Proceedings of the 3rd World Congress on Intelligent Control and Automation June 28—July 2, 2000, Hefei, P.R. China

Digital Finite-Precision Controller Realizations with Sparseness Considerations[1]

Jun Wu
National Lab. of Industrial Control Tech.
Institute of Advanced Process Control
Zhejiang University
Hangzhou, 310027, P. R. China
jwu@iipc.zju.edu.cn

Sheng Chen
Dep. of Electronics and Computer Sci.
University of Southampton
Highfield, Southampton SO17 1BJ, UK
sqc@ecs.soton.ac.uk

Gang Li
The school of EEE
Nanyang Technological University
Singapore
egli@ntu.edu.sg

Jian Chu
National Lab. of Industrial Control Tech.
Institute of Advanced Process Control
Zhejiang University
Hangzhou, 310027, P. R. China
chuj@iipc.zju.edu.cn

Abstract–**In this paper, we present a study on the Finite Word Length (FWL) implementation of digital controller structures. The relevant FWL closed-loop stability related measures are investigated, and an algorithm is provided to search for the sparse controller realization that yields a computationally efficient structure with good FWL closed-loop stability performance. A numerical example is included to illustrate the proposed design procedure.**

1 Introduction

For reasons of speed, memory space and ease-of-programming, the use of fixed-point processors is more desired for many industrial and consumer applications. However, a designed stable closed-loop system may become unstable when the "infinite-precision" controller is implemented using a fixed-point processor due to the FWL (Finite Word Length) effect. It is well known that a linear digital controller can be implemented in different realizations and different controller realizations have different FWL closed-loop stability behavior. Many studies have addressed the problem of digital controller realizations with finite-precision considerations [1]-[6]. In particular, computationally tractable FWL closed-loop stability related measures have recently been derived, and the design procedures have been developed to search for optimal finite-precision controller realizations with maximum tolerance to FWL errors [5]. However, few study has investigated an important issue in FWL implementation, namely the sparseness consideration of controller realizations [4]. A controller realization that possesses many trivial parameters of 0, +1 and -1 is called a sparse realization. Sparse realizations are preferred in real-time control applications, as they are computationally more efficient and produce less FWL errors. This paper address the complex problem of finding sparse realizations with good FWL closed-loop stability performance.

[1]This work was supported by Zhejiang Provincial Natural Science Foundation of China under Grant 699085

2 Measure on stability and sparseness

Consider the discrete-time control system depicted in Fig. 1, where the discrete-time plant model $P(z)$ is assumed to be strictly causal and $C(z)$ denotes the digital controller. Let $(A_z, B_z, C_z, 0)$ be a state-space description of $P(z)$ with $A_z \in R^{m\times m}$, $B_z \in R^{m\times l}$ and $C_z \in R^{q\times m}$, and (A_c, B_c, C_c, D_c) be a state-space description of $C(z)$ with $A_c \in R^{n\times n}$, $B_c \in R^{n\times q}$, $C_c \in R^{l\times n}$ and $D_c \in R^{l\times q}$. Then the stability of the closed-loop control system depends on the poles of the closed-loop system matrix

$$\overline{A} = \begin{bmatrix} A_z + B_z D_c C_z & B_z C_c \\ B_c C_z & A_c \end{bmatrix} \quad (1)$$

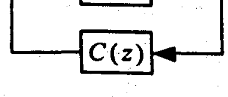

Fig. 1. Discrete-time control system

0-7803-5995-X/00/$10.00 ©2000 IEEE.

If $(A_c^0, B_c^0, C_c^0, D_c^0)$ is a state-space description of the digital controller $C(z)$, all the state-space descriptions of $C(z)$ form a set

$$\mathcal{S}_C \triangleq \{(A_c, B_c, C_c, D_c) : A_c = T^{-1}A_c^0 T, \\ B_c = T^{-1}B_c^0, C_c = C_c^0 T, D_c = D_c^0\} \quad (2)$$

where $T \in R^{n\times n}$ is any non-singular matrix, called a similarity transformation. Any $(A_c, B_c, C_c, D_c) \in \mathcal{S}_C$ is a *realization* of $C(z)$. Denote $N \triangleq (l+n)(q+n)$ and

$$X \triangleq \begin{bmatrix} D_c & C_c \\ B_c & A_c \end{bmatrix} = \begin{bmatrix} p_1 & p_{l+n+1} & \cdots & p_{N-l-n+1} \\ p_2 & p_{l+n+2} & \cdots & p_{N-l-n+2} \\ \vdots & \vdots & \cdots & \vdots \\ p_{l+n} & p_{2l+2n} & \cdots & p_N \end{bmatrix} \quad (3)$$

