Constructing Sparse Realizations of Finite-Precision Digital Controllers Based on a Closed-Loop Stability Related Measure

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

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

‡ Department 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

We present a study of the finite word length (FWL) implementation for digital controller structures with sparseness consideration. A new closed-loop stability related measure is derived, taking into account the number of trivial elements in a controller realization. A practical design procedure is presented, which first obtains a controller realization that maximizes a lower bound of the proposed measure, and then uses a stepwise algorithm to make the realization sparse. Simulation results show that the proposed design procedure yields computationally efficient controller realizations with enhanced FWL closed-loop stability performance.

Index Terms — digital controller, finite word length, closed-loop stability, sparse realization, optimization, stepwise algorithm, real-time computation.

1 Introduction

It is well-known that a designed stable control system may achieve a lower than predicted performance or even become unstable when the control law is implemented with a finite-precision device due to FWL effects. In real-time applications where computational efficiency is critical, a digital controller implemented in fixed-point arithmetic has certain advantages. With a fixed-point processor, the detrimental FWL effects are markedly increased due to a reduced precision. As the FWL effects on the closed-loop stability depend on the controller realization structure,

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

many studies have addressed the problem of finding "optimal" realizations of finite-precision controller structures based on various FWL stability measures [1]-[7]. Except [5], these design methods usually yield fully parameterized controller structures, that is, they generally do not produce sparse controller realizations.

It is highly desirable that a controller realization has a sparse structure, containing many trivial elements of 0, 1 or -1. This is particularly important for real-time applications with high-order controllers, as it will achieve better computational efficiency. It is known that canonical controller realizations have sparse structures but may not have the required FWL stability robustness. This poses a complex problem of finding sparse controller realizations with good FWL closed-loop stability characteristics. In [8], sparseness consideration is imposed as constraints in optimizing a FWL stability measure using an adaptive simulated annealing (ASA) algorithm. This approach is difficult to extend to high-order controllers due to high computational requirements. In our previous works [9],[10], a design procedure has been given to obtain sparse controller realizations based on a FWL pole-sensitivity stability related measure.

In this study we derive a new improved FWL closed-loop stability related measure, which takes into account the number of trivial elements in a controller realization. The true optimal realization that maximizes this measure will possess an optimal trade-off between robustness to FWL errors and sparse structure. However, it is not known how to obtain such an optimal realization. We extend an iterative algorithm [2],[11] to search for a suboptimal solution. Specifically, we first obtain the realization that maximizes a lower bound of the proposed stability measure. This can easily be done [5],[7] but the resulting realization is not sparse. A stepwise algorithm is then applied to make the realization sparse without sacrificing FWL stability robustness too much. The proposed method has some advantages over the existing methods [5],[9],[10]: it is less conservative in estimating the robustness of the FWL closed-loop stability and the computational complexity is considerably reduced. Numerical examples are used to test this design procedure and to compare its performance with the previous method [9],[10].

2 A stability related measure with sparseness considerations

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 $(\mathbf{A}_P, \mathbf{B}_P, \mathbf{C}_P)$, where $\mathbf{A}_P \in \mathcal{R}^{m\times m}$, $\mathbf{B}_P \in \mathcal{R}^{m\times l}$ and $\mathbf{C}_P \in \mathcal{R}^{q\times m}$.

Let $(\mathbf{A}_C, \mathbf{B}_C, \mathbf{C}_C, \mathbf{D}_C)$ be a state-space description of the controller $C(z)$, 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 $\mathbf{D}_C \in \mathcal{R}^{l\times q}$. A linear system with a given transfer function matrix has an infinite number of state-space descriptions. In fact, if $(\mathbf{A}_C^0, \mathbf{B}_C^0, \mathbf{C}_C^0, \mathbf{D}_C^0)$ is a state-space description of $C(z)$, all the state-space descriptions of $C(z)$ form a *realization* set

$$\mathcal{S}_C \stackrel{\triangle}{=} \left\{(\mathbf{A}_C, \mathbf{B}_C, \mathbf{C}_C, \mathbf{D}_C) | \mathbf{A}_C = \mathbf{T}^{-1}\mathbf{A}_C^0\mathbf{T}, \mathbf{B}_C = \mathbf{T}^{-1}\mathbf{B}_C^0, \mathbf{C}_C = \mathbf{C}_C^0\mathbf{T}, \mathbf{D}_C = \mathbf{D}_C^0\right\} \quad (1)$$

where $\mathbf{T} \in \mathcal{R}^{n\times n}$ is any non-singular matrix. Denote $N \stackrel{\triangle}{=} (l+n)(q+n)$ and

$$\mathbf{X} \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{D}_C & \mathbf{C}_C \\ \mathbf{B}_C & \mathbf{A}_C \end{bmatrix} = \begin{bmatrix} x_1 & x_{l+n+1} & \cdots & x_{N-l-n+1} \\ x_2 & x_{l+n+2} & \cdots & x_{N-l-n+2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{l+n} & x_{2l+2n} & \cdots & x_N \end{bmatrix} \quad (2)$$

The stability of the closed-loop control system depends on the eigenvalues of the closed-loop system matrix

$$\overline{\mathbf{A}}(\mathbf{X}) = \begin{bmatrix} \mathbf{A}_P + \mathbf{B}_P\mathbf{D}_C\mathbf{C}_P & \mathbf{B}_P\mathbf{C}_C \\ \mathbf{B}_C\mathbf{C}_P & \mathbf{A}_C \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{A}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix} \mathbf{X} \begin{bmatrix} \mathbf{C}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix} \stackrel{\triangle}{=} \mathbf{M}_0 + \mathbf{M}_1\mathbf{X}\mathbf{M}_2 \quad (3)$$

where $\mathbf{0}$ denotes the zero matrix of appropriate dimension and $\mathbf{I}_n$ the $n \times n$ identity matrix. 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 $\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))$, $1 \le i \le m+n$, are within the unit disk.

When a $\mathbf{X}$ is implemented with a fixed-point processor, it is perturbed to $\mathbf{X} + \Delta\mathbf{X}$ due to the FWL effect. Each element of $\Delta\mathbf{X}$ is bounded by $\pm\varepsilon/2$, that is,

$$\mu(\Delta\mathbf{X}) \stackrel{\triangle}{=} \max_{j\in\{1,\cdots,N\}} |\Delta x_j| \le \varepsilon/2 \quad (4)$$

For a fixed-point processor of B_s bits, let $B_s = B_i + B_f$, where 2^{B_i} is a "normalization" factor to make the absolute value of each element of $2^{-B_i}\mathbf{X}$ no larger than 1. Thus, B_i are bits required for the integer part of a number and B_f are bits used to implement the fractional part of a number. It can easily be seen that $\varepsilon = 2^{-B_f}$. With the perturbation $\Delta\mathbf{X}$, $\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))$ is moved to $\lambda_i(\overline{\mathbf{A}}(\mathbf{X} + \Delta\mathbf{X}))$. 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 B_s-bit implemented $\mathbf{X}$. It is therefore critical to choose a realization $\mathbf{X}$ that has a good closed-loop stability robustness to the FWL error. Another important consideration is the sparseness of $\mathbf{X}$. Those elements of $\mathbf{X}$, which have values 0, 1 and -1, are called *trivial* parameters. A trivial parameter requires no operations in the fixed-point implementation and does not cause any computational error at all. Thus $\Delta x_j = 0$ when $x_j = 0, 1$ or -1. In order to take into account this property of trivial controller parameters, we define an indicator function as

$$\delta(x) = \begin{cases} 0, & \text{if } x = 0, 1 \text{ or } -1 \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

We emphasize that in this paper a trivial element is referred to as 0, 1 or -1. A natural extension could also consider "semi-trivial" elements of $\mathbf{X}$, which are a power of two, $x = 2^{-i}$, such as $x = 0.5$, 0.25 and so on. These elements can be realized with simple shift operations in the fixed-point implementation. The design of such kind of sparse controller realizations are however much more difficult (see for example [12]).

