Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2011, Article ID 893760, 15 pages
doi:10.1155/2011/893760

Research Article

Sensitivity-Based Pole and Input-Output Errors of Linear Filters as Indicators of the Implementation Deterioration in Fixed-Point Context

Thibault Hilaire[1] and Philippe Chevrel[2]

[1] *Laboratory of Computer Science (LIP6), University Pierre & Marie Curie, 75005 Paris, France*
[2] *Institut de Recherche en Cybernétique et Communication de Nantes (UMR CNRS 6597), École des Mines de Nantes, 44321 Nantes Cedex, France*

Correspondence should be addressed to Thibault Hilaire, thibault.hilaire@lip6.fr

Received 30 June 2010; Accepted 19 November 2010

Academic Editor: Juan A. Löpez

Input-output or poles sensitivity is widely used to evaluate the resilience of a filter realization to coefficients quantization in an FWL implementation process. However, these measures do not exactly consider the various implementation schemes and are not accurate in general case. This paper generalizes the classical transfer function sensitivity and pole sensitivity measure, by taking into consideration the exact fixed-point representation of the coefficients. Working in the general framework of the specialized implicit descriptor representation, it shows how a statistical quantization error model may be used in order to define stochastic sensitivity measures that are definitely pertinent and normalized. The general framework of MIMO filters and controllers is considered. All the results are illustrated through an example.

1. Introduction

The majority of control or signal processing systems is implemented in digital general purpose processors, DSPs (Digital Signal Processors), FPGAs (Field Programmable Gate-Array), and so forth. Since these devices cannot compute with infinite precision and approximate real-number parameters with a finite binary representation, the numerical implementation of controllers (filters) leads to deterioration in characteristics and performance. This has two separate origins, corresponding to the quantization of the embedded coefficients and the round-off errors occurring during the computations. They can be formalized as parametric errors and numerical noises, respectively. This paper is focused on parametric errors, but one can refer to [1–4] for round-off noises, where measures with fixed-point consideration already exist or to [5] for interval-based characterization.

It is also well known that these Finite Word Length (FWL) effects depend on the structure of the realization. In state-space form, the realization depends on the choice of the basis of the state vector. This motivates us to investigate the coefficient sensitivity minimization problem. It has been well studied with the L_2-measure [1, 6]. However, this measure only considers how sensitive to the coefficients the transfer function is and does not investigate the coefficients quantization, which depends on the fixed-point representation used. In [6], the transfer function error is exhibited for the first time, however, only for quantized coefficients with the same binary-point position.

A common assumption in FWL error analysis is that the perturbations on the coefficients are independent and uniformly distributed random variables in the interval $[-\epsilon/2;\epsilon/2]$ with ϵ some constant depending on the wordlength. As shown in Section 4.1, this range can be different for each coefficient and depends on the coefficient itself and some fixed-point choices for the implementation. In that sense, this paper takes in consideration the different binary-point position of the coefficients in order to define a new stochastic error measure.

Making use of the Specialized Implicit Framework proposed by the authors in [7], this paper extends the stochastic approach of [8] to a much larger class of realizations, in

order to define and compute the transfer function and poles sensitivity (in both context of open- and closed-loop schemes).

The classical sensitivity analysis is introduced in Section 2 whereas the Specialized Implicit Framework is presented in Section 3. Section 4 exhibits the fixed-point implementation scheme and the new transfer function error, and Section 5 presents the pole error. A brief extension to closed-loop cases is shown in Section 6. The optimal realization problem is discussed in Section 7 with an example to illustrate theoretical results. Finally, some concluding remarks are given in Section 8.

Notations. Throughout this paper, real numbers are in lowercase, column vectors in lowercase boldface, and matrices in uppercase boldface. $\mathbf{A}^*$ will denote the conjugate, $\mathbf{A}^\top$ the transpose, $\mathbf{A}^H$ the transpose-conjugate, $\mathrm{tr}(\mathbf{A})$ the trace operator, $E\{\mathbf{A}\}$ the mean operator, $\mathrm{Re}(\mathbf{A})$ the real part, and $\mathbf{A} \times \mathbf{B}$ the Schur product of $\mathbf{A}$ and $\mathbf{B}$, respectively.

2. Classical Sensitivity Analysis

Classically, in the literature, the sensitivity analysis is performed on a state-space realization. Some other extended structures (like direct form, ρ-modal, δ-operator state-space, etc.) have been also studied, and specific sensitivity analysis has been performed for each structure.

Let $(\mathbf{A}, \mathbf{b}, \mathbf{c}, d)$ be a stable, controllable, and observable linear discrete time Single Input Single Output (SISO) state-space system, that is,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \\ y(k) &= \mathbf{c}\mathbf{x}(k) + du(k), \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$, $\mathbf{c} \in \mathbb{R}^{1\times n}$, and $d \in \mathbb{R}$. $u(k)$ is the scalar input, $y(k)$ is the scalar output, and $\mathbf{x}(k) \in \mathbb{R}^{n\times 1}$ is the state vector at time k.

Its input-output relationship is given by the scalar transfer function $h : \mathbb{C} \to \mathbb{C}$ defined by

$$h : z \longmapsto \mathbf{c}(z\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{b} + d. \tag{2}$$

2.1. Transfer Function Sensitivity Measure. The quantization of the coefficients $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, and d introduces some uncertainties leading to $\mathbf{A} + \mathbf{\Delta A}$, $\mathbf{b} + \mathbf{\Delta b}$, $\mathbf{c} + \mathbf{\Delta c}$, and $d + \Delta d$, respectively. It is common to consider the sensitivity of the transfer function with respect to the coefficients [1, 9, 10], based on the following definitions.

