Jointly Optimized Error-Feedback and Realization for Roundoff Noise Minimization in State-Space Digital Filters

Wu-Sheng Lu[1] and Takao Hinamoto[2]

1. Dept. of Electrical and Computer Engineering, University of Victoria,
Victoria, BC, Canada, V8W 3P6.
Tel.: 250 721 8692, Fax: 250 721 6052, E-mail: wslu@ece.uvic.ca
2. Graduate School of Engineering, Hiroshima University,
Higashi-Hiroshima, 739-8527, Japan.
Tel.: +81 824 24 7672, Fax: +81 824 22 7195, E-mail: hinamoto@hiroshima-u.ac.jp

Abstract

Roundoff noise (RN) is known to exist in digital filters and systems under finite-precision operations and can become a critical factor for severe performance degradation in IIR filters and systems. In the literature, two classes of methods are available for RN reduction or minimization — one uses state-space coordinate transformation, the other uses error feedback/feed-forward of state variables. In this paper, we propose a method for the joint optimization of error feedback/feed-forward and state-space realization. It is shown that the problem at hand can be solved in an unconstrained optimization setting. With a closed-form formula for gradient evaluation and an efficient quasi-Newton solver, the unconstrained minimization problem can be solved efficiently. With the infinite-precision solution as a reference point, we then move on to derive a semidefinite programming (SDP) relaxation method for an approximate solution of optimal error-feedback matrix with sum-of-power-of-two entries under a given state-space realization. Simulations are presented to illustrate the proposed algorithms and demonstrate the performance of optimized systems.

Keywords: roundoff noise in digital filters, state-space transformation, error-feedback matrix, unconstrained optimization.

EDICS: 2-QUAN (Quantization Effects and Roundoff Analysis)

I. INTRODUCTION

Since the work of [1][2], it has been well understood that the roundoff noise (RN) of infinite-impulse-response (IIR) digital filters under fixed-point arithmetic operations can be substantially reduced by using adequately chosen state-space realizations [3]–[5]. It has also been known that RN reduction can be accomplished by feeding the quantization error of state variables back to the filter's input through a memoryless (constant) *error-feedback* matrix without affecting the filter's input-output characteristics [6]–[8]. In addition, error feed-forward helps further reduce the RN [6]. Alternatively, the RN reduction problem has also been studied in various different settings [9]–[18]. Naturally, the success of these techniques leads one to the consideration of a *joint optimization* of error feedback and state-space realization so as to achieve greater reduction in RN. It turns out that obtaining such a jointly optimized error feedback and realization requires the solution of a sophisticated constrained nonlinear minimization problem (see Sec. III). In [8], an iterative algorithm for the above optimization problem was proposed for IIR filters with *scalar* error-feedback matrices, but it appears to be inherently difficult to extend the algorithm to the cases where the error-feedback matrices are diagonal or general.

The objectives of this paper are two-fold. First, the problem of joint optimization of error-feedback/feed-forward and state-space realization for RN minimization is investigated in a general nonlinear optimization framework where the error-feedback matrix can be a scalar, diagonal, or general matrix. Using linear-algebraic techniques, we convert the constrained optimization problem at hand into an unconstrained problem which can be solved using powerful quasi-Newton algorithms [19]. A nice feature of employing a general optimization setting for our problem is that both the realization optimization [1][2] and the error-feedback-matrix optimization [8] become special cases of the proposed formulation that explains why digital filters with jointly optimal error feedback and realization always outperform previously reported systems. The second objective of the paper is to present a solution to the optimal discretization for the error-feedback matrix obtained by the above-mentioned method. Specifically, we are concerned with IIR digital filters whose (infinite precision) jointly optimized error-feedback matrix and state-space realization have been determined and the error-feedback matrix is required to be implemented using sum-of-power-of-two (SP2) entries, each with fixed number of bits. Because of the *discrete* nature of the problem, its optimal solution is essentially a combinatorial opti-

mization problem with exponential-time computational complexity. Using the infinite-precision solution obtained, the problem at hand is formulated as a $(-1,\ 1)$-quadratic programming problem and a semidefinite programming (SDP) relaxation [20] is applied to obtain an approximate solution of the problem. This approximate solution is of interest as it can be calculated using efficient interior-point SDP solvers [21][22] of polynomial-time complexity and, as demonstrated by our computer simulations, it is often near optimal.

The paper is organized as follows. Sec. II gives background materials including review of state-space structure of IIR digital filters with error feedback and feed-forward and basic elements of SDP. Sec. III describes a general optimization formulation for the problem at hand and presents an unconstrained minimization based solution method. Sec. IV presents a $(-1,\ 1)$-quadratic programming formulation for the optimization of discrete error-feedback matrix and then an SDP-relaxation based approximate solution, and computer simulation results are presented to illustrate the proposed algorithms as well as demonstrate system performance as compared to previously reported results.

In the rest of the paper, boldfaced characters denote matrices and vectors; $\boldsymbol{I}$ denotes the identity matrix of proper dimension; $\boldsymbol{K}^{1/2}$ denotes the symmetric square root of positive definite matrix $\boldsymbol{K}$; $\boldsymbol{A}^T$, $\boldsymbol{A}^*$, tr$(\boldsymbol{A})$ and $(\boldsymbol{A})_{ii}$ denotes the transpose, conjugate-transpose, trace, and the ith diagonal element of $\boldsymbol{A}$, respectively; and $\|\boldsymbol{v}\|$ denotes the standard Euclidean norm of vector $\boldsymbol{v}$.

II. Preliminaries

A. State-Space Digital Filters with Error Feed-Forward and Error Feedback [6]

Let $(\boldsymbol{A},\ \boldsymbol{b},\ \boldsymbol{c},\ d)_n$ be a minimal state-space realization of a stable IIR digital filter of order n. This realization can be expressed as

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \tag{1a}$$

$$y(k) = \boldsymbol{c}\boldsymbol{x}(k) + du(k) \tag{1b}$$

where $\boldsymbol{A} \in \mathcal{R}^{n\times n}$, $\boldsymbol{b} \in \mathcal{R}^{n\times 1}$, $\boldsymbol{c} \in \mathcal{R}^{1\times n}$, and $d \in \mathcal{R}$. Now assuming that (i) the filter is implemented subject to finite-word-length (FWL) constraint and quantization takes place before matrix-vector multiplications, and (ii) error-feedback and error-feed-forward for state variables

are used for the sake of RN reduction, then the filter's model becomes [8]

$$\tilde{\boldsymbol{x}}(k+1) = \boldsymbol{A}Q[\tilde{\boldsymbol{x}}(k)] + \boldsymbol{b}u(k) + \boldsymbol{D}\boldsymbol{e}(k) \tag{2a}$$

$$\tilde{y}(k) = \boldsymbol{c}Q[\tilde{\boldsymbol{x}}(k)] + du(k) + \boldsymbol{h}\boldsymbol{e}(k) \tag{2b}$$

where $Q[\cdot]$ denotes the quantizer that rounds the fraction of each input component to a b-bit number, $\boldsymbol{e}(k)$ is the quantization error defined by

$$\boldsymbol{e}(k) = \tilde{\boldsymbol{x}}(k) - Q[\tilde{\boldsymbol{x}}(k)]$$

and $\boldsymbol{D}$ and $\boldsymbol{h}$ are referred to error-feedback matrix and error-feed-forward vector, respectively. Fig. 1 shows a block diagram of the state-space filter described by (2).

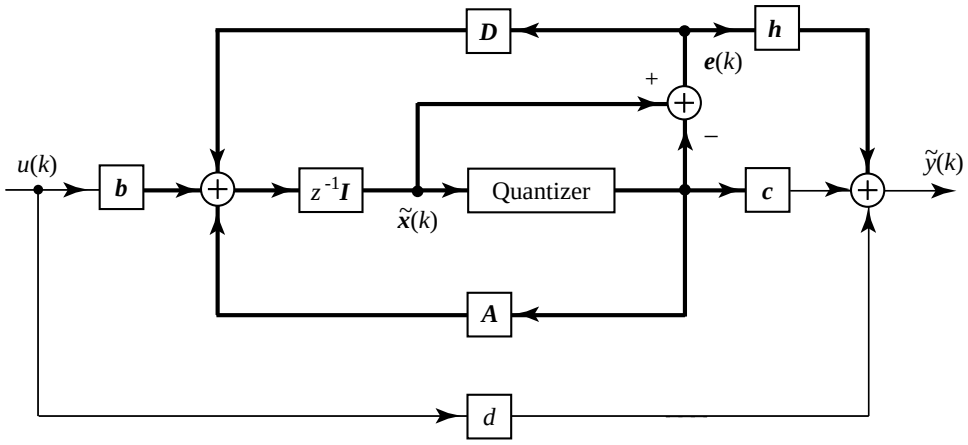

Fig. 1. Error feedback and error feed-forward in a state-space digital filter.