We will also refer to X as a realization of $C(z)$. From (1), we know that $\overline{A}$ is a function of X

$$\overline{A}(X) = \begin{bmatrix} A_z & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} B_z & 0 \\ 0 & I \end{bmatrix} X \begin{bmatrix} C_z & 0 \\ 0 & I \end{bmatrix} \triangleq M_0 + M_1 X M_2 \quad (4)$$

When the fixed-point format is used to implement the controller, X is perturbed into $X + \Delta X$ due to the FWL effect, where

$$\Delta X \triangleq \begin{bmatrix} \Delta p_1 & \Delta p_{l+n+1} & \cdots & \Delta p_{N-l-n+1} \\ \Delta p_2 & \Delta p_{l+n+2} & \cdots & \Delta p_{N-l-n+2} \\ \vdots & \vdots & \cdots & \vdots \\ \Delta p_{l+n} & \Delta p_{2l+2n} & \cdots & \Delta p_N \end{bmatrix} \quad (5)$$

and each element of ΔX is bounded by $\frac{\varepsilon}{2}$ such that

$$\mu(\Delta X) \triangleq \max_{i\in\{1,\cdots,N\}} |\Delta p_i| \le \frac{\varepsilon}{2} \quad (6)$$

For a fixed-point processor that uses B_f bits to implement the fractional part of a number, $\varepsilon = 2^{-B_f}$, and $\mu(\Delta X)$ is a norm of the FWL error ΔX. With the perturbation ΔX, a closed-loop pole $\lambda_i(\overline{A}(X))$ of the originally stable system is moved to $\lambda_i(\overline{A}(X+\Delta X))$, which may be outside the open unit disk and hence causes the closed-loop to become unstable.

Notice that the parameters 0, +1 and -1 are *trivial*, since they require no operations in the fixed-point implementation of X and do not cause any computation error at all. Thus $\Delta p_i = 0$ when $p_i = 0, +1$ or -1. Let us define the function

$$\delta(p) = \begin{cases} 0, & \text{if } p = 0, +1 \text{ or } -1 \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

To derive an FWL stability related measure for X, we first notice that when ΔX is small

$$\Delta\lambda_i \triangleq \lambda_i(\overline{A}(X+\Delta X)) - \lambda_i(\overline{A}(X)) \approx \sum_{j=1}^{N} \frac{\partial \lambda_i}{\partial p_j} \Delta p_j \delta(p_j), \ \forall i \in \{1, \cdots, m+n\} \quad (8)$$

It follows from the inequality

$$\left(\sum_{j=1}^{N_s} a_j\right)^2 \le N_s \sum_{j=1}^{N_s} a_j^2, \quad (9)$$

derived easily from Cauchy inequality, that

$$|\Delta\lambda_i| \le \sqrt{N_s \sum_{j=1}^{N} \left|\frac{\partial \lambda_i}{\partial p_j}\right|^2 |\Delta p_j|^2 \delta(p_j)} \le \mu(\Delta X) \sqrt{N_s \sum_{j=1}^{N} \left|\frac{\partial \lambda_i}{\partial p_j}\right|^2 \delta(p_j)}, \ \forall i \quad (10)$$

where N_s is the number of the non-trivial elements in X. Define

$$\mu_1(X) = \min_{i\in\{1,\cdots,m+n\}} \frac{1 - |\lambda_i(\overline{A}(X))|}{\sqrt{N_s \sum_{j=1}^{N} \delta(p_j) \left|\frac{\partial \lambda_i}{\partial p_j}\right|^2}} \quad (11)$$