We are now ready to propose a new FWL closed-loop stability related measure which takes into account the sparseness of a controller realization. When the FWL error $\Delta\mathbf{X}$ is small,

$$\Delta|\lambda_i| \triangleq \left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}+\Delta\mathbf{X}))\right| - \left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right| \approx \sum_{j=1}^{N} \frac{\partial|\lambda_i|}{\partial x_j}\Delta x_j \delta(x_j), \quad \forall i \in \{1, \cdots, m+n\} \tag{6}$$

where $\frac{\partial|\lambda_i|}{\partial x_j}$ is evaluated at $\mathbf{X}$. It follows from the Cauchy inequality that

$$|\Delta|\lambda_i|| \le \sqrt{N_s \sum_{j=1}^{N} \left|\frac{\partial|\lambda_i|}{\partial x_j}\right|^2 |\Delta x_j|^2 \delta(x_j)} \le \mu(\Delta\mathbf{X}) \sqrt{N_s \sum_{j=1}^{N} \left|\frac{\partial|\lambda_i|}{\partial x_j}\right|^2 \delta(x_j)}, \quad \forall i \tag{7}$$

where N_s is the number of the nontrivial elements in $\mathbf{X}$. This leads to the following FWL closed-loop stability related measure

$$\mu_1(\mathbf{X}) = \min_{i \in \{1,\cdots,m+n\}} \frac{1 - \left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right|}{\sqrt{N_s \sum\limits_{j=1}^{N} \delta(x_j) \left|\frac{\partial|\lambda_i|}{\partial x_j}\right|^2}} \tag{8}$$

The rationale of this measure is obvious. If the norm of the FWL error $\Delta\mathbf{X}$ is smaller than $\mu_1(\mathbf{X})$, i.e. $\mu(\Delta\mathbf{X}) < \mu_1(\mathbf{X})$, it follows from (7) and (8) that $|\Delta|\lambda_i|| < 1 - \left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right|$. Therefore

$$\left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}+\Delta\mathbf{X}))\right| \le |\Delta|\lambda_i|| + \left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right| < 1 \tag{9}$$

which means that the closed-loop system remains stable under the FWL error $\Delta\mathbf{X}$. In other words, for a given controller realization $\mathbf{X}$, the closed-loop system can tolerate those FWL perturbations $\Delta\mathbf{X}$ whose norms, as defined in (4), are less than $\mu_1(\mathbf{X})$. The larger $\mu_1(\mathbf{X})$ is, the larger FWL errors the closed-loop system can tolerate. Hence, $\mu_1(\mathbf{X})$ is a stability related measure describing the FWL closed-loop stability performance of a controller realization $\mathbf{X}$. This measure clearly considers the number of trivial parameters in a controller realization. We can now discuss how to compute $\mu_1(\mathbf{X})$. First we have the following lemma from [5],[7].

Lemma 1 Let $\overline{\mathbf{A}}(\mathbf{X}) = \mathbf{M}_0 + \mathbf{M}_1\mathbf{X}\mathbf{M}_2$ given in (3) be diagonalisable, and have eigenvalues $\{\lambda_i\} = \{\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\}$. Denote $\mathbf{p}_i$ a right eigenvector of $\overline{\mathbf{A}}(\mathbf{X})$ corresponding to the eigenvalue λ_i. Define $\mathbf{M}_p \triangleq \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{m+n} \end{bmatrix}$ and $\mathbf{M}_y \triangleq \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{m+n} \end{bmatrix} = \mathbf{M}_p^{-H}$, where H is the transpose and conjugate operator and $\mathbf{y}_i$ the reciprocal left eigenvector related to λ_i. Then

$$\frac{\partial \lambda_i}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial \lambda_i}{\partial x_1} & \cdots & \frac{\partial \lambda_i}{\partial x_{N-l-n+1}} \\ \vdots & \cdots & \vdots \\ \frac{\partial \lambda_i}{\partial x_{l+n}} & \cdots & \frac{\partial \lambda_i}{\partial x_N} \end{bmatrix} = \mathbf{M}_1^T \mathbf{y}_i^* \mathbf{p}_i^T \mathbf{M}_2^T \tag{10}$$

where the superscript $*$ denotes the conjugate operation and T the transpose operator.

Next, we have the following result

Lemma 2 For $\mathbf{X}$, $\overline{\mathbf{A}}(\mathbf{X})$ and $\{\lambda_i\}$ as defined in lemma 1,

$$\frac{\partial |\lambda_i|}{\partial \mathbf{X}} = \frac{1}{|\lambda_i|} \mathrm{Re}\left[\lambda_i^* \frac{\partial \lambda_i}{\partial \mathbf{X}}\right] \tag{11}$$

where Re$[\cdot]$ denotes the real part.

Proof: Noting $|\lambda_i| = \sqrt{\lambda_i^* \lambda_i}$ leads to

$$\frac{\partial |\lambda_i|}{\partial \mathbf{X}} = \frac{1}{2\sqrt{\lambda_i^* \lambda_i}} \left(\frac{\partial \lambda_i^*}{\partial \mathbf{X}} \lambda_i + \lambda_i^* \frac{\partial \lambda_i}{\partial \mathbf{X}} \right) = \frac{1}{2|\lambda_i|} \left(\left(\frac{\partial \lambda_i}{\partial \mathbf{X}} \right)^* \lambda_i + \lambda_i^* \frac{\partial \lambda_i}{\partial \mathbf{X}} \right) = \frac{1}{|\lambda_i|} \mathrm{Re}\left[\lambda_i^* \frac{\partial \lambda_i}{\partial \mathbf{X}}\right] \tag{12}$$

Combining lemma 1 with lemma 2 results in the following proposition, which shows that, given a $\mathbf{X}$, the value of $\mu_1(\mathbf{X})$ can easily be calculated.

Proposition 1 For $\mathbf{X}$, $\mathbf{M}_1$, $\mathbf{M}_2$, $\overline{\mathbf{A}}(\mathbf{X})$, $\{\lambda_i\}$, $\mathbf{p}_i$ and $\mathbf{y}_i$ as defined in lemma 1,

$$\frac{\partial |\lambda_i|}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial |\lambda_i|}{\partial x_1} & \cdots & \frac{\partial |\lambda_i|}{\partial x_{N-l-n+1}} \\ \vdots & \cdots & \vdots \\ \frac{\partial |\lambda_i|}{\partial x_{l+n}} & \cdots & \frac{\partial |\lambda_i|}{\partial x_N} \end{bmatrix} = \frac{1}{|\lambda_i|} \mathbf{M}_1^T \mathrm{Re}\left[\lambda_i^* \mathbf{y}_i^* \mathbf{p}_i^T\right] \mathbf{M}_2^T \tag{13}$$

It should be emphasized that the FWL stability related measure (8) is different with the one used in [5],[9],[10], which is given by

$$\mu_2(\mathbf{X}) = \min_{i \in \{1,\cdots,m+n\}} \frac{1 - \left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right|}{\sqrt{N_s \sum_{j=1}^{N} \delta(x_j) \left|\frac{\partial \lambda_i}{\partial x_j}\right|^2}} \tag{14}$$

The key difference between $\mu_1(\mathbf{X})$ and $\mu_2(\mathbf{X})$ is that the former considers the sensitivity of $|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))|$ while the latter considers the sensitivity of $\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))$. It is well-known that the stability of a linear discrete-time system depends only on the moduli of its eigenvalues. As $\mu_2(\mathbf{X})$ includes the unnecessary eigenvalue arguments in consideration, it is generally conservative in comparison with $\mu_1(\mathbf{X})$. This can be verified strictly. From lemma 2,

$$\left|\frac{\partial\left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right|}{\partial x_j}\right| \leq \frac{\left|\lambda_i^*(\overline{\mathbf{A}}(\mathbf{X}))\frac{\partial\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))}{\partial x_j}\right|}{\left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right|} = \left|\frac{\partial\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))}{\partial x_j}\right| \tag{15}$$

which means that $\mu_2(\mathbf{X}) \leq \mu_1(\mathbf{X})$. The result given in [7] has confirmed that by considering the sensitivity of eigenvalue moduli rather than the sensitivity of eigenvalues, a better FWL closed-loop stability related measure can be obtained. It is worth pointing out that the proposed measure $\mu_1(\mathbf{X})$ also has considerable computational advantages over the existing $\mu_2(\mathbf{X})$. This is because $\frac{\partial|\lambda_i|}{\partial\mathbf{X}}$ is real-valued while $\frac{\partial\lambda_i}{\partial\mathbf{X}}$ is complex-valued. Thus the optimisation process and sparse transformation procedure, discussed in the next section, require much less computation than the previous approach [5],[9],[10], unless all the system eigenvalues are real-valued in which case $\mu_1(\mathbf{X})$ and $\mu_2(\mathbf{X})$ become identical.

