A Search Algorithm for a Class of Optimal Finite-Precision Controller Realization Problems with Saddle Points

Jun Wu †, Sheng Chen ‡ [0], Gang Li § and Jian Chu †

† National Key Laboratory of Industrial Control Technology
Institute of Advanced Process Control
Zhejiang University, Hangzhou, 310027, P. R. China

‡ School of Electronics and Computer Science
University of Southampton, Highfield, Southampton SO17 1BJ, U.K.

§ School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore

Abstract: With game theory, we review the optimal digital controller realization problems that maximize a finite word length (FWL) closed-loop stability measure. For a large class of these optimal FWL controller realization problems which have saddle points, a minimax-based search algorithm is derived for finding a global optimal solution. The algorithm consists of two stages. In the first stage, the closed-form of a transformation set is constructed which contains global optimal solutions. In the second stage, a subgradient approach searches this transformation set to obtain a global optimal solution. This algorithm does not suffer from the usual drawbacks associated with using direct numerical optimization methods to tackle these FWL realization problems. Furthermore, for a small class of optimal FWL controller realization problems which have no saddle point, the proposed algorithm also provides useful information to help solving them.

Key Words — closed-loop stability, digital controller, finite word length, game theory, optimization, saddle points

Mathematics Subject Classification — 93D99, 91A80, 15A18, 90C47, 90C90, 90C30

1 Introduction

There has been a growing awareness that finite-precision controller implementation can have a serious influence on the actual performance of a digital closed-loop control system [1],[2],[3]. Due to the FWL errors, a casual controller implementation may degrade the designed closed-loop performance or even

[0] Contact author. Tel/Fax: +44 (0)23 8059 6660/4508; Email: sqc@ecs.soton.ac.uk

destabilize the designed stable closed-loop system, if the controller implementation structure is not carefully chosen. The FWL effect has become more critical with the growing popularity of robust controller design methods which focus only on dealing with large plant uncertainty and result in controllers of much higher order and complexity than traditional classical control [2]. There are generally two types of FWL errors in the digital controller implementation. The first one is the rounding errors that occur in arithmetic operations [4],[5] and the second one is the controller parameter representation errors which have critical influence on closed-loop stability [6]–[12]. Typically, these two types of errors are investigated separately for the reason of mathematical tractability.

In general, there exist two different strategies, called the direct and indirect strategies, for constructing digital controllers that can tolerate FWL implementation errors. For the indirect strategy, the transfer function of the digital controller has been designed by some controller synthesis methods. It is well known that a transfer function can be fulfilled with different realizations and different realizations possess different degrees of robustness to FWL errors. This property can be utilized to select "optimal" realizations that optimize some FWL performance measures. Various FWL performance measures have been investigated, and these include the averaged roundoff noise gain [5], the complex stability radius measure [6], the transfer function sensitivity measure [7], the l_1 based stability measure [8], the Frobenius-norm pole sensitivity measure [9] and the 1-norm pole sensitivity measure [10],[11]. In the direct strategy, controller design involves explicitly the considerations of FWL implementation. By extending the standard H_∞ control design to include FWL controller parameter perturbations, the work of [12] developed a Riccati inequality approach, which directly obtains optimal controller realizations satisfying both the H_∞ robustness and FWL closed-loop stability requirements. Similarly, by extending the standard LQG control design to include the effects of FWL roundoff noise, the work of [4] developed a FWL-LQG controller design method. The direct strategy appears to be better than the indirect strategy, since the former does not make specific assumptions on the controller and in theory it should be a preferred approach. However, except for a few methods, such as H_∞ and LQG, it is very difficult to extend various controller design methods to this direct strategy. But this difficulty does not exist in the indirect strategy where controller synthesis and controller realization are two separate steps. Various existing controller design methods can be used to attain a transfer function or an initial realization of the controller, which can then be optimized to satisfy FWL implementation requirements.

This paper adopts the indirect strategy with the Frobenius-norm pole sensitivity measure proposed in [9]. Our motivation is as follows. The Frobenius-norm pole sensitivity measure was derived in [9] and the optimal controller realization problem was defined as the maximization of this measure over all the possible controller realizations. An analytical solution to this class of optimal realization problems was attempted in [9]. However, it was pointed out that the conditions presented in [9] are not sufficient to provide an optimal realization [13]. Consequently, the solution expression presented in [9] is in general a suboptimal solution, and numerical optimization methods have to be adopted [14] to find optimal solutions. Since these optimal FWL realization problems are highly complicated nonlinear and non-convex optimization problems, especially when the order of the controller is large, a direct numerical optimization is computationally very expensive. Moreover, chances of search being trapped at some bad local solutions increase for large-scale problems and it is impossible to tell whether a solution obtained is a global optimum or not. In this paper, these optimal FWL controller realization problems are reviewed with game theory [15],[16]. They are consequently divided into two types: optimization problems which have saddle points and optimization problems which do not have a saddle point.

For the class of optimal FWL realization problems with saddle points, this paper derives a minimax-based search algorithm for finding global optimal solutions. Our search algorithm is computationally much more efficient than usual numerical optimization for tackling this class of complicated optimization problems. Moreover, when this algorithm attains a solution, it is guaranteed to be a global optimal realization. Comments are made regarding why in practice the class of these optimization problems with saddle points is much larger than the class having no saddle point. It is shown that our proposed search algorithm is also useful in helping to solve the small class of these optimal FWL realization problems which have no saddle point. The remainder of the paper is organized as follows. Section 2 defines the optimal FWL controller realization problem considered in this study and introduces some necessary mathematical preliminaries. In Section 3, the proposed two-stage search algorithm is derived. Section 4 discusses the practical value of this algorithm. Section 5 presents several design examples, and the paper concludes at Section 6.

2 Problem definition and preliminaries

For a complex-valued matrix $\mathbf{M} = [m_{ij}]$, $\mathbf{M}^T$ is the transposed matrix of $\mathbf{M}$, $\mathbf{M}^H$ is the Hermitian adjoint matrix of $\mathbf{M}$, $\mathbf{M}^*$ is conjugate to $\mathbf{M}$,

$$\|\mathbf{M}\|_{\max} \triangleq \max_{i,j} |m_{ij}| \tag{1}$$

and the Frobenius-norm is defined as

$$\|\mathbf{M}\|_F \triangleq \left(\sum_{i,j} |m_{ij}|^2 \right)^{1/2} . \tag{2}$$

Let $\mathrm{Vec}(\cdot)$ be the column stacking operator such that Vec($\mathbf{M}$) is a vector. For a real-valued positive semi-definite matrix $\mathbf{D} \geq 0$, the matrix $\mathbf{D}^{1/2}$ satisfies $\mathbf{D}^{1/2}(\mathbf{D}^{1/2})^T = \mathbf{D}$. For two real-valued matrices $\mathbf{M} = [m_{ij}]$ and $\mathbf{N} = [n_{ij}]$ of the same dimension, denote

$$\langle \mathbf{M}, \mathbf{N} \rangle = \sum_{i,j} m_{ij} n_{ij}. \tag{3}$$

2.1 Problem definition

Consider the discrete-time closed-loop control system, consisting of a linear time-invariant plant $P(z)$ and a digital controller $C(z)$. The plant model $P(z)$ is assumed to be strictly proper with a state-space description

$$\left\{ \begin{array}{l} \mathbf{x}_P(t+1) = \mathbf{A}_P \mathbf{x}_P(t) + \mathbf{B}_P \mathbf{u}(t) \\ \mathbf{z}(t) = \mathbf{C}_P \mathbf{x}_P(t) \end{array} \right. \tag{4}$$

where $\mathbf{A}_P \in \mathcal{R}^{m \times m}$, $\mathbf{B}_P \in \mathcal{R}^{m \times l}$ and $\mathbf{C}_P \in \mathcal{R}^{q \times m}$. The digital controller $C(z)$ is described by

$$\left\{ \begin{array}{l} \mathbf{x}_C(t+1) = \mathbf{A}_C \mathbf{x}_C(t) + \mathbf{B}_C \mathbf{z}(t) \\ \mathbf{u}(t) = \mathbf{C}_C \mathbf{x}_C(t) + \mathbf{D}_C \mathbf{z}(t) \end{array} \right. \tag{5}$$

with $\mathbf{A}_C \in \mathcal{R}^{n \times n}$, $\mathbf{B}_C \in \mathcal{R}^{n \times q}$, $\mathbf{C}_C \in \mathcal{R}^{l \times n}$ and $\mathbf{D}_C \in \mathcal{R}^{l \times q}$. Denote the *realization* of $C(z)$ as

$$\mathbf{X} \triangleq \begin{bmatrix} \mathbf{D}_C & \mathbf{C}_C \\ \mathbf{B}_C & \mathbf{A}_C \end{bmatrix}. \tag{6}$$

Assume that an initial realization of $C(z)$

$$\mathbf{X}_0 \triangleq \begin{bmatrix} \mathbf{D}_C^0 & \mathbf{C}_C^0 \\ \mathbf{B}_C^0 & \mathbf{A}_C^0 \end{bmatrix} \tag{7}$$

has been given by some controller synthesis method. Then all the realizations of $C(z)$ form a set

$$\mathcal{S}_C \triangleq \left\{ \mathbf{X} : \mathbf{X} = \mathbf{X}(\mathbf{T}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-1} \end{bmatrix} \mathbf{X}_0 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \right\} \tag{8}$$

where the transformation $\mathbf{T} \in \mathcal{R}^{n \times n}$ is an arbitrary non-singular matrix, $\mathbf{0}$ and $\mathbf{I}$ denote the zero and identity matrices of appropriate dimensions, respectively. $\mathcal{S}_C$ is not a convex set, as

$$\lambda \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_1^{-1} \end{bmatrix} \mathbf{X}_0 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_1 \end{bmatrix} + (1-\lambda) \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2^{-1} \end{bmatrix} \mathbf{X}_0 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix} \tag{9}$$

may not belong to $\mathcal{S}_C$ for any non-singular $\mathbf{T}_1, \mathbf{T}_2 \in \mathcal{R}^{n \times n}$ and $0 < \lambda < 1$. The stability of the closed-loop control system depends on the eigenvalues of the closed-loop transition matrix

$$\begin{aligned} \overline{\mathbf{A}}(\mathbf{X}) = \begin{bmatrix} \mathbf{A}_P + \mathbf{B}_P \mathbf{D}_C \mathbf{C}_P & \mathbf{B}_P \mathbf{C}_C \\ \mathbf{B}_C \mathbf{C}_P & \mathbf{A}_C \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{X} \begin{bmatrix} \mathbf{C}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &\triangleq \mathbf{M}_0 + \mathbf{M}_1 \mathbf{X} \mathbf{M}_2. \end{aligned} \tag{10}$$

All the different realizations $\mathbf{X}$ in $\mathcal{S}_C$ have exactly the same set of closed-loop poles if they are implemented with infinite precision. Since the closed-loop system has been designed to be stable, all the eigenvalues $\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))$, $1 \leq k \leq m+n$, of $\overline{\mathbf{A}}(\mathbf{X})$ are within the unit disk.

When $\mathbf{X}$ is implemented with an FWL digital processor of fixed-point format, it is perturbed to $\mathbf{X} + \Delta\mathbf{X}$. Each element of $\Delta\mathbf{X}$ is bounded by $\pm\varepsilon$, that is, $\|\Delta\mathbf{X}\|_{\max} \leq \varepsilon$ where ε is the maximum representation error of the digital processor. With the perturbation $\Delta\mathbf{X}$, $\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))$ is moved to $\lambda_k(\overline{\mathbf{A}}(\mathbf{X}+\Delta\mathbf{X}))$. If an eigenvalue of $\overline{\mathbf{A}}(\mathbf{X}+\Delta\mathbf{X})$ is outside the open unit disk, the closed-loop system, designed to be stable, becomes unstable with the finite-precision implemented $\mathbf{X}$. It is therefore critical to know when the FWL error will cause closed-loop instability. This means that we would like to know the largest open "hypercube" in the perturbation space within which the closed-loop system remains stable. The size of this perturbation hypercube quantifies the FWL characteristics of $\mathbf{X}$ and is therefore a true FWL closed-loop stability measure for $\mathbf{X}$ [17].

