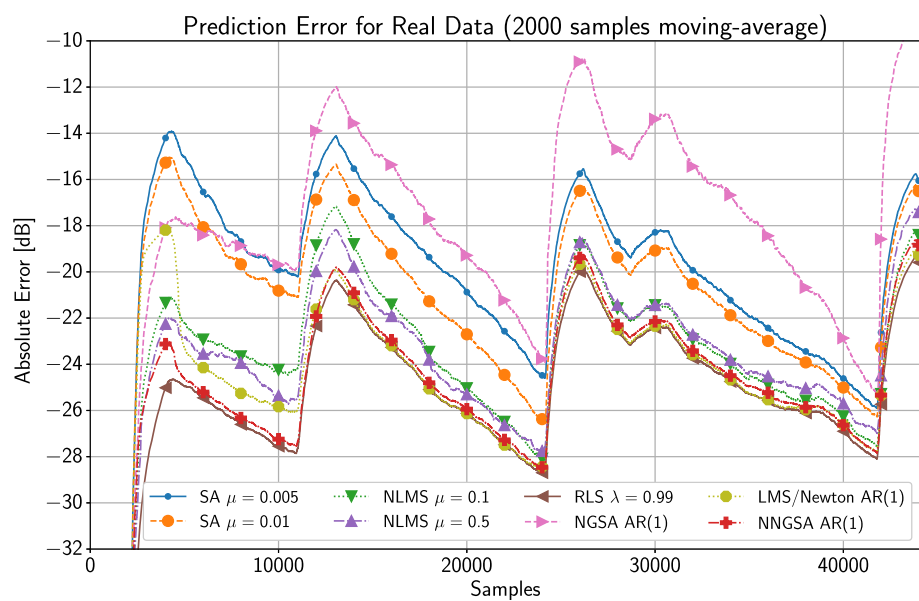


Fig. 6 Learning curves for white-Gaussian-noise input

**Fig. 7** Learning curves for correlated noise input**Fig. 8** AE comparison for real music data

MPEG4-ALS RM23

with default setting. We did not use the optimum compression option (-7) as the required encoding time was unrealistic.

NARU

The NGSa filter order was set to 64, the AR order was 1, and the SA filter order was 8.

There were two evaluation criteria:

compression ratio

$$= \frac{\text{compressed size (byte)}}{\text{original wav file size (byte)}} \times 100 [\%] \quad (36)$$

decoding speed

$$= \frac{\text{decoding time (sec)}}{\text{wav file length (sec)}} \times 100 [\%] \quad (37)$$

We employed the RWC music dataset [33] detailed in Table 1 and measured the root mean square (RMS) amplitude for each music data element. All the music data elements were formatted as Wav files, with 16-bit/sample, a stereo-channel setting, and a 44100 Hz sampling rate. The experiments were conducted on a Windows 10 OS PC having an Intel® Core™ i7-9750H 2.6 GHz CPU with 32 GB RAM.

The compression ratio and decoding speed results are presented in Tables 2 and 3, respectively.

5 Discussion

The proposed algorithms clearly achieved superior convergence performance to the SA and NLMS for correlated signal inputs. Furthermore, the NNGSA and LMS/Newton algorithms exhibited similar performance to the RLS, as indicated in [24]. The NNGSA is not superior to NLMS and RLS in both aspects of convergence speed and steady-state error. However, the NNGSA showed superior performance than the NLMS in highly correlated signals (Fig. 7). In general, digital audio signals exhibit high autocorrelation in small order. Hence, we suggested that the NNGSA showed superior convergence speed than the NLMS for empirical data. Furthermore, the

NNGSA time complexity for update gradient is $O(p)$ per adaptation; hence, its complexity is faster than RLS, which employs the Sharman–Morrison formula ($O(N^2)$). Therefore, we concluded that the NNGSA was a more accurate predictive algorithm than the SA and practical application to a lossless audio codec.

However, the proposed algorithms suffer from two major problems with regard to practical applications. First, matrix \mathbf{R} must be singular and dependent on input signals. For example, a static offset will be zero mean, variance, and autocorrelations by pre-emphasis. One approach to resolving this problem is to introduce regularization, which would involve calculation of the inverse matrix for $\mathbf{R} + \gamma \mathbf{I}$ ($\gamma > 0$) instead of \mathbf{R} . Second, the AR coefficients ψ_i ($i = 1, \dots, p$) must be calculated before the adaptation process, which can generate difficulties for streaming data processing.

As apparent from Tables 2 and 3, although Monkey's Audio yielded the best average compression performance, it also exhibited the lowest decoding speed. This is because Monkey's Audio uses a rich prediction/coding scheme, with a convolutional neural network for prediction and arithmetic coding. In addition, FLAC yielded an inverse trend, i.e., it exhibited the highest decoding speed and poorest compression performance.

NARU exhibited superior compression performance to FLAC, WavPack, TTA, and MPEG4-ALS. This method showed strength in the classical and jazz categories, whereas WavPack exhibited superior performance for popular music. We believe that NARU excels for quieter music, as classical and jazz music tends to have lower signal amplitudes than popular music (see Table 1).