3 Suboptimal controller realizations with sparse structures

The optimal sparse controller realization with a maximum tolerance to FWL perturbation in principle is the solution of the following optimization problem:

$$v \triangleq \max_{\mathbf{X}\in\mathcal{S}_C} \mu_1(\mathbf{X}) \tag{16}$$

However, it is difficult to solve for the above optimization problem because $\mu_1(\mathbf{X})$ includes $\delta(x_j)$ and is not a continuous function with respect to controller parameters x_j. To get around this difficulty, we consider a lower bound of $\mu_1(\mathbf{X})$ defined by

$$\underline{\mu_1}(\mathbf{X}) = \min_{i\in\{1,\cdots,m+n\}} \frac{1-\left|\lambda_i(\overline{\mathbf{A}}(\mathbf{X}))\right|}{\sqrt{N\sum_{j=1}^{N}\left|\frac{\partial|\lambda_i|}{\partial x_j}\right|^2}} \tag{17}$$

Obviously, $\underline{\mu_1}(\mathbf{X}) \leq \mu_1(\mathbf{X})$ and $\underline{\mu_1}(\mathbf{X})$ is a continuous function of controller parameters. It is relatively easy to optimize $\underline{\mu_1}(\mathbf{X})$ (e.g. [7]). Let the "optimal" controller realization $\mathbf{X}_{\text{opt}}$ be the solution of the optimization problem

$$\omega \triangleq \max_{\mathbf{X}\in\mathcal{S}_C} \underline{\mu_1}(\mathbf{X}) \tag{18}$$

Notice that $\mathbf{X}_{\text{opt}}$ is generally not the optimal solution of (16) and does not have a sparse structure. However, it can readily be attempted by the following optimization procedure.

3.1 Optimization of the lower-bound measure

Assume that an initial controller realization has been obtained by some design procedure and is denoted as $\mathbf{X}_0$. According to (1)–(3), a similarity transformation of $\mathbf{X}_0$ by $\mathbf{T}$ is

$$\mathbf{X} = \mathbf{X}(\mathbf{T}) = \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-1} \end{bmatrix} \mathbf{X}_0 \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \tag{19}$$

where $\det(\mathbf{T}) \neq 0$. The closed-loop system matrix for the realization $\mathbf{X}$ is

$$\overline{\mathbf{A}}(\mathbf{X}) = \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-1} \end{bmatrix} \overline{\mathbf{A}}(\mathbf{X}_0) \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \tag{20}$$

Obviously, $\overline{\mathbf{A}}(\mathbf{X})$ has the same set of eigenvalues as $\overline{\mathbf{A}}(\mathbf{X}_0)$, denoted as $\{\lambda_i^0\}$. From (20), applying proposition 1 results in

$$\left.\frac{\partial|\lambda_i|}{\partial \mathbf{X}}\right|_{\mathbf{X}(\mathbf{T})} = \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{T} \end{bmatrix} \left.\frac{\partial|\lambda_i|}{\partial \mathbf{X}}\right|_{\mathbf{X}_0} \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-T} \end{bmatrix} \tag{21}$$

For a complex-valued matrix $\mathbf{M} \in \mathcal{C}^{(l+n)\times(q+n)}$ with elements m_{sk}, denote the Frobenius norm

$$\|\mathbf{M}\|_F \triangleq \sqrt{\sum_{s=1}^{l+n} \sum_{k=1}^{q+n} m_{sk}^* m_{sk}} \tag{22}$$

Then the lower-bound measure (17) can be rewritten as

$$\begin{aligned} \underline{\mu_1}(\mathbf{X}) &= \min_{i\in\{1,\cdots,m+n\}} \frac{1-|\lambda_i^0|}{\sqrt{N}\left\| \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{T} \end{bmatrix} \left.\frac{\partial|\lambda_i|}{\partial \mathbf{X}}\right|_{\mathbf{X}_0} \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-T} \end{bmatrix} \right\|_F} \\ &= \min_{i\in\{1,\cdots,m+n\}} \frac{1}{\sqrt{N}\left\| \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{T} \end{bmatrix} \mathbf{\Phi}_i \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-T} \end{bmatrix} \right\|_F} \end{aligned} \tag{23}$$

where

$$\mathbf{\Phi}_i \triangleq \frac{\left.\frac{\partial|\lambda_i|}{\partial \mathbf{X}}\right|_{\mathbf{X}_0}}{1-|\lambda_i^0|} \tag{24}$$

are fixed matrices that are independent of $\mathbf{T}$. Thus, if we introduce the cost function

$$f(\mathbf{T}) = \min_{i\in\{1,\cdots,m+n\}} \frac{1}{\sqrt{N}\left\| \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{T} \end{bmatrix} \mathbf{\Phi}_i \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-T} \end{bmatrix} \right\|_F} = \underline{\mu_1}(\mathbf{X}) \tag{25}$$

the optimal similarity transformation $\mathbf{T}_{\text{opt}}$ can be obtained by solving for the following unconstrained optimization problem

$$\omega = \max_{\mathbf{T}\in\mathcal{R}^{n\times n}} f(\mathbf{T}) \tag{26}$$

with a measure of monitoring the singular values of $\mathbf{T}$ to make sure that $\det(\mathbf{T}) \neq 0$ [13]. The unconstrained optimization problem (26) can be solved, for example, using the simplex search algorithm [14], the simulated annealing algorithm [15], the ASA algorithm [16] or the genetic algorithm [17]. In our previous study, we have found that the ASA is very efficient in solving for this kind of optimization problems [7]. With $\mathbf{T}_{\text{opt}}$, the corresponding optimal realization $\mathbf{X}_{\text{opt}}$ that is the solution of (18) can readily be computed.

3.2 Stepwise transformation algorithm for sparse realizations

As the optimal sparse realization that maximizes μ_1 is difficult if not impossible to obtain, we will search for a suboptimal solution of (16). More precisely, we will search for a realization that is sparse with a large enough value of μ_1. Since $\mathbf{X}_{\text{opt}}$ maximizes $\underline{\mu_1}$ and $\underline{\mu_1}$ is a lower-bound of μ_1, $\mathbf{X}_{\text{opt}}$ will produce a satisfactory large value of μ_1, although it usually contains no trivial elements. We can make $\mathbf{X}_{\text{opt}}$ sparse by changing one nontrivial element of $\mathbf{X}_{\text{opt}}$ into a trivial one at a step, under the constraint that the value of $\underline{\mu_1}$ does not reduce too much. This process will produce a sparse realization $\mathbf{X}_{\text{spa}}$ with a satisfactory value of $\underline{\mu_1}$. Clearly such a $\mathbf{X}_{\text{spa}}$ is not a true optimal solution of (16). Notice that, even though $\underline{\mu_1}(\mathbf{X}_{\text{spa}}) \leq \underline{\mu_1}(\mathbf{X}_{\text{opt}})$, it is possible that $\mu_1(\mathbf{X}_{\text{spa}}) \geq \mu_1(\mathbf{X}_{\text{opt}})$. In other words, $\mathbf{X}_{\text{spa}}$ may actually achieve better FWL stability performance than $\mathbf{X}_{\text{opt}}$. The design procedure is similar to the one used in [9],[10]. We now describe the detailed stepwise procedure for obtaining $\mathbf{X}_{\text{spa}}$.

Step 1: Set τ to a very small positive real number (e.g. 10^{-5}). The transformation matrix $\mathbf{T} \in \mathcal{R}^{n \times n}$ is initially set to $\mathbf{T}_{\text{opt}}$ so that $\mathbf{X}(\mathbf{T}) = \mathbf{X}_{\text{opt}}$.

Step 2: Find out all the trivial elements $\{\eta_1, \cdots, \eta_r\}$ in $\mathbf{X}(\mathbf{T})$ (a parameter is considered to be trivial if its distance to 0, 1 or -1 is less than a tolerance value, say 10^{-8}). Denote ξ the nontrivial element in $\mathbf{X}(\mathbf{T})$ that is the nearest to 0, 1 or -1.