Computing the size of this largest stable perturbation hypercube, however, is an unsolved open problem. An approximation to this true FWL closed-loop stability measure is the following Frobenius-norm pole sensitivity measure defined in [9]:

$$f(\mathbf{X}) \overset{\triangle}{=} \min_{k\in\{1,\cdots,m+n\}} \frac{1 - |\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))|}{\sqrt{(l+n)(q+n)}\left\| \frac{\partial \lambda_k(\overline{\mathbf{A}}(\mathbf{X}))}{\partial \mathbf{X}} \right\|_F}. \tag{11}$$

Rigorous discussions regarding the rationality of $f(\mathbf{X})$ as an FWL closed-loop stability measure can be found in [9],[11]. Basically, under some mild assumptions and using a first-order approximation, it can be shown that the closed-loop system remains stable if $\|\Delta\mathbf{X}\|_{\max} < f(\mathbf{X})$. It has been argued in [18] that estimates obtained from first-order perturbation theory are often more realistic than rigorous bounds obtained by other means. Thus, the larger $f(\mathbf{X})$ is, the larger an FWL error $\Delta\mathbf{X}$ that the closed-loop system can tolerate. Moreover, $f(\mathbf{X})$ is computationally tractable as is summarized in the following lemma given by [19].

Lemma 1 Let $\overline{\mathbf{A}}(\mathbf{X}) = \mathbf{M}_0 + \mathbf{M}_1\mathbf{X}\mathbf{M}_2$ given in (10) be diagonalizable. Denote $\mathbf{p}_k$ a right eigenvector of $\overline{\mathbf{A}}(\mathbf{X})$ corresponding to the eigenvalue $\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))$. The reciprocal left eigenvector $\mathbf{y}_k$ related to $\mathbf{p}_k$ is obtained from $[\mathbf{y}_1\ \mathbf{y}_2 \cdots \mathbf{y}_{m+n}] = [\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_{m+n}]^{-H}$. Then

$$\frac{\partial \lambda_k(\overline{\mathbf{A}}(\mathbf{X}))}{\partial \mathbf{X}} = \mathbf{M}_1^T \mathbf{y}_k^* \mathbf{p}_k^T \mathbf{M}_2^T, \ \ \forall k \in \{1, \cdots, m+n\}. \tag{12}$$

As different controller realizations $\mathbf{X}$ result in different values of $f(\mathbf{X})$. It is natural to search for "optimal" controller realizations that maximize the measure defined in (11). This leads to the following optimal FWL realization problem [9]:

$$\upsilon \overset{\triangle}{=} \max_{\mathbf{X}\in\mathcal{S}_C} f(\mathbf{X}). \tag{13}$$

Numerical optimization methods have been used to attain solutions of this optimal realization problem (e.g. [14]). In general, the optimization problem (13) is highly nonlinear and non-convex. Thus, numerical optimization methods do not guarantee to attain a global optimal solution and suffer from high costs, particularly for large-scale systems.

Now, let us define

$$g(\mathbf{X},k) \overset{\triangle}{=} \frac{1-|\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))|}{\sqrt{(l+n)(q+n)}\left\|\frac{\partial\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))}{\partial\mathbf{X}}\right\|_F}. \tag{14}$$

Obviously, the optimal FWL realization problem (13) can be viewed as

$$\upsilon = \max_{\mathbf{X}\in\mathcal{S}_C} \min_{k\in\{1,\cdots,m+n\}} g(\mathbf{X},k). \tag{15}$$

2.2 Saddle points and minimax theorem

This subsection introduces without proofs some properties of saddle points and the minimax theorem, which are useful in solving the optimization problem (15). The detailed discussion of this topic can be found in the standard game theory textbooks, such as [15],[16].

Definition 1 $(\mathbf{X}',k')\in\mathcal{S}_C\times\{1,\cdots,m+n\}$ is said to be a *saddle point* of $g(\mathbf{X},k)$ if

$$g(\mathbf{X},k')\le g(\mathbf{X}',k')\le g(\mathbf{X}',k),\ \ \forall\mathbf{X}\in\mathcal{S}_C,\ \forall k\in\{1,\cdots,m+n\}. \tag{16}$$

Theorem 1 If both $(\mathbf{X}',k')$ and $(\mathbf{X}'',k'')$ are saddle points of $g(\mathbf{X},k)$, then

$$g(\mathbf{X}',k') = g(\mathbf{X}'',k''). \tag{17}$$

The following theorem is the well-known Minimax Theorem in game theory.

Theorem 2 If and only if there exists at least a saddle point $(\mathbf{X}',k')$ of $g(\mathbf{X},k)$, then

$$\max_{\mathbf{X}\in\mathcal{S}_C}\min_{k\in\{1,\cdots,m+n\}} g(\mathbf{X},k) = \min_{k\in\{1,\cdots,m+n\}}\max_{\mathbf{X}\in\mathcal{S}_C} g(\mathbf{X},k) = g(\mathbf{X}',k'). \tag{18}$$

A direct corollary of Theorem 2 is stated as follows.

Corollary 1 If $g(\mathbf{X}, k)$ has no saddle point, then

$$\max_{\mathbf{X}\in\mathcal{S}_C} \min_{k\in\{1,\cdots,m+n\}} g(\mathbf{X}, k) < \min_{k\in\{1,\cdots,m+n\}} \max_{\mathbf{X}\in\mathcal{S}_C} g(\mathbf{X}, k). \tag{19}$$

Theorems 1 and 2 show that for the optimal FWL realization problem (15) which has saddle points, any saddle point of $g(\mathbf{X}, k)$ is a global optimal solution of (15). Define

$$\rho_k \stackrel{\triangle}{=} \max_{\mathbf{X}\in\mathcal{S}_C} g(\mathbf{X}, k) \tag{20}$$

for $k \in \{1, \cdots, m+n\}$ and the index

$$k' = \arg \min_{k\in\{1,\cdots,m+n\}} \rho_k. \tag{21}$$

There exist an infinite number of $\mathbf{X} \in \mathcal{S}_C$ such that $g(\mathbf{X}, k') = \rho_{k'}$. Define

$$\mathcal{X} \stackrel{\triangle}{=} \{\mathbf{X} : g(\mathbf{X}, k') = \rho_{k'}, \mathbf{X} \in \mathcal{S}_C\}. \tag{22}$$

Fig. 1 depicts a simple illustration for a case of ρ_k with $k \in \{1, 2, 3\}$. It is easily seen that in this case $\mathcal{X}$ is the segment between q_1 and q_4 on $\mathbf{X}$ axis. It can also be observed in Fig. 1 that the points between q_2 and q_3 (a subset of $\mathcal{X}$) are the realizations corresponding to saddle points. This observation accords with the following theorem, which provides a method for finding a saddle point.

Theorem 3 If and only if $\mathbf{X}' \in \mathcal{X}$ satisfies

$$g(\mathbf{X}', k) \geq \rho_{k'}, \ \forall k \in \{1, \cdots, m+n\} \setminus \{k'\}, \tag{23}$$

then $(\mathbf{X}', k')$ is a saddle point of $g(\mathbf{X}, k)$.

3 Search algorithm

A main objective of this paper is how to find a global optimal solution to the optimal FWL realization problem (15) which has saddle points. In other words, we assume that there exist saddle points for $g(\mathbf{X}, k)$ in the problem (15). What happens if the problem has no saddle point and how to deal with it will be discussed later in Section 4. Based on Theorem 3, a two-stage algorithm is developed to find a saddle point of the optimal FWL controller realization problem (15). The first stage focuses the attention on solving the optimization problem (20) for $k \in \{1, \cdots, m+n\}$, and the index k' and the closed-form expression of $\mathcal{X}$ are obtained in this stage. The second stage searches $\mathcal{X}$ for a controller realization $\mathbf{X}_{\text{opt}}$ that meets the condition $g(\mathbf{X}_{\text{opt}}, k) \geq \rho_{k'}$, $\forall k \in \{1, \cdots, m+n\} \setminus \{k'\}$. Such an $\mathbf{X}_{\text{opt}}$ is a global optimal solution to the optimal FWL controller realization problem (13). We now discuss this two-stage algorithm in detail.

3.1 Stage 1 of the algorithm

It is known easily from (8) and (10) that

$$\overline{\mathbf{A}}(\mathbf{X})=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{T}^{-1}\end{bmatrix}\overline{\mathbf{A}}(\mathbf{X}_0)\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{T}\end{bmatrix}. \tag{24}$$

This means that, $\forall\mathbf{X}\in\mathcal{S}_C$, $\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))=\lambda_k(\overline{\mathbf{A}}(\mathbf{X}_0))$. Thus, from (14), solving the maximization problem (20) is equivalent to solving the following minimization problem:

$$\eta_k\overset{\triangle}{=}\min_{\mathbf{X}\in\mathcal{S}_C}\left\|\frac{\partial\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))}{\partial\mathbf{X}}\right\|_F. \tag{25}$$

Combining Lemma 1 with the definition of $\|\cdot\|_F$, one has

$$\left\|\frac{\partial\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))}{\partial\mathbf{X}}\right\|_F=\|\mathbf{M}_1^T\mathbf{y}_k\|_F\,\|\mathbf{M}_2\mathbf{p}_k\|_F\,. \tag{26}$$

Let $\mathbf{p}_k$ and $\mathbf{y}_k$ be partitioned into

$$\mathbf{p}_k=\begin{bmatrix}\mathbf{p}_k(1)\\ \mathbf{p}_k(2)\end{bmatrix},\ \ \mathbf{y}_k=\begin{bmatrix}\mathbf{y}_k(1)\\ \mathbf{y}_k(2)\end{bmatrix},\ \ \mathbf{p}_k(1),\mathbf{y}_k(1)\in\mathcal{C}^m,\ \ \mathbf{p}_k(2),\mathbf{y}_k(2)\in\mathcal{C}^n. \tag{27}$$

Then it follows from (24) that

$$\begin{bmatrix}\mathbf{p}_k(1)\\ \mathbf{p}_k(2)\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{T}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{p}_{0k}(1)\\ \mathbf{p}_{0k}(2)\end{bmatrix},\quad\begin{bmatrix}\mathbf{y}_k(1)\\ \mathbf{y}_k(2)\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{T}^{T}\end{bmatrix}\begin{bmatrix}\mathbf{y}_{0k}(1)\\ \mathbf{y}_{0k}(2)\end{bmatrix}, \tag{28}$$

where $\left[\mathbf{p}_{0k}^T(1)\ \mathbf{p}_{0k}^T(2)\right]^T$ and $\left[\mathbf{y}_{0k}^T(1)\ \mathbf{y}_{0k}^T(2)\right]^T$ are the right and reciprocal left eigenvectors of $\overline{\mathbf{A}}(\mathbf{X}_0)$ corresponding to $\lambda_k(\overline{\mathbf{A}}(\mathbf{X}_0))$, respectively. Combining (10) and (26)–(28), we have

$$\left\|\frac{\partial\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))}{\partial\mathbf{X}}\right\|_F^2=\|\mathbf{T}^{-1}\mathbf{p}_{0k}(2)\|_F^2\|\mathbf{T}^T\mathbf{y}_{0k}(2)\|_F^2+\alpha_k^2\|\mathbf{T}^T\mathbf{y}_{0k}(2)\|_F^2+\beta_k^2\|\mathbf{T}^{-1}\mathbf{p}_{0k}(2)\|_F^2+\alpha_k^2\beta_k^2 \tag{29}$$

where the constants $\alpha_k=\|\mathbf{C}_P\mathbf{p}_{0k}(1)\|_F$ and $\beta_k=\|\mathbf{B}_P^T\mathbf{y}_{0k}(1)\|_F$. It is easy to see that, in order to attain ρ_k, we need to minimize the function

$$\xi(\mathbf{T},\alpha,\beta,\mathbf{p},\mathbf{y})\overset{\triangle}{=}\|\mathbf{T}^{-1}\mathbf{p}\|_F^2\|\mathbf{T}^T\mathbf{y}\|_F^2+\alpha^2\|\mathbf{T}^T\mathbf{y}\|_F^2+\beta^2\|\mathbf{T}^{-1}\mathbf{p}\|_F^2+\alpha^2\beta^2. \tag{30}$$

There are three different cases on minimizing $\xi(\mathbf{T},\alpha,\beta,\mathbf{p},\mathbf{y})$, depending on whether $\mathbf{p}$ and $\mathbf{y}$ are real-valued or complex-valued.