6 Conclusions

We proposed two novel adaptive algorithms that introduce a natural gradient to the SA. The adaptive step-size algorithm, NNGSA, exhibits certain similarities with well-known algorithms such as NLMS and RLS. Furthermore, we demonstrated the superior performance of the proposed algorithms compared with the SA via toy-data and real-music-data experiments. In a future study, we will introduce an iterative method for estimation of the AR

Table 1 Dataset description

Category	Identifier (category suffix)	# of tracks	Total size (MB)	Average RMS (dB)
Classical	RWC-MDB-C-2001 (M01-M04)	34	2157.0	-25.4
Music genre	RWC-MDB-G-2001 (M01-M04)	48	1862.3	-19.3
Jazz	RWC-MDB-J-2001 (M01-M04)	50	2260.0	-19.6
Popular music	RWC-MDB-P-2001 (M01-M04)	64	2587.9	-13.0
Royalty-free	RWC-MDB-R-2001 (M01)	15	331.9	-15.5
Total	-	211	9199.0	-18.2

Table 2 Compression ratio (Eq. (36)) comparison

Category	Codec					
	FLAC	WavPack	TTA	MPEG4-ALS	Monkey's	NARU
Classical	44.22	43.91	43.96	43.67	41.28	43.30
Music genre	58.43	57.64	58.20	58.17	56.21	57.69
Jazz	48.12	47.43	47.64	47.51	45.51	47.27
Popular music	67.91	67.18	67.72	67.70	65.71	67.54
Royalty-free	61.03	60.47	61.00	60.82	58.59	60.25
Total	56.76	56.10	56.49	56.39	54.32	56.07

coefficients and expansion methods for affine projection algorithms [22].

We also proposed a novel lossless audio codec scheme based on the NGSA, namely NARU, which exhibited superior compression performance to existing codecs such as FLAC, WavPack, TTA, and MPEG4-ALS. The NARU decoding speed was lower than those of the other codecs, excluding Monkey's Audio. We found that the filter prediction and coefficient updating processes occupied the majority of the CPU time. Thus, we expect an acceleration of this process through optimization, e.g., though loop unrolling and explicit use of SIMD instructions. Finally, it is remarkable that NARU achieves competitive performance compared to other state-of-the-art codecs despite its simple implementation.

In future work, we will add support for a high-resolution bit (24-bit or higher) Wav and perform further optimization for practical applications, including hardware support. We also plan to employ multichannel decorrelation methods [34] to compression rate improvement for multichannel audio.

We believe that the proposed methods are acceptable to other signal processing tasks, e.g., noise cancellation, audio enhancement, and system identification.

Appendix 1: Proposition proofs

For convenience in the following proofs, we employ the residual vector $\theta[n]$ between an unknown parameter \mathbf{h}^* and a current parameter $\mathbf{h}[n]$, as

$$\theta[n] := \mathbf{h}^* - \mathbf{h}[n], \quad (38)$$

and we define an exponent for the autocorrelation matrix \mathbf{R} as

$$\mathbf{R}^\alpha := \mathbf{Q} \mathbf{\Lambda}^\alpha \mathbf{Q}^\top, \quad \alpha \in \mathbb{Q}, \quad (39)$$

where \mathbf{Q} is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix for which the diagonal elements are eigenvalues of \mathbf{R} .

NGSA inequality

When Eq. (15) is employed,

$$\theta[n+1] = \theta[n] - \mu \text{sgn}(\varepsilon[n]) \mathbf{R}^{-1} \mathbf{x}[n], \quad (40)$$

where $\mu := \mu_{\text{NGSA}}$. Multiplying both sides by $\mathbf{R}^{\frac{1}{2}}$ from the left, and taking the square of the L2 norm $\|\cdot\|_2^2$, we obtain

$$\begin{aligned} & \|\mathbf{R}^{\frac{1}{2}} \theta[n+1]\|_2^2 \\ &= \|\mathbf{R}^{\frac{1}{2}} \theta[n]\|_2^2 - 2\mu \text{sgn}(\varepsilon[n]) \theta[n]^\top \mathbf{x}[n] \\ & \quad + \mu^2 \|\mathbf{R}^{-\frac{1}{2}} \mathbf{x}[n]\|_2^2 \end{aligned} \quad (41)$$

$$\leq \|\mathbf{R}^{\frac{1}{2}} \theta[n]\|_2^2 - 2\mu |\varepsilon[n]| + 2\mu |\nu[n]| + \mu^2 \frac{\|\mathbf{x}[n]\|_2^2}{\lambda_{\min}}. \quad (42)$$

Taking the mean of Eq. (42) yields

$$\begin{aligned} & \mathbb{E} \left[\|\mathbf{R}^{\frac{1}{2}} \theta[n+1]\|_2^2 \right] \\ & \leq \mathbb{E} \left[\|\mathbf{R}^{\frac{1}{2}} \theta[n]\|_2^2 \right] - 2\mu \mathbb{E} [|\varepsilon[n]|] + 2\mu \varepsilon_{\min} \\ & \quad + 2\mu^2 \frac{h}{\lambda_{\min}} \end{aligned} \quad (43)$$