From (1) and (2), the roundoff noise for the filter can be modeled as

$$\Delta\boldsymbol{x}(k+1) = \boldsymbol{A}\Delta\boldsymbol{x}(k) + (\boldsymbol{A} - \boldsymbol{D})\boldsymbol{e}(k) \tag{3a}$$

$$\Delta y(k) = \boldsymbol{c}\Delta\boldsymbol{x}(k) + (\boldsymbol{c} - \boldsymbol{h})\boldsymbol{e}(k) \tag{3b}$$

where $\Delta\boldsymbol{x}(k) = \boldsymbol{x}(k) - \tilde{\boldsymbol{x}}(k)$ and $\Delta y(k) = y(k) - \tilde{y}(k)$. In the frequency domain, the noise process is modeled by

$$\Delta Y(z) = \boldsymbol{G}_{D,h}(z)\boldsymbol{E}(z) \tag{4a}$$

$$\boldsymbol{G}_{D,h}(z) = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{A})^{-1}(\boldsymbol{A} - \boldsymbol{D}) + \boldsymbol{c} - \boldsymbol{h} \tag{4b}$$

where $\Delta Y(z)$ and $\boldsymbol{E}(z)$ are the z-transforms of $\Delta y(k)$ and $\boldsymbol{e}(k)$, respectively, and $\boldsymbol{G}_{D,h}(z)$ denotes the transfer function from the quantization error to output roundoff noise. Therefore, a

noise gain due to quantization error can be defined as

$$I(\boldsymbol{D}) = \text{tr}(\boldsymbol{W}_{D,h}) \tag{5a}$$

where

$$\boldsymbol{W}_{D,h} = \frac{1}{2\pi j} \oint\limits_{|z|=1} \boldsymbol{G}^*_{D,h}(z)\boldsymbol{G}_{D,h}(z)\frac{dz}{z} \tag{5b}$$

It is known that the matrix $\boldsymbol{W}_{D,h}$ in (5b) can be expressed as [8]

$$\boldsymbol{W}_{D,h} = (\boldsymbol{A}-\boldsymbol{D})^T\boldsymbol{W}_o(\boldsymbol{A}-\boldsymbol{D}) + (\boldsymbol{c}-\boldsymbol{h})^T(\boldsymbol{c}-\boldsymbol{h}) \tag{6}$$

where $\boldsymbol{W}_o$ is the observability Gramian of the filter and can be computed by solving the Lyapunov equation [23]

$$\boldsymbol{W}_o - \boldsymbol{A}^T\boldsymbol{W}_o\boldsymbol{A} = \boldsymbol{c}^T\boldsymbol{c} \tag{7}$$

B. Semidefinite Programming

Semidefinite programming (SDP) is concerned with a class of constrained optimization problems where a linear objective function is minimized subject to matrix constraints which affinely depend on the variable vector. Typically an SDP problem can be formulated as

$$\text{minimize} \qquad \boldsymbol{c}^T\boldsymbol{x} \tag{8a}$$

$$\text{subject to:} \qquad \boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + \sum_{i=1}^{r} x_i\boldsymbol{F}_i \succeq \boldsymbol{0} \tag{8b}$$

where matrices $\boldsymbol{F}_i$ for $0 \leq i \leq r$ are symmetric and $\succeq \boldsymbol{0}$ means positive semidefinite. Since linear function $\boldsymbol{c}^T\boldsymbol{x}$ is always convex and the feasible region defined by the linear matrix inequality (LMI) in (8b) is convex, SDP forms an important subclass of *convex programming* problems that includes both linear and convex quadratic programming as special cases. Efficient polynomial-time interior-point algorithms have been extended to SDP [24][25] and software implementations of the algorithms are available, including the LMI Control Toolbox [21], SeDuMi [22], and SDPT3 [26].

III. Joint Optimization of Error-Feedback and Realization: The Infinite Precision Case

This section presents a solution of the problem at hand, where the entries of error-feedback matrix and error-feed-forward vector are assumed to have infinite precision. In the context

of numerical optimization, this simply means that the entries of matrix $\boldsymbol{D}$ can be treated as continuous variables.

A. An Optimization Formulation

As is well-known [23], the state-space realizations that are equivalent to a particular realization of a given digital filter, say $(\boldsymbol{A},\ \boldsymbol{b},\ \boldsymbol{c},\ d)_n$, are characterized by $(\overline{\boldsymbol{A}},\ \overline{\boldsymbol{b}},\ \overline{\boldsymbol{c}},\ \overline{d})_n = (\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T},\ \boldsymbol{T}^{-1}\ \boldsymbol{b},\ \boldsymbol{c}\boldsymbol{T},\ d)_n$ where $\boldsymbol{T} \in \mathcal{R}^{n\times n}$ is a nonsingular coordinate transformation matrix. Under a transformation $\boldsymbol{T}$, the noise gain defined in (5) becomes

$$I(\boldsymbol{D}, \boldsymbol{h}, \boldsymbol{T}) = \text{tr}(\overline{\boldsymbol{W}}_{D,h}) \tag{9a}$$

where

$$\overline{\boldsymbol{W}}_{D,h} = (\overline{\boldsymbol{A}} - \boldsymbol{D})^T \overline{\boldsymbol{W}}_o (\overline{\boldsymbol{A}} - \boldsymbol{D}) + (\overline{\boldsymbol{c}} - \boldsymbol{h})^T (\overline{\boldsymbol{c}} - \boldsymbol{h}) \tag{9b}$$

$$\overline{\boldsymbol{W}}_o = \boldsymbol{T}^T \boldsymbol{W}_o \boldsymbol{T} \tag{9c}$$

A basic constraint imposed on RN minimization is the l_2-norm dynamic range of the state variables [1][2]. Under a coordinate transformation, this constraint can be expressed as

$$(\overline{\boldsymbol{K}}_c)_{ii} = 1 \qquad \text{for } 1 \le i \le n \tag{10a}$$

where

$$\overline{\boldsymbol{K}}_c = \boldsymbol{T}^{-1} \boldsymbol{K}_c \boldsymbol{T}^{-T} \tag{10b}$$

and $\boldsymbol{K}_c$ is the controllability Gramian of the original realization and can be computed by solving the Lyapunov equation [23]

$$\boldsymbol{K}_c - \boldsymbol{A}\boldsymbol{K}_c\boldsymbol{A}^T = \boldsymbol{b}\boldsymbol{b}^T \tag{11}$$

The constrained optimization problem for the minimization of the noise gain subject to l_2-norm dynamic range constraints can now be described as

$$\underset{\boldsymbol{D},\ \boldsymbol{h},\ \boldsymbol{T}}{\text{minimize}} \quad J(\boldsymbol{D}, \boldsymbol{h}, \boldsymbol{T}) = \text{tr}[\overline{\boldsymbol{W}}_{D,h}] \tag{12a}$$

$$\text{subject to:} \quad (\boldsymbol{T}^{-1}\boldsymbol{K}_c\boldsymbol{T}^{-T})_{ii} = 1 \qquad \text{for } 1 \le i \le n \tag{12b}$$

where $\overline{\boldsymbol{W}}_{D,h}$ and $\overline{\boldsymbol{W}}_o$ are given in (9).

The problem formulation in (12) is rather general. As a matter of fact, it includes the following two special cases: If error feedback and error feed-forward are not used, then $\boldsymbol{D}$ and $\boldsymbol{h}$ are set

to null, which in conjunction with (9b) and (7) implies that

$$J(\mathbf{0},\mathbf{0},\boldsymbol{T}) = \text{tr}(\overline{\boldsymbol{W}}_o) \tag{13}$$

Several methods of minimizing $J(\mathbf{0},\mathbf{0},\boldsymbol{T})$ in (13) subject to (12b) were investigated in [1][2]. Another special case of (12) is to minimize the objective function $J(\boldsymbol{D},\boldsymbol{h},\boldsymbol{T})$ for a fixed $\boldsymbol{T}$ (i.e., for a given state-space realization). This problem has been addressed in [8]. Consequently, the solution of the general optimization problem in (12) that finds the jointly optimized $\boldsymbol{D},\boldsymbol{h}$ and $\boldsymbol{T}$ is expected to be superior to the solutions obtained from these two special cases.

B. An Equivalent Unconstrained Problem

Since the IIR filter at hand is assumed to be stable, controllable and observable, the controllability matrix $\boldsymbol{K}_c$ is positive definite [23]. Let $\boldsymbol{K}_c^{1/2}$ denote the symmetric square root of $\boldsymbol{K}_c$, i.e., $\boldsymbol{K}_c^{1/2}$ is a symmetric matrix satisfying $\boldsymbol{K}_c^{1/2}\boldsymbol{K}_c^{1/2} = \boldsymbol{K}_c$, then $\boldsymbol{K}_c^{1/2}$ is also positive definite and we can define

$$\hat{\boldsymbol{T}} = \boldsymbol{T}^T\boldsymbol{K}_c^{-1/2} \tag{14}$$

which implies that $\boldsymbol{T}^{-1} = \hat{\boldsymbol{T}}^{-T}\boldsymbol{K}_c^{-1/2}$ and the constraints in (12b) become

$$(\hat{\boldsymbol{T}}^{-T}\hat{\boldsymbol{T}}^{-1})_{ii} = 1 \qquad \text{for } 1 \le i \le n \tag{15}$$

The constraints in (15) simply mean that each column in $\hat{\boldsymbol{T}}^{-1}$ must be a unity vector. This can be satisfied if $\hat{\boldsymbol{T}}^{-1}$ assumes the form

$$\hat{\boldsymbol{T}}^{-1} = \left[\frac{\boldsymbol{t}_1}{\|\boldsymbol{t}_1\|} \;\; \frac{\boldsymbol{t}_2}{\|\boldsymbol{t}_2\|} \;\; \cdots \;\; \frac{\boldsymbol{t}_n}{\|\boldsymbol{t}_n\|}\right] \tag{16}$$

with $\boldsymbol{t}_i \in \mathcal{R}^{n\times 1}$. To complete our problem conversion, we use (9) and (14) to re-write the objective function in (12a) in terms of $\boldsymbol{D},\boldsymbol{h}$ and $\hat{\boldsymbol{T}}$ as

$$J(\boldsymbol{D},\boldsymbol{h},\hat{\boldsymbol{T}}) = J_1 + J_2 \tag{17a}$$

where

$$J_1 = \text{tr}[\hat{\boldsymbol{T}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{T}}^T\boldsymbol{D}\hat{\boldsymbol{T}}^{-T})^T\hat{\boldsymbol{W}}_o(\hat{\boldsymbol{A}} - \hat{\boldsymbol{T}}^T\boldsymbol{D}\hat{\boldsymbol{T}}^{-T})\hat{\boldsymbol{T}}^T] \tag{17b}$$

$$J_2 = \text{tr}[\hat{\boldsymbol{T}}(\hat{\boldsymbol{c}} - \boldsymbol{h}\hat{\boldsymbol{T}}^{-T})^T(\hat{\boldsymbol{c}} - \boldsymbol{h}\hat{\boldsymbol{T}}^{-T})\hat{\boldsymbol{T}}^T] \tag{17c}$$

$$\hat{\boldsymbol{A}} = \boldsymbol{K}_c^{-1/2}\boldsymbol{A}\boldsymbol{K}_c^{1/2} \tag{17d}$$

$$\hat{\boldsymbol{c}} = \boldsymbol{c}\boldsymbol{K}_c^{1/2} \tag{17e}$$

$$\hat{\boldsymbol{W}}_o = \boldsymbol{K}_c^{1/2}\boldsymbol{W}_o\boldsymbol{K}_c^{1/2} \tag{17f}$$

