ACCELERATION OF ADAPTIVE NORMALIZED QUASI-NEWTON ALGORITHM WITH IMPROVED UPPER BOUNDS OF THE CONDITION NUMBER

Kenji Kakimoto *Masao Yamagishi* *Isao Yamada*

Department of Information and Communications Engineering, Tokyo Institute of Technology, Japan

ABSTRACT

In 2013, Nguyen and Yamada proposed Adaptive normalized quasi-Newton algorithm and its adaptive step size for accurate and stable extraction of the first generalized eigenvector. The adaptive step size is determined by an upper bound of the condition number of a time-varying matrix. However, the employed upper bound is fairly tight only when the size of matrix is small, which degrades the performance of the algorithm for general case. In this paper, we propose new adaptive step sizes with aid of tighter upper bounds of the condition number. The proposed adaptive step sizes can be implemented efficiently, which are the same calculation order with the original adaptive step size. Numerical experiments show that the proposed adaptive step sizes succeed in extending the applicability of the algorithm.

Index Terms— Generalized Hermitian eigenvalue problem (GHEP), Adaptive normalized quasi-Newton algorithm, Upper bound of condition number, adaptive step size

1. INTRODUCTION

Generalized Hermitian eigenvalue problem (GHEP) is a problem of finding vectors $\boldsymbol{v}_i \in \mathbb{C}^N \setminus \{\boldsymbol{0}\}$, for a pair of Hermitian positive definite matrices $(\boldsymbol{R}_y, \boldsymbol{R}_x) \in \mathbb{C}^{N\times N} \times \mathbb{C}^{N\times N}$, satisfying

$$\boldsymbol{R}_y\boldsymbol{v}_i = \lambda_i \boldsymbol{R}_x\boldsymbol{v}_i \text{ s.t. } \boldsymbol{v}_i^H\boldsymbol{R}_x\boldsymbol{v}_j = \delta_{i,j}\ (i,j=1,2,\ldots,N),$$

where λ_i ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$) is called the ith principal generalized eigenvalue, $(\cdot)^H$ stands for the conjugate transpose of a vector or a matrix, and $\delta_{i,j}$ is the Kronecker delta function. We call $(\boldsymbol{R}_y, \boldsymbol{R}_x)$ and $\boldsymbol{v}_i$ respectively a matrix pencil and the ith principal generalized eigenvector of $(\boldsymbol{R}_y, \boldsymbol{R}_x)$. The GHEP has been attracting great attention in many branches of signal processing, e.g., subspace tracking [1], [2], blind source separation [3], fault detection [4], pattern recognition [5], and array signal processing [6]–[10].

Since for many applications of signal processing, the pair of covariance matrices $(\boldsymbol{R}_y, \boldsymbol{R}_x)$ of input signal are unknown a priori, we have to estimate simultaneously the matrix pencil $(\boldsymbol{R}_y, \boldsymbol{R}_x)$ and generalized eigenvectors. For example $(\boldsymbol{R}_y, \boldsymbol{R}_x)$ is estimated as covariance matrices of input sequences $(\boldsymbol{y}(k))_{k\geq 0}, (\boldsymbol{x}(k))_{k\geq 0}$, where k denotes discrete time index, by

$$\begin{cases} \boldsymbol{R}_y(k+1) = \beta\boldsymbol{R}_y(k) + \boldsymbol{y}(k+1)\boldsymbol{y}(k+1)^H, \\ \boldsymbol{R}_x(k+1) = \alpha\boldsymbol{R}_x(k) + \boldsymbol{x}(k+1)\boldsymbol{x}(k+1)^H. \end{cases} \quad (1)$$

To solve such problems, adaptive estimator of $\boldsymbol{v}_i$ is required but very few effective adaptive estimators are proposed, e.g., [11]–[14]. In [14], Adaptive normalized quasi-Newton algorithm with adaptive step size was proposed, for accurate and stable estimation of the first principal (or minor) generalized eigenvector, where a best upper bound of the condition number [15] of $\boldsymbol{R}(k) := \boldsymbol{R}_x^{-1}(k)\boldsymbol{R}_y(k)$ is employed to determine adaptive step size. This bound is a best in the sense that it achieves the supremum of condition numbers of all positive definite matrices of given trace and determinant of $\boldsymbol{R}(k)$ and N. However, since the upper bound increases exponentially w.r.t. N, the adaptive step size [14] works effectively only when N is small.

In this paper, we propose new adaptive step sizes with aid of tighter upper bounds of the condition number of $\boldsymbol{R}(k)$. Tighter upper bounds are derived from both $\boldsymbol{R}(k)$ and its inverse matrix $\boldsymbol{R}^{-1}(k)$ ($= \boldsymbol{R}_y^{-1}(k)\boldsymbol{R}_x(k)$). The first proposed adaptive step size is designed based on upper bounds in terms of both Frobenius norms and traces of these matrices and N. The calculation cost for this upper bound is a bit large compared with the original step size in [14] but the increasing speed is linear w.r.t. N. The second proposed adaptive step size is designed based on a best upper bound in terms of traces and N only. Indeed, it achieves the supremum of condition numbers of all positive definite matrices of given traces of $\boldsymbol{R}(k)$ and $\boldsymbol{R}^{-1}(k)$ and N. Though the calculation cost of this bound is almost the same as that of the original adaptive step size in [14], the increasing speed is in proportion to N^2. Numerical experiments show that, in a scenario of subspace tracking, estimation with proposed adaptive step sizes is applicable to larger N than estimation with the original adaptive step size.