Step 3: Choose $\mathbf{S} \in \mathcal{R}^{n \times n}$ such that

i) $\underline{\mu_1}(\mathbf{X}(\mathbf{T} + \tau\mathbf{S}))$ is close to $\underline{\mu_1}(\mathbf{X}(\mathbf{T}))$.

ii) $\{\eta_1, \cdots, \eta_r\}$ in $\mathbf{X}(\mathbf{T})$ remain unchanged in $\mathbf{X}(\mathbf{T} + \tau\mathbf{S})$.

iii) ξ in $\mathbf{X}(\mathbf{T})$ is changed as nearer as possible to 0, 1 or -1 in $\mathbf{X}(\mathbf{T} + \tau\mathbf{S})$.

iv) $\|\mathbf{S}\|_F = 1$.

If $\mathbf{S}$ does not exist, $\mathbf{T}_{\text{spa}} = \mathbf{T}$ and terminate the algorithm.

Step 4: $\mathbf{T} = \mathbf{T} + \tau\mathbf{S}$. If ξ in $\mathbf{X}(\mathbf{T})$ is nontrivial, go to step 3. If ξ becomes trivial, go to step 2.

The key of the above algorithm is **Step 3** which guarantees that $\mathbf{X}(\mathbf{T}_{\text{spa}})$ has good performance as measured by $\underline{\mu_1}$ and contains many trivial parameters. We now discuss how to obtain $\mathbf{S}$. Denote Vec$(\cdot)$ the column stacking operator. With a very small τ, condition i) means that

$$\left(\text{Vec}\left(\frac{d\underline{\mu_1}}{d\mathbf{T}}\right)\right)^T \text{Vec}\left(\mathbf{S}\right) = 0 \tag{27}$$

and condition ii) means that

$$\begin{cases} \left(\text{Vec}\left(\frac{d\eta_1}{d\mathbf{T}}\right)\right)^T \text{Vec}\left(\mathbf{S}\right) = 0 \\ \qquad\qquad\vdots \\ \left(\text{Vec}\left(\frac{d\eta_r}{d\mathbf{T}}\right)\right)^T \text{Vec}\left(\mathbf{S}\right) = 0 \end{cases} \tag{28}$$

Denote the matrix

$$\mathbf{E} \triangleq \begin{bmatrix} \left(\text{Vec}\left(\frac{d\underline{\mu_1}}{d\mathbf{T}}\right)\right)^T \\ \left(\text{Vec}\left(\frac{d\eta_1}{d\mathbf{T}}\right)\right)^T \\ \vdots \\ \left(\text{Vec}\left(\frac{d\eta_r}{d\mathbf{T}}\right)\right)^T \end{bmatrix} \in \mathcal{R}^{(r+1)\times n^2} \tag{29}$$

Vec$(\mathbf{S})$ must belong to the null space $\mathcal{N}(\mathbf{E})$ of $\mathbf{E}$. If $\mathcal{N}(\mathbf{E})$ is empty, Vec$(\mathbf{S})$ does not exist and the algorithm is terminated. If $\mathcal{N}(\mathbf{E})$ is not empty, it must have basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_t\}$, assuming that the dimension of $\mathcal{N}(\mathbf{E})$ is t. Condition iii) requires moving ξ to its desired value (0, 1 or -1) as fast as possible, and we should choose Vec$(\mathbf{S})$ as the orthogonal projection of Vec $\left(\frac{d\xi}{d\mathbf{T}}\right)$ onto $\mathcal{N}(\mathbf{E})$. Noting condition iv), we can compute Vec$(\mathbf{S})$ as follows:

$$a_i = \mathbf{b}_i^T \text{Vec}\left(\frac{d\xi}{d\mathbf{T}}\right) \in \mathcal{R}, \quad \forall i \in \{1, \cdots, t\} \tag{30}$$

$$\mathbf{v} = \sum_{i=1}^{t} a_i \mathbf{b}_i \in \mathcal{R}^{n^2} \tag{31}$$

$$\text{Vec}(\mathbf{S}) = \pm \frac{\mathbf{v}}{\sqrt{\mathbf{v}^T\mathbf{v}}} \in \mathcal{R}^{n^2} \tag{32}$$

The sign in (32) is chosen in the following way. If ξ is larger than its nearest desired value, the minus sign is taken; otherwise, the plus sign is used.

In the above algorithm, the derivatives $\frac{d\underline{\mu_1}}{d\mathbf{T}}$, $\frac{d\xi}{d\mathbf{T}}$, $\frac{d\eta_1}{d\mathbf{T}}, \cdots, \frac{d\eta_r}{d\mathbf{T}}$ are needed. For calculating these required derivatives, the following well-known fact is useful. Given any element y_{ij} in a nonsingular $\mathbf{Y} \in R^{n\times n}$ with $i \in \{1, \cdots, n\}$ and $j \in \{1, \cdots, n\}$,

$$\frac{\partial \mathbf{Y}}{\partial y_{ij}} = \mathbf{e}_i\mathbf{e}_j^T \quad \text{and} \quad \frac{\partial \mathbf{Y}^{-1}}{\partial y_{ij}} = -\mathbf{Y}^{-1}\mathbf{e}_i\mathbf{e}_j^T\mathbf{Y}^{-1} \tag{33}$$

where $\mathbf{e}_i$ denotes the ith coordinate vector. In (19), define

$$\mathbf{U}_1 = \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \quad \text{and} \quad \mathbf{U}_2 = \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \tag{34}$$

For any element x_{ks} in $\mathbf{X} = \mathbf{U}_1^{-1}\mathbf{X}_0\mathbf{U}_2$, where $k \in \{1, \cdots, l+n\}$ and $s \in \{1, \cdots, q+n\}$, and any t_{ij} in $\mathbf{T}$, where $i \in \{1, \cdots, n\}$ and $j \in \{1, \cdots, n\}$,

$$\begin{aligned}
\frac{\partial x_{ks}}{\partial t_{ij}} &= \mathbf{e}_k^T \frac{\partial \mathbf{U}_1^{-1}}{\partial t_{ij}} \mathbf{X}_0 \mathbf{U}_2 \mathbf{e}_s + \mathbf{e}_k^T \mathbf{U}_1^{-1} \mathbf{X}_0 \frac{\partial \mathbf{U}_2}{\partial t_{ij}} \mathbf{e}_s \\
&= -\mathbf{e}_k^T \mathbf{U}_1^{-1} \mathbf{e}_{l+i} \mathbf{e}_{l+j}^T \mathbf{U}_1^{-1} \mathbf{X}_0 \mathbf{U}_2 \mathbf{e}_s + \mathbf{e}_k^T \mathbf{U}_1^{-1} \mathbf{X}_0 \mathbf{e}_{q+i} \mathbf{e}_{q+j}^T \mathbf{e}_s \\
&= -\mathbf{e}_k^T \mathbf{U}_1^{-1} \mathbf{e}_{l+i} \mathbf{e}_{l+j}^T \mathbf{X} \mathbf{e}_s + \mathbf{e}_k^T \mathbf{U}_1^{-1} \mathbf{X}_0 \mathbf{e}_{q+i} \mathbf{e}_{q+j}^T \mathbf{e}_s
\end{aligned} \tag{35}$$

That is,

$$\begin{aligned}
\frac{dx_{ks}}{d\mathbf{T}} = & \begin{bmatrix} \mathbf{e}_k^T \mathbf{U}_1^{-1} & & \\ & \ddots & \\ & & \mathbf{e}_k^T \mathbf{U}_1^{-1} \end{bmatrix} \left(\begin{bmatrix} \mathbf{X}_0 \mathbf{e}_{q+1} \mathbf{e}_{q+1}^T & \cdots & \mathbf{X}_0 \mathbf{e}_{q+1} \mathbf{e}_{q+n}^T \\ \vdots & \cdots & \vdots \\ \mathbf{X}_0 \mathbf{e}_{q+n} \mathbf{e}_{q+1}^T & \cdots & \mathbf{X}_0 \mathbf{e}_{q+n} \mathbf{e}_{q+n}^T \end{bmatrix} \right. \\
& \left. - \begin{bmatrix} \mathbf{e}_{l+1} \mathbf{e}_{l+1}^T \mathbf{X} & \cdots & \mathbf{e}_{l+1} \mathbf{e}_{l+n}^T \mathbf{X} \\ \vdots & \cdots & \vdots \\ \mathbf{e}_{l+n} \mathbf{e}_{l+1}^T \mathbf{X} & \cdots & \mathbf{e}_{l+n} \mathbf{e}_{l+n}^T \mathbf{X} \end{bmatrix} \right) \begin{bmatrix} \mathbf{e}_s & & \\ & \ddots & \\ & & \mathbf{e}_s \end{bmatrix}
\end{aligned} \tag{36}$$