If $\mu(\Delta X) < \mu_1(X)$, it follows from (10) and (11) that $|\Delta\lambda_i| < 1 - |\lambda_i(\overline{A}(X))|$. Therefore

$$|\lambda_i(\overline{A}(X+\Delta X))| \le |\Delta\lambda_i| + |\lambda_i(\overline{A}(X))| < 1 \quad (12)$$

which means that the closed-loop system remains stable under the FWL error ΔX. In other words, for a given controller realization X, the closed-loop system can tolerate those FWL perturbations ΔX whose norms $\mu(\Delta X)$ are less than $\mu_1(X)$. The larger $\mu_1(X)$ is, the bigger FWL error ΔX that the closed-loop system can tolerate. Hence $\mu_1(X)$ is a stability related measure describing the FWL closed-loop stability performance of a controller realization X. Furthermore, $\mu_1(X)$ is computationally tractable, as shown in the following theorem which was proved in [6].

Theorem 1 Assume that $\overline{A}(X) = M_0 + M_1 X M_2$ given in (4) is diagonalizable with $\{\lambda_i\} = \{\lambda_i(\overline{A}(X))\}$ as its eigenvalues. Let x_i be a right eigenvector of $\overline{A}(X)$ corresponding to the eigenvalue λ_i. Denote $M_x \triangleq [\, x_1 \ \cdots \ x_{m+n} \,]$ and $M_y \triangleq [\, y_1 \ \cdots \ y_{m+n} \,] = M_x^{-\mathcal{H}}$, where $\mathcal{H}$ represents the transpose and conjugate operation and y_i is called the reciprocal left eigenvector corresponding to λ_i. Then

$$\frac{\partial \lambda_i}{\partial X} = M_1^T y_i^* x_i^T M_2^T \quad (13)$$

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

3 Optimal controller realizations with sparse structures

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

$$v \triangleq \max_{X \in S_C} \mu_1(X) \tag{14}$$

However, we do not know how to solve (14) because $\mu_1(X)$ includes $\delta(p_j)$ is not a continuous function with respect to controller elements p_j. To get around this difficulty, we consider a lower bound of $\mu_1(X)$

$$\underline{\mu_1}(X) = \min_{i \in \{1,\cdots,m+n\}} \frac{1 - |\lambda_i(\overline{A}(X))|}{\sqrt{N \sum_{j=1}^{N} \left|\frac{\partial \lambda_i}{\partial p_j}\right|^2}} \tag{15}$$

Obviously, $\underline{\mu_1}(X) \leq \mu_1(X)$ and $\underline{\mu_1}(X)$ is a continuous function. It is relatively easy to optimize $\underline{\mu_1}(X)$. Let the "optimal" controller realization $X_{\rm opt}$ be the solution of the problem

$$\omega \triangleq \max_{X \in S_C} \underline{\mu_1}(X) \tag{16}$$

Notice that $X_{\rm opt}$ is generally not the optimal solution of the problem of (14) and may not have a sparse structure. However, it can easily be obtained by the following optimization procedure.

3.1 Optimization of $\underline{\mu_1}$

Assume that an initial controller realization is given as

$$X_0 = \begin{bmatrix} D_c^0 & C_c^0 \\ B_c^0 & A_c^0 \end{bmatrix} \tag{17}$$

From (2) and (4), we have

$$X = X(T) = \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} X_0 \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix} \tag{18}$$

and

$$\overline{A}(X) = \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} \overline{A}(X_0) \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix} \tag{19}$$

Obviously, $\overline{A}(X)$ has the same eigenvalues as $\overline{A}(X_0)$, denoted as $\{\lambda_i^0\}$. From (19), applying theorem 1 results in

$$\left.\frac{\partial \lambda_i}{\partial X}\right|_{X=X(T)} = \begin{bmatrix} I & 0 \\ 0 & T^T \end{bmatrix} \left.\frac{\partial \lambda_i}{\partial X}\right|_{X=X_0} \begin{bmatrix} I & 0 \\ 0 & T^{-T} \end{bmatrix} \tag{20}$$

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

$$\|M\|_F \triangleq \sqrt{\sum_{i=1}^{l+n} \sum_{j=1}^{q+n} m_{ij}^* m_{ij}} \tag{21}$$

Define the cost function

$$f(T) = \min_{i \in \{1,\cdots,m+n\}} \frac{1}{\sqrt{N} \left\| \begin{bmatrix} I & 0 \\ 0 & T^T \end{bmatrix} \Phi_i \begin{bmatrix} I & 0 \\ 0 & T^{-T} \end{bmatrix} \right\|_F} \tag{22}$$

where

$$\Phi_i \triangleq \frac{\left.\frac{\partial \lambda_i}{\partial X}\right|_{X=X_0}}{1 - |\lambda_i^0|} \tag{23}$$