This work was supported in part by JSPS Grants-in-Aid (15K13986)
(e-mail: {kakimoto, myamagi, isao}@sp.ce.titech.ac.jp).

2. PRELIMINARIES

Let $\mathbb{R}$ and $\mathbb{C}$ be the sets of all real and complex numbers. Bold face capital and bold face small letters respectively express a matrix and a vector. The the $\boldsymbol{B}$-norm of $\boldsymbol{x} := (x_1, x_2, \ldots, x_N)^T \in \mathbb{C}^N$ is defined as $\|\boldsymbol{x}\|_{\boldsymbol{B}} := \sqrt{\boldsymbol{x}^H\boldsymbol{B}\boldsymbol{x}}$, where $\boldsymbol{B} \in \mathbb{C}^{N\times N}$ is a Hermitian positive definite matrix and $(\cdot)^T$ stands for the transpose. Let $\boldsymbol{A} \in \mathbb{C}^{N\times N}$ be a positive definite matrix (not necessarily Hermitian matrix), with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N > 0$. We denote the trace, determinant, and Frobenius norm of $\boldsymbol{A}$ respectively by $\mathrm{tr}(\boldsymbol{A})$, $\det(\boldsymbol{A})$, and $\|\boldsymbol{A}\|_F := \sqrt{\sum_{i,j=1}^N |a_{i,j}|^2}$, where $a_{i,j}$ is the (i,j) component of $\boldsymbol{A}$. Sum and product of the eigenvalues of $\boldsymbol{A}$ coincide with $\mathrm{tr}(\boldsymbol{A})$ and $\det(\boldsymbol{A})$, i.e., $\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^N \lambda_i$ and $\det(\boldsymbol{A}) = \prod_{i=1}^N \lambda_i$. The condition number of $\boldsymbol{A}$ is denoted by $\kappa(\boldsymbol{A}) := \lambda_1/\lambda_N$. For self-containedness, we present a summary of the adaptive eigenvector extraction in [14].

2.1. Normalized quasi-Newton algorithm

Algorithm 1 (Normalized quasi-Newton algorithm [14]) is an estimator of the first principal (or minor) generalized eigenvector of a matrix pencil $(\boldsymbol{R}_y, \boldsymbol{R}_x)$. This algorithm is known for stability and high accuracy of estimation. For convergence of the estimate, the step size η must be selected as

$$\eta \in (0,\ 2/(\kappa(\boldsymbol{R}) - 1))\,, \quad (2)$$

where $\boldsymbol{R} := \boldsymbol{R}_x^{-1}\boldsymbol{R}_y$. In this paper, we only focus on the estimation of the first principal generalized eigenvector extraction because we can estimate first r ($\leq N$) principal generalized eigenvectors $\boldsymbol{v}_i$ ($1 \leq i \leq r$) by combining Algorithm 1 with [16].

2.2. Adaptive estimation of generalized eigenpair

For many applications of signal processing, it is important to solve the generalized eigenvalue problem of the matrix pencil defined by the pairs of covariance matrices. Let $\boldsymbol{R}_y, \boldsymbol{R}_x$ be covariance matrices of input sequences $(\boldsymbol{y}(k))_{k\geq 0}$ and $(\boldsymbol{x}(k))_{k\geq 0}$, where k denotes the discrete time index. In general, the covariance matrices are unknown a priori and we have to estimate them, e.g., by (1), where $(\boldsymbol{R}_y(k), \boldsymbol{R}_x(k))$ is an estimate of the matrix pencil $(\boldsymbol{R}_y, \boldsymbol{R}_x)$. Then, we must estimate the matrix pencil $(\boldsymbol{R}_y, \boldsymbol{R}_x)$ and its first principal generalized eigenvector simultaneously. We can apply Algorithm 1 to adaptive estimation of the first principal generalized eigenvector by replacing $\boldsymbol{R}_y$ and $\boldsymbol{R}_x$ in Algorithm 1 with their estimates $\boldsymbol{R}_y(k)$ and $\boldsymbol{R}_x(k)$. In such a case, their inversion matrices $\boldsymbol{Q}_y(k) := \boldsymbol{R}_y^{-1}(k)$ and $\boldsymbol{Q}_x(k) := \boldsymbol{R}_x^{-1}(k)$ can be obtained recursively with low computational complexity by the matrix inversion lemma [17] as

$$\boldsymbol{Q}_y(k+1) = \frac{1}{\beta}\left[\boldsymbol{Q}_y(k) - \frac{\boldsymbol{Q}_y(k)\boldsymbol{y}(k+1)(\boldsymbol{y}(k+1))^H\boldsymbol{Q}_y(k)}{\beta + (\boldsymbol{y}(k+1))^H\boldsymbol{Q}_y(k)\boldsymbol{y}(k+1)}\right], \tag{3}$$

$$\boldsymbol{Q}_x(k+1) = \frac{1}{\alpha}\left[\boldsymbol{Q}_x(k) - \frac{\boldsymbol{Q}_x(k)\boldsymbol{x}(k+1)(\boldsymbol{x}(k+1))^H\boldsymbol{Q}_x(k)}{\alpha + (\boldsymbol{x}(k+1))^H\boldsymbol{Q}_x(k)\boldsymbol{x}(k+1)}\right], \tag{4}$$

where $\alpha, \beta \in (0,\ 1)$. In adaptive estimation, since the matrix pencil is time-varying, step size η must be chosen adaptively. From (2), for choosing appropriate adaptive step size, the condition number $\kappa(\boldsymbol{R}(k))$ is required, where $\boldsymbol{R}(k) := \boldsymbol{Q}_x(k)\boldsymbol{R}_y(k)$. In [14], a best upper bound of $\kappa(\boldsymbol{R}(k))$ [15] is used, and its additional calculation cost is $5N^2 + \mathcal{O}(N)$ at every k [14].