Case 1: $\mathbf{p},\mathbf{y}\in\mathcal{R}^n$ and $\mathbf{y}^T\mathbf{p}\neq 0$;

Case 2: $\mathbf{p},\mathbf{y}\in\mathcal{C}^n$ and $\det((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p}))>0$, where

$$\Upsilon(\mathbf{y})\overset{\triangle}{=}[\mathrm{Re}(\mathbf{y})\ \mathrm{Im}(\mathbf{y})] \tag{31}$$

with Re$(\mathbf{y})$ and Im$(\mathbf{y})$ denoting the real and imaginary parts of $\mathbf{y}$, respectively;

Case 3: $\mathbf{p},\mathbf{y}\in\mathcal{C}^n$ and $\det((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p}))<0$.

Let $\mathbf{e}_i$ denote the ith coordinate vector, and define

$$\mathbf{r}\triangleq\begin{cases}\mathbf{y}, & \text{for } \textit{Case 2},\\ \mathbf{y}^*, & \text{for } \textit{Case 3}.\end{cases} \tag{32}$$

The following theorem gives the results on minimizing $\xi(\mathbf{T},\alpha,\beta,\mathbf{p},\mathbf{y})$ for *Cases 2* and *3*. *Case 1* is much simpler than *Cases 2* and *3*, and the result for *Case 1* can easily be obtained in a similar way.

Theorem 4 Given positive $\alpha,\beta\in\mathcal{R}$, $\mathbf{p}$ and $\mathbf{y}$ are of *Case 2* or *3*, we have

$$\min_{\substack{\mathbf{T}\in\mathcal{R}^{n\times n}\\ \det\mathbf{T}\neq 0}}\xi(\mathbf{T},\alpha,\beta,\mathbf{p},\mathbf{y})=(|\mathbf{r}^H\mathbf{p}|+\alpha\beta)^2, \tag{33}$$

and $\xi(\mathbf{T},\alpha,\beta,\mathbf{p},\mathbf{y})$ achieves the minimum if and only if

$$\mathbf{T}=\mathbf{Q}\begin{bmatrix}\mathbf{H}^{1/2} & \mathbf{0}\\ \mathbf{F}(\mathbf{H}^{1/2})^{-T} & \boldsymbol{\Omega}\end{bmatrix}\mathbf{V} \tag{34}$$

where $\mathbf{V}\in\mathcal{R}^{n\times n}$ is an arbitrary orthogonal matrix, $\boldsymbol{\Omega}\in\mathcal{R}^{(n-2)\times(n-2)}$ is an arbitrary nonsingular matrix, the orthogonal matrix $\mathbf{Q}$ can be obtained from the QR factorization of $\Upsilon(\mathbf{r})$, that is,

$$\Upsilon(\mathbf{r})=\mathbf{Q}\begin{bmatrix}\gamma_{11} & 0 & 0 & \cdots & 0\\ \gamma_{12} & \gamma_{22} & 0 & \cdots & 0\end{bmatrix}^T, \tag{35}$$

the matrices

$$\mathbf{H}=\frac{\beta}{\alpha}\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix}^{-T}(\Upsilon(\mathbf{r}))^T\Upsilon(\mathbf{p})\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix}^{-1} \tag{36}$$

and

$$\mathbf{F}=\frac{\beta}{\alpha}\begin{bmatrix}\mathbf{e}_3^T\\ \vdots\\ \mathbf{e}_n^T\end{bmatrix}\mathbf{Q}^T\Upsilon(\mathbf{p})\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix}^{-1} \tag{37}$$

with $\theta\in[0,2\pi)$ which is solved from

$$\begin{cases}\tan\theta=\frac{a_{21}-a_{12}}{a_{11}+a_{22}}\\ a_{11}\cos\theta-a_{12}\sin\theta>0\end{cases} \tag{38}$$

and

$$\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\triangleq(\Upsilon(\mathbf{r}))^T\Upsilon(\mathbf{p}). \tag{39}$$

Proof: See Appendix.

Using Theorem 4, the single-pole peak ρ_k for $k \in \{1, \cdots, m+n\}$ can be computed. For example, when $\mathbf{p}_{0k}(2), \mathbf{y}_{0k}(2) \in \mathcal{C}^n$ and $\det((\Upsilon(\mathbf{y}_{0k}(2)))^T \Upsilon(\mathbf{p}_{0k}(2))) > 0$, we have

$$\rho_k = \frac{1 - |\lambda_k(\overline{\mathbf{A}}(\mathbf{X}_0))|}{\sqrt{(l+n)(q+n)}(|\mathbf{y}_{0k}^H(2)\mathbf{p}_{0k}(2)| + \|\mathbf{C}_P\mathbf{p}_{0k}(1)\|_F\|\mathbf{B}_P^T\mathbf{y}_{0k}(1)\|_F)}. \tag{40}$$

Thus, the index k' is readily given from $\rho_{k'} = \min_{k \in \{1,\cdots,m+n\}} \rho_k$. In addition, Theorem 4 with (34)–(39) provides the closed-form transformation set:

$$\mathcal{T} \triangleq \{\mathbf{T} : g(\mathbf{X}(\mathbf{T}), k') = \rho_{k'}, \mathbf{T} \in \mathcal{R}^{n\times n}, \det \mathbf{T} \neq 0\}. \tag{41}$$

Since $\mathbf{X}$ depends on $\mathbf{T}$ as is defined in (8), the realization set $\mathcal{X}$ given in (22) is defined on the transformation set $\mathcal{T}$ as:

$$\mathcal{X} = \left\{\mathbf{X} : \mathbf{X} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-1} \end{bmatrix} \mathbf{X}_0 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix}, \mathbf{T} \in \mathcal{T}\right\}. \tag{42}$$

3.2 Stage 2 of the algorithm

This stage searches in $\mathcal{T}$ for an optimal transformation $\mathbf{T}_{\text{opt}}$ that satisfies $g(\mathbf{X}(\mathbf{T}_{\text{opt}}), k) \geq \rho_{k'}, \forall k \in \{1, \cdots, m+n\} \setminus \{k'\}$. According to Theorem 3 the corresponding realization $\mathbf{X}_{\text{opt}} = \mathbf{X}(\mathbf{T}_{\text{opt}})$ is a global optimal solution for the optimal realization problem (13). Without any loss of generality, we will assume that $\mathbf{p}_{k'}$ and $\mathbf{y}_{k'}$ is of *Case 2*. From Theorem 4, the transformation set (41) is specified by

$$\mathcal{T} = \left\{\mathbf{T} : \mathbf{T} = \mathbf{Q} \begin{bmatrix} \mathbf{H}^{1/2} & \mathbf{0} \\ \mathbf{F}(\mathbf{H}^{1/2})^{-T} & \mathbf{\Omega} \end{bmatrix} \mathbf{V}\right\} \tag{43}$$

where $\mathbf{Q}$, $\mathbf{H}$ and $\mathbf{F}$ are determined in Theorem 4 by setting $\alpha = \|\mathbf{C}_P\mathbf{p}_{0k'}(1)\|_F$, $\beta = \|\mathbf{B}_P^T\mathbf{y}_{0k'}(1)\|_F$, $\mathbf{p} = \mathbf{p}_{0k'}(2)$ and $\mathbf{r} = \mathbf{y} = \mathbf{y}_{0k'}(2)$, $\mathbf{\Omega} \in \mathcal{R}^{(n-2)\times(n-2)}$ is an arbitrary nonsingular matrix and $\mathbf{V} \in \mathcal{R}^{n\times n}$ is an arbitrary orthogonal matrix. From (14), (29) and the definition of $\|\cdot\|_F$, it can be seen that $g(\mathbf{X}(\mathbf{T}), k) = g(\mathbf{X}(\mathbf{TV}), k)$ for any orthogonal $\mathbf{V} \in \mathcal{R}^{n\times n}$ and nonsingular $\mathbf{T} \in \mathcal{R}^{n\times n}$. This means that $\mathbf{V}$ plays no role in computing the value of $g(\mathbf{X}, k)$ and hence we simply set $\mathbf{V} = \mathbf{I}$. Thus we only explore those

$$\mathbf{T} = \mathbf{T}(\mathbf{\Omega}) = \mathbf{Q} \begin{bmatrix} \mathbf{H}^{1/2} & \mathbf{0} \\ \mathbf{F}(\mathbf{H}^{1/2})^{-T} & \mathbf{\Omega} \end{bmatrix}, \tag{44}$$

and the objective becomes to search for a nonsingular $\mathbf{\Omega}_{\text{opt}} \in \mathcal{R}^{(n-2)\times(n-2)}$ such that $g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega}_{\text{opt}})), k) \geq \rho_{k'}, \forall k \in \{1, \cdots, m+n\} \setminus \{k'\}$. The detailed search procedure is as follows.

Initialization: Arbitrarily select a nonsingular $\mathbf{\Omega} \in \mathcal{R}^{(n-2)\times(n-2)}$ to obtain an initial point $\mathbf{X}(\mathbf{T}(\mathbf{\Omega}))$, let N be a large enough integer and τ a small positive number, and set $N_t = 1$.

Step 1: Find out

$$e = \arg \min_{k \in \{1, \cdots, m+n\}} g(\mathbf{X}, k).$$

If $g(\mathbf{X}, e) = \rho_{k'}$, which means that (23) holds, then $\mathbf{\Omega}_{\text{opt}} = \mathbf{\Omega}$ and terminate the routine. If $g(\mathbf{X}, e) > \rho_{k'}$ but $N_t \geq N$, which means that no saddle point is found after a large number of iterations, then the routine is also terminated for practical consideration.

Step 2: $\mathbf{\Omega} = \mathbf{\Omega} + \tau \frac{\partial g(\mathbf{X}, e)}{\partial \mathbf{\Omega}} \| \frac{\partial g(\mathbf{X}, e)}{\partial \mathbf{\Omega}} \|_F^{-1}$, $N_t = N_t + 1$, and go to *Step 1*.

For calculating $\frac{\partial g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega})), e)}{\partial \mathbf{\Omega}}$, let $\mathbf{e}_i$ denote the ith coordinate vector. The following well-known fact is useful: given any element y_{ij} in a nonsingular $\mathbf{Y} \in \mathcal{R}^{n \times n}$ with $i \in \{1, \cdots, n\}$ and $j \in \{1, \cdots, n\}$,

$$\frac{\partial \mathbf{Y}}{\partial y_{ij}} = \mathbf{e}_i \mathbf{e}_j^T \quad \text{and} \quad \frac{\partial \mathbf{Y}^{-1}}{\partial y_{ij}} = -\mathbf{Y}^{-1} \mathbf{e}_i \mathbf{e}_j^T \mathbf{Y}^{-1}. \tag{45}$$

From (10), (14), (28) and Lemma 1, we know that

$$g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega})), e) = \frac{(1 - |\lambda_e|)/\sqrt{(l+n)(q+n)}}{\left\| \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^T(\mathbf{\Omega}) \end{bmatrix} \mathbf{M}_1^T \mathbf{y}_{0e}^* \mathbf{p}_{0e}^T \mathbf{M}_2^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-T}(\mathbf{\Omega}) \end{bmatrix} \right\|_F}. \tag{46}$$

From (44), we have

$$\left\| \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^T(\mathbf{\Omega}) \end{bmatrix} \mathbf{M}_1^T \mathbf{y}_{0e}^* \mathbf{p}_{0e}^T \mathbf{M}_2^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-T}(\mathbf{\Omega}) \end{bmatrix} \right\|_F = \left\| \mathbf{U}_1^T \mathbf{\Phi}_e \mathbf{U}_2^{-T} \right\|_F \tag{47}$$

where $\mathbf{U}_1$, $\mathbf{U}_2$ and $\mathbf{\Phi}_e$ are given respectively by ($\mathbf{I}$ in $\mathbf{U}_1$ and $\mathbf{U}_2$ have different dimensions.):

$$\mathbf{U}_1 = \left[\begin{array}{c|cc} \mathbf{I} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & \mathbf{H}^{1/2} & \mathbf{0} \\ & \mathbf{F}(\mathbf{H}^{1/2})^{-T} & \mathbf{\Omega} \end{array} \right], \tag{48}$$

$$\mathbf{U}_2 = \left[\begin{array}{c|cc} \mathbf{I} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & \mathbf{H}^{1/2} & \mathbf{0} \\ & \mathbf{F}(\mathbf{H}^{1/2})^{-T} & \mathbf{\Omega} \end{array} \right], \tag{49}$$

$$\mathbf{\Phi}_e = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^T \end{bmatrix} \mathbf{M}_1^T \mathbf{y}_{0e}^* \mathbf{p}_{0e}^T \mathbf{M}_2^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{-T} \end{bmatrix}. \tag{50}$$