Definition 1 (Transfer Function Derivative). Consider $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $f : \mathbb{R}^{m\times n} \to \mathbb{C}$ differentiable with respect to all the entries of $\mathbf{X}$. The derivative of f with respect to $\mathbf{X}$ is defined by the matrix $\mathbf{S}_\mathbf{X} \in \mathbb{R}^{m\times n}$ such as

$$\frac{\partial f}{\partial \mathbf{X}} \triangleq \mathbf{S}_\mathbf{X} \quad \text{with } (\mathbf{S}_\mathbf{X})_{i,j} \triangleq \frac{\partial f}{\partial \mathbf{X}_{i,j}}. \tag{3}$$

Applied to a scalar transfer function h where $h(z)$ depends on a given matrix $\mathbf{X}$, $\partial h/\partial \mathbf{X}$ is a Multiple Inputs Multiple Outputs (MIMO) transfer function, defined by

$$\frac{\partial h}{\partial \mathbf{X}}(z) \triangleq \frac{\partial h(z)}{\partial \mathbf{X}}, \quad \forall z \in \mathbb{C}. \tag{4}$$

Definition 2 (L_2-Norm). Let $\mathbf{H} : \mathbb{C} \to \mathbb{C}^{k\times l}$ be a function of the scalar complex variable z (i.e., a MIMO transfer function). Its L_2-norm, denoted $\|\mathbf{H}\|_2$ is defined by

$$\|\mathbf{H}\|_2 \triangleq \sqrt{\frac{1}{2\pi}\int_0^{2\pi} \left\|\mathbf{H}(e^{j\omega})\right\|_F^2 d\omega}, \tag{5}$$

where $\|\mathbf{Y}\|_F$ is the Frobenius norm of the matrix $\mathbf{Y}$ defined by

$$\|\mathbf{Y}\|_F \triangleq \sqrt{\sum_{ij} \left|\mathbf{Y}_{ij}\right|^2} = \sqrt{\mathrm{tr}\,\mathbf{Y}^H\mathbf{Y}}. \tag{6}$$

In [1], Gevers and Li have proposed the L_2-sensitivity measure (denoted M_{L_2}) to *evaluate* the coefficient roundoff errors.

Definition 3 (Transfer Function Sensitivity Measure). The Transfer Function Sensitivity Measure is defined by

$$M_{L_2} \triangleq \left\|\frac{\partial h}{\partial \mathbf{A}}\right\|_2^2 + \left\|\frac{\partial h}{\partial \mathbf{b}}\right\|_2^2 + \left\|\frac{\partial h}{\partial \mathbf{c}}\right\|_2^2 + \left\|\frac{\partial h}{\partial d}\right\|_2^2. \tag{7}$$

It can be computed with Proposition 4 and the following equations

$$\begin{aligned} \frac{\partial h}{\partial \mathbf{A}}(z) &= \mathbf{G}^\top(z)\mathbf{F}^\top(z), & \frac{\partial h}{\partial \mathbf{b}}(z) &= \mathbf{G}^\top(z), \\ \frac{\partial h}{\partial \mathbf{c}}(z) &= \mathbf{F}(z), & \frac{\partial h}{\partial d}(z) &= 1 \end{aligned} \tag{8}$$

with

$$\mathbf{F}(z) \triangleq (z\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{b}, \qquad \mathbf{G}(z) \triangleq \mathbf{c}(z\mathbf{I}_n - \mathbf{A})^{-1}. \tag{9}$$

$\mathbf{F}$ and $\mathbf{G}$ can be seen as the MIMO state-space systems $(\mathbf{A}, \mathbf{b}, \mathbf{I}_n, \mathbf{0})$ and $(\mathbf{A}, \mathbf{I}_n, \mathbf{c}, \mathbf{0})$, respectively.

Proposition 4. *If* $\mathbf{H}$ *is the MIMO state-space system* $(\mathbf{K}, \mathbf{L}, \mathbf{M}, \mathbf{N})$*, then its* L_2*-norm can be computed by*

$$\begin{aligned} \|\mathbf{H}\|_2^2 &= \mathrm{tr}(\mathbf{N}\mathbf{N}^\top + \mathbf{M}\mathbf{W}_c\mathbf{M}^\top), \\ &= \mathrm{tr}(\mathbf{N}^\top\mathbf{N} + \mathbf{L}^\top\mathbf{W}_o\mathbf{L}), \end{aligned} \tag{10}$$

where $\mathbf{W}_c$ *and* $\mathbf{W}_o$ *are the controllability and observability Gramians, respectively. They are solutions to the Lyapunov equations*

$$\mathbf{W}_c = \mathbf{K}\mathbf{W}_c\mathbf{K}^\top + \mathbf{L}\mathbf{L}^\top, \qquad \mathbf{W}_o = \mathbf{K}^\top\mathbf{W}_o\mathbf{K} + \mathbf{M}^\top\mathbf{M}. \tag{11}$$

Proof. See [1]. □

Remark 5. This measure is an extension of the more tractable but less natural L_1/L_2 sensitivity measure proposed by Tavsanoglu and Thiele [10] ($\|\partial h/\partial \mathbf{A}\|_1^2$ instead of $\|\partial h/\partial \mathbf{A}\|_2^2$ in (7)).

Applying a coordinate transformation, defined by $\widetilde{\mathbf{x}}(k) \triangleq \mathcal{U}^{-1}\mathbf{x}(k)$ to the state-space system $(\mathbf{A}, \mathbf{b}, \mathbf{c}, d)$, leads to a new equivalent realization $(\mathcal{U}^{-1}\mathbf{A}\mathcal{U}, \mathcal{U}^{-1}\mathbf{b}, \mathbf{c}\mathcal{U}, d)$.

Since these two realizations are equivalent in infinite precision but are no more equivalent in finite precision (fixed-point arithmetic, floating-point arithmetic, etc.), the L_2-sensitivity then depends on $\mathcal{U}$ and is denoted $M_{L_2}(\mathcal{U})$.

It is natural to define the following problem.

Problem 1 (Optimal L_2-sensitivity problem). Considering a state-space realization $(\mathbf{A}, \mathbf{b}, \mathbf{c}, d)$, the optimal L_2-sensitivity problem consists of finding the coordinate transformation $\mathcal{U}_{\text{opt}}$ that minimizes the transfer function sensitivity measure

$$\mathcal{U}_{\text{opt}} = \underset{\mathcal{U}\ \text{invertible}}{\arg\min}\ M_{L_2}(\mathcal{U}). \tag{12}$$

In [1], it is shown that the problem has one unique solution, and a gradient method can be used to solve it.