Then the optimization problem (16) is equivalent to

$$\omega = \max_{\substack{T \in R^{n\times n} \\ \det(T) \neq 0}} f(T) \tag{24}$$

Furthermore, the optimal similarity transformation $T_{\rm opt}$ can be obtained by solving for the following unconstrained optimization problem

$$\omega = \max_{T \in R^{n\times n}} f(T) \tag{25}$$

with a consideration that $\det(T) = 0$ is very rare. The unconstrained optimization problem (25) can be solved, for example, using the simplex search algorithm. The corresponding controller realization is then given by $X_{\rm opt} = X(T_{\rm opt})$.

3.2 Stepwise transformation algorithm

As the optimal sparse realization that maximizes μ_1 is difficult to obtain, we will search for a *suboptimal* solution of (14). More precisely, we will search for a realization that is sparse with a large enough value of μ_1. Since $X_{\rm opt}$ maximizes $\underline{\mu_1}$ and $\underline{\mu_1}$ is a lower bound of μ_1, $X_{\rm opt}$ will produce a satisfactory value of μ_1, although it usually contains no trivial elements. We can make $X_{\rm opt}$ sparse by changing one non-trivial element of $X_{\rm opt}$ 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 $X_{\rm spa}$ with a satisfactory value of $\underline{\mu_1}$. Notice that, even though $\underline{\mu_1}(X_{\rm spa}) < \underline{\mu_1}(X_{\rm opt})$, it is possible that $\mu_1(X_{\rm spa}) > \mu_1(X_{\rm opt})$. In other words, $X_{\rm spa}$ may achieve better FWL stability performance than $X_{\rm opt}$. We now describe the detailed stepwise procedure for obtaining $X_{\rm spa}$.

Step 1: Set τ to a very small positive real number (e.g. 10^{-5}). The transformation matrix T is initially set to $T_{\rm opt}$ so that $X(T) = X_{\rm opt}$.

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

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

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

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

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

iv) $\|S\|_F = 1$.

If S does not exist, $T_{\rm spa} = T$ and terminate the algorithm.

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

The key of the above algorithm is **step 3**, which guarantees that $X_{spa} = X(T_{spa})$ contains many trivial elements and has good performance as measured by μ_1. We now discuss how to obtain S. First, denote $Vec(M)$ the vector containing the columns of the matrix M stacked in column order. With a very small τ, condition i) means

$$\left(Vec\left(\frac{d\mu_1}{dT}\right)\right)^T Vec(S) = 0 \quad (26)$$

Condition ii) means

$$\begin{cases} \left(Vec\left(\frac{d\eta_1}{dT}\right)\right)^T Vec(S) = 0 \\ \vdots \\ \left(Vec\left(\frac{d\eta_r}{dT}\right)\right)^T Vec(S) = 0 \end{cases} \quad (27)$$

Denote the matrix

$$E \triangleq \begin{bmatrix} \left(Vec\left(\frac{d\mu_1}{dT}\right)\right)^T \\ \left(Vec\left(\frac{d\eta_1}{dT}\right)\right)^T \\ \vdots \\ \left(Vec\left(\frac{d\eta_r}{dT}\right)\right)^T \end{bmatrix} \in R^{(r+1)\times n^2} \quad (28)$$

$Vec(S)$ must belong to the null space $\mathcal{N}(E)$ of E. If $\mathcal{N}(E)$ is empty, $Vec(S)$ does not exist and the algorithm is terminated. If $\mathcal{N}(E)$ is not empty, it must have basis $\{b_1, \cdots, b_t\}$, assuming that the dimension of $\mathcal{N}(E)$ is t. Condition iii) requires moving ξ closer to its desired value (0, +1 or -1) as fast as possible, and we should choose $Vec(S)$ as the orthogonal projection of $Vec\left(\frac{d\xi}{dT}\right)$ onto $\mathcal{N}(E)$. Noting condition iv), we can compute $Vec(S)$ as follows