Fact 1 *(Best upper bound in [15]) For a positive definite matrix $\boldsymbol{A}$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N > 0$ (not necessarily Hermitian matrix), $\kappa(\boldsymbol{A}) := \lambda_1/\lambda_N$ satisfies*

$$\kappa(\boldsymbol{A}) \leq \frac{1 + \sqrt{1 - (N/\mathrm{tr}(\boldsymbol{A}))^N \det(\boldsymbol{A})}}{1 - \sqrt{1 - (N/\mathrm{tr}(\boldsymbol{A}))^N \det(\boldsymbol{A})}} =: \hat{\kappa}_1(\boldsymbol{A}), \tag{5}$$

Moreover, $\hat{\kappa}_1(\boldsymbol{A})$ is the best upper bound of $\kappa(\boldsymbol{A})$ in terms of $\mathrm{tr}(\boldsymbol{A}), \det(\boldsymbol{A}), N$ only, i.e., for any positive definite matrix $\boldsymbol{X}$,

$$\hat{\kappa}_1(\boldsymbol{A}) = \sup_{\boldsymbol{X}\in\mathbb{C}^{N\times N}} \{\kappa(\boldsymbol{X}) \mid \mathrm{tr}(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{A}), \det(\boldsymbol{X}) = \det(\boldsymbol{A})\}. \tag{6}$$

Eq. (6) implies that for any other upper bound $\hat{\kappa}_1'(\boldsymbol{A})$ in terms of $\mathrm{tr}(\boldsymbol{A})$, $\det(\boldsymbol{A})$ and N only, $\hat{\kappa}_1(\boldsymbol{A}) \leq \hat{\kappa}_1'(\boldsymbol{A})$.

2.3. Rapid increasing of existing upper bound

The upper bound (5) increases exponentially w.r.t N, which is confirmed easily from

$$\hat{\kappa}_1(\boldsymbol{A}) = 2c^N\left(1 + \sqrt{1 - c^{-N}}\right) - 1, \tag{7}$$

where

$$c = \frac{\mathrm{tr}(\boldsymbol{A})/N}{\det(\boldsymbol{A})^{1/N}} = \frac{\left(\sum_{i=1}^N \lambda_i\right)/N}{\left(\prod_{i=1}^N \lambda_i\right)^{1/N}} \geq 1. \tag{8}$$

The inequality (8) is derived from arithmetic-geometric-mean inequality. This fact indicates that the range $(0, 2/(\hat{\kappa}_1(\boldsymbol{R}(k)) - 1))$ from which we select adaptive step size shrinks rapidly, and we cannot select adaptive step size large enough for estimation (see blue line in Fig. 1(b)(c) in Sect. 4). Consequently, the update of estimate vanishes, and hence the estimate becomes unchanged (see blue lines in Fig. 2(b)(c) in Sect. 4).

Algorithm 1 Normalized quasi-Newton algorithm [14]

With $\boldsymbol{R}_x$-normalized vector $\boldsymbol{w}(0) \in \mathbb{C}^N$ and $\lambda(0) > 0$, generate the sequence $(\boldsymbol{w}(k), \lambda(k)) \in (\mathbb{C}^N \setminus \{\boldsymbol{0}\}) \times \mathbb{R}$ $(k = 0, 1, \dots)$ by

$$\begin{aligned}
\widehat{\boldsymbol{w}}(k+1) &:= \boldsymbol{w}(k) + \frac{\eta}{\lambda(k)}\Big(\boldsymbol{R}_x^{-1}\boldsymbol{R}_y\boldsymbol{w}(k) \\
&\qquad + (\boldsymbol{w}(k))^H\boldsymbol{R}_y\boldsymbol{w}(k)\boldsymbol{w}(k)\Big) \\
\boldsymbol{w}(k+1) &:= \widehat{\boldsymbol{w}}(k+1)/\|\widehat{\boldsymbol{w}}(k+1)\|_{\boldsymbol{R}_x} \\
\lambda(k+1) &:= (1-\gamma)\lambda(k) + \gamma(\boldsymbol{w}(k+1))^H\boldsymbol{R}_y\boldsymbol{w}(k+1)
\end{aligned}$$

with step sizes $\eta > 0$ and $\gamma \in (0, 1]$, where $(\boldsymbol{w}(k), \lambda(k))$ are the estimates of the first principal generalized eigenvector and eigenvalue.