$\leq \dots$

$$\leq r - 2\mu \sum_{k=1}^n \mathbb{E} [|\varepsilon[k]|] + 2n\mu \varepsilon_{\min} + 2n\mu^2 \frac{h}{\lambda_{\min}}, \quad (44)$$

Table 3 Decoding speed (Eq. (37)) comparison

Category	Codec					
	FLAC	WavPack	TTA	MPEG4-ALS	Monkey's	NARU
Total average	0.11	0.41	0.31	0.43	1.06	0.68

where $r = E[\|\mathbf{R}^{\frac{1}{2}}\boldsymbol{\theta}[1]\|_2^2]$. Dividing both sides by $2n\mu$ and rearranging, we obtain

$$\frac{1}{n} \sum_{k=1}^n E[|\varepsilon[k]|] \leq \varepsilon_{\min} + \mu \frac{h}{\lambda_{\min}} + \frac{r}{2n\mu}. \quad (45)$$

Hence, we obtain Eq. (16) by $n \rightarrow \infty$.

NNGSA convergence condition

In the case where Eq. (21) is employed,

$$\boldsymbol{\theta}[n+1] = \mathbf{h}^* - \mathbf{h}[n] - \frac{\mu \varepsilon[n]}{\mathbf{x}[n]^T \mathbf{R}^{-1} \mathbf{x}[n]} \mathbf{R}^{-1} \mathbf{x}[n] \quad (46)$$

$$= (\mathbf{I} - \mu \mathbf{P}[n]) \boldsymbol{\theta}[n] + \frac{\mu \varepsilon^*[n]}{\mathbf{x}[n]^T \mathbf{R}^{-1} \mathbf{x}[n]} \mathbf{R}^{-1} \mathbf{x}[n], \quad (47)$$

where $\varepsilon^*[n] = d[n] - \mathbf{h}^{*T} \mathbf{x}[n]$, $\mathbf{P}[n] = \frac{\mathbf{R}^{-1} \mathbf{x}[n] \mathbf{x}[n]^T}{\mathbf{x}[n]^T \mathbf{R}^{-1} \mathbf{x}[n]}$ and $\mu := \mu_{\text{NNGSA}}$. Taking the mean of Eq. (47), we obtain

$$E[\boldsymbol{\theta}[n+1]] = E[(\mathbf{I} - \mu \mathbf{P}[n]) \boldsymbol{\theta}[n]], \quad (48)$$

as the mean gradient for the unknown parameter is $\mathbf{0}$. Furthermore, $\mathbf{h}[n]$ and $\mathbf{x}[n]$ are statistically independent, such that

$$E[\boldsymbol{\theta}[n+1]] = E[\mathbf{I} - \mu \mathbf{P}[n]] E[\boldsymbol{\theta}[n]]. \quad (49)$$

We can denote $E[\mathbf{P}[n]]$ as

$$E[\mathbf{P}[n]] = \mathbf{Q} \boldsymbol{\Lambda}^{-\frac{1}{2}} \mathbf{R}_q \boldsymbol{\Lambda}^{\frac{1}{2}} \mathbf{Q}^T, \quad (50)$$

where $\mathbf{q}[n] = \boldsymbol{\Lambda}^{-\frac{1}{2}} \mathbf{Q}^T \mathbf{x}[n]$, $\mathbf{R}_q = E\left[\frac{\mathbf{q}[n] \mathbf{q}[n]^T}{\mathbf{q}[n]^T \mathbf{q}[n]}\right]$. Furthermore,

$$\mathbf{R}_q = \mathbf{Q}_q \boldsymbol{\Lambda}_q \mathbf{Q}_q^T \quad (51)$$

holds as \mathbf{R}_q is symmetric, where \mathbf{Q}_q is an orthogonal matrix and $\boldsymbol{\Lambda}_q$ is the diagonal matrix in which the elements are eigenvalues of \mathbf{R}_q . Hence, Eq. (49) is rewritten as

$$E[\mathbf{I} - \mu \mathbf{P}[n]] = \mathbf{Q} \boldsymbol{\Lambda}^{-\frac{1}{2}} \mathbf{Q}_q (\mathbf{I} - \mu \boldsymbol{\Lambda}_q) \mathbf{Q}_q^T \boldsymbol{\Lambda}^{\frac{1}{2}} \mathbf{Q}^T. \quad (52)$$

Therefore, to satisfy $\lim_{n \rightarrow \infty} E[\boldsymbol{\theta}[n]] = \mathbf{0}$,

$$|1 - \mu \lambda_{qi}| < 1 \quad (i = 1, \dots, N) \quad (53)$$

is required, where λ_{qi} is the eigenvalue of \mathbf{R}_q . Here, \mathbf{R}_q is a positive semi-definite matrix and

$$\text{tr}[\boldsymbol{\Lambda}_q] = \text{tr}[\mathbf{R}_q] = 1 \quad (54)$$

holds. Hence, the eigenvalue range is

$$0 \leq \lambda_{qi} \leq 1 \quad (i = 1, \dots, N). \quad (55)$$

Therefore, the convergence condition is obtained when $\max_{i \in \{1, \dots, N\}} \lambda_{qi} = 1$.