$$a_i = b_i^T Vec\left(\frac{d\xi}{dT}\right) \in R, \ \forall i \in \{1, \cdots, t\} \quad (29)$$

$$v = \sum_{i=1}^{t} a_i b_i \in R^{n^2} \quad (30)$$

$$Vec(S) = \pm \frac{v}{\sqrt{v^T v}} \in R^{n^2} \quad (31)$$

The sign in (31) 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 $\frac{d\mu_1}{dT}, \frac{d\xi}{dT}, \frac{d\eta_1}{dT}, \cdots, \frac{d\eta_r}{dT}$ are needed. Denote e_i as the ith elementary vector with the ith unit element and the rest of the elements being all zero. For matrix M, denote $\mathcal{D}(M) = \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix}$. We provide the following lemmas without giving proofs.

Lemma 1 For $H \in R^{m\times n}$ with elements h_{jk}, $J \in C^{n\times q}$ and $G = HJ$ with elements g_{lr},

$$\frac{\partial g_{lr}}{\partial h_{jk}} = e_l^T e_j e_k^T J e_r \quad (32)$$

$$\frac{dg_{lr}}{dH} = \mathcal{D}(e_l^T) \begin{bmatrix} e_1 e_1^T & \cdots & e_1 e_n^T \\ \vdots & \vdots & \vdots \\ e_m e_1^T & \cdots & e_m e_n^T \end{bmatrix} \mathcal{D}(J e_r) \quad (33)$$

Lemma 2 For $H \in R^{n\times q}$ with elements h_{jk}, $J \in C^{m\times n}$ and $G = JH$ with elements g_{lr},

$$\frac{\partial g_{lr}}{\partial h_{jk}} = e_l^T J e_j e_k^T e_r \quad (34)$$

$$\frac{dg_{lr}}{dH} = \mathcal{D}(e_l^T J) \begin{bmatrix} e_1 e_1^T & \cdots & e_1 e_q^T \\ \vdots & \vdots & \vdots \\ e_n e_1^T & \cdots & e_n e_q^T \end{bmatrix} \mathcal{D}(e_r) \quad (35)$$

Lemma 3 For nonsingular $H \in R^{m\times m}$ with elements h_{jk} and its inverse H^{-1} with elements $\hat{h}_{lr}$,

$$\frac{\partial \hat{h}_{lr}}{\partial h_{jk}} = -e_l^T H^{-1} e_j e_k^T H^{-1} e_r \quad (36)$$

$$\frac{d\hat{h}_{lr}}{dH} = -\mathcal{D}(e_l^T H^{-1}) \begin{bmatrix} e_1 e_1^T & \cdots & e_1 e_m^T \\ \vdots & \vdots & \vdots \\ e_m e_1^T & \cdots & e_m e_m^T \end{bmatrix} \mathcal{D}(H^{-1} e_r) \quad (37)$$

Now let the elements of U^{-1} be $\hat{u}_{jk}$ for $j \in \{1, \cdots, m+n\}$ and $k \in \{1, \cdots, m+n\}$, where

$$U = \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix} \quad (38)$$

For any element x_{lr} in $X = U^{-1} X_0 U$,

$$\frac{dx_{lr}}{dU} = \sum_{j=1}^{m+n} \sum_{k=1}^{m+n} \frac{\partial x_{lr}}{\partial \hat{u}_{jk}} \frac{d\hat{u}_{jk}}{dU} + \frac{\partial x_{lr}}{\partial U} \quad (39)$$

can be calculated from lemma 1 to 3. Considering

$$\frac{dx_{lr}}{dT} = [0 \quad I] \frac{dx_{lr}}{dU} \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (40)$$

we can calculate $\frac{\partial \xi}{\partial T}, \frac{\partial \eta_1}{\partial T}, \cdots, \frac{\partial \eta_r}{\partial T}$.