2.2. Pole Sensitivity Measure. In addition to the transfer function sensitivity measure, some other sensitivity-based measures have been developed: the perturbations of the system poles is specially studied [11–14]. Poles are not only structuring parameters, but also indicators of the stability.

Let $(\lambda_k)_{1\leqslant k\leqslant n}$ denote the poles of the system (they are the eigenvalues of $\mathbf{A}$). The partial pole sensitivity measure Ψ_k is defined as follows:

$$\Psi_k \triangleq \left\| \frac{\partial |\lambda_k|}{\partial \mathbf{A}} \right\|_F^2. \tag{13}$$

Remark 6. The eigenvalues λ_k does not depend on $\mathbf{b}$, $\mathbf{c}$, and d, so the terms $\partial|\lambda_k|/\partial\mathbf{b}$, $\partial|\lambda_k|/\partial\mathbf{c}$, and $\partial|\lambda_k|/\partial d$ are not considered in the definition (13) (they are null).

Moreover, the moduli of the poles is considered because the FWL error that can cause a stable system to become unstable is determined by how close the pole are to 1 and how sensitive they are to the parameter perturbations. So, the partial pole sensitivities are combined in a global Pole Sensitivity Measure [15].

Definition 7 (Pole Sensitivity Measure). The Pole Sensitivity Measure Ψ is defined by

$$\Psi \triangleq \sum_{k=1}^{n} \omega_k \Psi_k, \tag{14}$$

where $(\omega_k)_{1\leqslant k\leqslant n}$ are the weighting coefficients. Generally

$$\omega_k = \frac{1}{1-|\lambda_k|}, \quad \forall 1 \leqslant k \leqslant n \tag{15}$$

to give more weight for the poles closed to the unit circle [15].

Table 1: M_{L_2}-sensitivity measure and transfer function error for different realizations.

Realization	M_{L_2}	$\|h - h^\dagger\|_2$
$\mathbf{X}_1$	$3.521e+5$	1.8323
$\mathbf{X}_2$	$1.142e+6$	1.4697
$\mathbf{X}_3$	$4.287e+5$	1.9852

The pole sensitivity measure is also used in closed-loop context, in some stability-related measures [14, 16], see Section 6.

2.3. Limitations. The classical measures are based on the sensitivity with respect to the coefficients. Since it was classically assumed [1, 6, 12] that the perturbations on the coefficients were independent and uniformly distributed random variable in the interval $[-\epsilon/2; \epsilon/2]$ with ϵ some positive constant depending on the wordlength only, it was natural to consider the sensitivity as a good evaluation of the overall deterioration (transfer function moving or pole moving). But this is a reasonable consideration only if the coefficients all have the same magnitude order. It is generally not the case in practice.

To illustrate this point, let us consider the first-order transfer function $h : z \mapsto 100/(z - 0.8)$. The three following realizations are state-space realizations of this transfer function, with coefficient quantized in 8-bit fixed-point (in bold are the integer values coding for the coefficients, the exponent part being implicit, see Section 4.1)

$$\begin{aligned}
\mathbf{X}_1 &= \left(\begin{array}{c|c} \mathbf{102}\cdot 2^{-7} & \mathbf{80}\cdot 2^{-3} \\ \hline \mathbf{80}\cdot 2^{-3} & 0 \end{array}\right), \\
\mathbf{X}_2 &= \left(\begin{array}{c|c} \mathbf{102}\cdot 2^{-7} & \mathbf{66}\cdot 2^{3} \\ \hline \mathbf{96}\cdot 2^{-9} & 0 \end{array}\right), \\
\mathbf{X}_3 &= \left(\begin{array}{c|c} \mathbf{102}\cdot 2^{-7} & \mathbf{76}\cdot 2^{-7} \\ \hline \mathbf{83}\cdot 2^{1} & 0 \end{array}\right).
\end{aligned} \tag{16}$$

One can remark that all the coefficients do not have the same exponent (these realizations are classical realizations, that is, balanced, arbitrary-scaled, and L_2-scaled, resp.). The quantization error of these coefficients will be completely different, since his quantization error is equal to their power-of-2 part, for example,

$$\mathbf{\Delta X_1} = \left(\begin{array}{c|c} 2^{-7} & 2^{-7} \\ \hline 2^{1} & 0 \end{array}\right). \tag{17}$$

So, for the same sensitivity, the quantization of coefficients with higher magnitude will more affect the transfer function and the poles.

But the sensitivity measures previously presented cannot take this into consideration. Table 1 exhibits the transfer function sensitivity measure and the transfer function error $\|h - h^\dagger\|_2$ (where $h^\dagger$ is the transfer function with quantized coefficients) for these three different realizations. In that case, $\mathbf{X}_2$ has the highest L_2-sensitivity, but is yet the most resilient to the fixed-point implementation considered.

3. Specialized Implicit Framework

3.1. Definitions. Many controller/filter forms, such as lattice filters and δ-operator controllers, make use of intermediate variables, and hence cannot be expressed in the traditional state-space form. The SIF has been proposed in order to model a much wider class of discrete-time linear time-invariant controller implementations than the classical state-space form. It is presented here for MIMO filters/controllers.

The model takes the form of an implicit state-space realization [17] specialized according to

$$\begin{pmatrix} \mathbf{J} & \mathbf{0} & \mathbf{0} \\ -\mathbf{K} & \mathbf{I}_n & \mathbf{0} \\ -\mathbf{L} & \mathbf{0} & \mathbf{I}_p \end{pmatrix} \begin{pmatrix} \mathbf{t}(k+1) \\ \mathbf{x}(k+1) \\ \mathbf{y}(k) \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{M} & \mathbf{N} \\ \mathbf{0} & \mathbf{P} & \mathbf{Q} \\ \mathbf{0} & \mathbf{R} & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{t}(k) \\ \mathbf{x}(k) \\ \mathbf{u}(k) \end{pmatrix}, \tag{18}$$