Thus, we can readily calculate $\frac{d\xi}{d\mathbf{T}}$, $\frac{d\eta_1}{d\mathbf{T}}$, $\cdots$, $\frac{d\eta_r}{d\mathbf{T}}$. Next, define

$$i_0 = \arg \min_{i \in \{1, \cdots, m+n\}} \frac{1}{\sqrt{N} \left\| \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^T \end{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{i}} \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-T} \end{bmatrix} \right\|_F} \tag{37}$$

Similar to the derivation of $\frac{dx_{ks}}{d\mathbf{T}}$, for any element w_{ks} in $\mathbf{W} = \mathbf{U}_1^T \boldsymbol{\Phi}_{\boldsymbol{i_0}} \mathbf{U}_2^{-T}$, where $k \in \{1, \cdots, l+n\}$ and $s \in \{1, \cdots, q+n\}$, we have

$$\begin{aligned}
\frac{dw_{ks}}{d\mathbf{T}} = & \begin{bmatrix} \mathbf{e}_k^T & & \\ & \ddots & \\ & & \mathbf{e}_k^T \end{bmatrix} \left(\begin{bmatrix} \mathbf{e}_{l+1} \mathbf{e}_{l+1}^T \boldsymbol{\Phi}_{\boldsymbol{i_0}} & \cdots & \mathbf{e}_{l+n} \mathbf{e}_{l+1}^T \boldsymbol{\Phi}_{\boldsymbol{i_0}} \\ \vdots & \cdots & \vdots \\ \mathbf{e}_{l+1} \mathbf{e}_{l+n}^T \boldsymbol{\Phi}_{\boldsymbol{i_0}} & \cdots & \mathbf{e}_{l+n} \mathbf{e}_{l+n}^T \boldsymbol{\Phi}_{\boldsymbol{i_0}} \end{bmatrix} \right. \\
& \left. - \begin{bmatrix} \mathbf{W} \mathbf{e}_{q+1} \mathbf{e}_{q+1}^T & \cdots & \mathbf{W} \mathbf{e}_{q+n} \mathbf{e}_{q+1}^T \\ \vdots & \cdots & \vdots \\ \mathbf{W} \mathbf{e}_{q+1} \mathbf{e}_{q+n}^T & \cdots & \mathbf{W} \mathbf{e}_{q+n} \mathbf{e}_{q+n}^T \end{bmatrix} \right) \begin{bmatrix} \mathbf{U}_2^{-T} \mathbf{e}_s & & \\ & \ddots & \\ & & \mathbf{U}_2^{-T} \mathbf{e}_s \end{bmatrix}
\end{aligned} \tag{38}$$

Since

$$\underline{\mu_1} = \frac{1}{\sqrt{N} \sqrt{\sum_{k=1}^{l+n} \sum_{s=1}^{q+n} w_{ks}^* w_{ks}}} \tag{39}$$

We can calculate

$$\frac{d\underline{\mu_1}}{d\mathbf{T}} = -\frac{1}{\sqrt{N} \|\mathbf{W}\|_F^3} \mathrm{Re} \left[\sum_{k=1}^{l+n} \sum_{s=1}^{q+n} w_{ks}^* \frac{dw_{ks}}{d\mathbf{T}} \right] \tag{40}$$

Before presenting some simulation results, we point out that given a FWL pole-sensitivity measure, such as $\underline{\mu_1}(\mathbf{X})$, an estimated minimum bit length for guaranteeing closed-loop stability can be estimated using [6],[7]

$$\hat{B}_{s,\min} = B_i + \mathrm{Int}[-\log_2(\underline{\mu_1}(\mathbf{X}))] - 1 \tag{41}$$

where the integer $\mathrm{Int}[x] \geq x$.

4 Numerical examples

We present two design examples to show how our approach can be used efficiently to search for sparse controller realizations with satisfactory FWL closed-loop stability performance.

Example 1. This was a single-input single-output fluid power speed control system studied in [18],[19]. The plant model was in the continuous-time form and a continuous-time H_∞ optimal controller was designed in [18]. In this study, we obtained a discrete-time plant $P(z)$ and a discrete-time controller $C(z)$ by sampling the continuous-time plant and H_∞ controller using a sampling rate of 2 kHz. The discrete-time plant $P(z)$ was given by

$$\mathbf{A}_P = \begin{bmatrix} 9.9988e-01 & 1.9432e-05 & 5.9320e-05 & -6.2286e-05 \\ -4.9631e-07 & 2.3577e-02 & 2.3709e-05 & 2.3672e-05 \\ -1.5151e-03 & 2.3709e-02 & 2.3751e-05 & 2.3898e-05 \\ 1.5908e-03 & 2.3672e-02 & 2.3898e-05 & 2.3667e-05 \end{bmatrix},$$

$$\mathbf{B}_P = \begin{bmatrix} 3.0504e-03 \\ -1.2373e-02 \\ -1.2375e-02 \\ -8.8703e-02 \end{bmatrix}, \quad \mathbf{C}_P = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

The initial realization of the controller $C(z)$ given in a controllable canonical form was

$$\mathbf{X}_0 = \begin{bmatrix} -8.0843e-04 & -1.6112e-03 & -1.5998e-03 & -1.5885e-03 & -1.5773e-03 \\ 1 & 0 & 0 & 0 & -3.3071e-01 \\ 0 & 1 & 0 & 0 & 1.9869e+00 \\ 0 & 0 & 1 & 0 & -3.9816e+00 \\ 0 & 0 & 0 & 1 & 3.3255e+00 \end{bmatrix}$$

Notice that the controllable canonical form was very sparse, containing only 9 non-trivial elements. The closed-loop transition matrix $\overline{\mathbf{A}}(\mathbf{X}_0)$ was then formed using (3), from which the eigenvalues and the corresponding eigenvectors of the ideal (infinite-precision) closed-loop system were computed. The closed-loop eigenvalues were:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \end{bmatrix} = \begin{bmatrix} 9.9956e-01 \; + \; j\,2.5674e-04 \\ 9.9956e-01 \; - \; j\,2.5674e-04 \\ 9.9955e-01 \\ 9.9333e-01 \\ 3.3333e-01 \\ 2.3625e-02 \\ 2.7819e-19 \\ -3.8735e-09 \end{bmatrix}$$

The optimisation problem (26) was constructed, and the ASA algorithm [16] obtained the following solution

$$\mathbf{T}_{\text{opt}} = \begin{bmatrix} 2.3644e+07 & 2.0268e+06 & 1.0498e+08 & -4.7194e+06 \\ -1.1839e+08 & -9.9623e+06 & -5.2570e+08 & 2.3636e+07 \\ 1.6622e+08 & 1.3872e+07 & 7.3801e+08 & -3.3191e+07 \\ -7.1475e+07 & -5.9364e+06 & -3.1729e+08 & 1.4274e+07 \end{bmatrix}$$

The corresponding controller realization, which maximises the lower-bound measure $\underline{\mu_1}$, was

$$\mathbf{X}_{\mathrm{opt}} = \begin{bmatrix} -8.0843e-04 & 6.4378e-02 & -1.1974e-02 & -1.1493e-02 & -2.2104e-01 \\ 2.7588e-03 & 1.0010e+00 & -1.4054e-02 & 1.0924e-03 & -8.9552e-03 \\ -2.2776e-04 & -5.8175e-02 & 3.3649e-01 & 7.5457e-02 & 1.3962e-03 \\ -2.5200e-04 & 1.0668e-03 & 1.6778e-02 & 9.9766e-01 & 1.5423e-03 \\ 8.1179e-03 & 5.1520e-03 & 3.1311e-02 & -3.8681e-03 & 9.9031e-01 \end{bmatrix}$$