3. APPLICABLE BOUNDS FOR ADAPTIVE STEP SIZES

We propose to employ tighter upper bounds using partial information of inverse matrix $\boldsymbol{R}^{-1}(k)$. This is because the largest eigenvalue of $\boldsymbol{R}^{-1}$ is $1/\lambda_N$, which implies that $\mathrm{tr}(\boldsymbol{R}^{-1})$ and $\|\boldsymbol{R}^{-1}\|_F$ are dominated by $1/\lambda_N$ and they give nice estimates of $1/\lambda_N$. Moreover, the estimates $\boldsymbol{R}(k)$ and $\boldsymbol{R}^{-1}(k)$ can be obtained efficiently with $\mathcal{O}(N^2)$ in the adaptive generalized eigenvector extraction.

3.1. Bounds in terms of Frobenius norms and traces

We propose to use two upper bounds in terms of Frobenius norms and traces of $\boldsymbol{R}(k)$ and $\boldsymbol{R}^{-1}(k)$ for selecting adaptive step size.

As the first upper bound, we employ an upper bound in [18]

Fact 2 *(Upper bound introduced in [18]) For a positive definite matrix $\boldsymbol{A}$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N > 0$ (not necessarily Hermitian matrix), $\kappa(\boldsymbol{A}) := \lambda_1/\lambda_N$,*

$$\kappa(\boldsymbol{A}) \leq \sqrt{\left[\frac{\overline{t}}{N} + \sqrt{\frac{N-1}{N}\left(\overline{f}^2 - \frac{\overline{t}^2}{N}\right)}\right]\left[\frac{\underline{t}}{N} + \sqrt{\frac{N-1}{N}\left(\underline{f}^2 - \frac{\underline{t}^2}{N}\right)}\right]}$$

$=: \hat{\kappa}_{ft}(\boldsymbol{A})$*, where*

$$\overline{t} := \mathrm{tr}(\boldsymbol{A}),\ \underline{t} := \mathrm{tr}(\boldsymbol{A}^{-1}),\ \overline{f} := \|\boldsymbol{A}\|_F,\ \underline{f} := \|\boldsymbol{A}^{-1}\|_F. \tag{9}$$

This bound $\hat{\kappa}_{ft}(\boldsymbol{A})$ coincides with the true condition number if $N = 2$ or $\kappa(\boldsymbol{A}) = 1$ for $N \geq 3$. As the second upper bound, we derive a best upper bound in terms of Frobenius norms and N only.

Theorem 1 *For a positive definite matrix $\boldsymbol{A}$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N > 0$ (not necessarily Hermitian matrix), $\kappa(\boldsymbol{A}) := \lambda_1/\lambda_N$ satisfies*

$$\kappa(\boldsymbol{A}) \leq \sqrt{\frac{\overline{f}\underline{f} - N + 2 + \sqrt{(\overline{f}\underline{f} - N + 2)^2 - 4}}{\overline{f}\underline{f} - N + 2 - \sqrt{(\overline{f}\underline{f} - N + 2)^2 - 4}}} =: \hat{\kappa}_f(\boldsymbol{A}), \tag{10}$$

where

$$\overline{f} := \|\boldsymbol{A}\|_F,\ \underline{f} := \|\boldsymbol{A}^{-1}\|_F.$$

Moreover, $\hat{\kappa}_f(\boldsymbol{A})$ is the best upper bound of $\kappa(\boldsymbol{A})$ in terms of $\overline{f}, \underline{f}, N$ only, i.e., for any positive definite matrix $\boldsymbol{X}$,

$$\hat{\kappa}_f(\boldsymbol{A}) = \sup_{\boldsymbol{X}\in\mathbb{C}^{N\times N}} \{\kappa(\boldsymbol{X}) \mid \|\boldsymbol{X}\|_F = \overline{f},\ \|\boldsymbol{X}^{-1}\| = \underline{f}\}.$$

Equality holds in (10) *if and only if*

$$\lambda_1 = \sqrt{\frac{\overline{f}}{2\underline{f}}\left(\overline{f}\underline{f} - N + 2 + \sqrt{(\overline{f}\underline{f} - N + 2)^2 - 4}\right)},$$

$$\lambda_N = \sqrt{\frac{\overline{f}}{2\underline{f}}\left(\overline{f}\underline{f} - N + 2 - \sqrt{(\overline{f}\underline{f} - N + 2)^2 - 4}\right)},$$

and

$$\lambda_i = \overline{f}/\underline{f} \quad (i = 2, \ldots, N-1).$$

The word "best" in Theorem 1 is the same meaning of that in Fact 1. The proposed bound $\hat{\kappa}_f(\boldsymbol{A})$ equals to the true condition number even when $\hat{\kappa}_{ft}(\boldsymbol{A})$ does not, for $N \geq 3$. To take advantages of the above two bounds, we propose to use

$$\hat{\kappa}_2(\boldsymbol{R}(k)) = \min(\hat{\kappa}_{ft}(\boldsymbol{R}(k)), \hat{\kappa}_f(\boldsymbol{R}(k))) \tag{11}$$

for selecting adaptive step size from $(0,\ 2/(\hat{\kappa}_2(\boldsymbol{R}(k)) - 1))$.