where $\mathbf{J} \in \mathbb{R}^{l\times l}$, $\mathbf{K} \in \mathbb{R}^{n\times l}$, $\mathbf{L} \in \mathbb{R}^{p\times l}$, $\mathbf{M} \in \mathbb{R}^{l\times n}$, $\mathbf{N} \in \mathbb{R}^{l\times m}$, $\mathbf{P} \in \mathbb{R}^{n\times n}$, $\mathbf{Q} \in \mathbb{R}^{n\times m}$, $\mathbf{R} \in \mathbb{R}^{p\times n}$, $\mathbf{S} \in \mathbb{R}^{p\times m}$, $\mathbf{t}(k) \in \mathbb{R}^l$, $\mathbf{x}(k) \in \mathbb{R}^n$, $\mathbf{u}(k) \in \mathbb{R}^m$, $\mathbf{y}(k) \in \mathbb{R}^p$, and the matrix $\mathbf{J}$ is lower triangular with 1's on the main diagonal. Note that $\mathbf{x}(k+1)$ is the state-vector and is stored from one step to the next, whilst the vector $\mathbf{t}$ plays a particular role as $\mathbf{t}(k+1)$ is independent of $\mathbf{t}(k)$ (it is here defined as the vector of intermediary variables). The particular structure of $\mathbf{J}$ allows the expression of how the computations are decomposed with intermediates results that could be reused.

Remark 8. In that sense, the SIF can be seen as an extension of the factored state-space representation (FSSR) proposed by Roberts and Mullis [18] as

$$\begin{pmatrix} \mathbf{x}(k+1) \\ \mathbf{y}(k) \end{pmatrix} = \prod_{i=1}^{N} \begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix} \begin{pmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{pmatrix}. \tag{19}$$

Indeed, the factored expression

$$\mathbf{v} = \mathbf{M}_1\mathbf{M}_0\mathbf{w} \tag{20}$$

can be rewritten by decomposing the computations $\mathbf{M}_0\mathbf{w}$ and introducing intermediate vector (and left term)

$$\begin{pmatrix} \mathbf{I} & 0 \\ -\mathbf{M}_1 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_0 \\ \mathbf{0} \end{pmatrix} \mathbf{w}. \tag{21}$$

So, the left term of the implicit state space (18) can represent factored state space. But it could also represent not only linear but also affine expression like $\mathbf{v} = \mathbf{M}_1(\mathbf{M}_0\mathbf{w} + \mathbf{n}_0) + \mathbf{n}_1$ and more. In fact, all the algorithms with additions, shifts, and multiplication by a constant can be represented.

It is implicitly assumed throughout the paper that the computations associated with the realization (18) are executed in row order, giving the following algorithm:

$$\begin{aligned} &\text{(i)} \quad \mathbf{J}\cdot\mathbf{t}(k+1) \longleftarrow \mathbf{M}\cdot\mathbf{x}(k) + \mathbf{N}\cdot\mathbf{u}(k), \\ &\text{(ii)} \quad \mathbf{x}(k+1) \longleftarrow \mathbf{K}\cdot\mathbf{t}(k+1) + \mathbf{P}\cdot\mathbf{x}(k) + \mathbf{Q}\cdot\mathbf{u}(k), \\ &\text{(iii)} \quad \mathbf{y}(k) \longleftarrow \mathbf{L}\cdot\mathbf{t}(k+1) + \mathbf{R}\cdot\mathbf{x}(k) + \mathbf{S}\cdot\mathbf{u}(k). \end{aligned} \tag{22}$$

Note that in practice, steps (ii) and (iii) could be exchanged to reduce the computational delay. Also note that there is no need to compute $\mathbf{J}^{-1}$ because the computations are executed in row order and $\mathbf{J}$ is lower triangular with 1's on the main diagonal.

Equation (18) is equivalent in infinite precision to the state-space system $(\mathbf{A}_\mathbf{Z}, \mathbf{B}_\mathbf{Z}, \mathbf{C}_\mathbf{Z}, \mathbf{D}_\mathbf{Z})$ with $\mathbf{A}_\mathbf{Z} \in \mathbb{R}^{n\times n}$, $\mathbf{B}_\mathbf{Z} \in \mathbb{R}^{n\times m}$, $\mathbf{C}_\mathbf{Z} \in \mathbb{R}^{p\times n}$, and $\mathbf{D}_\mathbf{Z} \in \mathbb{R}^{p\times m}$, where

$$\begin{aligned} \mathbf{A}_\mathbf{Z} &\triangleq \mathbf{K}\mathbf{J}^{-1}\mathbf{M} + \mathbf{P}, & \mathbf{B}_\mathbf{Z} &\triangleq \mathbf{K}\mathbf{J}^{-1}\mathbf{N} + \mathbf{Q}, \\ \mathbf{C}_\mathbf{Z} &= \mathbf{L}\mathbf{J}^{-1}\mathbf{M} + \mathbf{R}, & \mathbf{D}_\mathbf{Z} &\triangleq \mathbf{L}\mathbf{J}^{-1}\mathbf{N} + \mathbf{S}. \end{aligned} \tag{23}$$

This state-space system corresponds to a different parametrization than (18) (the finite-precision implementation of the state-space $(\mathbf{A}_\mathbf{Z}, \mathbf{B}_\mathbf{Z}, \mathbf{C}_\mathbf{Z}, \mathbf{D}_\mathbf{Z})$ will cause different numerical deterioration than for (18)). The associated system transfer function $\mathbf{H}$ is given by

$$\mathbf{H} : z \longmapsto \mathbf{C}_\mathbf{Z}(z\mathbf{I}_n - \mathbf{A}_\mathbf{Z})^{-1}\mathbf{B}_\mathbf{Z} + \mathbf{D}_\mathbf{Z}. \tag{24}$$

A complete framework for the description of all digital controller implementations can be developed by using the following definitions. For further details, see [7].

Definition 9. A *realization* of a transfer matrix H is entirely defined by the data $\mathbf{Z}$, l, m, n, and p, where $\mathbf{Z} \in \mathbb{R}^{(l+n+p)(l+n+m)}$ is partitioned according to

$$\mathbf{Z} \triangleq \begin{pmatrix} -\mathbf{J} & \mathbf{M} & \mathbf{N} \\ \mathbf{K} & \mathbf{P} & \mathbf{Q} \\ \mathbf{L} & \mathbf{R} & \mathbf{S} \end{pmatrix} \tag{25}$$

and l, m, n, and p are the matrix dimensions given previously.