Notice that for any $\boldsymbol{D}, \boldsymbol{h}$, and $\hat{\boldsymbol{T}}$, both J_1 and J_2 are nonnegative, hence we always have $J(\boldsymbol{D}, \boldsymbol{h}, \hat{\boldsymbol{T}}) \geq 0$. With $\hat{\boldsymbol{T}}^{-1}$ assuming the form in (16), the constraints on dynamic range are eliminated and the joint optimization problem in (12) can be stated as

$$\min_{\boldsymbol{D},\, \boldsymbol{h},\, \hat{\boldsymbol{T}}} J(\boldsymbol{D}, \boldsymbol{h}, \hat{\boldsymbol{T}}) = J_1 + J_2 \tag{18}$$

In what follows we examine the problem in (18) for several cases of particular interest, where the entries of $\boldsymbol{D}$ and $\boldsymbol{h}$ assume different degrees of freedom.

Case 1: Both $\boldsymbol{D}$ and $\boldsymbol{h}$ have full degree of freedom. In this case, the optimal choices of $\boldsymbol{D}$ and $\boldsymbol{h}$ are

$$\boldsymbol{D} = \overline{\boldsymbol{A}}, \quad \boldsymbol{h} = \overline{\boldsymbol{c}} \tag{19}$$

which leads to the absolute minimum $J(\boldsymbol{D}, \boldsymbol{h}, \hat{\boldsymbol{T}}) = 0$. Although this solution is hardly of use in practical implementation, it serves as a reference point when an optimal integer-valued (or sum-of-power-of-two (SP2) valued) $\boldsymbol{D}$ and $\boldsymbol{h}$ are sought, see Sec. IV.

Case 2: $\boldsymbol{D}$ is a scalar matrix, i.e., $\boldsymbol{D} = \alpha\boldsymbol{I}$, and $\boldsymbol{h}$ is a general vector. In this case, the choice of $\boldsymbol{h} = \overline{\boldsymbol{c}}$ makes J_2 vanish and the objective function in (18) becomes

$$J(\alpha\boldsymbol{I}, \overline{\boldsymbol{c}}, \hat{\boldsymbol{T}}) = \text{tr}[\hat{\boldsymbol{T}}(\hat{\boldsymbol{A}} - \alpha\boldsymbol{I})^T\hat{\boldsymbol{W}}_o(\hat{\boldsymbol{A}} - \alpha\boldsymbol{I})\hat{\boldsymbol{T}}^T] \tag{20}$$

Hence the variables in the minimization are $\{\boldsymbol{t}_i,\ 1 \leq i \leq n\}$ plus a scalar α.

Case 3: $\boldsymbol{D}$ contains certain number of zero entries in fixed places but is free elsewhere, and $\boldsymbol{h}$ is a general vector. This obviously includes the case where $\boldsymbol{D}$ is a diagonal matrix. With the choice $\boldsymbol{h} = \overline{\boldsymbol{c}}$, the objective function in (18) assumes the form

$$J(\boldsymbol{D}, \overline{\boldsymbol{c}}, \hat{\boldsymbol{T}}) = \text{tr}[\hat{\boldsymbol{T}}(\hat{\boldsymbol{A}} - \hat{\boldsymbol{T}}^T\boldsymbol{D}\hat{\boldsymbol{T}}^{-T})^T\hat{\boldsymbol{W}}_o(\hat{\boldsymbol{A}} - \hat{\boldsymbol{T}}^T\boldsymbol{D}\hat{\boldsymbol{T}}^{-T})\hat{\boldsymbol{T}}^T] \tag{21}$$

The variables involved in the minimization are $\{\boldsymbol{t}_i,\ 1 \leq i \leq n\}$ and the nonzero entries in $\boldsymbol{D}$.

Again in both Case 2 and Case 3 the solution obtained will serve as a reference point for the search of optimal integer-valued (or SP2-valued) $\boldsymbol{D}$ and $\boldsymbol{h}$, see Sec. IV for the details.

Finally, we remark that although the vectors $\{\boldsymbol{t}_i,\ 1 \leq i \leq n\}$ have to be such that $\hat{\boldsymbol{T}}$ is non-singular, this type of "constraint" needs not to be imposed explicitly because a near singular $\hat{\boldsymbol{T}}$ would make the value of $J(\boldsymbol{D}, \boldsymbol{h}, \hat{\boldsymbol{T}})$ very large, hence the process of minimizing $J(\boldsymbol{D}, \boldsymbol{h}, \hat{\boldsymbol{T}})$ automatically avoids considering ill-conditioned $\hat{\boldsymbol{T}}$. Consequently, the problem in (18) is practically an *unconstrained* minimization problem.

C. A Quasi-Newton Algorithm for Problem (18)

For the sake of simplicity we assume that the error feed-forward vector $\boldsymbol{h}$ is set to be equal to $\overline{\boldsymbol{c}}$ so as to eliminate term J_2 in (18). For the cases where $\boldsymbol{h}$ is not available or contains a number of zeros, the method described below still applies with straightforward modifications. Let $\boldsymbol{x}$ be the column vector that collects the variables in $\boldsymbol{D}$ and $\hat{\boldsymbol{T}}$, thus $J(\boldsymbol{D}, \hat{\boldsymbol{T}})$ is a function of $\boldsymbol{x}$, denoted by $J(\boldsymbol{x})$. The algorithm starts with a trivial initial point $\boldsymbol{x}_0$ obtained by letting $\boldsymbol{D} = \boldsymbol{I}$ and $\hat{\boldsymbol{T}} = \boldsymbol{I}$. Suppose we are in the kth iteration to update the most recent point $\boldsymbol{x}_k$. A quasi-Newton algorithm updates $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$ as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k \tag{22}$$

in two steps: (i) Determine a search direction $\boldsymbol{d}_k = -\boldsymbol{S}_k \boldsymbol{g}_k$ where $\boldsymbol{g}_k = \nabla J(\boldsymbol{x})$ is the gradient of the objective function and $\boldsymbol{S}_k$ is a positive-definite approximation of the inverse Hessian matrix of $J(\boldsymbol{x})$. A popular quasi-Newton algorithm is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [19] which updates $\boldsymbol{S}_k$ through the recursive relation

$$\boldsymbol{S}_{k+1} = \boldsymbol{S}_k + \left(1 + \frac{\boldsymbol{\gamma}_k^T \boldsymbol{S}_k \boldsymbol{\gamma}_k}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k}\right) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} - \frac{(\boldsymbol{\delta}_k \boldsymbol{\gamma}_k^T \boldsymbol{S}_k + \boldsymbol{S}_k \boldsymbol{\gamma}_k \boldsymbol{\delta}_k^T)}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} \tag{23}$$

where $\boldsymbol{S}_0 = \boldsymbol{I}$, $\boldsymbol{\delta}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$, and $\boldsymbol{\gamma}_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$. (ii) Once the search direction $\boldsymbol{d}_k$ is computed, the one-dimensional optimization (often called line search)

$$\lambda_k = \arg \underset{\lambda}{\text{minimize}}\ J(\boldsymbol{x}_k + \lambda \boldsymbol{d}_k) \tag{24}$$

is carried out to determine the value of λ_k. If the iteration progress measured by $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|$, is greater than a prescribed tolerance ε, then set $k := k + 1$ and repeat from Step (i), otherwise the iteration is terminated and $\boldsymbol{x}_{k+1}$ is claimed to be a solution point.

The implementation of (22) requires the gradient of $J(\boldsymbol{x})$. Closed form expressions for $J(\boldsymbol{x})$ with scalar, diagonal, and other types error-feedback matrix $\boldsymbol{D}$ are given in Appendix A.

We stress that the objective function J in (18) is *not* convex with respect to variables in $\boldsymbol{D}$ and $\boldsymbol{t}_1,\ \ldots,\ \boldsymbol{t}_n$. As a result, a solution obtained by the BFGS algorithm can only be claimed to be locally optimal. On the other hand, our computer simulations have indicated that the proposed algorithm appears to be rather insensitive to the choice of initial point. The reader is referred to Sec. III.D for more details.

D. Examples

We present two examples to illustrate the proposed optimization method. The first example concerns a 3rd-order lowpass IIR digital filter which was also used in [8]. The second example is about a 9th-order lowpass IIR filter which is used to demonstrate the ability of the proposed algorithm to deal with relatively large number of variables.