The stepwise transformation algorithm was then applied to make $\mathbf{X}_{\mathrm{opt}}$ sparse, which yielded the following similarity transformation matrix and corresponding controller realization

$$\mathbf{T}_{\mathrm{spa}} = \begin{bmatrix} -1.7499e+05 & -4.5848e+05 & 2.1159e+08 & 3.0140e+02 \\ 8.1616e+05 & 1.8611e+06 & -1.0592e+09 & -1.2931e+03 \\ -1.0789e+06 & -2.3503e+06 & 1.4869e+09 & 1.8162e+03 \\ 4.3753e+05 & 9.4770e+05 & -6.3921e+08 & -7.8105e+02 \end{bmatrix}$$

$$\mathbf{X}_{\mathrm{spa}} = \begin{bmatrix} -8.0843e-04 & 1.6372e-02 & -5.4228e-04 & -1.8348e-03 & -6.9866e-02 \\ 0 & 1 & 0 & 0 & -1.4073e-03 \\ 0 & -6.8678e-02 & 3.3285e-01 & 4.2230e-01 & 5.8895e-04 \\ 0 & -5.6623e-06 & -7.6002e-04 & 1 & 0 \\ 2.3061e-02 & -8.1961e-06 & 0 & 4.5476e-05 & 9.9262e-01 \end{bmatrix}$$

As the controller order is not large for this example, the computational effort in solving the optimisation problem (26) is relatively low. In a typical workstation network, $\mathbf{X}_{\mathrm{opt}}$ was obtained within a few minutes. The complexity of the sparse procedure obviously depends on how sparse one wants to force a realization to be. Typically a few hundreds of iterations are sufficient. For this example, $\mathbf{X}_{\mathrm{spa}}$ was obtained from $\mathbf{X}_{\mathrm{opt}}$ within a few minutes.

Table 1 compares the FWL closed-loop stability performance and the number of non-trivial elements for the three controller realizations $\mathbf{X}_0$, $\mathbf{X}_{\mathrm{opt}}$ and $\mathbf{X}_{\mathrm{spa}}$, respectively. For a comparison purpose, the values of the previous stability related measure μ_2 and its lower-bound $\underline{\mu_2}$ together with their corresponding estimated minimum bit lengths [9],[10] are also given in Table 1 for the three realizations. We also exploited the true minimum bit length that guaranteed closed-loop stability for a controller realization $\mathbf{X}$ using the following computer simulation. Starting with a large enough bit length, e.g. $B_s = 100$, we rounded the controller $\mathbf{X}$ to B_s bits and checked the stability of the closed-loop system, i.e. observing whether the closed-loop poles were within the open unit disk. Reduced B_s by 1 and repeated the process until there appeared to be closed-loop instability at B_u bits. Then $B_{s,\min} = B_u + 1$. The values of $B_{s,\min}$ for the three realizations are given in Table 1. Notice that for $B_s \geq B_{s,\min}$, the B_s-bit implemented controller will always guarantee closed-loop stability. However, there may exist some $B_s < B_u$, which regains closed-loop stability. For example, for the initial realization $\mathbf{X}_0$, $B_u = 32$, i.e. when the bit length is smaller than 33, the closed-loop becomes unstable. At $B_s = 16$ or 15, the closed-loop becomes stable again. With $B_s < 15$ instability is observed again.

For this example, the canonical realization $\mathbf{X}_0$ is the most sparse with only 9 non-trivial parameters, but its FWL closed-loop stability related measure $\mu_1(\mathbf{X}_0)$ is very poor. The realization $\mathbf{X}_{\text{opt}}$ has a much better FWL stability robustness as indicated by $\mu_1(\mathbf{X}_{\text{opt}})$, but its all 25 elements are non-trivial. The realization $\mathbf{X}_{\text{spa}}$ has the largest $\mu_1(\mathbf{X}_{\text{spa}})$ and, moreover, it is sparse with only 16 non-trivial parameters. This example only has a pair of complex eigenvalues. Even so, the results shown in Table 1 indicate that the proposed μ_1 ($\underline{\mu_1}$ respectively) is less conservative in estimating the robustness of FWL closed-loop stability than the previous measure μ_2 ($\underline{\mu_2}$ respectively)[1]. We also computed the unit impulse response of the closed-loop control system when the controllers were the infinite-precision implemented $\mathbf{X}_0$ and 16-bit implemented three different controller realizations. Notice that any realization $\mathbf{X} \in \mathcal{S}_C$ implemented in infinite precision will achieve the exact performance of the infinite-precision implemented $\mathbf{X}_0$, which is the *designed* controller performance. For this reason, the the infinite-precision implemented $\mathbf{X}_0$ is referred to as the *ideal* controller realization $\mathbf{X}_{\text{ideal}}$. Fig. 1 compares the unit impulse response of the plant output $y(k)$ for the ideal controller $\mathbf{X}_{\text{ideal}}$ with those of the 16-bit implemented $\mathbf{X}_0$, $\mathbf{X}_{\text{opt}}$ and $\mathbf{X}_{\text{spa}}$. It can be seen that the performance of the 16-bit implemented $\mathbf{X}_{\text{spa}}$ is almost identical to that of the 16-bit implemented $\mathbf{X}_{\text{opt}}$, which is very close to the ideal performance.

Example 2. This was a dual wrist assembly which was a prototype telerobotic system used in micro-surgery experiments [20]. This dual wrist assembly is a two-input ($l = 2$) two-output ($q = 2$) system with a plant order $m = 4$, and the digital controller designed using $\mathcal{H}_\infty$ method had an order of $n = 10$ [20]. The total number of controller parameters was $N = 144$. The $\mathcal{H}_\infty$ controller designed in [20], which was fully parameterised with $N_s = N$, was used as the initial controller realization $\mathbf{X}_0$, and the realization $\mathbf{X}_{\text{opt}}$ that maximized the lower-bound measure $\underline{\mu_1}$ was obtained using the ASA algorithm. This realization was then made sparse using the algorithm given in subsection 3.2 to yield $\mathbf{X}_{\text{spa}}$. As the controller was a high-order one, the computational cost was much higher, compared with the previous example, and the entire design process was completed in 50 minutes in a typical workstation network. Table 2 summarizes the performance of these three different controller realizations. It can be seen that the proposed measure μ_1 ($\underline{\mu_1}$ respectively) yielded less conservative results in estimating the robustness of FWL closed-loop stability than the previous measure μ_2 ($\underline{\mu_2}$ respectively).

Fig. 2 compares the first-input to first-output unit impulse response of the closed-loop system

[1]If arg $\mu_1 =$ arg $\mu_2 = i_0$ (arg $\underline{\mu_1} =$ arg $\underline{\mu_2}$ respectively) and λ_{i_0} is real valued, then obviously $\mu_1 = \mu_2$ ($\underline{\mu_1} = \underline{\mu_2}$ respectively).

obtained using the ideal controller $\mathbf{X}_{\rm ideal}$ with those obtained using the 20-bit implemented controller realizations $\mathbf{X}_{\rm opt}$ and $\mathbf{X}_{\rm spa}$. The 20-bit implemented $\mathbf{X}_0$ is unstable and therefore is not shown. It can be seen that the performance of the 20-bit implemented $\mathbf{X}_{\rm opt}$ is close to the ideal performance, and the 20-bit implemented $\mathbf{X}_{\rm spa}$, although deviating from the ideal one, achieves a stable closed-loop performance. Fig. 3 compares the second-input to second-output ideal unit impulse response of the closed-loop system with those of the 24-bit implemented $\mathbf{X}_0$, $\mathbf{X}_{\rm opt}$ and $\mathbf{X}_{\rm spa}$. It can be seen that the performance of the 24-bit implemented $\mathbf{X}_{\rm spa}$ closely matches that of the 24-bit implemented $\mathbf{X}_{\rm opt}$, which itself is almost identical to the ideal performance. Deviation from the ideal performance by the 24-bit implemented $\mathbf{X}_0$ can clearly be seen from Fig. 3. This example clearly demonstrates the effectiveness of the proposed design procedure. The sparse controller realization $\mathbf{X}_{\rm spa}$ obtained has almost half of its parameters being trivial, and it has a much improved FWL closed-loop stability robustness over the initial controller realization $\mathbf{X}_0$.

5 Conclusions

We have studied FWL implementation of digital controller structures with sparseness consideration. A new FWL closed-loop stability related measure has been derived, which takes into account the number of trivial parameters in a controller realization. It has been shown that this new measure yields a more accurate estimate for the robustness of FWL closed-loop stability. A practical procedure has been presented to obtain sparse controller realizations with satisfactory FWL closed-loop stability characteristics. Two examples demonstrate that the proposed design procedure yields computationally efficient controller structures suitable for FWL implementation in real-time applications.