The notation $\mathbf{Z}$ is introduced to make the further developments more compact (see (44), (70), etc.).

3.2. Equivalent Realizations. In order to exploit the potential offered by the specialized implicit form in improving implementations, it is necessary to describe sets of equivalent system realizations. The *Inclusion Principle* introduced by Ikeda and Siljak [19] in the context of decentralized control, has been extended to the Specialized Implicit Form in order to characterize equivalent classes of realizations [7]. Although this extension gives the formal description of equivalent classes, it is of practical interest to consider only realizations with the same dimensions, where transformation from one realization to another is only a similarity transformation.

Proposition 10. *Consider a realization* $\mathbf{Z}_0$.

All the realizations $\mathbf{Z}_1$ *with*

$$\mathbf{Z}_1 = \begin{pmatrix} \mathcal{Y} & & \\ & \mathcal{U}^{-1} & \\ & & \mathbf{I}_p \end{pmatrix} \mathbf{Z}_0 \begin{pmatrix} \mathcal{W} & & \\ & \mathcal{U} & \\ & & \mathbf{I}_m \end{pmatrix} \tag{26}$$

and $\mathcal{U}$, $\mathcal{W}$, $\mathcal{Y}$ *are nonsingular matrices, are equivalent to* $\mathbf{Z}_0$, *and share the same complexity (i.e., generically the same amount of computation).*

It is also possible to just consider a subset of similarity transformations that preserve a particular structure, by adding specific constraints on $\mathcal{U}$, $\mathcal{W}$, or $\mathcal{Y}$.

This will allow us to consider all the realizations $\mathbf{Z}$ with a given transfer function as input-output relationship and a given structure, and find the most suitable for the implementation.

3.3. Examples. Here are some examples of structured realizations expressed with the SIF.

3.3.1. Cascaded State-Space. The cascade form is a common realization for filter implementation. It generally has good FWL properties compared to the direct forms. For cascade form, the filter is decomposed into a number of lower order (usually first- and second-order) transfer function blocks connected in series. For the next example, we consider two standard q-operator state-space blocks connected in series as shown in Figure 1.

If two state-space realizations $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$ and $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{D}_2)$ are cascaded together, then it leads to the following realization

$$\mathbf{Z} = \left(\begin{array}{c|cc|c} -\mathbf{I} & \mathbf{C}_1 & \mathbf{0} & \mathbf{D}_1 \\ \hline \mathbf{0} & \mathbf{A}_1 & \mathbf{0} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{0} & \mathbf{A}_2 & \mathbf{0} \\ \hline \mathbf{D}_2 & \mathbf{0} & \mathbf{C}_2 & \mathbf{0} \end{array}\right). \tag{27}$$

The output of first block is computed in the intermediate variable and used as the input of the second block.

The main point is that if we consider the equivalent state-space realization, with parameters

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2\mathbf{D}_1 \end{pmatrix}, \\ \mathbf{C} &= \begin{pmatrix} \mathbf{D}_2\mathbf{C}_1 & \mathbf{C}_2 \end{pmatrix}, \qquad \mathbf{D} = \mathbf{D}_2\mathbf{D}_1, \end{aligned} \tag{28}$$

the parametrization is not the one used in the computations, and the FWL effects will not be the one of the implemented version.

Remark 11. The cascade structuration can be easily extended to a series of specialized implicit forms and to general multiple cascaded systems.

3.3.2. δ-Realizations. Consider the δ-state-space realization

$$\begin{aligned} \delta[\mathbf{x}(k)] &= \mathbf{A}_\delta \mathbf{x}(k) + \mathbf{B}_\delta \mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C}_\delta \mathbf{x}(k) + \mathbf{D}_\delta \mathbf{u}(k), \end{aligned} \tag{29}$$

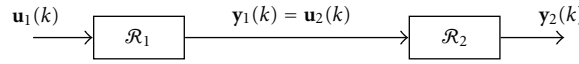

Figure 1: Cascade form.

with $\delta = (q-1)/\Delta$, $\Delta \in \mathbb{R}_{+*}$, and q is the shift operator [1, 20, 21]. This operator has been introduced as a unifying time operator, between discrete and continuous time. But it is used in practice for its interesting numerical properties in FWL context.

This realization should be implemented with the following algorithm:

$$\begin{aligned} &\text{(i)} \quad \mathbf{t} \longleftarrow \mathbf{A}_\delta \cdot \mathbf{x}(k) + \mathbf{B}_\delta \cdot \mathbf{u}(k), \\ &\text{(ii)} \quad \mathbf{x}(k+1) \longleftarrow \mathbf{x}(k) + \Delta \cdot \mathbf{t}, \\ &\text{(iii)} \quad \mathbf{y}(k) \longleftarrow \mathbf{C}_\delta \cdot \mathbf{x}(k) + \mathbf{D}_\delta \cdot \mathbf{u}(k), \end{aligned} \tag{30}$$

where $\mathbf{t}$ is an intermediate variable. This could be modelled with the specialized implicit form as

$$\begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ -\Delta\mathbf{I}_n & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_p \end{pmatrix} \begin{pmatrix} \mathbf{t}(k+1) \\ \mathbf{x}(k+1) \\ \mathbf{y}(k) \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{A}_\delta & \mathbf{B}_\delta \\ \mathbf{0} & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_\delta & \mathbf{D}_\delta \end{pmatrix} \begin{pmatrix} \mathbf{t}(k) \\ \mathbf{x}(k) \\ \mathbf{u}(k) \end{pmatrix}. \tag{31}$$

3.3.3. ρ Direct-Form II Transposed (ρDFIIt). Li et al. [22–24] have presented a new sparse structure called ρDFIIt. This is a generalization of the transposed direct-form II structure with the conventional shift and the δ-operator and is similar to that of [25]. It is a sparse realization (with $3n+1$ parameters when n is the order of the controller), leading so to an economic (few computations) implementation that could be very numerically efficient. As we will see later, this realization has n extra degrees of freedom that can be used to find an *optimal* realization within its particular structuration.