Derivation of efficient natural-gradient update method

Employing Eq. (10), the elements of $\mathbf{m}[n]$ can be calculated as

$$\begin{aligned} m_1[n] &= x_{n-N+1} - \psi_1 x_{n-N+2} - \dots - \psi_p x_{n-N+p+1}, \\ m_2[n] &= -\psi_1 x_{n-N+1} + (1 + \psi_1^2) \psi_1 x_{n-N+2} \\ &\quad + \dots + (\psi_1 \psi_p - \psi_{p-1}) x_{n-N+p+1} - \psi_p x_{n-N+p+2}, \\ &\vdots \\ m_{p+1}[n] &= -\psi_p x_{n-N+1} + (\psi_1 \psi_p - \psi_{p-1}) x_{n-N+2} \\ &\quad + \dots + (1 + \psi_1^2 + \dots + \psi_p^2) x_{n-N+p+1} \\ &\quad + \dots + (\psi_1 \psi_p - \psi_{p-1}) x_{n-N+2p} - \psi_p x_{n-N+2p+1}, \\ m_{p+2}[n] &= -\psi_p x_{n-N+2} + (\psi_1 \psi_p - \psi_{p-1}) x_{n-N+3} \\ &\quad + \dots + (1 + \psi_1^2 + \dots + \psi_p^2) x_{n-N+p+2} \\ &\quad + \dots + (\psi_1 \psi_p - \psi_{p-1}) x_{n-N+2p+1} - \psi_p x_{n-N+2p+2}, \\ &\vdots \\ m_{N-p}[n] &= -\psi_p x_{n-2p} + (\psi_1 \psi_p - \psi_{p-1}) x_{n-2p+1} \\ &\quad + \dots + (1 + \psi_1^2 + \dots + \psi_p^2) x_{n-p} \\ &\quad + \dots + (\psi_1 \psi_p - \psi_{p-1}) x_{n-1} - \psi_p x_n, \\ m_{N-p+1}[n] &= -\psi_p x_{n-2p+1} + (\psi_1 \psi_p - \psi_{p-1}) x_{n-2p+2} \\ &\quad + \dots + (1 + \psi_1^2 + \dots + \psi_p^2) x_{n-p+1} \\ &\quad + \dots + (\psi_1 \psi_{p-1} - \psi_{p-2}) x_{n-1} - \psi_p x_n, \\ &\vdots \\ m_{N-1}[n] &= -\psi_p x_{n-p+1} + (\psi_1 \psi_p - \psi_{p-1}) x_{n-p} \\ &\quad + \dots + (1 + \psi_1^2) x_{n-1} - \psi_1 x_n, \\ m_N[n] &= -\psi_p x_{n-p} - \psi_{p-1} x_{n-p+1} - \dots - \psi_1 x_{n-1} + x_n. \end{aligned} \quad (56)$$

Hence, we can denote $\mathbf{m}[n+1]$ as follows:

$$\begin{aligned}
 m_1[n+1] &= m_2[n] + \psi_1 m_1[n], \\
 m_2[n+1] &= m_3[n] + \psi_2 m_1[n], \\
 &\vdots \\
 m_p[n+1] &= m_{p+1}[n] + \psi_p m_1[n], \\
 m_{p+1}[n+1] &= m_{p+2}[n], \\
 &\vdots \\
 m_{N-p-1}[n+1] &= m_{N-p}[n], \\
 m_{N-p}[n+1] &= m_{N-p+1}[n] - \psi_p m_N[n+1], \\
 &\vdots \\
 m_{N-1}[n+1] &= m_N[n] - \psi_1 m_N[n+1], \\
 m_N[n+1] &= -\psi_p x_{n-p+1} - \dots - \psi_1 x_n + x_{n+1},
 \end{aligned} \tag{57}$$

and the Mahalanobis norm $\mathbf{x}[n]^T \mathbf{K}_p^{-1} \mathbf{x}[n] = \mathbf{m}[n]^T \mathbf{x}[n]$ can be updated as follows:

$$\mathbf{x}[n+1]^T \mathbf{K}_p^{-1} \mathbf{x}[n+1] = \mathbf{m}[n+1]^T \mathbf{x}[n+1] \tag{58}$$

$$\begin{aligned}
 &= \sum_{i=2}^N m_i[n] x_{n-N+i} + m_1[n] \sum_{i=1}^p \psi_i x_{n-N+1+i} \\
 &\quad - m_N[n+1] \sum_{i=1}^p \psi_i x_{n+1-i} + m_N[n+1] x_{n+1} \\
 &= \mathbf{m}[n]^T \mathbf{x}[n] \\
 &\quad + m_1[n] \left(-x_{n-N+1} + \sum_{i=1}^p \psi_i x_{n-N+1+i} \right) \\
 &\quad - m_N[n+1] \left(\sum_{i=1}^p \psi_i x_{n+1-i} - x_{n+1} \right) \\
 &= \mathbf{x}[n]^T \mathbf{K}_p^{-1} \mathbf{x}[n] - m_1[n]^2 + m_N[n+1]^2.
 \end{aligned} \tag{59}$$

$$\begin{aligned}
 &= \mathbf{m}[n]^T \mathbf{x}[n] \\
 &\quad + m_1[n] \left(-x_{n-N+1} + \sum_{i=1}^p \psi_i x_{n-N+1+i} \right) \\
 &\quad - m_N[n+1] \left(\sum_{i=1}^p \psi_i x_{n+1-i} - x_{n+1} \right) \\
 &= \mathbf{x}[n]^T \mathbf{K}_p^{-1} \mathbf{x}[n] - m_1[n]^2 + m_N[n+1]^2.
 \end{aligned} \tag{60}$$

$$= \mathbf{x}[n]^T \mathbf{K}_p^{-1} \mathbf{x}[n] - m_1[n]^2 + m_N[n+1]^2. \tag{61}$$