Example 1 Consider a 3rd-order stable IIR lowpass digital filter whose controllable canonical realization is denoted by $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)_3$ with

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.339377 & -1.152652 & 1.520167 \end{bmatrix} \tag{25a}$$

$$\boldsymbol{b} = [0 \quad 0 \quad 0.437881]^T \tag{25b}$$

$$\boldsymbol{c} = [0.212964 \quad 0.293733 \quad 0.718718] \tag{25c}$$

$$d = 6.59592 \times 10^{-2} \tag{25d}$$

The controllability Gramian $\boldsymbol{K}_c$ of the above filter has been normalized to satisfy the constraints $(\boldsymbol{K}_c)_{ii} = 1$ for $i = 1,\ 2,\ 3$. Without error feedback and error feed-forward (i.e., $\boldsymbol{D} = \boldsymbol{0}$ and $\boldsymbol{h} = \boldsymbol{0}$), the noise gain of filter (25) was found to be [8]

$$\text{tr}(\boldsymbol{W}_o) = 11.1332$$

If one applies the method of [1][2] to (25) to obtain a realization $(\overline{\boldsymbol{A}},\ \overline{\boldsymbol{b}},\ \overline{\boldsymbol{c}},\ d)_3$ for roundoff noise minimization (without error feedback and error feed-forward), then the noise gain was reduced to [8]

$$\text{tr}(\overline{\boldsymbol{W}}_o) = 2.3554$$

Next, the feed-forward vector $\boldsymbol{h}$ is assumed to be equal to $\overline{\boldsymbol{c}}$, and we compute jointly optimal error-feedback and state-space realization, with $\boldsymbol{D}$ being scalar and diagonal matrices, by

solving the respective minimization problem (18) using BFGS updates. The total number of variables involved in the optimization are 10 (for a scalar $\boldsymbol{D}$) and 12 (for a diagonal $\boldsymbol{D}$). The initial point used corresponds to $\boldsymbol{D} = \boldsymbol{I}$ and $\hat{\boldsymbol{T}} = \boldsymbol{I}$, and with $\varepsilon = 10^{-8}$ the algorithm converges with less than 20 iterations. The minimized noise gains obtained with separately and jointly optimized error-feedback matrix $\boldsymbol{D}$ and feed-forward vector $\boldsymbol{h}$ of integer quantized, 3-bit quantized, and infinite precision are given in Table I. For comparison purposes, Table I also includes the noise gains obtained in [8] where the error-feedback matrix is optimized for a fixed state-space realization that is optimal (without error feedback) for roundoff noise minimization. These values are listed in Table I in the lines where "Separate" is indicated for the column "Joint/Separate". Form the simulation results, it is evident that the proposed joint optimization offers consistent performance improvement for RN reduction.

In addition to the above initial point, as many as 20 randomly chosen initial points were also used to test the robustness of the proposed algorithm. The solutions obtained with 18 of these initial points were identical to the one described above. The solutions obtained with the other 2 initial points are different from the above-described solution with slightly degraded performance.

TABLE I

PERFORMANCE COMPARISON FOR EXAMPLE 1

Matrix $\boldsymbol{D}$	Optimization Method	Accuracy of $\boldsymbol{D}$ and $\boldsymbol{h}$		
		Infinite Precision	3-Bit Quantization	Integer Quantization
Null	—	2.3554		
Scalar	Separate	0.7552	0.7607	1.3697
	Joint	0.7537	0.7596	1.2825
Diagonal	Separate	0.6246	0.6303	1.0108
	Joint	0.6164	0.6268	1.0005
General	—	0	0.0088	1.1468

Example 2 Now we consider a 9th-order stable IIR lowpass filter whose transfer function is

denoted by

$$H(z) = \frac{b_1 z^9 + b_2 z^8 + \cdots + b_9 z + b_{10}}{a_1 z^9 + a_2 z^8 + \cdots + a_9 z + a_{10}}$$

where the coefficients are given in Table II.

TABLE II

COEFFICIENTS OF $H(z)$

i	a_i	b_i
1	1	0.002198
2	−3.640015	−0.007993
3	7.148374	0.010478
4	−9.521133	−0.007251
5	9.297299	0.011297
6	−6.830931	−0.010582
7	3.754299	−0.017728
8	−1.485109	0.023996
9	0.384233	0.006712
10	−0.049851	0.046116

Next we obtain the controllable canonical realization of the filter and normalize its controllability matrix by scaling so as to satisfy the constraints $(\boldsymbol{K}_c)_{ii} = 1$ for $i = 1,\ 2,\ \ldots,\ 9$. The state-space realization obtained is denoted by $(\boldsymbol{A},\ \boldsymbol{b},\ \boldsymbol{c},\ d)_9$ and its noise gain (without error feedback and error feed-forward) was found to be

$$\text{tr}(\boldsymbol{W}_o) = 3.1354 \times 10^3$$

Without using error feedback and error feed-forward, the method of [1][2] was applied to obtain a realization $(\overline{\boldsymbol{A}},\ \overline{\boldsymbol{b}},\ \overline{\boldsymbol{c}},\ \overline{d})_9$ for RN minimization, whose noise gain was reduced to

$$\text{tr}(\overline{\boldsymbol{W}}_o) = 2.5315$$

The proposed joint optimization method was then applied to the controllable canonical realization with error-feedback matrix $\boldsymbol{D}$ being scalar and diagonal matrices. As in Example 1,

$\boldsymbol{h} = \overline{\boldsymbol{c}}$ was assumed, and for both cases the same initial point, which corresponds to the choice of $\boldsymbol{D} = \boldsymbol{I}$, $\hat{\boldsymbol{T}} = \boldsymbol{I}$, was used. With $\varepsilon = 10^{-4}$, it took the algorithm 35 (for a scalar $\boldsymbol{D}$) and 196 (for a diagonal $\boldsymbol{D}$) iterations to converge. The minimized noise gains obtained with separately and jointly optimized error-feedback matrix $\boldsymbol{D}$ and feed-forward vector $\boldsymbol{h}$ of integer-quantized, 3-bit quantized, and infinite precision are given in Table III. Again, for comparison purposes, Table III also lists the noise gains obtained using separate realization and error-feedback matrix optimization proposed in [8]. From the table, it is observed that the performance improvement provided by the proposed joint optimization appears to be more pronounced. Based on this and a large number of simulations conducted so far, we conclude that the proposed joint optimization can offer improved performance gain for IIR state-space digital filters of relatively high order.

TABLE III

PERFORMANCE COMPARISON FOR EXAMPLE 2

Matrix $\boldsymbol{D}$	Optimization Method	Accuracy of $\boldsymbol{D}$ and $\boldsymbol{h}$		
		Infinite Precision	3-Bit Quantization	Integer Quantization
Null	—	2.5315		
Scalar	Separate	1.0846	1.0994	1.6220
	Joint	0.9545	0.9680	1.3650
Diagonal	Separate	1.0219	1.0326	1.5070
	Joint	0.7770	0.7958	1.1892
General	—	0	0.0379	1.4509

IV. DISCRETE OPTIMIZATION OF ERROR-FEEDBACK MATRIX AND ERROR-FEED-FORWARD VECTOR

A. Problem Statement

In the preceding section, state-space realization, error-feedback matrix and error-feed-forward vector are optimized under the assumption of infinite-precision implementation. In [8], after a (separately) optimized error-feedback matrix is obtained, its discrete counterpart with sum-of-power-of-two (SP2) entries is generated by rounding. A question that naturally arises is whether

or not "optimal" error-feedback matrix $\boldsymbol{D}$ and error-feed-forward vector $\boldsymbol{h}$ with SP2 components can be computed based on the optimal infinite-precision $\boldsymbol{D}$ and $\boldsymbol{h}$ with much reduced computational complexity compared to what is required for a solution of an integer programming (IP) problem. More specifically, here the term "an SP2 entry", say γ, is referred to a real number which can be expressed as

$$\gamma = \sum_{i=-L}^{U} \gamma_i 2^i \tag{26}$$

where $\gamma_i \in \{0, 1, -1\}$, and L and U are nonnegative integers defining the number of bits available for the representation of the integer and fractional parts of γ, respectively. In particular, with $L = 0,\ \gamma$ in (26) represents an integer between $-(2^{U+1} - 1)$ and $2^{U+1} - 1$.

Suppose $(\boldsymbol{A},\ \boldsymbol{b},\ \boldsymbol{c},\ \boldsymbol{d})_n$ is the state-space realization that, together with error-feedback matrix $\boldsymbol{D}_{opt}$ and error-feed-forward vector $\boldsymbol{h}_{opt}$ are obtained by the proposed joint optimization method. We seek for a general matrix $\boldsymbol{D}$ and a general vector $\boldsymbol{h}$ that solve the discrete optimization problem

$$\begin{aligned}
\underset{\boldsymbol{D},\boldsymbol{h}}{\text{minimize}} \quad & I(\boldsymbol{D}) = \text{tr}[(\boldsymbol{D} - \boldsymbol{A})^T \boldsymbol{W}_o (\boldsymbol{D} - \boldsymbol{A}) + (\boldsymbol{c} - \boldsymbol{h})^T (\boldsymbol{c} - \boldsymbol{h})] && \text{(27a)} \\
\text{subject to:} \quad & \text{all entries of } \boldsymbol{D} \text{ and } \boldsymbol{h} \text{ are SP2} && \text{(27b)} \\
& (\boldsymbol{D}, \boldsymbol{h}) \text{ is in the vicinity of } (\boldsymbol{D}_{opt}, \boldsymbol{h}_{opt}) && \text{(27c)}
\end{aligned}$$

The precise meaning of the term "vicinity" in (27c) will become transparent shortly. The reason we constrain our search to the vicinity of $(\boldsymbol{D}_{opt},\ \boldsymbol{h}_{opt})$ is purely technical: Since $(\boldsymbol{D}_{opt}, \boldsymbol{h}_{opt})$ is the optimal solution for the infinite precision case, it is reasonable to expect a good discrete candidate nearby $(\boldsymbol{D}_{opt},\ \boldsymbol{h}_{opt})$; but more importantly, it is this constraint that allows one to convert the problem at hand into a $(-1, 1)$-quadratic programming problem which admits a semidefinite programming (SDP) relaxation [20] — a key step towards a near-optimal solution with reduced computational complexity. Details of this development are given next.