For any element ψ_{ts} in $\mathbf{\Psi}_e = \mathbf{U}_1^T \mathbf{\Phi}_e \mathbf{U}_2^{-T}$, where $t \in \{1, \cdots, l+n\}$ and $s \in \{1, \cdots, q+n\}$, and any ω_{ij} in $\mathbf{\Omega}$, where $i \in \{1, \cdots, n-2\}$ and $j \in \{1, \cdots, n-2\}$,

$$\begin{aligned} \frac{\partial \psi_{ts}}{\partial \omega_{ij}} &= \mathbf{e}_t^T \frac{\partial \mathbf{U}_1^T}{\partial \omega_{ij}} \mathbf{\Phi}_e \mathbf{U}_2^{-T} \mathbf{e}_s + \mathbf{e}_t^T \mathbf{U}_1^T \mathbf{\Phi}_e \frac{\partial \mathbf{U}_2^{-T}}{\partial \omega_{ij}} \mathbf{e}_s \\ &= \mathbf{e}_t^T \mathbf{e}_{l+2+j} \mathbf{e}_{l+2+i}^T \mathbf{\Phi}_e \mathbf{U}_2^{-T} \mathbf{e}_s - \mathbf{e}_t^T \mathbf{U}_1^T \mathbf{\Phi}_e \mathbf{U}_2^{-T} \mathbf{e}_{q+2+j} \mathbf{e}_{q+2+i}^T \mathbf{U}_2^{-T} \mathbf{e}_s \\ &= \mathbf{e}_t^T \mathbf{e}_{l+2+j} \mathbf{e}_{l+2+i}^T \mathbf{\Phi}_e \mathbf{U}_2^{-T} \mathbf{e}_s - \mathbf{e}_t^T \mathbf{\Psi}_e \mathbf{e}_{q+2+j} \mathbf{e}_{q+2+i}^T \mathbf{U}_2^{-T} \mathbf{e}_s. \end{aligned} \tag{51}$$

That is,

$$\begin{aligned}\frac{\partial \psi_{ts}}{\partial \mathbf{\Omega}} &= \begin{bmatrix} \mathbf{e}_t^T & & \\ & \ddots & \\ & & \mathbf{e}_t^T \end{bmatrix} \left(\begin{bmatrix} \mathbf{e}_{l+3}\mathbf{e}_{l+3}^T \mathbf{\Phi}_e & \cdots & \mathbf{e}_{l+n}\mathbf{e}_{l+3}^T \mathbf{\Phi}_e \\ \vdots & \cdots & \vdots \\ \mathbf{e}_{l+3}\mathbf{e}_{l+n}^T \mathbf{\Phi}_e & \cdots & \mathbf{e}_{l+n}\mathbf{e}_{l+n}^T \mathbf{\Phi}_e \end{bmatrix} \right. \\ & \left. - \begin{bmatrix} \mathbf{\Psi}_e \mathbf{e}_{q+3}\mathbf{e}_{q+3}^T & \cdots & \mathbf{\Psi}_e \mathbf{e}_{q+n}\mathbf{e}_{q+3}^T \\ \vdots & \cdots & \vdots \\ \mathbf{\Psi}_e \mathbf{e}_{q+3}\mathbf{e}_{q+n}^T & \cdots & \mathbf{\Psi}_e \mathbf{e}_{q+n}\mathbf{e}_{q+n}^T \end{bmatrix} \right) \begin{bmatrix} \mathbf{U}_2^{-T}\mathbf{e}_s & & \\ & \ddots & \\ & & \mathbf{U}_2^{-T}\mathbf{e}_s \end{bmatrix}. \end{aligned} \tag{52}$$

Since

$$g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega})), e) = \frac{(1 - |\lambda_e|)/\sqrt{(l+n)(q+n)}}{\sqrt{\sum_{t=1}^{l+n} \sum_{s=1}^{q+n} \psi_{ts}^* \psi_{ts}}}, \tag{53}$$

we can readily calculate

$$\frac{\partial g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega})), e)}{\partial \mathbf{\Omega}} = -\frac{1 - |\lambda_e|}{\sqrt{(l+n)(q+n)}\, \|\mathbf{\Psi}_e\|_F^3} \mathrm{Re}\left[\sum_{t=1}^{l+n} \sum_{s=1}^{q+n} \psi_{ts}^* \frac{\partial \psi_{ts}}{\partial \mathbf{\Omega}} \right]. \tag{54}$$

Comment 1: In a way, the above search procedure solves

$$\min_{\mathbf{\Omega} \in \mathcal{R}^{(n-2)\times(n-2)}} \max_{k \in \{1,\cdots,m+n\}} \left(-g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega})), k) \right). \tag{55}$$

The function $h(\mathbf{\Omega}) = \max\limits_{k \in \{1,\cdots,m+n\}} \left(-g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega})), k) \right)$ to be minimized has corners where differentiability fails, although $g(\mathbf{X}(\mathbf{T}(\mathbf{\Omega})), k)$ is differentiable for any $k \in \{1, \cdots, m+n\}$. In fact, the problem (55) is a classical optimization problem which requires nondifferentiable optimization approaches, such as subgradient methods [22]. Subdifferentiation of h at $\mathbf{\Omega}$ is defined as

$$\aleph h(\mathbf{\Omega}) = Conv \left\{ \mathbf{J} \in \mathcal{R}^{(n-2)\times(n-2)} \;\middle|\; \begin{array}{c} \mathbf{J} = \lim \frac{\partial h(\mathbf{\Omega}_i)}{\partial \mathbf{\Omega}_i}, \mathbf{\Omega}_i \to \mathbf{\Omega} \\ \frac{\partial h(\mathbf{\Omega}_i)}{\partial \mathbf{\Omega}_i} \text{ exists, } \frac{\partial h(\mathbf{\Omega}_i)}{\partial \mathbf{\Omega}_i} \text{ converges} \end{array} \right\} \tag{56}$$

where $Conv$ denotes the convex hull. The elements of $\aleph h(\mathbf{\Omega})$ are called subgradients. Denote the directional derivative

$$h^\circ(\mathbf{\Omega}, \mathbf{\Gamma}) = \lim_{\substack{t \to 0 \\ t > 0}} \frac{h(\mathbf{\Omega} + t\mathbf{\Gamma}) - h(\mathbf{\Omega})}{t} \tag{57}$$

in every direction $\mathbf{\Gamma} \in \mathcal{R}^{(n-2)\times(n-2)}$. A relationship between subgradients and the directional derivative is given in [22], which is re-stated in the following lemma.

Lemma 2 $h^\circ(\mathbf{\Omega}, \mathbf{\Gamma}) = \max\limits_{\mathbf{J} \in \aleph h(\mathbf{\Omega})} \langle \mathbf{J}, \mathbf{\Gamma} \rangle$.

It is easily seen that $-\frac{\partial g(\mathbf{X}, e)}{\partial \mathbf{\Omega}}$ is a subgradient of $h(\mathbf{\Omega})$ and hence our method is a subgradient algorithm. Since $h(\mathbf{\Omega})$ is differentiable almost everywhere when $\mathbf{\Omega}$ is not a local optimal point, there must exist a neighborhood $\mathcal{B}_r = \left\{ \mathbf{\Theta} \in \mathcal{R}^{(n-2)\times(n-2)} \mid \|\mathbf{\Theta} - \mathbf{\Omega}\|_F < r \right\}$ such that

$$h^\circ(\mathbf{\Omega}, \mathbf{\Xi} - \mathbf{\Omega}) < 0 \tag{58}$$

and

$$\boldsymbol{\Xi} = \min_{\boldsymbol{\Theta} \in \mathcal{B}_r} h(\boldsymbol{\Omega}). \tag{59}$$

Then we have the following theorem.

Theorem 5 There exists $\tau_m > 0$ such that for *Step 2* of the above search algorithm

$$\left\| \boldsymbol{\Omega} + \tau \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \left\| \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \right\|_F^{-1} - \boldsymbol{\Xi} \right\|_F < \|\boldsymbol{\Omega} - \boldsymbol{\Xi}\|_F \tag{60}$$

for all $\tau \in (0, \tau_m)$.

Proof: By the definition of Frobenius-norm,

$$\begin{aligned} & \left\| \boldsymbol{\Xi} - \boldsymbol{\Omega} - \tau \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \left\| \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \right\|_F^{-1} \right\|_F^2 \\ = \ & \|\boldsymbol{\Xi} - \boldsymbol{\Omega}\|_F^2 + 2\tau \left\langle -\frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \left\| \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \right\|_F^{-1}, \boldsymbol{\Xi} - \boldsymbol{\Omega} \right\rangle + \tau^2. \end{aligned} \tag{61}$$

Since $-\frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}}$ is a subgradient, from Lemma 2 and (58), one has

$$\left\langle -\frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \left\| \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \right\|_F^{-1}, \boldsymbol{\Xi} - \boldsymbol{\Omega} \right\rangle \leq h^\circ(\boldsymbol{\Omega}, \boldsymbol{\Xi} - \boldsymbol{\Omega}) \left\| \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \right\|_F^{-1} < 0. \tag{62}$$

Thus, for $0 < \tau < \tau_m = 2 \left\langle \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \left\| \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \right\|_F^{-1}, \boldsymbol{\Xi} - \boldsymbol{\Omega} \right\rangle$,

$$2\tau \left\langle -\frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \left\| \frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}} \right\|_F^{-1}, \boldsymbol{\Xi} - \boldsymbol{\Omega} \right\rangle + \tau^2 < 0. \tag{63}$$

This together with (61) proves the assertion.

The above result shows that, for sufficiently small $\tau > 0$, $\frac{\partial g(\mathbf{X}, e)}{\partial \boldsymbol{\Omega}}$ is a good direction along which to update $\boldsymbol{\Omega}$ so that it becomes closer to $\boldsymbol{\Xi}$, although occasionally the updated $h(\boldsymbol{\Omega})$ may be worse. Therefore, $h(\boldsymbol{\Omega})$ will be improved significantly after some iterations. Our numerical examples listed in Section 5 show that this simplest subgradient optimization algorithm behaves satisfactorily in practice, provided that τ is chosen appropriately. Of course, if this simplest subgradient algorithm fails in some cases, various enhanced subgradient algorithms [22]-[24] can be adopted to tackle the problem.

Comment 2: The termination at $N_t \geq N$ does not mean that the problem (55) has no saddle point. As $h(\boldsymbol{\Omega})$ may be nonconvex, our subgradient search sequence may possibly oscillate around a local optimum which is worst than $\rho_{k'}$. Regardless whether the problem (55) has saddle points or not, when the routine does not find a saddle point, we can further increase the value of $\min\limits_{k \in \{1, \cdots, m+n\}} g(\mathbf{X}, k)$ by a direct numerical optimization. This is further discussed in the next section.

4 Discussions

The function $g(\mathbf{X}, k)$ having saddle points is the main assumption in this paper. Here we explain heuristically that for many practical control systems this assumption is valid. Firstly, from Section 3.1, it is known that k', $\rho_{k'}$ and $\mathcal{X}$ exist regardless whether $g(\mathbf{X}, k)$ has saddle points or not. Secondly, Theorem 3 shows that if and only if there exist $\mathbf{T} \in \mathcal{T}$ satisfying

$$g(\mathbf{X}(\mathbf{T}), k) \geq \rho_{k'} \ \ \forall k \in \{1, \cdots, m+n\} \setminus \{k'\}, \tag{64}$$

the saddle points of $g(\mathbf{X}, k)$ exist. From the definition of $g(\mathbf{X}, k)$ in (14), $g(\mathbf{X}, k)$ is proportional to the single-pole stability margin $1 - |\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))|$, which is a fixed value, and inverse proportional to its eigenvalue sensitivity, which depends on $\mathbf{X}$. For practical digital closed-loop control systems, there exist usually only a few dominant poles which are near the unit circle and/or have relatively high eigenvalue sensitivities, compared with all the other non-dominant poles. For this reason, the index k' defined in (21) is usually the index of a dominant pole, and the values of $g(\mathbf{X}, k)$ for those non-dominant poles at $\mathbf{X}(\mathbf{T})$ are larger than $\rho_{k'}$ for most $\mathbf{T} \in \mathcal{T}$. Therefore, to satisfy condition (64), one needs only to consider the few dominant poles whose indices are not k'. It should be observed that $\mathbf{T}$ in $\mathcal{T}$ has a fairly large degree of freedom. Specifically, the free parameter $\mathbf{\Omega}$ in (44) can be any nonsingular matrix in $\mathcal{R}^{(n-2)\times(n-2)}$. This large degree of freedom together with the fact that there are typically just a few dominant poles to consider means that most likely there exist $\mathbf{T} \in \mathcal{T}$ satisfying (64). Thus $g(\mathbf{X}, k)$ has saddle points for many practical problems. We conjecture without a rigorous proof that the class of optimal FWL controller realization problems (15) which have saddle points is much larger than the class having no saddle point. Empirically, we have tested a total of six FWL controller design examples that we can found in the FWL controller design literature. Only one example, which is given in [14], was shown to possibly have no saddle point.