Convergence condition for LMS/Newton algorithm

In the case that Eq. (27) is used,

$$\begin{aligned}
 \mathbf{h}[n+1] &= \mathbf{h}[n] + \mu \mathbf{R}_1^{-1} (d[n] \mathbf{x}[n] - \mathbf{x}[n] \mathbf{x}[n]^T \mathbf{h}[n]), \\
 &= \mathbf{h}[n] + \mu \mathbf{R}_1^{-1} (d[n] \mathbf{x}[n] - \mathbf{x}[n] \mathbf{x}[n]^T \mathbf{h}[n]),
 \end{aligned} \tag{62}$$

where $\mu := \mu_{\text{LMSN}}$. Taking the mean of both sides, we obtain

$$\begin{aligned}
 E[\mathbf{h}[n+1]] &= E[\mathbf{h}[n]] \\
 &\quad + \mu \mathbf{R}_1^{-1} (E[d[n] \mathbf{x}[n]] - E[\mathbf{x}[n] \mathbf{x}[n]^T] E[\mathbf{h}[n]])
 \end{aligned} \tag{63}$$

$$\begin{aligned}
 &= E[\mathbf{h}[n]] \\
 &\quad + \mu \mathbf{R}_1^{-1} (E[d[n] \mathbf{x}[n]] - E[\mathbf{x}[n] \mathbf{x}[n]^T] E[\mathbf{h}[n]])
 \end{aligned} \tag{64}$$

$$= E[\mathbf{h}[n]] + \mu \mathbf{R}_1^{-1} (\mathbf{R} \mathbf{h}^* - \mathbf{R} E[\mathbf{h}[n]]) \tag{65}$$

$$= E[\mathbf{h}[n]] + \mu \mathbf{R}_1^{-1} \mathbf{R} E[\boldsymbol{\theta}[n]]. \tag{66}$$

Here, Eq. (64) exploits the statistical independence between $\mathbf{x}[n]$ and $\mathbf{h}[n]$, and Eq. (65) utilizes the Wiener-Hopf solution. Subtracting \mathbf{h}^* from both sides, we have

$$E[\boldsymbol{\theta}[n+1]] = (\mathbf{I} - \mu \mathbf{R}_1^{-1} \mathbf{R}) E[\boldsymbol{\theta}[n]]. \tag{67}$$

Hence, for $\mathbf{h}[n]$ to converge to \mathbf{h}^* ,

$$0 < \mu < \frac{2}{\eta_{\max}} \tag{68}$$

is required [21], where η_{\max} is the maximum eigenvalue of $\mathbf{R}_1^{-1} \mathbf{R}$. Furthermore, the eigenvalue range of \mathbf{R}_1 satisfies [29] the following:

$$\begin{aligned}
 \lambda_k &= \frac{\sigma^2(1 - \psi_1^2)}{1 - 2\psi_1 \cos \theta_k + \psi_1^2}, \quad (k = 1, \dots, N), \\
 \frac{(k-1)\pi}{N+1} &< \theta_k < \frac{k\pi}{N+1}.
 \end{aligned} \tag{69}$$

More roughly, eigenvalues λ_k ($k = 1, \dots, N$) satisfy

$$\frac{\sigma^2(1 - \psi_1)}{1 + \psi_1} < \lambda_k < \frac{\sigma^2(1 + \psi_1)}{1 - \psi_1}. \tag{70}$$

Therefore, employing the Rayleigh quotient,

$$\eta_{\max} = \max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T \mathbf{R} \mathbf{x}}{\mathbf{x}^T \mathbf{R}_1 \mathbf{x}} = \max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T \mathbf{R} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \frac{\mathbf{x}^T \mathbf{x}}{\mathbf{x}^T \mathbf{R}_1 \mathbf{x}} \tag{71}$$

$$\leq \left(\max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T \mathbf{R} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right) \left(\min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T \mathbf{R}_1 \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right)^{-1} \tag{72}$$

$$< N \sigma^2 \left\{ \frac{\sigma^2(1 - \psi_1)}{1 + \psi_1} \right\}^{-1} \tag{73}$$

$$= \frac{N(1 + \psi_1)}{1 - \psi_1}. \tag{74}$$

Here, Eq. (73) exploits the fact that the maximum eigenvalue of \mathbf{R} is smaller than $\text{tr}[\mathbf{R}] = N\sigma^2$.