In the worst case, $\hat{\kappa}_2(\boldsymbol{A})$ increases linear w.r.t. N. This fact is confirmed as

$$\begin{aligned}\hat{\kappa}_2(\boldsymbol{A}) &\leq \hat{\kappa}_f(\boldsymbol{A}) \\ &= \frac{\overline{f}\underline{f} - N + 2 + \sqrt{(\overline{f}\underline{f} - N + 2)^2 - 4}}{2} \\ &< \overline{f}\underline{f} - N + 2 \leq N\left(\kappa(\boldsymbol{A}) - 1\right) + 2.\end{aligned}$$

The last inequality is derived from

$$\overline{f}\underline{f} = \|\boldsymbol{A}\|_F \|\boldsymbol{A}^{-1}\|_F = \sqrt{\left(\sum_{i=1}^{N} \lambda_i^2\right)\left(\sum_{i=1}^{N} \frac{1}{\lambda_i^2}\right)} \leq N\kappa(\boldsymbol{A}).$$

We evaluate the calculation cost of $\hat{\kappa}_2(\boldsymbol{R}(k))$ accurately. For calculation of the upper bound, $\hat{\kappa}_2(\boldsymbol{R}(k))$, $\|\boldsymbol{R}(k)\|_F$ and $\|\boldsymbol{R}^{-1}(k)\|_F$ are required. We evaluate the additional multiplications for obtaining the Frobenius norms. From (1) and (4), the matrix $\boldsymbol{R}(k+1)$ can be recursively calculated from $\boldsymbol{R}(k)$ as

$$\begin{aligned}\boldsymbol{R}(k+1) &= \boldsymbol{Q}_x(k+1)\boldsymbol{R}_y(k+1) \\ &= \frac{\beta}{\alpha}\boldsymbol{R}(k) - \frac{\beta \boldsymbol{l}_x(k+1)(\boldsymbol{l}_x(k+1))^H \boldsymbol{R}_y(k)}{\alpha(\alpha + (\boldsymbol{x}(k+1))^H \boldsymbol{l}_x(k+1))} \\ &\quad + \frac{1}{\alpha}\boldsymbol{l}_y(k+1)(\boldsymbol{y}(k+1))^H \\ &\quad - \frac{((\boldsymbol{x}(k+1))^H \boldsymbol{l}_y(k+1))\boldsymbol{l}_x(k+1)(\boldsymbol{y}(k+1))^H}{\alpha(\alpha + (\boldsymbol{x}(k+1))^H \boldsymbol{l}_x(k+1))},\end{aligned} \tag{12}$$

where $\boldsymbol{l}_x(k+1) = \boldsymbol{Q}_x(k)\boldsymbol{x}(k+1)$ and $\boldsymbol{l}_y(k+1) = \boldsymbol{Q}_x(k)\boldsymbol{y}(k+1)$. For obtaining $\boldsymbol{R}(k)$, we have to calculate $\boldsymbol{l}_y(k+1)$ and (12) ($\boldsymbol{l}_x(k+1)$ is already calculated in (4)). Therefore, to calculate $\boldsymbol{R}(k)$ costs $6N^2 + \mathcal{O}(N)$ and we can obtain $\|\boldsymbol{R}(k)\|_F$ with $7N^2 + \mathcal{O}(N)$ multiplications. In the same way $\|\boldsymbol{R}^{-1}(k)\|_F$ also requires additional $7N^2 + \mathcal{O}(N)$ multiplications. Finally, since we need to additionally calculate $\boldsymbol{Q}_y(k)$ (it costs $3N^2 + \mathcal{O}(N)$ from (3)) for $\boldsymbol{R}^{-1}(k)$ ($\boldsymbol{R}_y(k)$, $\boldsymbol{R}_x(k)$ and $\boldsymbol{Q}_x(k)$ are already calculated in Algorithm 1 at every k), the total additional multiplications to combine $\hat{\kappa}_2(\boldsymbol{R}(k))$ with Algorithm 1 are $17N^2 + \mathcal{O}(N)$ ($\mathrm{tr}(\boldsymbol{R}(k))$ and $\mathrm{tr}(\boldsymbol{R}^{-1}(k))$ can be calculated with no additional multiplications). This additional calculation cost is a slight large compared with that of adaptive step size used in [14].

3.2. Proposed best upper bound with use of traces

We also propose to use another best possible upper bound for $\kappa(\boldsymbol{R}(k))$ with use of $\mathrm{tr}(\boldsymbol{R}(k))$, $\mathrm{tr}(\boldsymbol{R}^{-1}(k))$ and N only. This bound does not increase exponentially w.r.t. N. Moreover, this bound can be calculated with almost the same multiplications of the existing adaptive step size, i.e., $5N^2 + \mathcal{O}(N)$, in a scenario of adaptive estimation.

Table 1. Additional multiplications for upper bounds

	additional multiplications
$\hat{\kappa}_1(\boldsymbol{R}(k))$ (Existing [15])	$5N^2 + \mathcal{O}(N)$ [14]
$\hat{\kappa}_2(\boldsymbol{R}(k))$ (Proposed bound 1)	$17N^2 + \mathcal{O}(N)$
$\hat{\kappa}_3(\boldsymbol{R}(k))$ (Proposed bound 2)	$5N^2 + \mathcal{O}(N)$