The routine presented in Section 3.2 is computationally much more attractive than a direct numerical optimization of (13). Actually, all the needed is to find a $\mathbf{T} \in \mathcal{T}$ such that $g(\mathbf{X}(\mathbf{T}), k) \geq \rho_{k'}$ for $k \in \{1, \cdots, m+n\} \setminus \{k'\}$, rather than to directly maximize $f(\mathbf{X}(\mathbf{T}))$ over $\mathcal{R}^{n\times n}$ (and of course $\det \mathbf{T} \neq 0$). The former objective can be attained often easily even for large-scale problems. In addition, the number of saddle points is infinite when $g(\mathbf{X}, k)$ has saddle points. Hence our algorithm can find global optimal solutions for most practical problems which have saddle points even though we do not strictly prove the convergence of the subgradient routine. An additional advantage of the algorithm presented, which is particularly important in practical applications, is that when the algorithm attains a solution the user knows for sure that it is a global optimal solution to the optimal realization problem

(13). This should be compared with direct numerical optimization of (13) where even when it converges to a solution, there is no way to tell whether the solution is a global optimal one or not.

It should be pointed out that our algorithm, presented for the problems having saddle points, is also useful in helping to solve those optimal FWL realization problems which do not have saddle point. Actually, the algorithm given in Section 3 can be executed even for the problems which do not have saddle point. Using the results of Section 3.1, k' and $\rho_{k'}$ can be computed, and $\mathcal{X}$ is obtained in closed-form. Corollary 1 tells us that $\rho_{k'}$ is an upper bound of the optimal value of the realization problem having no saddle point. After executing N iterations of the routine given in Section 3.2, the resulting realization $\mathbf{X}_t$ obviously does not satisfy (64). But through these N iterations, $\min\limits_{k\in\{1,\cdots,m+n\}} g(\mathbf{X},k)$ has been increased to as close to $\rho_{k'}$ as possible under $\mathbf{X}\in\mathcal{X}$. Therefore the value of $f(\mathbf{X}_t)$ is not much less than $\rho_{k'}$. This provides a small region $[f(\mathbf{X}_t),\rho_{k'}]$ within which the optimal value of the FWL controller realization problem lies. Of course, this also provide an excellent guess from which a direct numerical optimization approach can be used to find a (local) optimal solution for those optimization problems having no saddle point.

Obviously, the same idea is equally applicable to the problems whose saddle points are not found after N iterations of the search routine. In fact, when the subgradient routine is terminated after N iterations but the condition (64) is not met, one cannot answer the question of whether the problem (55) has any saddle point or not. However, one knows the small region within which the global optimal value lies and the solution obtained after N iterations provides an excellent initial guess for a direct numerical optimization.

5 Design examples

Six examples are used to illustrate the effectiveness of the proposed design algorithm.

Example 1. The example in [25] is discretized with a sampling frequency of 5 Hz to obtain the discrete-time plant model

$$\mathbf{A}_P=\begin{bmatrix} 3.2439e-1 & -4.5451e+0 & -4.0535e+0 & -2.7003e-3 & 0\\ 1.4518e-1 & 4.9477e-1 & -4.6945e-1 & -3.1274e-4 & 0\\ 1.6814e-2 & 1.6491e-1 & 9.6681e-1 & -2.2114e-5 & 0\\ 1.1889e-3 & 1.8209e-2 & 1.9829e-1 & 1.0000e+0 & 0\\ 6.1301e-5 & 1.2609e-3 & 1.9930e-2 & 2.0000e-1 & 1.0000e+0 \end{bmatrix},$$

$$\mathbf{B}_P=\begin{bmatrix}1.4518e-1 & 1.6814e-2 & 1.1889e-3 & 6.1301e-5 & 2.4979e-6\end{bmatrix}^T,$$

$$\mathbf{C}_P=\begin{bmatrix}0 & 0 & 1.6188e+0 & -1.5750e-1 & -4.3943e+1\end{bmatrix};$$

and the initially designed digital controller

$$\mathbf{A}_C^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -4.7086e-1 \\ 1 & 0 & 0 & 0 & 0 & 2.6885e+0 \\ 0 & 1 & 0 & 0 & 0 & -6.6649e+0 \\ 0 & 0 & 1 & 0 & 0 & 9.4410e+0 \\ 0 & 0 & 0 & 1 & 0 & -8.2537e+0 \\ 0 & 0 & 0 & 0 & 1 & 4.2600e+0 \end{bmatrix}, \quad \mathbf{B}_C^0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{D}_C^0 = [4.6000e-2],$$

$$\mathbf{C}_C^0 = [\, 2.1187e-1 \quad 9.4498e-2 \quad 1.0887e-2 \quad -4.4171e-2 \quad -7.6000e-2 \quad -8.8562e-2 \,].$$

The corresponding closed-loop transition matrix $\overline{\mathbf{A}}(\mathbf{X}_0)$ is then formed using (10), from which the eigenvalues and the eigenvectors of the ideal closed-loop system are computed. These 11 eigenvalues and their absolute values are

$$\begin{bmatrix} \lambda_{1,2} \\ \lambda_{3,4} \\ \lambda_{5,6} \\ \lambda_{7,8} \\ \lambda_{9,10} \\ \lambda_{11} \end{bmatrix} = \begin{bmatrix} 4.8368e-1 \pm j8.5569e-1 \\ 4.8135e-1 \pm j8.5363e-1 \\ 9.9993e-1 \pm j3.7887e-4 \\ 8.3967e-1 \pm j1.6514e-1 \\ 8.0884e-1 \pm j1.2026e-1 \\ 8.1905e-1 \end{bmatrix}, \quad \begin{bmatrix} |\lambda_{1,2}| \\ |\lambda_{3,4}| \\ |\lambda_{5,6}| \\ |\lambda_{7,8}| \\ |\lambda_{9,10}| \\ |\lambda_{11}| \end{bmatrix} \begin{bmatrix} 9.8293e-1 \\ 9.7999e-1 \\ 9.9993e-1 \\ 8.5575e-1 \\ 8.1774e-1 \\ 8.1905e-1 \end{bmatrix}.$$

This closed-loop system has five pairs of conjugate complex-valued eigenvalues and one real-valued eigenvalue. Using the method developed in section 3.1, the single-pole peak for each eigenvalue is computed, and they are

$$\begin{bmatrix} \rho_{1,2} \\ \rho_{3,4} \\ \rho_{5,6} \\ \rho_{7,8} \\ \rho_{9,10} \\ \rho_{11} \end{bmatrix} = \begin{bmatrix} 2.5072e-3 \\ 2.1295e-3 \\ 6.7344e-6 \\ 2.8586e-3 \\ 3.0832e-3 \\ 4.3181e-3 \end{bmatrix}.$$

Obviously, the minimum value of all the ρ_ks is ρ_5 (or ρ_6). Therefore, $k' = 5$ and the corresponding matrices **Q**, **H** and **F** in the set (44) are given by

$$\mathbf{Q} = \begin{bmatrix} -6.6011e-2 & -8.4915e-2 & -4.3670e-1 & -5.1206e-1 & -5.2972e-1 & -5.0490e-1 \\ -3.7006e-1 & -4.3518e-1 & -4.9156e-1 & -2.2314e-1 & 1.7033e-1 & 5.9434e-1 \\ -5.0566e-1 & -3.8025e-1 & 7.1063e-1 & -2.5387e-1 & -1.6560e-1 & -5.3367e-2 \\ -5.2127e-1 & -8.6900e-2 & -2.2452e-1 & 7.4814e-1 & -2.4759e-1 & -2.2204e-1 \\ -4.5786e-1 & 3.1775e-1 & -1.0190e-1 & -2.0850e-1 & 6.8322e-1 & -4.1079e-1 \\ -3.4878e-1 & 7.4183e-1 & 4.3249e-2 & -1.4270e-1 & -3.6725e-1 & 4.1345e-1 \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} 2.6322e+0 & -3.9258e+2 \\ -3.9258e+2 & 6.9856e+6 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 4.8432e+4 & -8.8104e+8 \\ -5.2079e+4 & 9.4682e+8 \\ 2.4998e+4 & -4.5374e+8 \\ -2.4644e+4 & 4.4816e+8 \end{bmatrix}.$$

Set $\tau = 0.1$ and the initial $\mathbf{\Omega} = \mathbf{I}$. Fig. 2 illustrates the changes of $g(\mathbf{X}, k)$ in each iteration. From Fig. 2, it can be seen that at the 37th iteration, the optimal controller realization is found, since at this iteration the conditions of Theorem 3 are met and the algorithm terminates. The resulting matrix $\mathbf{\Omega}_{\rm opt}$ is

$$\mathbf{\Omega}_{\rm opt} = \begin{bmatrix} 2.3184e+0 & -1.6411e+0 & 5.5681e-1 & -7.6953e-1 \\ -1.6411e+0 & 2.4047e+0 & -8.2094e-1 & 7.0079e-1 \\ 5.5680e-1 & -8.2095e-1 & 1.2097e+0 & -3.7643e-1 \\ -7.6954e-1 & 7.0078e-1 & -3.7643e-1 & 1.3454e+0 \end{bmatrix}$$

and the corresponding global optimal transformation matrix is

$$\mathbf{T}_{\rm opt} = \begin{bmatrix} -6.9470e+1 & -3.2765e+4 & -7.8507e-2 & -4.3363e-1 & -2.7354e-1 & -5.0267e-1 \\ 3.0977e+2 & 1.5431e+5 & -1.1360e+0 & 5.4680e-1 & -1.0820e-1 & 9.5739e-1 \\ -6.0267e+2 & -3.0945e+5 & 2.0130e+0 & -1.6781e+0 & 4.2386e-1 & -7.3423e-1 \\ 6.5537e+2 & 3.4747e+5 & -1.7153e+0 & 2.2151e+0 & -9.5513e-1 & 4.9153e-1 \\ -4.1530e+2 & -2.2683e+5 & 8.0247e-1 & -1.1829e+0 & 1.0956e+0 & -8.7755e-1 \\ 1.1931e+2 & 6.9580e+4 & -1.8821e-1 & 1.7712e-1 & -4.5868e-1 & 5.6121e-1 \end{bmatrix}.$$

Example 2. The second example is taken from [14]. In this example, $m = 4$, $n = 10$, $l = 2$, $q = 2$ and hence it is a closed-loop system of order 14. The initial controller realization $\mathbf{X}_0$ of $C(z)$ has been given [20]. The corresponding closed-loop transition matrix $\overline{\mathbf{A}}(\mathbf{X}_0)$ is formed using (10), from which the eigenvalues and the eigenvectors of the ideal closed-loop system are computed. This closed-loop system has six pairs of conjugate complex-valued eigenvalues and two real-valued eigenvalues given by

$$\begin{bmatrix} \lambda_{1,2} \\ \lambda_{3,4} \\ \lambda_5 \\ \lambda_{6,7} \\ \lambda_{8,9} \\ \lambda_{10,11} \\ \lambda_{12} \\ \lambda_{13,14} \end{bmatrix} = \begin{bmatrix} -8.4482e-1 \pm j7.8204e-2 \\ -3.7557e-1 \pm j3.3602e-1 \\ 2.1624e-1 \\ 7.1567e-1 \pm j9.6631e-3 \\ 9.2895e-1 \pm j1.2923e-1 \\ 9.8506e-1 \pm j7.5831e-2 \\ 8.3133e-1 \\ 8.8267e-1 \pm j3.7235e-2 \end{bmatrix}.$$