Appendix 2: Toy-data experiment results for other configurations

Figures 9 and 10 show learning curves for toy-data experiments for -20 and -60 dB variance configurations.

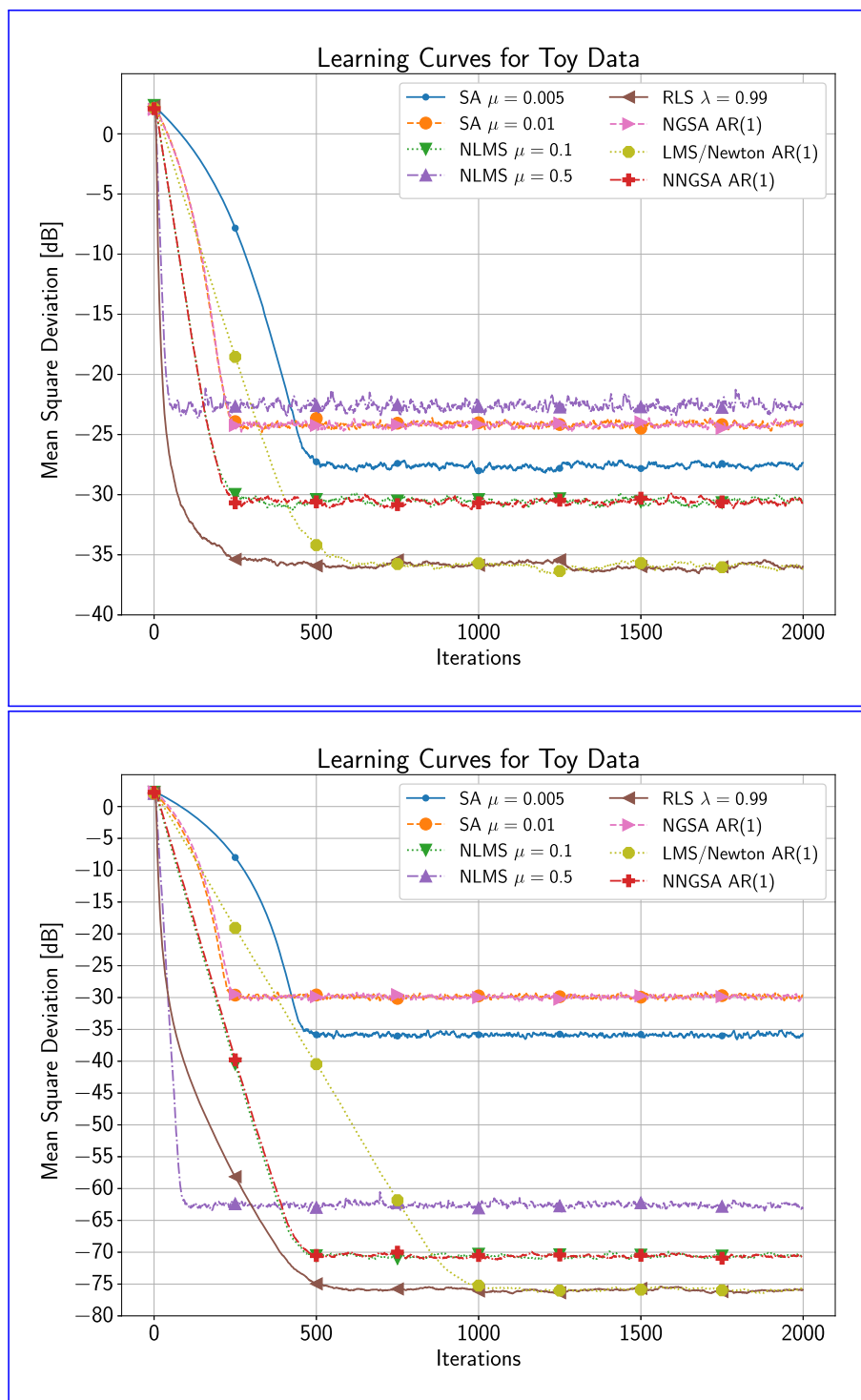


Fig. 9 Learning curves for white-Gaussian-noise input (above: -20 dB, below: -60 dB)