Let us define

$$\begin{aligned} \rho_i &: z \longmapsto \frac{z - \gamma_i}{\Delta_i}, \quad 1 \leqslant i \leqslant n, \\ \varrho_i &: z \longmapsto \prod_{j=1}^{i} \rho_j(z), \quad 1 \leqslant i \leqslant n, \end{aligned} \tag{32}$$

where $(\gamma_i)_{1\leqslant i\leqslant n}$ and $(\Delta_i > 0)_{1\leqslant i\leqslant n}$ are two sets of constants. Let $(a_i)_{1\leqslant i\leqslant n}$ and $(b_i)_{0\leqslant i\leqslant n}$ be the coefficient sets of the transfer function, using the shift operator

$$h : z \longmapsto \frac{b_0 + b_1 z^{-1} + \cdots + b_{n-1} z^{-n+1} + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_{n-1} z^{-n+1} + a_n z^{-n}}. \tag{33}$$

Therefore, h can be *reparametrized* with $(\alpha_i)_{1\leqslant i\leqslant n}$ and $(\beta_i)_{0\leqslant i\leqslant n}$ as follows:

$$h(z)=\frac{\beta_0+\beta_1\varrho_1^{-1}(z)+\cdots+\beta_{n-1}\varrho_{n-1}^{-1}(z)+\beta_n\varrho_n^{-1}(z)}{1+\alpha_1\varrho_1^{-1}(z)+\cdots+\alpha_{n-1}\varrho_{n-1}^{-1}(z)+\alpha_n\varrho_n^{-1}(z)}. \tag{34}$$

Denoting

$$\mathbf{v}_a\triangleq\begin{pmatrix}1\\a_1\\\vdots\\a_n\end{pmatrix},\qquad \mathbf{v}_b\triangleq\begin{pmatrix}b_0\\b_1\\\vdots\\b_n\end{pmatrix},$$
$$\mathbf{v}_\alpha\triangleq\begin{pmatrix}1\\\alpha_1\\\vdots\\\alpha_n\end{pmatrix},\qquad \mathbf{v}_\beta\triangleq\begin{pmatrix}\beta_0\\\beta_1\\\vdots\\\beta_n\end{pmatrix}, \tag{35}$$

the parameters $(a_i)_{1\leqslant i\leqslant n}$, $(b_i)_{0\leqslant i\leqslant n}$, $(\alpha_i)_{1\leqslant i\leqslant n}$, and $(\beta_i)_{0\leqslant i\leqslant n}$ are related [23] according to

$$\begin{aligned}\mathbf{v}_a&=\kappa\boldsymbol{\Omega}\mathbf{v}_\alpha,\\ \mathbf{v}_b&=\kappa\boldsymbol{\Omega}\mathbf{v}_\beta,\end{aligned} \tag{36}$$

where $\kappa\triangleq\prod_{i=1}^{n}\Delta_i$ and $\boldsymbol{\Omega}\in\mathbb{R}^{n+1\times n+1}$ is a lower triangular matrix whose ith column is determined by the coefficients of the z-polynomial $\prod_{j=i}^{n}\rho_j(z)$ for $1\leqslant i\leqslant n$ and with $\boldsymbol{\Omega}_{n+1,n+1}=1$.

Equation (34) can be, for example, implemented with a transposed direct form II (see Figure 2), and each operator ρ_i^{-1} can be implemented as shown in Figure 3 (each ϱ_k^{-1} is obtained by cascading the $(\rho_i^{-1})_{1\leqslant i\leqslant k}$). Clearly, when $\gamma_i=0$, $\Delta_i=1$ $(1\leqslant i\leqslant n)$, Figure 2 is the conventional transposed direct form II. When $\gamma_i=1$, $\Delta_i=\Delta$ $(1\leqslant i\leqslant n)$, one gets the δ transposed direct form II. This form was first proposed as an unification for the shift-direct form II transposed and the δ-direct form II transposed. It is now used to exploit the n extradegrees of freedom given by the choice of the parameters $(\gamma_i)_{1\leqslant i\leqslant n}$.

The corresponding algorithm is

$$\begin{aligned}&\text{(i)}\quad y(k)\longleftarrow\beta_0u(k)+w_1(k),\\ &\text{(ii)}\quad w_i(k)\longleftarrow\rho_i^{-1}[\beta_iu(k)-\alpha_iy(k)+w_{i+1}(k)],\\ &\text{(iii)}\quad w_n(k)\longleftarrow\rho_n^{-1}[\beta_nu(k)-\alpha_ny(k)].\end{aligned} \tag{37}$$

By introducing the intermediate variables needed to realize the ρ_i^{-1} operator (according to $\rho_i^{-1}=(1/(q^{-1}-\gamma_i))\Delta_i$, with the multiplication by Δ_i done last, see Figure 3), the ρDFIIt can be rewritten as

$$\begin{aligned}\mathbf{t}&=\begin{pmatrix}\Delta_1&&&\\&\Delta_2&&\\&&\ddots&\\&&&\Delta_n\end{pmatrix}\mathbf{x}(k)+\begin{pmatrix}\beta_0\\0\\\vdots\\0\end{pmatrix}u(k),\\ \mathbf{x}(k+1)&=\begin{pmatrix}-\alpha_1&1&&\\-\alpha_2&0&\ddots&\\\vdots&&\ddots&1\\-\alpha_n&&&0\end{pmatrix}\mathbf{t},\\ &\quad+\begin{pmatrix}\gamma_1&&&\\&\gamma_2&&\\&&\ddots&\\&&&\gamma_n\end{pmatrix}\mathbf{x}(n)+\begin{pmatrix}\beta_1\\\beta_2\\\vdots\\\beta_n\end{pmatrix}u(k),\\ y(k)&=\begin{pmatrix}1&0&\cdots&0\end{pmatrix}\mathbf{t}.\end{aligned} \tag{38}$$

Within the SIF Framework, the ρDFIIt form is described by

$$\mathbf{Z}=\left(\begin{array}{cccc|ccccc}-1&&&&\Delta_1&&&&\beta_0\\&\ddots&&&&\Delta_2&&&0\\&&\ddots&&&&\ddots&&\vdots\\&&&-1&&&&\Delta_n&0\\\hline-\alpha_1&1&&&\gamma_1&&&&\beta_1\\-\alpha_2&0&\ddots&&&\gamma_2&&&\beta_2\\\vdots&&\ddots&1&&&\ddots&&\vdots\\-\alpha_n&&&0&&&&\gamma_n&\beta_n\\\hline1&0&\cdots&0&0&\cdots&\cdots&0&0\end{array}\right). \tag{39}$$

Remark 12. Thanks to the SIF, there is no need to use another operator unlike the shift operator.