Theorem 2 *For a positive definite matrix $\boldsymbol{A}$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N > 0$ (not necessarily Hermitian matrix), $\kappa(\boldsymbol{A}) := \lambda_1/\lambda_N$ satisfies*

$$\kappa(\boldsymbol{A}) \leq \frac{\sqrt{\overline{t}\underline{t}} - N + 2 + \sqrt{(\sqrt{\overline{t}\underline{t}} - N + 2)^2 - 4}}{\sqrt{\overline{t}\underline{t}} - N + 2 - \sqrt{(\sqrt{\overline{t}\underline{t}} - N + 2)^2 - 4}} =: \hat{\kappa}_3(\boldsymbol{A}), \tag{13}$$

where

$$\overline{t} := \mathrm{tr}(\boldsymbol{A}),\ \underline{t} := \mathrm{tr}(\boldsymbol{A}^{-1}).$$

Moreover, $\hat{\kappa}_3(\boldsymbol{A})$ is the best upper bound of $\kappa(\boldsymbol{A})$ in terms of $\overline{t}, \underline{t}$ and N only, i.e., for any positive definite matrix $\boldsymbol{X}$,

$$\hat{\kappa}_3(\boldsymbol{A}) = \sup_{\boldsymbol{X} \in \mathbb{C}^{N \times N}} \{\kappa(\boldsymbol{X}) \mid \mathrm{tr}(\boldsymbol{X}) = \overline{t},\ \mathrm{tr}(\boldsymbol{X}^{-1}) = \underline{t}\}.$$

Equality holds in (13) *if and only if*

$$\lambda_1 = \frac{\sqrt{\overline{t}}}{2\sqrt{\underline{t}}}\left(\sqrt{\overline{t}\underline{t}} - N + 2 + \sqrt{(\sqrt{\overline{t}\underline{t}} - N + 2)^2 - 4}\right),$$
$$\lambda_N = \frac{\sqrt{\overline{t}}}{2\sqrt{\underline{t}}}\left(\sqrt{\overline{t}\underline{t}} - N + 2 - \sqrt{(\sqrt{\overline{t}\underline{t}} - N + 2)^2 - 4}\right),$$

and

$$\lambda_i = \sqrt{\overline{t}/\underline{t}} \quad (i = 2, \ldots, N-1).$$

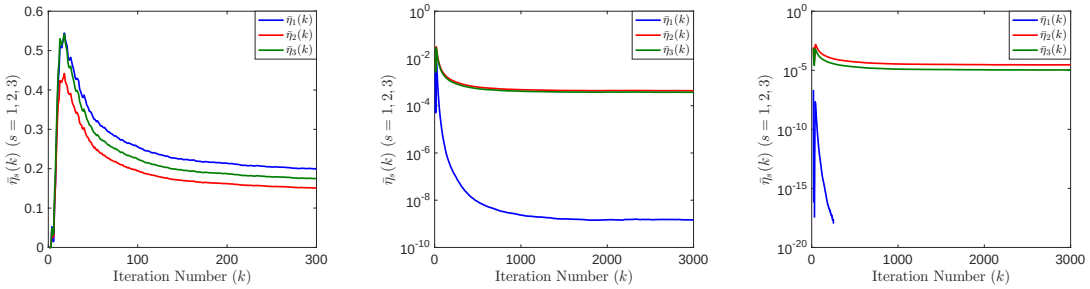
The word "best" in Theorem 2 is the same meaning of that in Fact 1. We propose to select adaptive step size from the range $(0,\ 2/(\hat{\kappa}_3(\boldsymbol{R}(k)) - 1))$.

In the worst case, the upper bound $\hat{\kappa}_3(\boldsymbol{R}(k))$ increases in proportion to N^2 by similar discussion in Sect 3.1. The increasing speed of this bound is much slower compared with the existing bound [15], and hence this bound is applicable for selecting adaptive step size.

We evaluate additional multiplications for $\mathrm{tr}(\boldsymbol{R}(k))$ and $\mathrm{tr}(\boldsymbol{R}^{-1}(k))$. For proposed bound we additionally need to calculate $\boldsymbol{Q}_y(k)$ for $\mathrm{tr}(\boldsymbol{R}^{-1}(k))$ ($\boldsymbol{Q}_x(k)$ is already calculated in Algorithm 1 at every time index k). From (4), additional $3N^2 + \mathcal{O}(N)$ multiplications are needed to obtain $\boldsymbol{Q}_y(k)$. Next, we evaluate additional multiplications for $\mathrm{tr}(\boldsymbol{R}(k))$. It is sufficient to calculate only diagonal components of $\boldsymbol{R}(k)$ for obtaining $\mathrm{tr}(\boldsymbol{R}(k))$. The (i,i) $(1 \leq i \leq N)$ component of $\boldsymbol{R}(k)$ equals to the inner product of ith row vector of $\boldsymbol{Q}_x(k)$ and ith column vector of $\boldsymbol{R}_y(k)$. Since inner product needs N multiplications, we can obtain the sum of diagonal components of $\boldsymbol{R}(k)$, i.e., $\mathrm{tr}(\boldsymbol{R}(k))$ with N^2 multiplications. In the same way, $\mathrm{tr}(\boldsymbol{R}^{-1}(k))$ can be obtained with additional N^2 multiplications. Then, we can calculate adaptive step size with $5N^2 + \mathcal{O}(N)$ additional multiplications, which is almost the same calculation cost of the existing adaptive step size. Finally, we summarize the additional calculation costs of upper bounds for adaptive step sizes in Table 1.