B. Problem Conversion

First, note that the term $(\boldsymbol{c} - \boldsymbol{h})^T (\boldsymbol{c} - \boldsymbol{h})$ in (27a) does not depend on $\boldsymbol{D}$, hence the problem in (27) can be addressed by splitting it into two sub-problems as follows:

$$\begin{aligned}
\underset{\boldsymbol{D}}{\text{minimize}} \quad & \text{tr}(\boldsymbol{D}^T \boldsymbol{W}_o \boldsymbol{D} - 2\boldsymbol{D}^T \boldsymbol{W}_o \boldsymbol{A}) && \text{(28a)} \\
\text{subject to:} \quad & \text{entries of } \boldsymbol{D} \text{ are SP2} && \text{(28b)}
\end{aligned}$$

$$\boldsymbol{D} \text{ in the vicinity of } \boldsymbol{D}_{opt} \tag{28c}$$

where the term $\boldsymbol{A}^T\boldsymbol{W}_o\boldsymbol{A}$ contained in the first term in (27a) has been dropped, and

$$\begin{aligned} \underset{\boldsymbol{h}}{\text{minimize}} \quad & \text{tr}[(\boldsymbol{c}-\boldsymbol{h})^T(\boldsymbol{c}-\boldsymbol{h})] && \text{(29a)} \\ \text{subject to:} \quad & \text{all entries of } \boldsymbol{h} \text{ are SP2} && \text{(29b)} \\ & \boldsymbol{h} \text{ in the vicinity of } \boldsymbol{h}_{opt} && \text{(29c)} \end{aligned}$$

If we denote the vectors generated by stacking the columns of $\boldsymbol{D}$, $\boldsymbol{D}_{opt}$ and $\boldsymbol{W}_o\boldsymbol{A}$ by $\boldsymbol{d}, \boldsymbol{d}_{opt}$, and $\boldsymbol{p}$, respectively, and denote the block diagonal matrix diag$\{\boldsymbol{W}_o,\ \boldsymbol{W}_o,\ \ldots,\ \boldsymbol{W}_o\} \in \mathcal{R}^{n^2\times n^2}$ by $\boldsymbol{Q}$, then the problem in (28) becomes

$$\begin{aligned} \underset{\boldsymbol{d}}{\text{minimize}} \quad & \boldsymbol{d}^T\boldsymbol{Q}\boldsymbol{d} - 2\boldsymbol{d}^T\boldsymbol{p} && \text{(30a)} \\ \text{subject to:} \quad & \text{entries of } \boldsymbol{d} \text{ are SP2} && \text{(30b)} \\ & \boldsymbol{d} \text{ in the vicinity of } \boldsymbol{d}_{opt} && \text{(30c)} \end{aligned}$$

Let $\boldsymbol{d}$ be denoted by $\boldsymbol{d} = [d_1 \ \ d_2 \ \ \cdots \ \ d_{n^2}]^T$. For a given bit number for the representation of each component d_k, the least SP2 upper bound $\overline{d}_k$ and largest lower bound $\underline{d}_k$ of the infinite-precision d_k can be identified. It follows that $\underline{d}_k \le d_k \le \overline{d}_k$ and in each open interval $(\underline{d}_k, \overline{d}_k)$ no SP2 terms with the given number of bits exist. Fig. 2 illustrates the first several components d_k and their bounds. The SP2 representation based on rounding is given by

$$d_k^{(r)} = \begin{cases} \overline{d}_k & \text{if } \overline{d}_k - d_k \le d_k - \underline{d}_k \\ \underline{d}_k & \text{otherwise} \end{cases} \tag{31}$$

We denote $\boldsymbol{d}_r = [d_1^{(r)} \ \ d_2^{(r)} \ \ \cdots \ \ d_{n^2}^{(r)}]^T$ and remark that although $\boldsymbol{d}_r$ satisfies the constraints in (30b) and (30c), in general $\boldsymbol{d}_r$ does not minimize the objective function in (30a). As will be shown in Sec. IV.C, however, the SP2 representation obtained by rounding is indeed the solution of (30) when $\boldsymbol{D}_{opt}$ is scalar or diagonal.

Now denote the midpoint of each interval $[\underline{d}_k,\ \overline{d}_k]$ as $d_{mk} = (\overline{d}_k + \underline{d}_k)/2$ and a half of the interval length as $\delta_k = (\overline{d}_k - \underline{d}_k)/2$, the bounds $\overline{d}_k$ and $\underline{d}_k$ can then be selected as $d_{mk} + r_k\delta_k$ with $r_k = 1$ and $r_k = -1$, respectively. This in conjunction with Fig. 2 explains the meaning of the term "vicinity" in (30c). The vector $\boldsymbol{d}$ in (30) can now be expressed as

$$\boldsymbol{d} = \boldsymbol{d}_m + \boldsymbol{\Delta r} \tag{32}$$

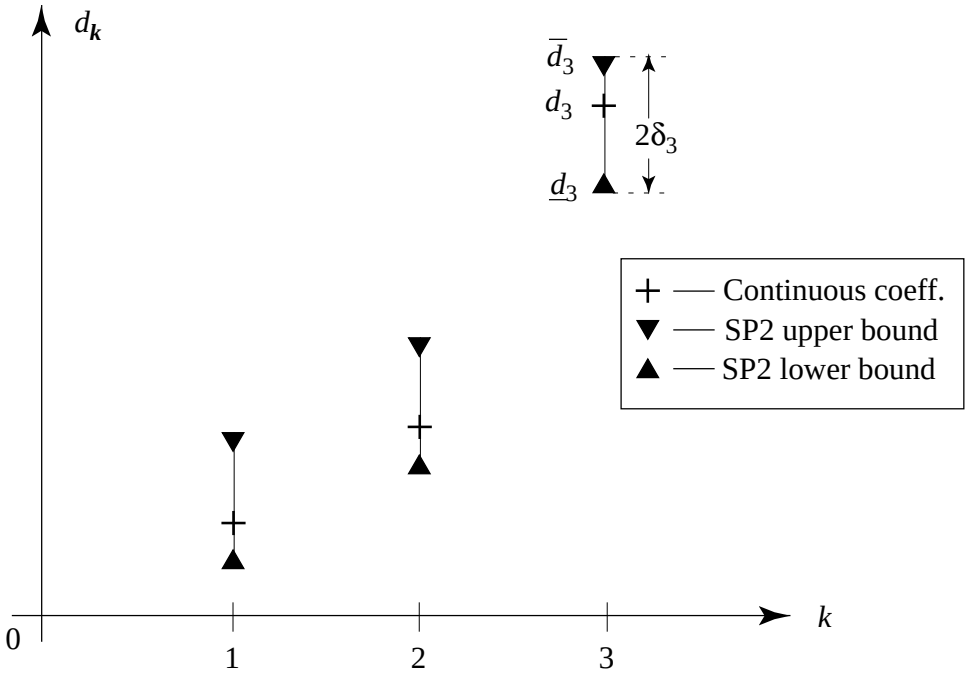

Fig. 2. Continuous components and their least SP2 upper bounds and largest SP2 lower bounds.

where $\boldsymbol{d}_m = [d_{m1},\ d_{m2},\ \cdots,\ d_{m,n^2}]^T$, $\boldsymbol{\Delta} = \text{diag}\{\delta_1,\ \delta_2,\ \cdots,\ \delta_{n^2}\}$, and $\boldsymbol{r} = [r_1,\ r_2,\ \cdots,\ r_{n^2}]^T$, and up to a constant, the objective function in (30a) becomes $\boldsymbol{r}^T\boldsymbol{Q}_\delta\boldsymbol{r} - 2\boldsymbol{r}^T\boldsymbol{p}_\delta$ where $\boldsymbol{Q}_\delta = \boldsymbol{\Delta}\boldsymbol{Q}\boldsymbol{\Delta}$ and $\boldsymbol{p}_\delta = \boldsymbol{\Delta}(\boldsymbol{p} - \boldsymbol{Q}\boldsymbol{d}_m)$. Hence the discrete optimization problem in (30) can be formulated as

$$\underset{\boldsymbol{r}}{\text{minimize}} \quad \boldsymbol{r}^T\boldsymbol{Q}_\delta\boldsymbol{r} - 2\boldsymbol{r}^T\boldsymbol{p}_\delta \tag{33a}$$

$$\text{subject to:} \quad r_k \in \{-1,\ 1\} \qquad \text{for } 1 \le k \le n^2 \tag{33b}$$

Since the components of $\boldsymbol{Q}_\delta$ and $\boldsymbol{p}_\delta$ are continuous-valued, the problem in (33) is a $(-1,\ 1)$-mixed integer quadratic programming (MIQP) problem.

C. Two Simple Cases: $\boldsymbol{D}_{opt}$ is Scalar or Diagonal

The formulation in (33) applies to the case where the error-feedback matrix $\boldsymbol{D}_{opt}$ is a general matrix. When $\boldsymbol{D}_{opt}$ is a diagonal or scalar matrix, the problem at hand is considerably simplified in terms of the number of variables involved as well as problem complexity.