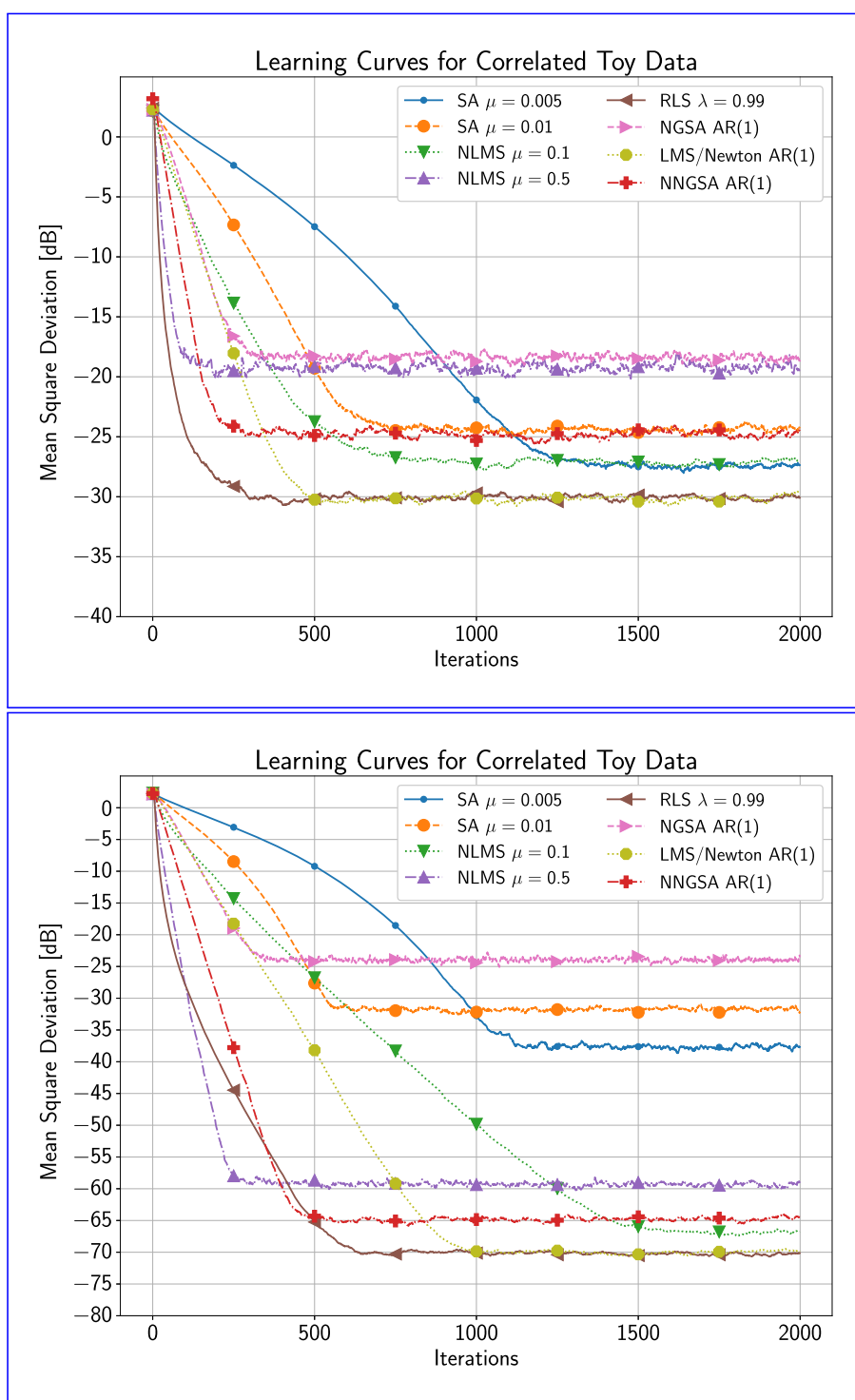


Fig. 10 Learning curves for correlated noise input (above: -20 dB, below: -60 dB)

Abbreviations

SA: Sign algorithm; NSGA: Natural-gradient sign algorithm; LMS: Least mean square; NLMS: Normalized least mean square; NNGSA: Normalized natural-gradient sign algorithm; FIR: Finite impulse response; RLS: Recursive least squares; PCM: Pulse code modulation; MSD: Mean square deviation; AE: Absolute error; RWC: Real-world computing; RMS: Root mean square

Acknowledgements

The authors thank the associate editor and the anonymous reviewers for their constructive comments and useful suggestions.

Authors' contributions

Taiyo Mineo: software and writing—original draft. Hayaru Shouno: writing, review, and editing. Both authors read and approved the final manuscript.

Authors' information

Taiyo Mineo received a B. Eng. from the University of Electro-Communications, Tokyo, in 2014, and received an M. Eng. from the Tokyo Institute of Technology in 2016. He was employed by CRI Middleware Co., Ltd., from 2016 to 2020, and is currently pursuing a Ph.D. in information engineering at the University of Electro-Communications. His research interests include signal processing and machine learning.

Hayaru Shouno received a Ph.D. in Engineering from Osaka University, Osaka, in 1999. He is currently a Professor at the Graduate School of Informatics and Engineering, the University of Electro-Communications, Tokyo. His research interests include computer vision and machine learning involving neural networks. He is an Action Editor of *Neural Networks* and an elected governor of the Asia Pacific Neural Network Society (APNNS).

Funding

Not applicable.