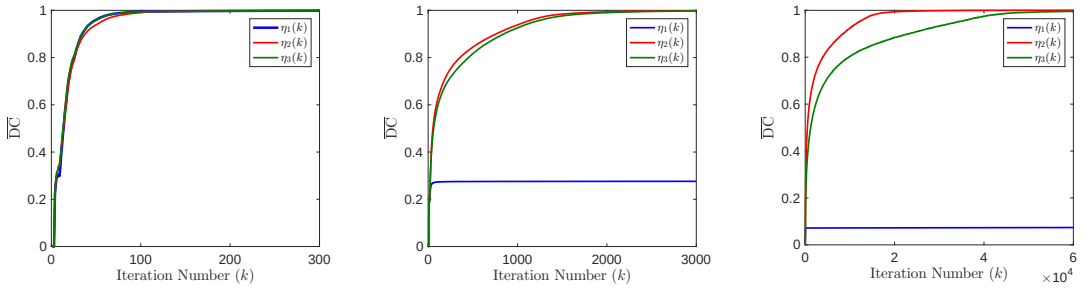
4. NUMERICAL EXPERIMENT

We evaluate proposed adaptive step size selections in a scenario of subspace tracking. We track the principal generalized eigensubspace $\mathrm{span}\{\boldsymbol{v}_i\}_{i=1}^2$ of $(\boldsymbol{R}_y, \boldsymbol{R}_x)$ from input sequences



(a) $N = 4$ (b) $N = 8$ (c) $N = 20$

Fig. 1. The averages of adaptive step sizes $\bar{\eta}_s(k)$ $(s = 1, 2, 3)$ over $L = 100$ runs



(a) $N = 4$ (b) $N = 8$ (c) $N = 20$

Fig. 2. The averages of Direction Cosine $\overline{\mathrm{DC}}$ over $L = 100$ runs

$(\boldsymbol{y}(k))_{k\geq 0}, (\boldsymbol{x}(k))_{k\geq 0}$ by combining Adaptive normalized quasi-Newton algorithm (Algorithm 1) and *Nested orthogonal complement structure* [16]. We compare three adaptive step sizes $\eta_s(k) := (2-\epsilon)/(\hat{\kappa}_s(\boldsymbol{R}(k)) - 1)$ $(s = 1, 2, 3)$, where ϵ is a small positive value. As comparison criteria, we observe the similarity between two subspaces $\mathrm{span}\{\boldsymbol{v}_i\}_{i=1}^2$ and $\mathrm{span}\{\boldsymbol{w}_{i,j}(k),\}_{i=1}^2$ (the estimate of $\mathrm{span}\{\boldsymbol{v}_i\}_{i=1}^2$ at time k in the jth independent run) in terms of Direction Cosine of the angle between subspaces

$$\mathrm{DC}_j(k) := \cos\left(\mathrm{angle}(\mathrm{span}\{\boldsymbol{v}_i\}_{i=1}^2, \mathrm{span}\{\boldsymbol{w}_{i,j}(k)\}_{i=1}^2)\right).$$

Define the averages of $\mathrm{DC}_j(k)$ in L independent runs as $\overline{\mathrm{DC}}(k) := \frac{1}{L}\sum_{j=1}^{L}\mathrm{DC}_j(k)$. We also observe the averages of adaptive step sizes $\bar{\eta}_s(k) = \frac{1}{L}\sum_{j=1}^{L}\eta_{s,j}(k)$, where $\eta_{s,j}(k)$ is $\eta_s(k)$ at jth independent run $(s = 1, 2, 3)$.

The input samples are generated respectively as

$$y(k) = \sqrt{2}\sin(0.62\pi k + \theta_1) + n_1(k),$$

and

$$x(k) = \sqrt{2}\sin(0.46\pi k + \theta_2) + \sqrt{2}\sin(0.74\pi k + \theta_3) + n_2(k),$$

where the initial phase θ_i $(i = 1, 2, 3)$ has the uniform distribution in $[0, 2\pi]$, $n_1(k)$ and $n_2(k)$ are white Gaussian noise with variance $\sigma^2 = 0.1$. The input vectors $\boldsymbol{y}(k) \in \mathbb{C}^N$ and $\boldsymbol{x}(k) \in \mathbb{C}^N$ ($N = 4, 8, 20$) are defined as $\boldsymbol{y}(k) := (y(k), y(k-1), \ldots, y(k-N+1))^T$ and $\boldsymbol{x}(k) := (x(k), x(k-1), \ldots, x(k-N+1))^T$ $(k \geq N)$. [1] The matrix pencil is estimated by (1) with the parameters $\alpha = \beta = 0.998$, and the initial values $\boldsymbol{R}_y(0) = \boldsymbol{R}_x(0) = \boldsymbol{I}_N$. The parameters for generalized eigenvalue estimation and adaptive step size is respectively set as $\gamma = 0.998$ and $\epsilon = 10^{-15}$.