Case 1: If $\boldsymbol{D}_{opt} = \alpha_{opt}\boldsymbol{I}$, then $\boldsymbol{D}$ assumes the form $\boldsymbol{D} = \alpha\boldsymbol{I}$ and the objective function in (28a)

becomes

$$\text{tr}(\boldsymbol{W}_o)\alpha^2 - 2\,\text{tr}(\boldsymbol{W}_o\boldsymbol{A})\alpha$$

where scalar α can be expressed as

$$\alpha = \alpha_m + \delta r \tag{34}$$

with

$$\begin{aligned}
\alpha_m &= (\overline{\alpha} + \underline{\alpha})/2 \\
\overline{\alpha} &= \text{ least upper bound of } \alpha_{opt} \\
\underline{\alpha} &= \text{ largest lower bound of } \alpha_{opt} \\
\delta &= (\overline{\alpha} - \underline{\alpha})/2 \\
r &\in \{-1,\ 1\}
\end{aligned}$$

Hence, up to a constant, the objective function is given by

$$-2\delta r\,\text{tr}[\boldsymbol{W}_o(\boldsymbol{A} - \alpha_m\boldsymbol{I})]$$

Because $\delta \geq 0$, the objective function achieves its minimum if

$$r = \text{sign}\{\text{tr}[\boldsymbol{W}_o(\boldsymbol{A} - \alpha_m\boldsymbol{I})]\} \tag{35}$$

It is known [8] that the infinite-precision α_{opt} is given by

$$\alpha_{opt} = \frac{\text{tr}(\boldsymbol{W}_o\boldsymbol{A})}{\text{tr}(\boldsymbol{W}_o)} \tag{36}$$

It follows that the value of r in (35) can be evaluated as

$$r = \text{sign}[\alpha_{opt}\text{tr}(\boldsymbol{W}_o) - \alpha_m\text{tr}(\boldsymbol{W}_o)] = \text{sign}(\alpha_{opt} - \alpha_m) \tag{37}$$

where the second equality is based on the fact that $\text{tr}(\boldsymbol{W}_o) > 0$. Consequently, if $\alpha_{opt} \geq \alpha_m$, then $r = 1$ and $\alpha = \alpha_m + \delta = \overline{\alpha}$; and if $\alpha_{opt} < \alpha_m$, then $r = -1$ and $\alpha = \alpha_m - \delta = \underline{\alpha}$. This means that α *can be obtained by simply rounding the value of* α_{opt}.

Case 2: If $\boldsymbol{D}_{opt} = \text{diag}\{d_1^{(opt)},\ d_2^{(opt)},\ \ldots,\ d_n^{(opt)}\}$, then $\boldsymbol{D}$ assumes the form $\boldsymbol{D} = \text{diag}\{d_1,\ d_2,\ \ldots,\ d_n\}$ and the objective function in (28a) becomes

$$\sum_{k=1}^{n}[(\boldsymbol{W}_o)_{kk}d_k^2 - 2(\boldsymbol{W}_o\boldsymbol{A})_{kk}d_k] \tag{38}$$

Let $d_k = d_{mk} + \delta_k r_k$ with $d_{mk} = (\overline{d}_k + \underline{d}_k)/2$, $\delta_k = (\overline{d}_k - \underline{d}_k)/2$, and $r_k \in \{-1,\ 1\}$, (38) can be written (up to a constant) as

$$-2\sum_{k=1}^{n} \delta_k r_k[(\boldsymbol{W}_o\boldsymbol{A})_{kk} - d_{mk}(\boldsymbol{W}_o)_{kk}]$$

whose minimum is achieved if

$$r_k = \text{sign}[(\boldsymbol{W}_o\boldsymbol{A})_{kk} - d_{mk}(\boldsymbol{W}_o)_{kk}] \qquad \text{for } 1 \le k \le n \tag{39}$$

because $\delta_k \ge 0$ for all k. From [8], it is known that the infinite-precision $\{d_k^{(opt)},\ \ 1 \le k \le n\}$ are given by

$$d_k^{(opt)} = \frac{(\boldsymbol{W}_o\boldsymbol{A})_{kk}}{(\boldsymbol{W}_o)_{kk}} \qquad 1 \le k \le n \tag{40}$$

Hence r_k in (39) can be evaluated as

$$r_k = \text{sign}\{(\boldsymbol{W}_o)_{kk}[d_k^{(opt)} - d_{mk}]\} = \text{sign}(d_k^{(opt)} - d_{mk})$$

where the second equality holds because $(\boldsymbol{W}_o)_{kk} > 0$. If $d_k^{(opt)} \ge d_{mk}$, then we have $r_k = 1$ and $d_k = d_{mk} + \delta_k = \overline{d}_k$; if $d_k^{(opt)} < d_{mk}$ then $r_k = -1$ and $d_k = d_{mk} - \delta_k = \underline{d}_k$. Consequently, d_k *can be obtained by rounding the value of* $d_k^{(opt)}$.

In words, we conclude that in the cases of $\boldsymbol{D}_{opt}$ being a scalar or diagonal matrix, the optimal error-feedback matrix $\boldsymbol{S}$ with SP2 entries can be obtained by simply rounding the infinite-precision $\boldsymbol{D}_{opt}$.

D. An Approximate Solution of Problem (33)

In this section, the problem in (33) is *relaxed* to a semidefinite programming (SDP) problem so as to obtain a satisfactory approximate solution in polynomial time. Goemans and Williamson [20] was among the first to propose an SDP relaxation of the MAX-CUT problem — a well-known integer quadratic programming problem in graph theory. Following [20], SDP relaxation of various combinatorial optimization problems have been reported in graph optimization, network management, scheduling [27], filter designs [28][30], and other applications [29]. To the knowledge of the authors, however, to date SDP relaxation has not been applied to obtain an improved error-feedback matrix with SP2 entries for RN reduction. The first step towards a

relaxed problem formulation is to write (33) as

$$\begin{aligned} \text{minimize} \quad & \text{tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{R}}) && \text{(41a)} \\ & \hat{\boldsymbol{R}} \succeq \mathbf{0} && \text{(41b)} \\ & (\hat{\boldsymbol{R}})_{kk} = 1 \qquad \text{for } 1 \le k \le N && \text{(41c)} \\ & \text{rank}(\hat{\boldsymbol{R}}) = 1 && \text{(41d)} \end{aligned}$$

where

$$\hat{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{Q}_\delta & -\boldsymbol{p}_\delta \\ -\boldsymbol{p}_\delta^T & 0 \end{bmatrix}, \; \hat{\boldsymbol{R}} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{r} \\ \boldsymbol{r}^T & 1 \end{bmatrix} \tag{41e}$$

and $N = n^2 + 1$. To see the equivalence of (41) to (33), we note that (33b) means

$$r_k^2 = 1 \qquad \text{for } 1 \le k \le n^2 \tag{42}$$

On the other hand, the rank condition in (41d) implies that the $n^2 \times n^2$ matrix $\boldsymbol{R}$ in (41e) must have the form $\boldsymbol{R} = \boldsymbol{r}\boldsymbol{r}^T$, i.e.,

$$\hat{\boldsymbol{R}} = \begin{bmatrix} \boldsymbol{r}\boldsymbol{r}^T & \boldsymbol{r} \\ \boldsymbol{r}^T & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}^T & 1 \end{bmatrix} \tag{43}$$

which leads to the equivalence of (41c) to (42). Moreover, (43) implies (41b) and the objective function in (41a) can be expressed as

$$\text{tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{R}}) = \boldsymbol{r}^T\boldsymbol{Q}_\delta\boldsymbol{r} - 2\boldsymbol{r}^T\boldsymbol{p}_\delta$$

which is exactly the objective function in (33a).

The second step does the "relaxation" by dropping the rank condition in (41d), which leads to the following optimization problem

$$\begin{aligned} \text{minimize} \quad & \text{tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{R}}) && \text{(44a)} \\ \text{subject to:} \quad & \hat{\boldsymbol{R}} \succeq \mathbf{0} && \text{(44b)} \\ & (\hat{\boldsymbol{R}})_{kk} = 1 \qquad \text{for } 1 \le k \le N && \text{(44c)} \end{aligned}$$

The equality constraints in (44c) are automatically satisfied if $\hat{\boldsymbol{R}}$ assumes the form

$$\hat{\boldsymbol{R}} = \begin{bmatrix} 1 & * & \cdots & * \\ * & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ * & \cdots & * & 1 \end{bmatrix} \tag{45}$$

which contains $N(N-1)/2$ independent parameters because $\hat{\boldsymbol{R}}$ is symmetric. Let $\hat{\boldsymbol{R}} = \{\hat{r}_{ij}\}$. The positive semidefinite constraint in (44b) can be written as

$$\hat{\boldsymbol{R}} = \boldsymbol{I} + \sum_{i \neq j} r_{ij}(\boldsymbol{I}_{ij} + \boldsymbol{I}_{ji}) \succeq \boldsymbol{0} \tag{46}$$

where $\boldsymbol{I}_{ij}$ is the matrix with zero entries except its $(i,\ j)$th component which assumes the value of 1. With the form of $\hat{\boldsymbol{R}}$ in (46), the objective function in (44a) is obviously a linear function of $\{r_{ij}\}$. Consequently, (44) is an SDP problem (see Eq. (8)).

Once (44) is solved for $\hat{\boldsymbol{R}}$, the solution vector $\boldsymbol{r}$ with $r_i \in \{-1,\ 1\}$ can be obtained in several ways. A straightforward solution can be obtained based on (41e) which suggests

$$\boldsymbol{r} = \text{sign}[\hat{\boldsymbol{R}}(1 : n^2,\ N)] \tag{47}$$

where $\hat{\boldsymbol{R}}(1 : n^2,\ N)$ denotes the first n^2 components of the last column vector of $\hat{\boldsymbol{R}}$. At the cost of more computations, an often improved solution can be obtained using an optimal rank-one approximation of $\hat{\boldsymbol{R}}$ in the 2-norm sense. It is well-known that such an approximation is given by $\lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T$ where λ_1 and $\boldsymbol{u}_1$ are the largest eigenvalue of $\hat{\boldsymbol{R}}$ and the associated eigenvector, respectively. If we write

$$\boldsymbol{u}_1 = \begin{bmatrix} \boldsymbol{u} \\ u_N \end{bmatrix} \tag{48}$$

then

$$\hat{\boldsymbol{R}} \approx \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T = \frac{\lambda_1}{u_N^2} \begin{bmatrix} \hat{\boldsymbol{u}}\hat{\boldsymbol{u}}^T & \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{u}}^T & 1 \end{bmatrix} \tag{49}$$

where $\hat{\boldsymbol{u}} = \boldsymbol{u}/u_N$. On comparing (49) with (41e), it is obvious that, up to a positive factor, $\hat{\boldsymbol{u}}$ is a close resemblance of $\boldsymbol{r}$. This suggests the solution

$$\boldsymbol{r} = \text{sign}(\boldsymbol{u}/u_N) \tag{50}$$

Once a binary vector $\boldsymbol{r}$ is obtained, vector $\boldsymbol{d}$ can be computed using (32) and a discrete error-feedback matrix $\boldsymbol{D}$ can be formed using $\boldsymbol{d}$.