Acknowledgements

J. Wu and S. Chen wish to thank the support of the U.K. Royal Society under a KC Wong fellowship (RL/ART/CN/XFI/KCW/11949).

References

[1] P. Moroney, A.S. Willsky and P.K. Houpt, "The digital implementation of control compensators: the coefficient wordlength issue," *IEEE Trans. Automatic Control*, Vol.25, No.8, pp.621–630, 1980.

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

[3] I.J. Fialho and T.T. Georgiou, "On stability and performance of sampled data systems subject to word length constraint," *IEEE Trans. Automatic Control*, Vol.39, No.12, pp.2476–2481, 1994.

[4] 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.

[5] 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.

[6] R.H. Istepanian, G. Li, J. Wu and J. Chu, "Analysis of sensitivity measures of finite-precision digital controller structures with closed-loop stability bounds," *IEE Proc. Control Theory and Applications*, Vol.145, No.5, pp.472–478, 1998.

[7] J. Wu, S. Chen, G. Li, R.H. Istepanian and J. Chu, "An improved closed-loop stability related measure for finite-precision digital controller realizations," *IEEE Trans. Automatic Control*, Vol.46, No.7, pp.1162–1166, 2001.

[8] S. Chen, R.H. Istepanian, J.F. Whidborne and J. Wu, "Adaptive simulated annealing for designing finite-precision PID controller structures," in *IEE Colloquium Optimisation in Control: Methods and Application* (London), Nov.10, 1998, pp.3/1–3/3.

[9] J. Wu, S. Chen, G. Li and J. Chu, "Digital finite-precision controller realizations with sparseness considerations," in *Proc. 3rd Chinese World Cong. Intelligent Control and Intelligent Automation* (Hefei, China), June 28-July 2, 2000, pp.2869–2873.

[10] R.H. Istepanian, J. Wu and S. Chen, "Sparse realizations of optimal finite-precision teleoperation controller structures," in *Proc. ACC'2000* (Chicago, IL), June 28-30, 2000, pp.687–691.

[11] D.S.K. Chan, "Constrained minimization of roundoff noise in fixed-point digital filters," in *Proc. ICASSP'79*, April 1979, pp.335–339.

[12] J.F. Whidborne and R.S.H. Istepanian, "Genetic algorithm approach to designing finite-precision controller structures," *IEE Proc. Control Theory and Applications*, Vol.148, No.5, pp.377–382, 2001.

[13] R.H. Istepanian, S. Chen, J. Wu and J.F. Whidborne, "Optimal finite-precision controller realization of sample data systems," *Int. J. Systems Science*, Vol.31, No.4, pp.429–438, 2000.

[14] J. Kowalik and M.R. Osborne, *Methods for Unconstrained Optimization Problems.* New York: Elsevier, 1968.

[15] E.H.L. Aarts and J.H.M. Korst, *Simulated Annealing and Boltzmann Machines.* John Wiley and Sons, 1989.

[16] S. Chen and B.L. Luk, "Adaptive simulated annealing for optimization in signal processing applications," *Signal Processing*, Vol.79, No.1, pp.117–128, 1999.

[17] D.E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison Wesley, 1989.

[18] I. Njabeleke, R.F. Pannett, P.K. Chawdhry and C.R. Burrows, "H_∞ control in fluid power," in *IEE Colloquium Robust Control – Theory, Software and Applications* (London, U.K.), 1997, pp.7/1–7/4.

[19] 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.

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

realization	$\mathbf{X}_0$	$\mathbf{X}_{\mathrm{opt}}$	$\mathbf{X}_{\mathrm{spa}}$
N_s	9	25	16
$\underline{\mu_1}$	2.604531e-12	6.862889e-05	6.108122e-05
$\hat{B}_{s,\min}$ based on $\underline{\mu_1}$	40	14	14
μ_1	4.417941e-12	6.862889e-05	1.348887e-04
$\hat{B}_{s,\min}$ based on μ_1	39	14	13
$\underline{\mu_2}$	2.604531e-12	5.500982e-05	6.108052e-05
$\hat{B}_{s,\min}$ based on $\underline{\mu_2}$	40	15	14
μ_2	4.417941e-12	5.500982e-05	1.348839e-04
$\hat{B}_{s,\min}$ based on μ_2	39	15	13
$B_{s,\min}$	33	11	11

Table 1: Performance comparison of the three different controller realizations for Example 1.

realization	$\mathbf{X}_0$	$\mathbf{X}_{\mathrm{opt}}$	$\mathbf{X}_{\mathrm{spa}}$
N_s	144	144	75
$\underline{\mu_1}$	4.306085e-04	3.224443e-03	1.279414e-03
$\hat{B}_{s,\min}$ based on $\underline{\mu_1}$	27	24	25
μ_1	4.306085e-04	3.224443e-03	2.331625e-03
$\hat{B}_{s,\min}$ based on μ_1	27	24	24
$\underline{\mu_2}$	1.173382e-04	1.057405e-03	4.393420e-04
$\hat{B}_{s,\min}$ based on $\underline{\mu_2}$	29	25	27
μ_2	1.173382e-04	1.057405e-03	9.249032e-04
$\hat{B}_{s,\min}$ based on μ_2	29	25	26
$B_{s,\min}$	22	20	20

Table 2: Performance comparison of the three different controller realizations for Example 2.

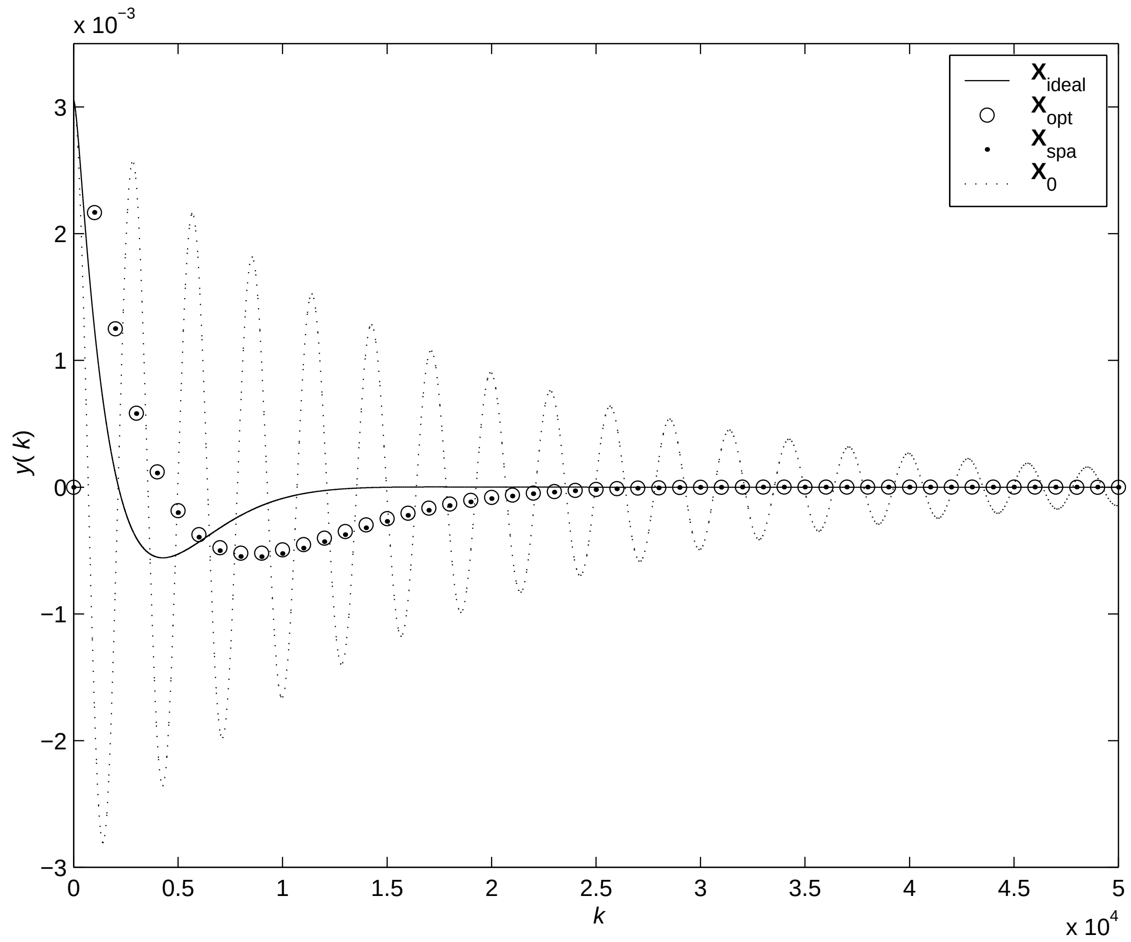

Figure 1: Comparison of unit impulse response of the infinite-precision controller implementation $\mathbf{X}_{\rm ideal}$ with those of the three 16-bit implemented controller realizations $\mathbf{X}_0$, $\mathbf{X}_{\rm opt}$ and $\mathbf{X}_{\rm spa}$ for Example 1.