Denote

$$i_0 = \arg \min_{i\in\{1,\cdots,m+n\}} \frac{1}{\sqrt{N}\left\|\begin{bmatrix} I & 0 \\ 0 & T^{\mathcal{T}} \end{bmatrix} \Phi_i \begin{bmatrix} I & 0 \\ 0 & T^{-\mathcal{T}} \end{bmatrix}\right\|_F} \quad (41)$$

and

$$W = \begin{bmatrix} I & 0 \\ 0 & T^{\mathcal{T}} \end{bmatrix} \Phi_{i_0} \begin{bmatrix} I & 0 \\ 0 & T^{-\mathcal{T}} \end{bmatrix} = U^{\mathcal{T}} \Phi_{i_0} U^{-\mathcal{T}} \quad (42)$$

Let w_{lr} be the elements of W, and $\tilde{u}_{jk}$ for $j \in \{1,\cdots,m+n\}$ and $k \in \{1,\cdots,m+n\}$ be the elements of $U^{-\mathcal{T}}$. Similar to the derivation of $\frac{dx_{lr}}{dT}$, we have

$$\frac{dw_{lr}}{dT} = [0 \quad I] \left(\sum_{j=1}^{m+n} \sum_{k=1}^{m+n} \frac{\partial w_{lr}}{\partial \tilde{u}_{jk}} \frac{d\tilde{u}_{jk}}{dU^{\mathcal{T}}} + \frac{\partial w_{lr}}{\partial U^{\mathcal{T}}} \right)^{\mathcal{T}} \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (43)$$

based on lemmas 1 to 3. Since

$$\underline{\mu_1} = \frac{1}{\sqrt{N}\sqrt{\sum_{l=1}^{m+n}\sum_{r=1}^{m+n} w_{lr}^* w_{lr}}} \quad (44)$$

we can calculate

$$\frac{d\underline{\mu_1}}{dT} = -\frac{1}{\sqrt{N}\,\|W\|_F^3} \sum_{l=1}^{m+n} \sum_{r=1}^{m+n} w_{lr}^* \frac{dw_{lr}}{dT}. \quad (45)$$

4 An illustrative example

The discrete-time plant model $P(z)$ is given by

$$A_z = 10^{-5} \times \begin{bmatrix} 1.00e5 & 1.94 & 5.93 & -6.23 \\ -4.96e-2 & 2.36e3 & 2.37 & 2.37 \\ -1.52e2 & 2.37e3 & 2.38 & 2.39 \\ 1.59e2 & 2.37e3 & 2.39 & 2.37 \end{bmatrix}$$

$$B_z = \begin{bmatrix} 3.05e-3 \\ -1.24e-2 \\ -1.24e-2 \\ -8.87e-2 \end{bmatrix} \qquad C_z = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

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

$$A_c^0 = \begin{bmatrix} 0 & 0 & 0 & -3.31e-1 \\ 1 & 0 & 0 & 1.99 \\ 0 & 1 & 0 & -3.98 \\ 0 & 0 & 1 & 3.33 \end{bmatrix},$$

$$B_c^0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, C_c^0 = \begin{bmatrix} -1.61e-3 \\ -1.60e-3 \\ -1.59e-3 \\ -1.57e-3 \end{bmatrix}^{\mathcal{T}}, D_c^0 = -8.08e-4$$

The optimization problem (25) is constructed, and the simplex search algorithm obtains the solution T_{opt} and the corresponding optimal realization X_{opt} that maximizes $\underline{\mu_1}$. The stepwise transformation algorithm is then applied to make X_{opt} sparse and obtain T_{spa} and X_{spa}.

Table 1: Comparison for different realizations

Realization	μ_1	$\underline{\mu_1}$	N_s
X_0	4.3890×10^{-12}	2.5854×10^{-12}	9
X_{opt}	6.6854×10^{-5}	6.6854×10^{-5}	25
X_{spa}	8.4007×10^{-5}	3.5478×10^{-5}	16

Table 1 compares the three different realizations X_0, X_{opt} and X_{spa} of the example, respectively. Obviously, the sparse realization X_{spa} has the best FWL stability performance.

5 Conclusions

Based on the FWL closed-loop stability related measure with sparseness considerations, we have addressed an optimial realization problem and given a solution strategy. A practical stepwise procedure has also been presented to obtain sparse controller realizations with satisfactory FWL closed-loop stability performance.

References

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

[2] M. Gevers and G. Li, *Parameterizations in Control, Estimation and Filtering Problems: Accuracy Aspects*, Springer Verlag, London; 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, 1994, pp.2476–2481.

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

[5] R.H. Istepanian, L. Gang, 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, 1998, pp.472–478.

[6] 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, 1999, pp.472–478.