Using the method developed in Section 3.1, the single-pole peaks for every eigenvalues are computed, and they are

$$\begin{bmatrix} \rho_{1,2} \\ \rho_{3,4} \\ \rho_5 \\ \rho_{6,7} \\ \rho_{8,9} \\ \rho_{10,11} \\ \rho_{12} \\ \rho_{13,14} \end{bmatrix} = \begin{bmatrix} 8.3118e-3 \\ 4.0562e-2 \\ 6.2954e-2 \\ 8.0984e-3 \\ 3.7768e-3 \\ 5.4246e-3 \\ 5.8442e-3 \\ 8.0773e-3 \end{bmatrix}.$$

Obviously, the minimum value of all the ρ_ks is ρ_8 (or ρ_9). Therefore, $k' = 8$ and the corresponding matrices $\mathbf{Q}$, $\mathbf{H}$ and $\mathbf{F}$ in (44) are computed according to Theorem 4. With $\mathbf{T}$ in (44), the second stage of our algorithm can be executed. Fig. 3 illustrates the changes of $g(\mathbf{X}, k)$ in each iteration of the second stage. From Fig. 3, it can be seen that after $N = 50000$ iteration, we still cannot find a realization satisfying (64). This suggests that this example most likely has no saddle point (although one cannot be sure). So we terminate the algorithm at 50000 iteration and obtain a realization $\mathbf{X}_t$. Although this $\mathbf{X}_t$ is not an optimal realization, it is much better than $\mathbf{X}_0$, since $f(\mathbf{X}_t) = 2.1539e - 3$ while $f(\mathbf{X}_0) = 1.1734e - 4$. In particular, we notice that $\mathbf{X}_t$ is also better than the "optimal" realization given in [14], which was found by a direct numerical optimization search using the simulated annealing algorithm and has a FWL measure value of $1.5844e - 3$ [14]. At this stage, we are sure that the optimal solution given in [14] is not a global optimal one at all. Using the realization $\mathbf{X}_t$ obtained by our search algorithm as the initial point, we then use a direct numerical optimization method to solve for the optimization problem (13) and obtain a new optimal realization whose FWL measure value is $3.1929e - 3$. This optimal value is more than double of that given in [14]. Obviously, we cannot tell whether this new optimal realization is a global optimal one or not. However, we know that the optimal value of the FWL realization problem for this example lies in the range of $[3.1929e - 3,\ 3.7768e - 3]$. For this example, no other design has found a controller realization whose FWL closed-loop stability measure $f(\mathbf{X})$ is larger than $3e - 3$. Our algorithm is the first one to achieve a $f(\mathbf{X}) > 3e - 3$.

The saddle points (or the global optimal solutions) of the following four examples are found successfully by our proposed method.

Example 3. This example is a fluid power speed controller given in [8], where $m = 4$, $n = 4$, $l = 1$ and $q = 1$.

Example 4. This example is a discretized version of an H_∞ robust controller given in [26] with a sampling frequency of 250 Hz, where $m = 2$, $n = 3$, $l = 1$ and $q = 1$.

Example 5. This example is taken from [6], where $m = 3$, $n = 2$, $l = 1$ and $q = 1$.

Example 6. This example is a steel rolling mill PID controller given in [8], where $m = 3$, $n = 2$, $l = 1$ and $q = 1$. "

As is mentioned previously, the realizations of $C(z)$ are not unique. For instance, in Example 1, the initially designed controller $(\mathbf{A}_C^0, \mathbf{B}_C^0, \mathbf{C}_C^0, \mathbf{D}_C^0)$ is the controllable companion-form realization for

$$C(z) = \frac{0.046z^6 + 0.0159z^5 - 0.4284z^4 + 0.9227z^3 - 1.0043z^2 + 0.5983z - 0.1503}{z^6 - 4.26z^5 + 8.2537z^4 - 9.441z^3 + 6.6649z^2 - 2.6885z + 0.4709}.$$

Apart from the controllable companion-form, denoted as $\mathbf{X}_c$, a controller is also often implemented in the parallel or series form in practice. Denote these two realizations of $C(z)$ as

$$\mathbf{X}_p = \begin{bmatrix} \mathbf{D}_C^p & \mathbf{C}_C^p \\ \mathbf{B}_C^p & \mathbf{A}_C^p \end{bmatrix} \tag{65}$$

and

$$\mathbf{X}_s = \begin{bmatrix} \mathbf{D}_C^s & \mathbf{C}_C^s \\ \mathbf{B}_C^s & \mathbf{A}_C^s \end{bmatrix}, \tag{66}$$

respectively. The parallel-form realization of $C(z)$ for Example 1 is given by

$$C(z) = 0.046 + \frac{1.8921e-7}{z-1} + \frac{-0.0024z+0.0013}{z^2-0.9670z+0.9589} + \frac{0.1056z-0.1487}{z^2-1.6016z+0.7103} + \frac{0.1087}{z-0.6913}$$

with

$$\begin{aligned} \mathbf{A}_C^p &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -9.5886e-1 & 9.6700e-1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -7.1030e-1 & 1.6016e0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6.9134e-1 \end{bmatrix}, \\ \mathbf{B}_C^p &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}^T, \ \mathbf{D}_C^p = 4.6000e-2, \\ \mathbf{C}_C^p &= \begin{bmatrix} 1.8921e-7 & 1.2816e-3 & -2.3654e-3 & -1.4868e-1 & 1.0555e-1 & 1.0869e-1 \end{bmatrix}; \end{aligned}$$

while the series-form realization is $C(z) =$

$$0.046\left(\frac{0.1812}{z-1}+1\right)\left(\frac{0.6344z+0.2556}{z^2-1.6016z+0.7103}+1\right)\left(\frac{4.8231}{z-0.6913}+1\right)\left(\frac{-1.0329z+0.0410}{z^2-0.9670z+0.9589}+1\right)$$

with

$$\begin{aligned} \mathbf{A}_C^s &= \begin{bmatrix} 1 & 0 & 1.8120e-1 & 1.8120e-1 & 0 & 1.8120e-1 \\ 0 & 0 & -7.1030e-1 & 2.5562e-1 & 0 & 2.5562e-1 \\ 0 & 1 & 1.6016e0 & 6.3442e-1 & 0 & 6.3442e-1 \\ 0 & 0 & 0 & 6.9134e-1 & 0 & 4.8231e0 \\ 0 & 0 & 0 & 0 & 0 & -9.5886e-1 \\ 0 & 0 & 0 & 0 & 1 & 9.6700e-1 \end{bmatrix}, \ \mathbf{B}_C^s = \begin{bmatrix} 1.8120e-1 \\ 2.5562e-1 \\ 6.3442e-1 \\ 4.8231e0 \\ 4.1007e-2 \\ -1.0329e0 \end{bmatrix}, \\ \mathbf{C}_C^s &= \begin{bmatrix} 4.6000e-2 & 0 & 4.6000e-2 & 4.6000e-2 & 0 & 4.6000e-2 \end{bmatrix}, \ \mathbf{D}_C^s = 4.6000e-2. \end{aligned}$$

The above three realizations, $\mathbf{X}_c$, $\mathbf{X}_p$ and $\mathbf{X}_s$, are sparse because they contain many trivial parameters (0,1 or -1). For Example 1, $\mathbf{X}_0$ has 13 nontrivial parameters, while $\mathbf{X}_p$ and $\mathbf{X}_s$ only have 12 nontrivial parameters (the repeated values, such as $1.8120e-1$ in $\mathbf{X}_s$, are counted as one nontrivial parameter). Clearly, a trivial parameter requires no arithmetic operation in a fixed-point implementation and does not cause any computational error. A sparse controller realization has computational advantages in practical implementations. An FWL closed-loop stability measure, which is similar to the one defined in (11) but takes into account the sparsity of controller realization, is defined in [9] as

$$f_{sp}(\mathbf{X}) \stackrel{\triangle}{=} \min_{k\in\{1,\cdots,m+n\}} \frac{1-|\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))|}{\sqrt{N_s \sum_{i,j} \delta(x_{ij}) \left|\frac{\partial\lambda_k(\overline{\mathbf{A}}(\mathbf{X}))}{\partial x_{ij}}\right|^2}} \tag{67}$$

where

$$\delta(x_{ij}) = \begin{cases} 1, & x_{ij} \text{ is nontrivial,} \\ 0, & x_{ij} \text{ is trivial,} \end{cases} \tag{68}$$

and N_s is the number of nontrivial parameters in $\mathbf{X}$. Comparing the definitions of $f_{sp}(\mathbf{X})$ and $f(\mathbf{X})$, it follows that

$$f_{sp}(\mathbf{X}) \geq f(\mathbf{X}). \tag{69}$$

Table 1 lists the values of $f(\mathbf{X})$, $f_{sp}(\mathbf{X})$ and N_s for $\mathbf{X}_{\text{opt}}$, $\mathbf{X}_p$, $\mathbf{X}_s$ and $\mathbf{X}_c$ of every examples except for Example 2. Example 2 is a multiple-input multiple-output system for which no parallel-form or series-form realization is defined. It can be seen that the optimal realization $\mathbf{X}_{\text{opt}}$ found by the proposed method has the best FWL closed-loop stability robustness as measured either by $f(\mathbf{X})$ or $f_{sp}(\mathbf{X})$, compared with the other three realizations. It can also be seen that the optimal realization obtained by the proposed search algorithm is a fully parameterized non-sparse one. The other three sparse realizations have similar numbers of nontrivial parameters, and thus have the same lighter computational load than the optimal one given here. However, it is worth pointing out that $\mathbf{X}_{\text{opt}}$ is not unique since $\mathbf{V}$ in (43) is an arbitrary orthogonal matrix. By choosing $\mathbf{V}$ in an appropriate way, one can obtain a sparse optimal realization $\mathbf{X}_{\text{opt}}$. The topic of how to make $\mathbf{X}_{\text{opt}}$ sparse is beyond the scope of this paper, and the interested readers are referred to the work [1] for details.

6 Conclusions

We have developed an efficient search algorithm for solving the class of optimal FWL controller realization problems based on the Frobenius-norm pole sensitivity measure, which have saddle points. Our approach first constructs the closed-form of a transformation matrix set which contains global optimal solutions, and then searches this set with a subgradient routine to find a global optimal solution. The proposed algorithm has considerable advantages over using direct numerical optimization methods to tackle this class of optimal FWL realization problems. In particular, when the subgradient routine converges to a solution, it is guaranteed to be a global optimal solution. It has been conjectured with some empirical supports that for many practical control systems the assumption of having saddle points is a valid one and the cases of optimal FWL controller realization problems which do not have saddle points are less common. It has been demonstrated that for this smaller class of optimal FWL realization problems without saddle points our algorithm also provides useful information to help solving them.

Appendix Proof of Theorem 4

We present the proof for *Case 2*. The proof for *Case 3* is similar and hence is omitted.

Lemma 3 (See [21]) Let real-valued matrices $\mathbf{M}_{22}$, $\mathbf{M}_{21}$ and $\mathbf{M}_{11} > 0$ be given with appropriate dimensions. Then

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{21}^T \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} > 0 \tag{70}$$

if and only if $\mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{21}^T > 0$.

Lemma 4 Given positive $\alpha, \beta \in \mathcal{R}$, $\mathbf{p}, \mathbf{y} \in \mathcal{C}^n$, and for any nonsingular $\mathbf{T} \in \mathcal{R}^{n\times n}$, we have

$$\xi(\mathbf{T}, \alpha, \beta, \mathbf{p}, \mathbf{y}) \geq (|\mathbf{y}^H\mathbf{p}| + \alpha\beta)^2. \tag{71}$$

The equality occurs if and only if there exist $\mathbf{W} \in \mathcal{R}^{n\times n}$, $\mathbf{W} > 0$ and $\theta \in [0, 2\pi)$ satisfying:

$$\mathbf{W}\Upsilon(\mathbf{y}) = \frac{\beta}{\alpha}\Upsilon(\mathbf{p}) \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{72}$$

When the above equation (72) has solutions, the equality in (71) occurs only at the transformation matrix $\mathbf{T} = \mathbf{W}^{1/2}\mathbf{V}$, where $\mathbf{V} \in \mathcal{R}^{n\times n}$ is an arbitrary orthogonal matrix.