Availability of data and materials

The NARU codec implementation is available at <https://github.com/aikiriao/NARU>.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 1 December 2021 Accepted: 12 April 2022

Published online: 21 May 2022

References

- K. Konstantinides, An introduction to super audio CD and DVD-audio. *IEEE Signal Proc. Mag.* **20**(4), 71–82 (2003)
- T. Moriya, N. Harada, Y. Kamamoto, H. Sekigawa, MPEG-4 ALS international standard for lossless audio coding. *NTT Tech. Rev.* **4**(8), 40–45 (2006)
- M. Hans, R. W. Schaffer, Lossless compression of digital audio. *IEEE Signal Proc. Mag.* **18**(4), 21–32 (2001)
- T. Robinson, Shorten: simple lossless and near-lossless waveform compression. Technical Report, Cambridge Univ., Eng. Dept. (1994)
- T. Liebchen, MPEG-4 ALS-the standard for lossless audio coding. *J. Acoust. Soc. Korea*. **28**(7), 618–629 (2009)
- Apple Lossless Audio Codec (2011). <https://macosforge.github.io/alac/>. Accessed 23 Apr 2022
- FLAC - free lossless audio codec (2011). <https://xiph.org/flac/>. Accessed 23 Apr 2022
- WavPack audio compression (2004). <http://www.wavpack.com>. Accessed 23 Apr 2022
- TTA lossless audio codec - true audio compressor algorithms (2005). http://tausoft.org/wiki/True_Audio_Codec_Overview. Accessed 23 Apr 2022
- Monkey's Audio - a fast and powerful lossless audio compressor (2000). <https://monkeysaudio.com>. Accessed 23 Apr 2022
- R. F. Rice, in *Appl. Digit. Image Process. III*. Practical universal noiseless coding, vol. 207, (1979), pp. 247–267
- H. Kameoka, Y. Kamamoto, N. Harada, T. Moriya, A linear predictive coding algorithm minimizing the Golomb-Rice code length of the residual signal. *Trans. Inst. Electron. Inf. Commun. Eng. A*. **91**, 1017–1025 (2008)
- P. S. Diniz, et al., *Adaptive Filtering*, vol. 4. (Springer, Massachusetts, 1997)
- A. Gersho, Adaptive filtering with binary reinforcement. *IEEE Trans. Inf. Theory*. **30**(2), 191–199 (1984)
- L. Lu, H. Zhao, K. Li, B. Chen, A novel normalized sign algorithm for system identification under impulsive noise interference. *Circ. Syst. Signal Proc.* **35**(9), 3244–3265 (2016)
- M. O. Sayin, N. D. Vanli, S. S. Kozat, A novel family of adaptive filtering algorithms based on the logarithmic cost. *IEEE Trans. Signal Process.* **62**(17), 4411–4424 (2014)
- S. L. Gay, S. C. Douglas, in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. Normalized natural gradient adaptive filtering for sparse and non-sparse systems (IEEE, New York, 2002), p. 1405
- S.-I. Amari, Natural gradient works efficiently in learning. *Neural Comput.* **10**(2), 251–276 (1998)
- T. Mineo, H. Shouno, in *2021 29th European Signal Processing Conference (EUSIPCO)*. Improving convergence rate of sign algorithm using natural gradient method (IEEE, New York, 2021), pp. 51–55
- M. Siddiqui, On the inversion of the sample covariance matrix in a stationary autoregressive process. *Ann. Math. Stat.* **29**(2), 585–588 (1958)
- W. Bernard, D. S. Samuel, *Adaptive signal processing*. (Prentice Hall, Englewood Cliffs, 1985)
- S. S. Haykin, *Adaptive Filter Theory*. (Pearson Education India, 2005)
- P. S. Diniz, L. W. Biscainho, Optimal variable step size for the LMS/Newton algorithm with application to subband adaptive filtering. *IEEE Trans. Signal Process.* **40**(11), 2825–2829 (1992)
- P. S. Diniz, M. L. de Campos, A. Antoniou, Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor. *IEEE Trans. Signal Process.* **43**(3), 617–627 (1995)
- S.-I. Amari, *Differential-Geometrical Methods in Statistics*, vol. 28. (Springer, New York, 2012)
- T. Petillon, A. Gilloire, S. Theodoridis, The fast newton transversal filter: an efficient scheme for acoustic echo cancellation in mobile radio. *IEEE Trans. Signal Process.* **42**(3), 509–518 (1994)
- B. Farhang-Boroujeny, Fast LMS/Newton algorithms based on autoregressive modeling and their application to acoustic echo cancellation. *IEEE Trans. Signal Process.* **45**(8), 1987–2000 (1997)
- J. E. Markel, A. H. Gray, *Linear Prediction of Speech*. (Springer, Berlin, 1982)
- U. Grenander, G. Szegö, *Toeplitz Forms and Their Applications*. (Univ of California Press, California, 1958)
- H. Huang, P. Franti, D. Huang, S. Rahardja, Cascaded RLS-LMS prediction in MPEG-4 lossless audio coding. *IEEE Trans. Audio Speech Lang. Process.* **16**(3), 554–562 (2008)
- D. Salomon, *Data compression: the complete reference*. (Springer Science & Business Media, Berlin/Heidelberg, 2007)
- (International Organization for Standardization, Geneva, 1990)
- M. Goto, H. Hashiguchi, T. Nishimura, R. Oka, RWC music database: popular, classical and jazz music databases. *Ismir*. **2**, 287–288 (2002)
- Y. Kamamoto, N. Harada, T. Moriya, N. Ito, N. Ono, T. Nishimoto, S. Sagayama, in *2009 IEEE 13th International Symposium on Consumer Electronics*. An efficient lossless compression of multichannel time-series signals by MPEG-4 ALS (IEEE, New York, 2009), pp. 159–163

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.