4. Sensitivity-Based Transfer Function Error

4.1. Fixed-Point Implementation. In this article, the notation (β,γ) is used for the fixed-point representation of a variable or coefficient (2's complement scheme), according to Figure 4. β is the total wordlength of the representation in bits, whereas γ is the wordlength of the fractional part (it determines the position of the binary-point). They are fixed for each variable (input, states, output) and each coefficient, and implicit (unlike the floating-point representation). β and γ will be suffixed by the variable/coefficient they refer to. These parameters could be scalars, vectors, or matrices, according to the variables they refer to.

Let us suppose that the coefficients wordlength $\beta_{\mathbf{Z}}$ is given (in FPGA or ASIC, it is of interest to consider

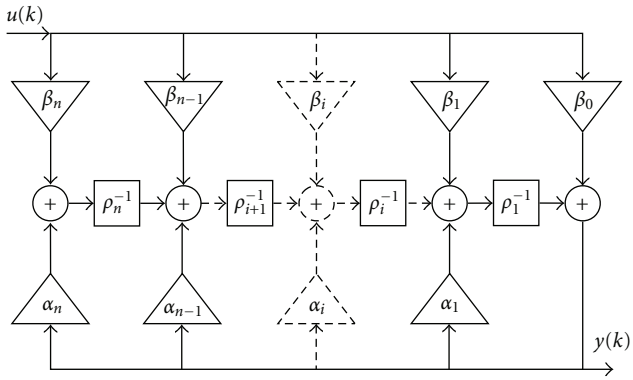

Figure 2: Generalized ρ Direct Form II.

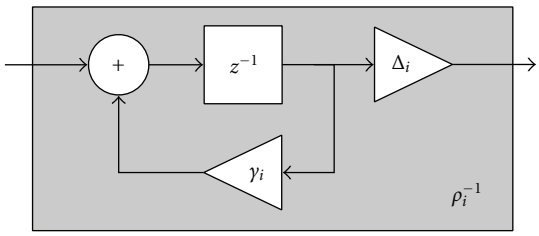

Figure 3: Realization of operator ρ_i^{-1}.

the wordlength as optimization variables, in order to find hardware realizations that minimize hardware criteria like power consumption or surface, under certain numerical accuracy constraints, like L_2-sensitivity ones [26]. This is not considered here). Then, the coefficient $\mathbf{Z}_{ij}$ is represented in fixed point by $(\beta_{\mathbf{Z}_{ij}}, \gamma_{\mathbf{Z}_{ij}})$ with

$$\gamma_{\mathbf{Z}_{ij}} = \beta_{\mathbf{Z}_{ij}} - 2 - \left\lfloor \log_2 \left| \mathbf{Z}_{ij} \right| \right\rfloor, \tag{40}$$

where the $\lfloor a \rfloor$ operation rounds a to the nearest integer less or equal to a (for positive numbers $\lfloor a \rfloor$ is the integer part).

Remark 13. The binary point position is not defined for null coefficients; however, this is no problem because these coefficients will not be represented in the final algorithm (the null multiplications are removed).

So, in order to consider coefficients that will be quantized without error, we introduced a *weighting* matrix $\delta_{\mathbf{Z}}$ such that

$$(\delta_{\mathbf{Z}})_{ij} \triangleq \begin{cases} 0 & \text{if } \mathbf{Z}_{ij} \text{ is exactly implemented} \\ 1 & \text{otherwise.} \end{cases} \tag{41}$$

The exactly implemented coefficients are 0 and the positive and negative powers of 2 (including ± 1).

Remark 14. In some specific computational cases the fixed-point representation chosen for the coefficients is not always the best one as defined in (40). For example, in the *Roundoff Before Multiplication* scheme, some extraquantizations are added to the coefficients, in order to avoid shift operations after multiplications [2]. Only the classical case (corresponding to the *Roundoff After Multiplication*) is considered here, as defined by (40).

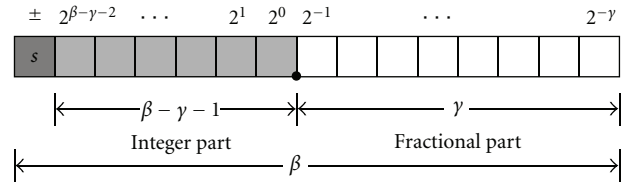

Figure 4: Fixed-point representation.

Remark 15. It is also possible to choose any $\gamma_{\mathbf{Z}_{ij}}$ such that $\gamma_{\mathbf{Z}_{ij}} \leqslant \beta_{\mathbf{Z}_{ij}} - 2 - \lfloor \log_2 |\mathbf{Z}_{ij}| \rfloor$ (e.g., choose the same binary-point position for all the the coefficients, given by the binary-point position of the coefficient with highest magnitude). But in that case, the coefficients could be coded with less meaningful bits and have a higher relative error. When the ratio between the greatest and lowest magnitude is too high, then underflows occur for the lowest coefficients that cannot be represented. For example, this is common for the Direct Form realizations with high (or low) L_2-gain.

During the quantization process, the coefficients are changed from $\mathbf{Z}$ into $\mathbf{Z}^{\dagger} \triangleq \mathbf{Z} + \mathbf{\Delta Z}$. For a rounding quantization, the $(\mathbf{\Delta Z}_{i,j})$ are independent centered random variables uniformly distributed [27, 28] within the ranges $-2^{-\gamma_{\mathbf{Z}_{ij}}-1} \leqslant \mathbf{\Delta Z}_{i,j} < 2^{-\gamma_{\mathbf{Z}_{ij}}-1}$, so their second-order moments are given by

$$\begin{aligned} \sigma^2_{\mathbf{\Delta Z}_{ij}} &\triangleq E\left\{ \left(\mathbf{\Delta Z}_{ij} \right)^2 \right\} \\ &= \frac{2^{-2\gamma_{\mathbf{Z}_{ij}}}}{12} \delta_{\mathbf{Z}_{ij}} \end{aligned} \tag{42}$$

(exactly implemented coefficients are not changed by the quantization).