Proof: First of all,

$$\|\mathbf{T}^{-1}\mathbf{p}\|_F^2\|\mathbf{T}^T\mathbf{y}\|_F^2 + \alpha^2\|\mathbf{T}^T\mathbf{y}\|_F^2 + \beta^2\|\mathbf{T}^{-1}\mathbf{p}\|_F^2 + \alpha^2\beta^2 \geq (\|\mathbf{T}^{-1}\mathbf{p}\|_F\|\mathbf{T}^T\mathbf{y}\|_F + \alpha\beta)^2. \tag{73}$$

The equality holds if and only if

$$\alpha\|\mathbf{T}^T\mathbf{y}\|_F = \beta\|\mathbf{T}^{-1}\mathbf{p}\|_F. \tag{74}$$

Using the Cauchy-Schwartz inequality, we have

$$(\|\mathbf{T}^{-1}\mathbf{p}\|_F\|\mathbf{T}^T\mathbf{y}\|_F + \alpha\beta)^2 \geq (\|(\mathbf{T}^T\mathbf{y})^H\mathbf{T}^{-1}\mathbf{p}\|_F + \alpha\beta)^2 \geq (|\mathbf{y}^H\mathbf{p}| + \alpha\beta)^2. \tag{75}$$

The equality holds if and only if

$$\mathbf{T}^T\mathbf{y} = c\mathbf{T}^{-1}\mathbf{p} \tag{76}$$

for some complex number c.

To achieve (73) and (75) with equality, one needs to satisfy both the conditions (74) and (76). This implies that $c = (\cos\theta + j\sin\theta)\frac{\beta}{\alpha}$ and $\theta \in [0, 2\pi)$. Thus,

$$\mathbf{T}^T\mathbf{y} = (\cos\theta + j\sin\theta)\frac{\beta}{\alpha}\mathbf{T}^{-1}\mathbf{p}. \tag{77}$$

As $\mathbf{T}$ is nonsingular, equality (77) is equivalent to

$$\mathbf{W}\mathbf{y} = (\cos\theta + j\sin\theta)\frac{\beta}{\alpha}\mathbf{p} \tag{78}$$

with $\mathbf{W} > 0$ and $\mathbf{T} = \mathbf{W}^{1/2}\mathbf{V}$. Noticing the map Υ defined in (31), condition (78) can be viewed as

$$\mathbf{W}\Upsilon(\mathbf{y}) = \frac{\beta}{\alpha}\Upsilon(\mathbf{p})\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{79}$$

Lemma 5 Given positive $\alpha, \beta \in \mathcal{R}$, $\mathbf{p}, \mathbf{y} \in \mathcal{C}^n$ and $\operatorname{rank}(\Upsilon(\mathbf{y})) = 2$, equation (79) has solutions if and only if $\det((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})) > 0$. Moreover, any solution to equation (79) can be expressed as

$$\left.\begin{array}{l} \tan\theta = \frac{a_{21}-a_{12}}{a_{11}+a_{22}} \\ a_{11}\cos\theta - a_{12}\sin\theta > 0 \\ \mathbf{W} = \mathbf{Q}\begin{bmatrix} \mathbf{H} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{G} \end{bmatrix}\mathbf{Q}^T \end{array}\right\} \tag{80}$$

where

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = (\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p}); \tag{81}$$

the orthogonal matrix $\mathbf{Q}$ can be obtained from the QR factorization of $\Upsilon(\mathbf{y})$, that is,

$$\Upsilon(\mathbf{y}) = \mathbf{Q}\begin{bmatrix} \gamma_{11} & 0 & 0 & \cdots & 0 \\ \gamma_{12} & \gamma_{22} & 0 & \cdots & 0 \end{bmatrix}^T; \tag{82}$$

$\mathbf{H}$ and $\mathbf{F}$ are determined by

$$\mathbf{H} = \frac{\beta}{\alpha}\begin{bmatrix} \gamma_{11} & \gamma_{12} \\ 0 & \gamma_{22} \end{bmatrix}^{-T}(\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} \gamma_{11} & \gamma_{12} \\ 0 & \gamma_{22} \end{bmatrix}^{-1}, \tag{83}$$

$$\mathbf{F} = \frac{\beta}{\alpha}\begin{bmatrix} \mathbf{e}_3^T \\ \vdots \\ \mathbf{e}_n^T \end{bmatrix}\mathbf{Q}^T\Upsilon(\mathbf{p})\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} \gamma_{11} & \gamma_{12} \\ 0 & \gamma_{22} \end{bmatrix}^{-1}; \tag{84}$$

and $\mathbf{G}$ is given as

$$\mathbf{G} = \mathbf{F}\mathbf{H}^{-1}\mathbf{F}^T + \mathbf{U} \tag{85}$$

with $\mathbf{U} \in \mathcal{R}^{(n-2)\times(n-2)}$ being an arbitrary positive definite matrix.

Proof: If $\det((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})) > 0$, it is easy to verify that $\mathbf{W}$ and θ given by (80)–(85) are a solution to equation (79). If on the other hand equation (79) has a solution $\mathbf{W}$ and θ, it can be seen that

$$(\Upsilon(\mathbf{y}))^T\mathbf{W}\Upsilon(\mathbf{y}) = \frac{\beta}{\alpha}(\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{86}$$

On account of $(\Upsilon(\mathbf{y}))^T\mathbf{W}\Upsilon(\mathbf{y}) > 0$, we have

$$(\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} > 0. \tag{87}$$

A necessary condition to satisfy (87) is that

$$\det\left((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\right) > 0. \tag{88}$$

Since the left side of the above inequality is equal to $\det((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p}))$, The condition (88) becomes $\det((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})) > 0$. This completes the proof of the first part of Lemma 5.

Now, when (81) is given, (87) holds if and only if all of the following three conditions are satisfied

$$\left.\begin{array}{l} a_{21}\cos\theta - a_{22}\sin\theta = a_{11}\sin\theta + a_{12}\cos\theta\\ a_{11}\cos\theta - a_{12}\sin\theta > 0\\ \det((\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})) > 0\end{array}\right\} \tag{89}$$

From the first line of (89), we directly obtain $\tan\theta = \frac{a_{21}-a_{12}}{a_{11}+a_{22}}$. Denote

$$\mathbf{S} = \mathbf{Q}^T\mathbf{W}\mathbf{Q}. \tag{90}$$

Then, from (79), (82) and (90), one has

$$\mathbf{S}\,[\,\mathbf{e}_1 \quad \mathbf{e}_2\,]\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix} = \mathbf{S}\begin{bmatrix}\gamma_{11} & 0 & 0 & \cdots & 0\\ \gamma_{12} & \gamma_{22} & 0 & \cdots & 0\end{bmatrix}^T = \frac{\beta}{\alpha}\mathbf{Q}^T\Upsilon(\mathbf{p})\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}. \tag{91}$$

Partition $\mathbf{S}$ into

$$\mathbf{S} = \begin{bmatrix}\mathbf{H} & \mathbf{F}^T\\ \mathbf{F} & \mathbf{G}\end{bmatrix} \tag{92}$$

where $\mathbf{H} \in \mathcal{R}^{2\times 2}$, $\mathbf{F} \in \mathcal{R}^{(n-2)\times 2}$ and $\mathbf{G} \in \mathcal{R}^{(n-2)\times(n-2)}$. Then from (91) and noticing

$$(\Upsilon(\mathbf{y}))^T = \begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix}^T [\,\mathbf{e}_1 \quad \mathbf{e}_2\,]^T\,\mathbf{Q}^T, \tag{93}$$

we have

$$\mathbf{H} = \begin{bmatrix}\mathbf{e}_1^T\\ \mathbf{e}_2^T\end{bmatrix}\mathbf{S}\,[\,\mathbf{e}_1 \quad \mathbf{e}_2\,] = \frac{\beta}{\alpha}\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix}^{-T}(\Upsilon(\mathbf{y}))^T\Upsilon(\mathbf{p})\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix}^{-1}, \tag{94}$$

$$\mathbf{F} = \begin{bmatrix}\mathbf{e}_3^T\\ \vdots\\ \mathbf{e}_n^T\end{bmatrix}\mathbf{S}\,[\,\mathbf{e}_1 \quad \mathbf{e}_2\,] = \frac{\beta}{\alpha}\begin{bmatrix}\mathbf{e}_3^T\\ \vdots\\ \mathbf{e}_n^T\end{bmatrix}\mathbf{Q}^T\Upsilon(\mathbf{p})\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\gamma_{11} & \gamma_{12}\\ 0 & \gamma_{22}\end{bmatrix}^{-1}. \tag{95}$$

From Lemma 3 and $\mathbf{S} > 0$, it is known that $\mathbf{G} = \mathbf{F}\mathbf{H}^{-1}\mathbf{F}^T + \mathbf{U}$, where $\mathbf{U} \in \mathcal{R}^{(n-2)\times(n-2)}$ is an arbitrary positive definite matrix.

Combining Lemmas 4 and 5 leads to Theorem 4 for *Case 2*.

Acknowledgements

J. Wu and J. Chu wish to thank the support of the National Natural Science Foundation of China (Grants No.60174026 and No.60374002) and 973 program of China (Grant No.2002CB312200). S. Chen wish to thank the support of the United Kingdom Royal Academy of Engineering.

References

[1] M. Gevers and G. Li, *Parameterizations in Control, Estimation and Filtering Problems: Accuracy Aspects*. London: Springer Verlag, 1993.

[2] R.S.H. Istepanian and J.F. Whidborne, eds., *Digital Controller Implementation and Fragility: A Modern Perspective*. London: Springer Verlag, 2001.

[3] G.F. Franklin, J.D. Powell and M.L. Workman, *Digital Control of Dynamic Systems*. 3rd Edition. Reading, MA: Addison-Wesley, 1998.

[4] K. Liu, R.E. Skelton and K. Grigoriadis, "Optimal controllers for finite wordlength implementation," *IEEE Trans. Automatic Control*, Vol.37, No.9, pp.1294–1304, 1992.

[5] G. Li, J. Wu, S. Chen and K.Y. Zhao, "Optimum structures of digital controllers in sampled-data systems: a roundoff noise analysis," *IEE Proc. Control Theory and Applications*, Vol.149, No.3, pp.247–255, 2002.

[6] I.J. Fialho and T.T. Georgiou, "Computational algorithms for sparse optimal digital controller realizations," in: R.S.H. Istepanian and J.F. Whidborne, eds., *Digital Controller Implementation and Fragility: A Modern Perspective*. London: Springer Verlag, 2001, pp.105–121.

[7] A.G. Madievski, B.D.O. Anderson and M. Gevers, "Optimum realizations of sampled-data controllers for FWL sensitivity minimization," *Automatica*, Vol.31, No.3, pp.367–379, 1995.

[8] J.F. Whidborne, J. Wu and R.S.H. Istepanian, "Finite word length stability issues in an l_1 framework," *Int. J. Control*, Vol.73, No.2, pp.166–176, 2000.

[9] G. Li, "On the structure of digital controllers with finite word length consideration," *IEEE Trans. Automatic Control*, Vol.43, No.5, pp.689–693, 1998.

[10] P.E. Mantey, "Eigenvalue sensitivity and state-variable selection," *IEEE Trans. Automatic Control*, Vol.13, No.3, pp.263–269, 1968.

[11] J. Wu, S. Chen, G. Li and J. Chu, "Optimal finite-precision state-estimate feedback controller realizations of discrete-time systems," *IEEE Trans. Automatic Control*, Vol.45, No.8, pp.1550–1554, 2000.

[12] G.-H. Yang, J.L. Wang and C. Lin, "H_∞ control for linear systems with additive controller gain variations," *Int. J. Control*, Vol.73, No.16, pp.1500–1506, 2000.

[13] J.F. Whidborne, J. Wu, R.S.H. Istepanian and J. Chu, “Comments on ‘On the structure of digital controllers with finite word length consideration’,” *IEEE Trans. Automatic Control*, Vol.45, No.2, pp.344–344, 2000.

[14] R.S.H. Istepanian, J. Wu and J.F. Whidborne, “Controller realizations of a teleoperated dual-wrist assembly system with finite word length considerations,” *IEEE Trans. Control Systems Technology*, Vol.9, No.4, pp.624–628, 2001.

[15] G. Owen, *Game Theory*. 3rd Edition. San Diego, CA: Academic Press, 1995.

[16] J. Szép and F. Forgó, *Introduction to the Theory of Games*. Dordrecht, Holland: D. Reidel Publishing Company, 1985.

[17] I.J. Fialho and T.T. Georgiou, “On stability and performance of sampled-data systems subject to wordlength constraint,” *IEEE Trans. Automatic Control*, Vol.39, No.12, pp.2476–2481, 1994.