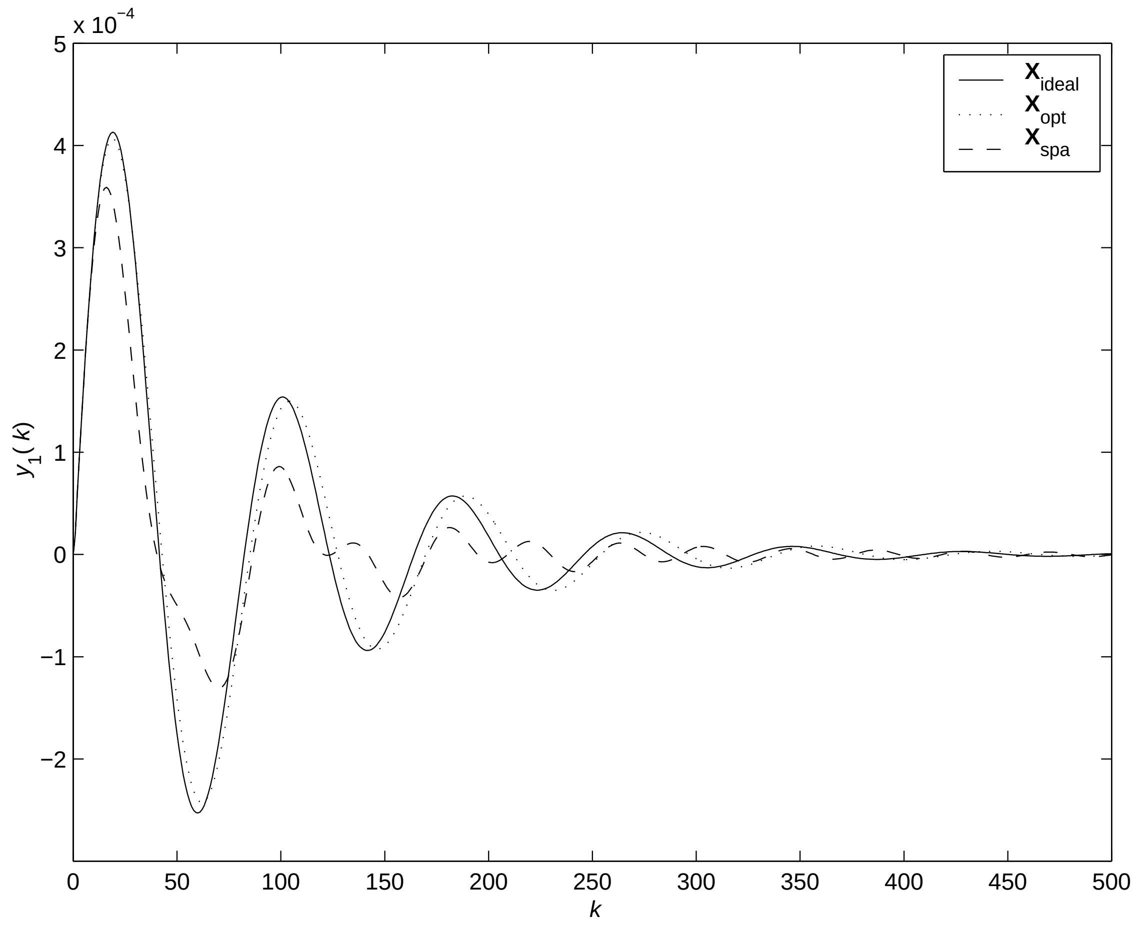

Figure 2: Comparison of first-input first-output unit impulse response of the infinite-precision controller implementation $\mathbf{X}_{\text{ideal}}$ with those of the 20-bit implemented controller realizations $\mathbf{X}_{\text{opt}}$ and $\mathbf{X}_{\text{spa}}$ for Example 2. The 20-bit implemented $\mathbf{X}_0$ is unstable and hence is not shown here.

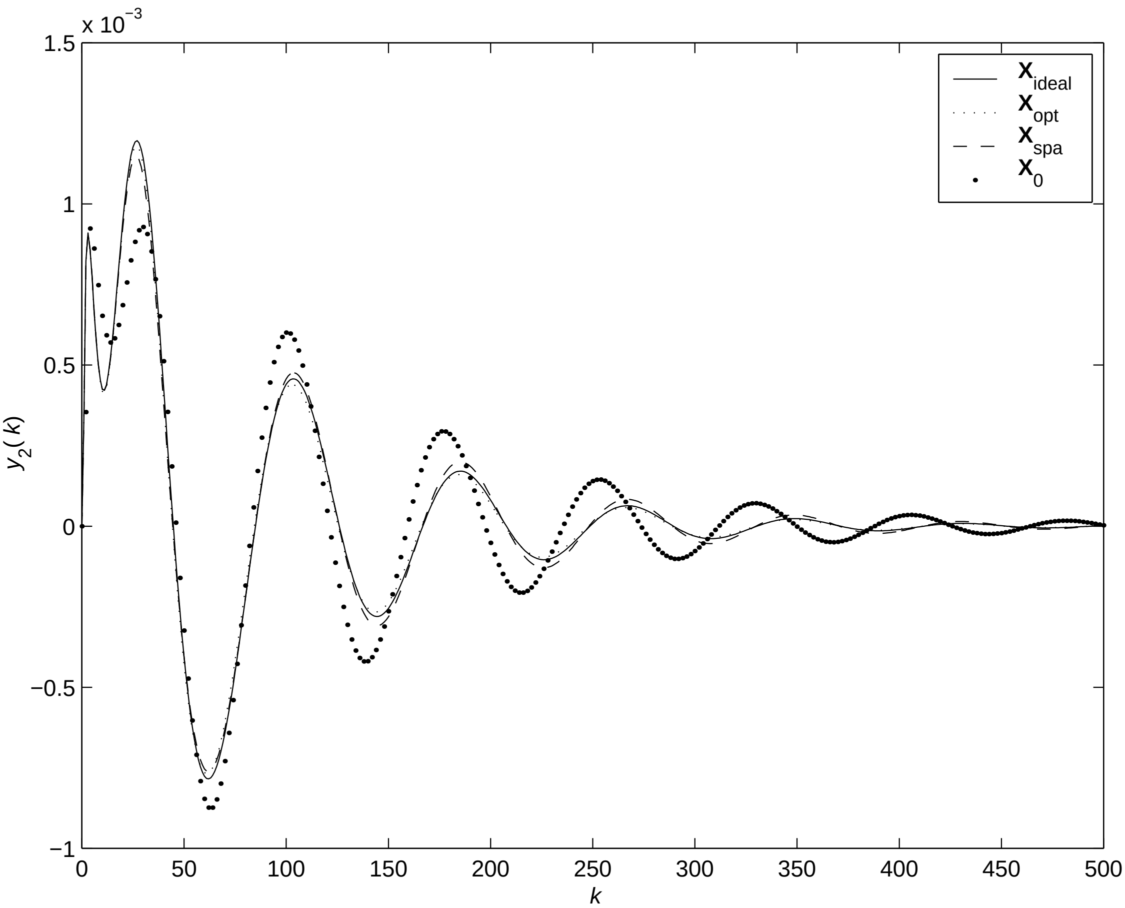

Figure 3: Comparison of second-input second-output unit impulse response of the infinite-precision controller implementation $\mathbf{X}_{\rm ideal}$ with those of the 24-bit implemented controller realizations $\mathbf{X}_0$, $\mathbf{X}_{\rm opt}$ and $\mathbf{X}_{\rm spa}$ for Example 2.