Two remarks are in order. It is well-known [25] that the MIQP problem in (33) is NP-hard, whose computational complexity grows exponentially with its problem size. On the other hand, efficient interior-point algorithms with polynomial-time complexity for SDP problems have been available since 1990's [27]. It is important to note that the number of variables in the SDP

problem (44) has increased to $N(N-1)/2 = n^2(1+n^2)/2$ while the number of variables in the original problem (33) is n^2. It is therefore expected that solving (44) becomes slow even for a system of moderate order. This problem can be largely overcome by converting the SDP problem into a dual SDP problem which involves only n^2 variables. The reader is referred to reference [30] for details. Our second remark is about the quality of the approximate solution obtained by the SDP relaxation. It is known that for certain types of matrices $\hat{\boldsymbol{Q}}$, the approximate solution by SDP relaxation is guaranteed to have excellent quality [20]. Although the matrix $\hat{\boldsymbol{Q}}$ for the problem at hand does not belong to the matrix class discussed in [20], our computer simulations have demonstrated that the proposed SDP-relaxation-based method offers in many cases near-optimal solutions.

E. Examples

We now apply the SDP-relaxation method to the two IIR filters examined in Sec. III.D.

Example 3 Consider the IIR filter in Example 1 where the state-space realization $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)_3$ satisfying the l_2-norm dynamic range constraints was obtained by the method in [1][2]. According to Sec. III.B, we have the infinite-precision solution $\boldsymbol{D}_{opt} = \boldsymbol{A}$ and $\boldsymbol{h}_{opt} = \boldsymbol{c}$, which are given by

$$\boldsymbol{D}_{opt} = \begin{bmatrix} 0.460807 & 0.621155 & 0.154614 \\ -0.436518 & 0.542428 & 0.491784 \\ 0.007819 & -0.232128 & 0.516932 \end{bmatrix}$$

$$\boldsymbol{h}_{opt} = [\,0.780485 \quad 0.335409 \quad 0.241041\,]$$

A case of particular interest is when both $\boldsymbol{D}$ and $\boldsymbol{h}$ assume integer entries. In this case above $\boldsymbol{D}_{opt}$ and $\boldsymbol{h}_{opt}$ suggest the range $U = 0$ and $L = 0$. Three approaches, i.e., rounding, SDP relaxation, and exhaustive search were used, where the exhaustive search evaluates the objective function in (28a) for all possible combinations of upper and lower bounds of d_{ij} and finds the $\boldsymbol{D}$ at which the function reaches its minimum. For the present example, a 9-entry $\boldsymbol{D}_{opt}$ means that the exhaustive search requires a total of 2^9 function evaluations. The integer-optimal $\boldsymbol{h}$ was obtained by rounding $\boldsymbol{h}_{opt}$ to $\boldsymbol{h} = [1 \;\; 0 \;\; 0]$. The $\boldsymbol{D}$ matrices obtained by rounding, SDP

relaxation, and exhaustive search were as follows:

$$\boldsymbol{D}_{rounding} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{D}_{SDPR} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{D}_{exh} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

which correspond to noise gains 1.1468, 0.6435, and 0.6435, respectively. Note that the SDP-relaxation-based method offered considerably improved performance over the rounding approach and it indeed reached the minimum noise gain.

Example 4 Here we consider the 9th-order IIR filter in Example 2 where the optimized state-space realization was obtained using the method in [1][2]. Since all entries of the infinite-precision $\boldsymbol{D}_{opt}$ and $\boldsymbol{h}_{opt}$ are less than one in magnitude, for the search of optimal integer-valued $\boldsymbol{D}$ and $\boldsymbol{h}$, $U = 0$ and $L = 0$ were assumed. Since there are 81 entries in $\boldsymbol{D}$, exhaustive search that would require 2^{81} function evaluations turned out to be infeasible. The integer-optimal $\boldsymbol{h}$ was obtained by rounding $\boldsymbol{h}_{opt}$ which yielded $\boldsymbol{h} = \boldsymbol{0}$, and the $\boldsymbol{D}$ matrices obtained by rounding and SDP relaxation were found to be

$$\boldsymbol{D}_{rounding} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{D}_{SDPR} = \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

which correspond to noise gains 1.4509 and 1.2792, respectively, representing a 12% improvement in the SDP-relaxation solution over that obtained by rounding.

V. Conclusions

In this paper, joint optimization of error feedback/feed-forward and state-space realization for RN minimization has been investigated in two different scenarios: Under the assumption of infinite precision for both error-feedback and coordinate transformation matrices, the problem at hand is converted into a general unconstrained problem in which the previously reported optimization for realization-only and error-feedback-only can be regarded as special cases; closed-form formula for gradient evaluation is derived; and efficient quasi-Newton algorithms are applicable. In the second scenario, discrete optimization of error-feedback matrix under a given state-space realization has been studied, in which the infinite-precision solution is utilized as a reference point, and an SDP-relaxation method is proposed to obtain an approximate solution of the NP-hard mixed integer quadratic programming problem. Computer simulations have demonstrated that in the case where a general error-feedback matrix is used, the approximate solution offers improved performance over the rounding-based solution, and can indeed be optimal or near optimal.

APPENDIX A EVALUATION OF $\nabla J(\boldsymbol{x})$

A.1 If $\boldsymbol{D}$ is a scalar matrix

The objective function in (20) has the form $J(\boldsymbol{x}) = \text{tr}(\hat{\boldsymbol{T}}\boldsymbol{M}\hat{\boldsymbol{T}}^T)$ which, in the light of (16), can be expressed as

$$J(\boldsymbol{x}) = \text{tr}\left\{ \begin{bmatrix} \dfrac{\boldsymbol{t}_1}{\|\boldsymbol{t}_1\|} & \cdots & \dfrac{\boldsymbol{t}_n}{\|\boldsymbol{t}_n\|} \end{bmatrix}^{-1} \boldsymbol{M} \begin{bmatrix} \dfrac{\boldsymbol{t}_1}{\|\boldsymbol{t}_1\|} & \cdots & \dfrac{\boldsymbol{t}_n}{\|\boldsymbol{t}_n\|} \end{bmatrix}^{-T} \right\} \tag{A1}$$

If $\boldsymbol{D}$ is a scalar matrix, $\boldsymbol{D} = \alpha\boldsymbol{I}$, then vector $\boldsymbol{x}$ contains a total of $n^2 + 1$ variables, i.e., $\alpha,\ \boldsymbol{t}_1,\ \boldsymbol{t}_2,\ \ldots,\ \boldsymbol{t}_n$. To compute $\partial J(\boldsymbol{x})/\partial t_{ij}$, we perturb the ith component of vector $\boldsymbol{t}_j$ by a small amount, say δ, and keep the rest of $\hat{\boldsymbol{T}}$ unchanged. If we denote the perturbed jth column of $\hat{\boldsymbol{T}}^{-1}$ by $\tilde{\boldsymbol{t}}_j/\|\tilde{\boldsymbol{t}}_j\|$, then we can write a linear approximation of $\tilde{\boldsymbol{t}}_j/\|\tilde{\boldsymbol{t}}_j\|$ as

$$\frac{\tilde{\boldsymbol{t}}_j}{\|\tilde{\boldsymbol{t}}_j\|} \approx \frac{\boldsymbol{t}_j}{\|\boldsymbol{t}_j\|} - \delta\boldsymbol{g}_{ij}$$

where $\boldsymbol{g}_{ij}$ is a vector given by

$$\boldsymbol{g}_{ij} = \frac{1}{\|\boldsymbol{t}_j\|^3}(t_{ij}\boldsymbol{t}_j - \|\boldsymbol{t}_j\|^2\boldsymbol{e}_i) \tag{A2}$$

and $\boldsymbol{e}_i$ is the ith coordinate vector. Now let $\hat{\boldsymbol{T}}_{ij}$ be the matrix obtained from $\hat{\boldsymbol{T}}$ with a perturbed (i,j)th component, then up to the first order the matrix inversion formula [23, p. 655] gives

$$\hat{\boldsymbol{T}}_{ij} = \hat{\boldsymbol{T}} + \frac{\delta(\hat{\boldsymbol{T}}\boldsymbol{g}_{ij})(\boldsymbol{e}_j^T\hat{\boldsymbol{T}})}{1 - \delta\boldsymbol{e}_j^T\hat{\boldsymbol{T}}\boldsymbol{g}_{ij}}$$

Consequently, we have

$$\begin{aligned} \frac{\partial J(\boldsymbol{x})}{\partial t_{ij}} &= \lim_{\delta\to 0}[\text{tr}(\hat{\boldsymbol{T}}_{ij}\boldsymbol{M}\hat{\boldsymbol{T}}_{ij}^T) - \text{tr}(\hat{\boldsymbol{T}}\boldsymbol{M}\hat{\boldsymbol{T}}^T)]/\delta \\ &= 2\text{tr}[(\hat{\boldsymbol{T}}\boldsymbol{g}_{ij})(\boldsymbol{e}_j^T\hat{\boldsymbol{T}})\boldsymbol{M}\hat{\boldsymbol{T}}^T] \\ &= 2\boldsymbol{e}_j^T(\hat{\boldsymbol{T}}\boldsymbol{M}\hat{\boldsymbol{T}}^T\hat{\boldsymbol{T}})\boldsymbol{g}_{ij} \qquad \text{for } 1 \le i,j \le n \end{aligned} \tag{A3}$$