Fig. 1 and Fig. 2 respectively show the averages of adaptive step sizes $\{\bar{\eta}_s(k)\}_{s=1}^3$ and $\overline{\mathrm{DC}}(k)$ for $N = 4, 8, 20$ over $L = 100$ runs. Fig. 1(a) depicts the existing and the proposed adaptive step sizes for the case $N = 4$. In this case, since adaptive step sizes are chosen large enough, the estimates of generalized eigensubspaces with adaptive step sizes $\{\eta_s(k)\}_{s=1}^3$ converge to the true eigensubspace (Fig. 2(a)). For the case $N = 8$ and $N = 20$, the existing adaptive step size is small (Fig. 1(b)(c) blue line) and fails to extract generalized eigensubspace (Fig. 2(b)(c) blue line) as mentioned in Sect. 2.3. In contrast, the proposed adaptive step sizes are large enough (Fig. 1 (b)(c) green and red lines), and estimates with the proposed adaptive step sizes converge (Fig. 2(b)(c), green and red lines). From these figures, we observe that proposed adaptive step sizes accelerate the convergence speed of Adaptive normalized quasi-Newton algorithm.

5. CONCLUSIONS

We found that the existing step size works only when N is small by analysing the behavior of the existing best upper bound of the condition number w.r.t. N. Attention to the fact that we can obtain the inverse matrix with $\mathcal{O}(N^2)$ in a scenario of subspace tracking, we proposed to utilize tighter upper bounds for selecting applicable adaptive step sizes and evaluated their introduction cost precisely. Numerical experiments showed that the proposed adaptive step sizes are large enough and accelerate the estimation of Adaptive normalized quasi-Newton algorithm.

[1] The covariance matrices $\boldsymbol{R}_y, \boldsymbol{R}_x \in \mathbb{C}^{N\times N}$ are given as

$$\begin{cases} (\boldsymbol{R}_y)_{i,j} := \cos(0.62\pi(j-i)) + \delta_{i,j}\sigma^2, \\ (\boldsymbol{R}_x)_{i,j} := \cos(0.46\pi(j-i)) + \cos(0.74\pi(j-i)) + \delta_{i,j}\sigma^2. \end{cases}$$

These matrices are used for evaluating the true principal generalized eigensubspace, i.e., $\mathrm{span}\{\boldsymbol{v}_i\}_{i=1}^2$.

6. REFERENCES

[1] B. Yang, "Projection approximation subspace tracking," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 95–107, 1995.

[2] Y. Hua, T. Chen, and W.-Y. Yan, "Global convergence of Oja's subspace algorithm for principal component extraction," *IEEE Transactions on Neural Networks*, vol. 9, no. 1, 1998.

[3] C. Chang, Z. Ding, S. F. Yau, and F. H. Chan, "A matrix-pencil approach to blind separation of colored nonstationary signals," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 900–907, 2000.

[4] H. Chen, G. Jiang, and K. Yoshihira, "Failure detection in large-scale internet services by principal subspace mapping," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 10, pp. 1308–1320, 2007.

[5] K. Fukunaga, *Introduction to statistical pattern recognition*. San Diego, CA, USA: Academic, 1990.

[6] T. F. Wong, T. M. Lok, J. S. Lehnert, and M. D. Zoltowski, "A linear receiver for direct-sequence spread-spectrum multiple-access systems with antenna arrays and blind adaptation," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 659–676, 1998.

[7] S. Choi, J. Choi, H.-J. Im, and B. Choi, "A novel adaptive beamforming algorithm for antenna array cdma systems with strong interferers," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 5, pp. 808–816, 2002.

[8] S. Shahbazpanahi, A. B. Gershman, Z.-Q. Luo, and K. M. Wong, "Robust adaptive beamforming for general-rank signal models," *IEEE Transactions on Signal Processing*, vol. 51, no. 9, pp. 2257–2269, 2003.

[9] D. R. Morgan, "Downlink adaptive array algorithms for cellular mobile communications," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 476–488, 2003.

[10] ——, "Adaptive algorithms for solving generalized eigenvalue signal enhancement problems," *Signal processing*, vol. 84, no. 6, pp. 957–968, 2004.

[11] J. Yang, H. Xi, F. Yang, and Y. Zhao, "RLS-based adaptive algorithms for generalized eigen-decomposition," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1177–1188, 2006.

[12] S. Attallah and K. Abed-Meraim, "A fast adaptive algorithm for the generalized symmetric eigenvalue problem," *IEEE Signal Processing Letters*, vol. 15, pp. 797–800, 2008.

[13] T. Tanaka, "Fast generalized eigenvector tracking based on the power method," *IEEE Signal Processing Letters*, vol. 16, no. 11, pp. 969–972, 2009.

[14] T. D. Nguyen and I. Yamada, "Adaptive normalized quasi-newton algorithms for extraction of generalized eigen-pairs and their convergence analysis," *IEEE Transactions on Signal Processing*, vol. 61, no. 6, pp. 1404–1418, 2013.

[15] J. K. Merikoski, U. Urpala, A. Virtanen, T.-Y. Tam, and F. Uhlig, "A best upper bound for the 2-norm condition number of a matrix," *Linear algebra and its applications*, vol. 254, no. 1, pp. 355–365, 1997.

[16] K. Kakimoto, D. Kitahara, M. Yamagishi, and I. Yamada, "Stabilization of adaptive eigenvector extraction by continuation in nested orthogonal complement structure," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4189–4193.

[17] W. W. Hager, "Updating the inverse of a matrix," *SIAM review*, vol. 31, no. 2, pp. 221–239, 1989.

[18] J. Lam, Z. Li, Y. Wei, J. Feng, and K. W. Chung, "Estimates of the spectral condition number," *Linear and Multilinear Algebra*, vol. 59, no. 3, pp. 249–260, 2011.