[18] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*. Oxford, U.K.: Clarendon Press, 1988.

[19] S. Chen, J. Wu, R.H. Istepanian, J. Chu and J.F. Whidborne, “Optimising stability bounds of finite-precision controller structures for sampled-data systems in the δ-operator domain,” *IEE Proc. Control Theory and Applications*, Vol.146, No.6, pp.517–526, 1999.

[20] J. Yan and S.E. Salcudean, “Teleoperation controller design using H_∞-optimization with application to motion-scaling,” *IEEE Trans. Control Systems Technology*, Vol.4, No.3, pp.244–258, 1996.

[21] S. Boyd, L.El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*. Philadelphia, PA: SIAM, 1994.

[22] J. Zowe, “Nondifferentiable optimization,” in: K. Schittkowski, eds., *Computational Mathematical Programming*. Berlin: Springer-Verlag, 1985.

[23] N.Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Berlin: Springer-Verlag, 1985.

[24] K.C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*. Berlin: Springer-Verlag, 1985.

[25] T. Chen and B.A. Francis, “Input-output stability of sampled-data systems,” *IEEE Trans. Automatic Control*, Vol.36, No.1, pp.50–58, 1991.

[26] L.H. Keel and S.P. Bhattacharyya, “Robust, fragile, or optimal,” *IEEE Trans. Automatic Control*, Vol.42, No.8, pp.1098–1105, 1997.

		$\mathbf{X}_c$	$\mathbf{X}_p$	$\mathbf{X}_s$	$\mathbf{X}_{\mathrm{opt}}$
Example 1	$f(\mathbf{X})$	$3.1797e-11$	$8.0156e-9$	$2.8727e-9$	$6.7344e-6$
	$f_{sp}(\mathbf{X})$	$7.4944e-11$	$1.8464e-8$	$7.1095e-9$	$6.7344e-6$
	N_s	13	12	12	49
Example 3	$f(\mathbf{X})$	$5.0963e-10$	$1.5234e-5$	$3.0949e-6$	$2.7321e-4$
	$f_{sp}(\mathbf{X})$	$8.5965e-10$	$2.7908e-5$	$5.4711e-6$	$2.7321e-4$
	N_s	9	8	8	25
Example 4	$f(\mathbf{X})$	$1.6555e-10$	$8.3351e-10$	$1.4611e-7$	$5.0786e-5$
	$f_{sp}(\mathbf{X})$	$6.1068e-10$	$1.5627e-7$	$3.0905e-7$	$5.0786e-5$
	N_s	7	7	7	16
Example 5	$f(\mathbf{X})$	$1.6699e-4$	$5.4326e-4$	$4.8802e-4$	$3.2716e-3$
	$f_{sp}(\mathbf{X})$	$2.5956e-4$	$2.4426e-3$	$7.3417e-4$	$3.2716e-3$
	N_s	5	4	4	9
Example 6	$f(\mathbf{X})$	$6.7163e-4$	$1.0775e-3$	$1.0774e-3$	$4.8968e-3$
	$f_{sp}(\mathbf{X})$	$9.5044e-4$	$3.5239e-3$	$1.6347e-3$	$4.8968e-3$
	N_s	5	4	4	9

Table 1: Comparison of performance measures for different realizations.

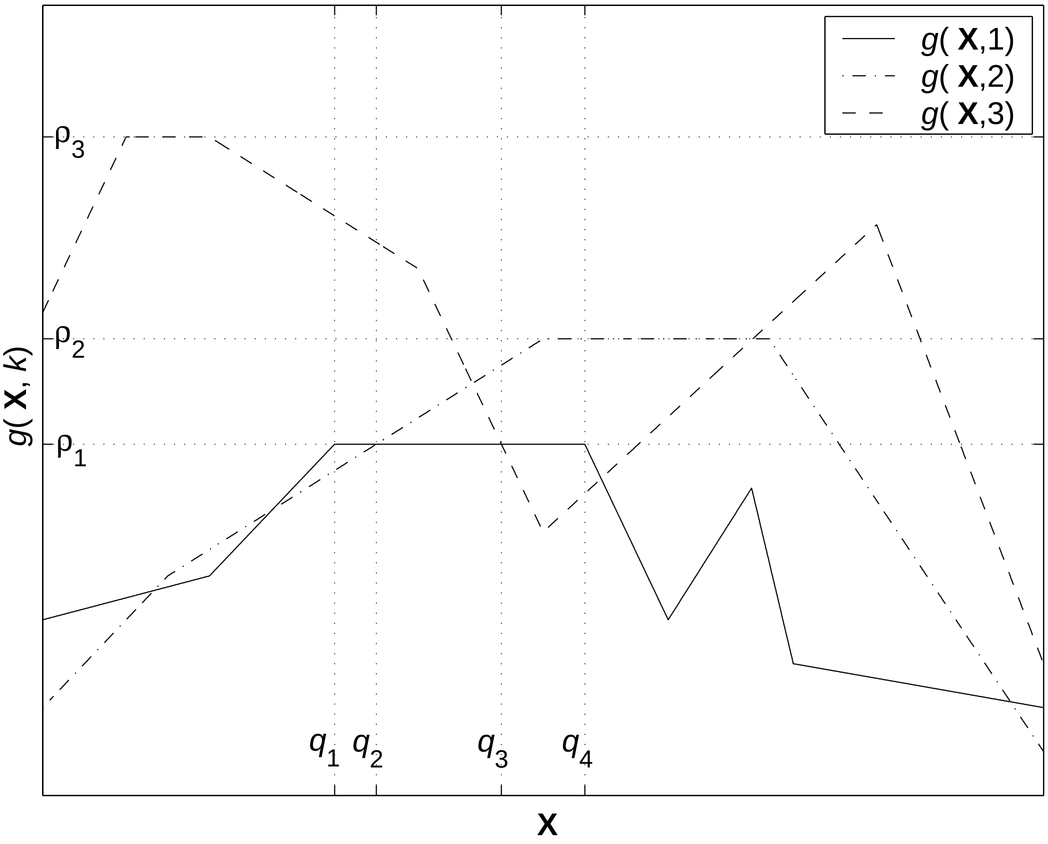

Figure 1: A simple illustration on ρ_k, $\mathcal{X}$ and saddle points.

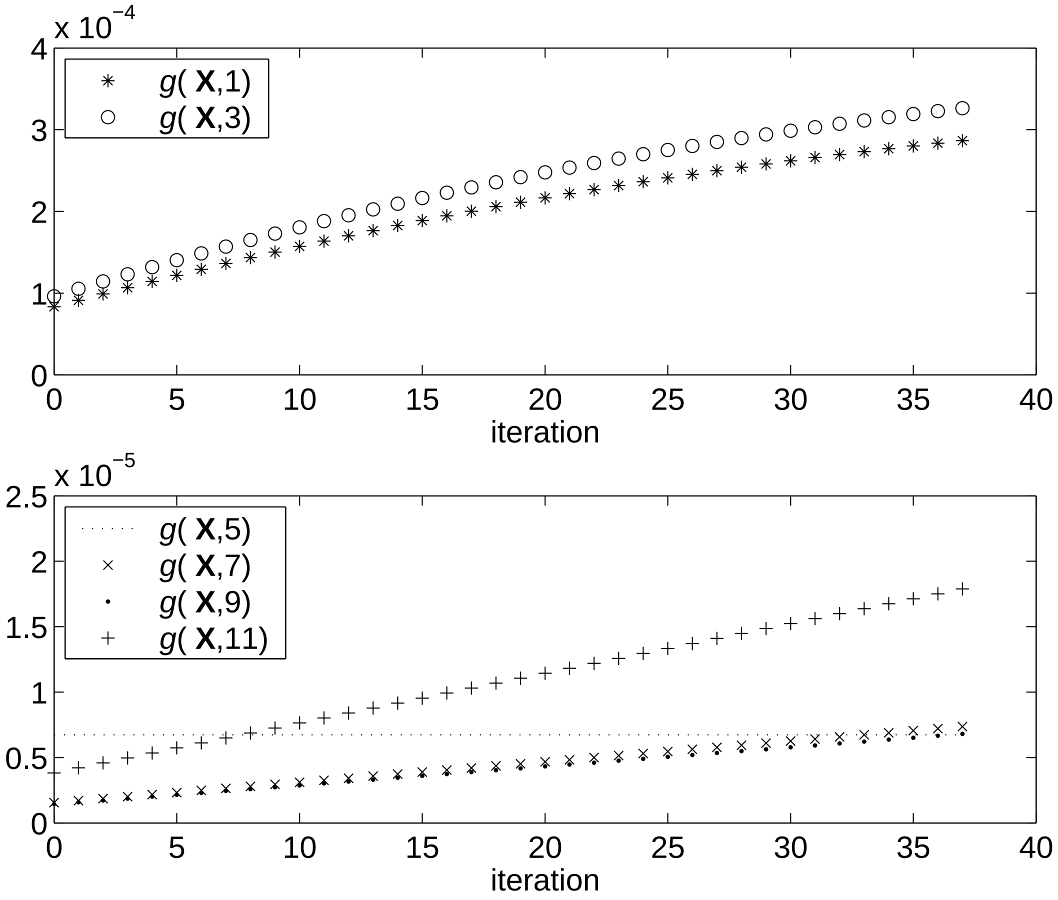

Figure 2: The values of $g(\mathbf{X}, k)$ in each iteration of the algorithm for Example 1.

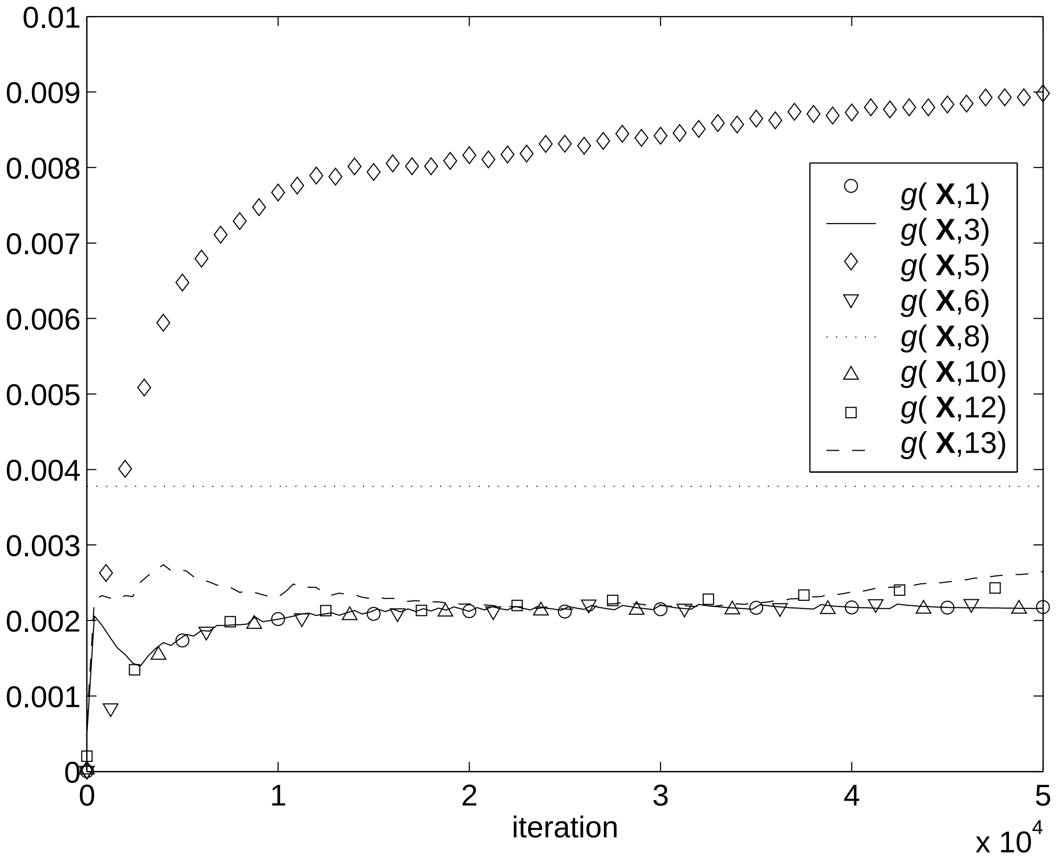

Figure 3: The values of $g(\mathbf{X}, k)$ in each iteration of the algorithm for Example 2.