Finally, we use (20) to compute

$$\frac{\partial J(\boldsymbol{x})}{\partial \alpha} = \text{tr}[\hat{\boldsymbol{T}}(2\alpha\hat{\boldsymbol{W}}_o - \hat{\boldsymbol{A}}^T\hat{\boldsymbol{W}}_o - \hat{\boldsymbol{W}}_o\hat{\boldsymbol{A}})\hat{\boldsymbol{T}}^T] \tag{A4}$$

A.2 If matrix $\boldsymbol{D}$ contains certain number of zero components in fixed places

The type of $\boldsymbol{D}$ matrices we deal with here obviously includes the case of $\boldsymbol{D}$ being a diagonal matrix. To evaluate the gradient of J in (18) where $J_2 = 0$ is assumed, we write it as

$$J(\boldsymbol{x}) = \text{tr}(\hat{\boldsymbol{T}}\boldsymbol{M}\hat{\boldsymbol{T}}^T) + \text{tr}(\boldsymbol{D}^T\hat{\boldsymbol{T}}\hat{\boldsymbol{W}}_o\hat{\boldsymbol{T}}^T\boldsymbol{D}) - 2\text{tr}(\hat{\boldsymbol{T}}\hat{\boldsymbol{A}}^T\hat{\boldsymbol{W}}_o\hat{\boldsymbol{T}}^T\boldsymbol{D}) \tag{A5}$$

where

$$\boldsymbol{M} = \hat{\boldsymbol{A}}^T\hat{\boldsymbol{W}}_o\hat{\boldsymbol{A}}$$

and $\boldsymbol{x}$ in this case contains the nonzero entries of $\boldsymbol{D}$ plus vectors $\boldsymbol{t}_1,\ \boldsymbol{t}_2,\ \ldots,\ \boldsymbol{t}_n$. To compute $\partial J(\boldsymbol{x})/\partial t_{ij}$, we treat all the quantities other than t_{ij} in (A5) including $\boldsymbol{D}$ as constant terms. It then follows from Sec. A.1 that

$$\frac{\partial J(\boldsymbol{x})}{\partial t_{ij}} = 2\beta_1 + 2(\beta_2 - \beta_3) \qquad \text{for } 1 \le i, j \le n \tag{A6}$$

with

$$\beta_1 = \boldsymbol{e}_j^T(\hat{\boldsymbol{T}}\boldsymbol{M}\hat{\boldsymbol{T}}^T\hat{\boldsymbol{T}})\boldsymbol{g}_{ij} \tag{A7}$$

$$\beta_2 = \boldsymbol{e}_j^T(\hat{\boldsymbol{T}}\hat{\boldsymbol{W}}_o\hat{\boldsymbol{T}}^T\boldsymbol{D}\boldsymbol{D}^T\hat{\boldsymbol{T}})\boldsymbol{g}_{ij} \tag{A8}$$

$$\beta_3 = \boldsymbol{e}_j^T\hat{\boldsymbol{T}}(\hat{\boldsymbol{A}}^T\hat{\boldsymbol{W}}_o\hat{\boldsymbol{T}}^T\boldsymbol{D} + \hat{\boldsymbol{W}}_o\hat{\boldsymbol{A}}\hat{\boldsymbol{T}}^T\boldsymbol{D}^T)\hat{\boldsymbol{T}}\boldsymbol{g}_{ij} \tag{A9}$$

Finally, using (A5) we compute for $\boldsymbol{D} = \{d_{ij}\}$ the derivative

$$\frac{\partial J(\boldsymbol{x})}{\partial d_{ij}} = 2\boldsymbol{e}_j^T(\boldsymbol{D}^T\hat{\boldsymbol{T}} - \hat{\boldsymbol{T}}\hat{\boldsymbol{A}}^T)\hat{\boldsymbol{W}}_o\hat{\boldsymbol{T}}^T\boldsymbol{g}_{ij} \tag{A10}$$

In particular, if $\boldsymbol{D}$ is a diagonal matrix, i.e., $\boldsymbol{D} = \text{diag}\{d_1, d_2, \ldots, d_n\}$, then

$$\frac{\partial J(\boldsymbol{x})}{\partial d_i} = 2\boldsymbol{e}_i^T(\boldsymbol{D}^T\hat{\boldsymbol{T}} - \hat{\boldsymbol{T}}\hat{\boldsymbol{A}}^T)\hat{\boldsymbol{W}}_o\hat{\boldsymbol{T}}^T\boldsymbol{g}_{ii} \tag{A11}$$

REFERENCES

[1] S. Y. Hwang, "Roundoff noise in state-space digital filtering: A general analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 24, pp. 254-262, June 1976.

[2] C. T. Mullis and R. A. Roberts, "Synthesis of minimum roundoff noise fixed-point digital filters," *IEEE Trans. Circuits syst.*, vol. 23, pp. 551-562, Sept. 1976.

[3] S. Y. Hwang, "Minimum uncorrelated unit noise in state-space digital filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, pp. 273-281, Aug. 1977.

[4] L. B. Jackson, A. G. Lindgren and Y. Kim, "Optimal synthesis of second-order state-space structures for digital filters," *IEEE Trans. Circuits Syst.*, vol. 26, pp. 149-153, Mar. 1979.

[5] B. W. Bomar, "State-space structure for the realization of low roundoff noise digital filters," *Dissertation*, University of Tennessee, 1983.

[6] D. Williamson, "Roundoff noise minimization and pole-zero sensitivity in fixed-point digital filter using residue feedback," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1210-1220, Oct. 1986.

[7] G. Li and M. Gevers, "Roundoff noise minimization using delta-operator realizations," *IEEE Trans. Signal Processing*, vol. 41, pp. 629-637, Feb. 1993.

[8] T. Hinamoto, H. Ohnishi, and W-S. Lu, "Roundoff noise minimization of state-space digital using separate and joint error feedback/coordinate transformation optimization," *IEEE Trans. Circuits Syst., I*, vol. 50, pp. 23-33, Jan. 2003.

[9] H. A. Spang, III and P. M. Shultheiss, "Reduction of quantizing noise by use of feedback," *IRE Trans. Comm. Syst.*, vol. 10, pp. 373-380, Dec. 1962.

[10] T. Thong and B. Liu, "Error spectrum shaping in narrowband recursive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, pp. 200-203, Apr. 1977.

[11] T. L. Chang and S. A. White, "An error cancellation digital filter structure and its distributed-arithmetic implementation," *IEEE Trans. Circuits Syst.*, vol. 28, pp. 339-342, Apr. 1981.

[12] D. C. Munson and D. Liu, "Narrowband recursive filters with error spectrum shaping," *IEEE Trans. Circuits Syst.* vol. 28, pp. 160-163, Feb. 1981.

[13] W. E. Higgins and D. C. Munson, "Noise reduction strategies for digital filters: Error spectrum shaping versus the optimal linear state-space formulation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 30, pp. 963-973, Dec. 1982.

[14] M. Renfors, "Roundoff noise in error-feedback state-space filters," *Proc. Int. Conf. Acoustics, Speech, Signal Processing* (*ICASSP'83*), pp. 619-622, Apr. 1983.

[15] W. E. Higgins and D. C. Munson, "Optimal and suboptimal error-spectrum shaping for cascade-form digital filters," *IEEE Trans. circuits Syst.*, vol. 31, pp. 429-437, May 1984.

[16] T. I. Laakso and I. O. Hartimo, "Noise reduction in recursive digital filters using high-order error feedback," *IEEE Trans. Signal Processing*, vol. 40, pp. 1096-1107, May 1992.

[17] P. P. Vaidyanathan, "On error-spectrum shaping in state-space digital filters," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 88-92, Jan. 1985.

[18] T. Hinamoto, S. Karino, N. Kuroda and T. Kuma, "Error spectrum shaping in two-dimensional recursive digital filters," *IEEE Trans. Circuits Syst.*, vol. 46, pp. 1203-1215, Oct. 1999.

[19] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 1987.

[20] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, pp. 1115-1145, 1995.

[21] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *Manual of LMI Control Toolbox*, The MathWorks Inc., Natick, MA, 1995.

[22] J. F. Sturm, "Using SeDuMe1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, pp. 625-653, 1999.

[23] T. Kailath, *Linear Systems*, Prentice Hall, 1980.

[24] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization", *SIAM J. Optimization*, vol. 5, pp. 13-51, 1995.

[25] L. Vandenberghe and S. Boyd, "Semidefinite programming", *SIAM Review*, vol. 38, pp. 49-95, March 1996.

[26] K. C. Toh, M. J. Tood, and R. H. Tütüncü, *SDPT3 Version 2.1 – A MATLAB Software for Semidefinite Programming*, Sept. 1999.

[27] H. Wolkowicz, R. Saigal, and L. Vandenberghe (ed.), *Handbook on Semidefinite Programming*, Kluwer Academic, 2000.

[28] W.-S. Lu, "Design of FIR filters with discrete coefficients: A semidefinite programming relaxation approach," *Proc. ISCAS 2001*, vol. 2, pp. 297-300, Sydney, Austalia, May 2001.

[29] X. M. Wang, W.-S. Lu, and A. Antoniou, "A near-optimal multiuser detector for CDMA channels using semidefinite programming relaxation," *Proc. ISCAS 2001*, vol. 4, pp. 298-301, Sydney, Australia, May 2001.

[30] W.-S. Lu, "A unified approach for the design of 2-D digital filters via semidefinite programming," *Trans. IEEE Trans. Circuits Syst. I*, vol. 49, pp. 814-826, June 2002.