4.2. Sensitivity-Based Transfer Function Error. As a consequence, the sensitivity of each coefficient should not be considered with the same weight, since there is no special reason for the $(\mathbf{\Delta Z}_{ij})$ to be all in the same range and share the same binary-point position. So it is interesting to evaluate how the transfer function is changed from $\mathbf{H}$ to $\mathbf{H}^{\dagger} \triangleq \mathbf{H} + \mathbf{\Delta H}$ by the coefficient quantization, rather than evaluate only its sensitivity.

By an extension of the SISO state-space definition given in [6], this degradation can be evaluated in a statistical way with the following definition.

Definition 16 (Sensitivity-Based Transfer Function Error). A measure of the transfer function error can be statistically defined by

$$\sigma^2_{\Delta \mathbf{H}} \triangleq \frac{1}{2\pi} \int_0^{2\pi} E\left\{ \left\| \mathbf{\Delta H}\left(e^{j\omega} \right) \right\|_F^2 \right\} d\omega. \tag{43}$$

Remark 17. This definition was introduced by Hinamoto et al. in [6], but under the assumption that the $\mathbf{\Delta Z}_{ij}$ all share the same variance. See Section 4.3.

The transfer function error is a tractable measure that can be evaluated with the two following propositions.

Proposition 18. *The sensitivity-based transfer function error of a realization* $\mathbf{Z}$*, with* $\mathbf{H}$ *as a transfer function, can be computed by*

$$\sigma^2_{\Delta\mathbf{H}} = \left\| \frac{\delta\mathbf{H}}{\delta\mathbf{Z}} \times \Xi_{\mathbf{Z}} \right\|^2_F, \tag{44}$$

where

(i) $\delta\mathbf{H}/\delta\mathbf{Z} \in \mathbb{R}^{(l+n+p)\times(l+n+m)}$ *is the transfer function sensitivity matrix (previously introduced in [7]) defined by*

$$\left(\frac{\delta\mathbf{H}}{\delta\mathbf{Z}}\right)_{ij} \triangleq \left\| \frac{\partial\mathbf{H}}{\partial\mathbf{Z}_{ij}} \right\|_2, \tag{45}$$

(ii) $\Xi_{\mathbf{Z}} \in \mathbb{R}^{(l+n+p)\times(l+n+m)}$ *is defined by*

$$\Xi_{\mathbf{Z}ij} \triangleq \begin{cases} \dfrac{2^{-\beta_{\mathbf{Z}_{ij}}+1}}{\sqrt{3}} \left\lfloor \mathbf{Z}_{ij} \right\rfloor_2 (\delta_{\mathbf{Z}})_{ij} & \text{if } \mathbf{Z}_{ij} \neq 0 \\ 0 & \text{if } \mathbf{Z}_{ij} = 0, \end{cases} \tag{46}$$

(iii) $\lfloor x \rfloor_2$ *is the nearest power of 2 lower than* $|x|$*:*

$$\lfloor x \rfloor_2 \triangleq 2^{\lfloor \log_2 |x| \rfloor}, \quad \forall x \in \mathbb{R}. \tag{47}$$

Proof. A first-order approximation gives

$$\Delta\mathbf{H}(z) = \sum_{i,j} \frac{\partial\mathbf{H}}{\partial\mathbf{Z}_{ij}}(z)\Delta\mathbf{Z}_{ij}, \quad \forall z \in \mathbb{C}. \tag{48}$$

Hence, for all $\omega \in [0, 2\pi]$,

$$\begin{aligned} E\left\{ \left\| \Delta\mathbf{H}\left(e^{j\omega}\right) \right\|^2_F \right\} &= E\left\{ \left\| \sum_{i,j} \frac{\partial\mathbf{H}}{\partial\mathbf{Z}_{ij}}\left(e^{j\omega}\right)\Delta\mathbf{Z}_{ij} \right\|^2_F \right\} \\ &= E\left\{ \sum_{k,l} \left| \sum_{i,j} \frac{\partial\mathbf{H}_{kl}}{\partial\mathbf{Z}_{ij}}\left(e^{j\omega}\right)\Delta\mathbf{Z}_{ij} \right|^2 \right\} \\ &= \sum_{i,j}\sum_{k,l} E\left\{ \left| \frac{\partial\mathbf{H}_{kl}}{\partial\mathbf{Z}_{ij}}\left(e^{j\omega}\right)\Delta\mathbf{Z}_{ij} \right|^2 \right\} \\ &\quad + \sum_{i,j}\sum_{k,l}\sum_{\substack{r,s \\ r\neq i \\ s\neq j}} E\left\{ \frac{\partial\mathbf{H}_{kl}}{\partial\mathbf{Z}_{ij}}\left(e^{j\omega}\right)\Delta\mathbf{Z}_{ij} \frac{\partial\mathbf{H}_{kl}}{\partial\mathbf{Z}_{rs}}\left(e^{j\omega}\right)\Delta\mathbf{Z}_{rs} \right\} \\ &= \sum_{i,j}\sum_{k,l} \left| \frac{\partial\mathbf{H}_{kl}}{\partial\mathbf{Z}_{ij}}\left(e^{j\omega}\right) \right|^2 \sigma^2_{\Delta\mathbf{Z}_{ij}}, \end{aligned} \tag{49}$$

because the random variables $(\Delta\mathbf{Z})_{ij}$ are all independent and centered. Then,

$$\begin{aligned} \sigma^2_{\Delta\mathbf{H}} &= \sum_{i,j} \sigma^2_{\Delta\mathbf{Z}_{ij}} \frac{1}{2\pi} \int_0^{2\pi} \left\| \frac{\partial\mathbf{H}}{\partial\mathbf{Z}_{ij}}\left(e^{j\omega}\right) \right\|^2_F d\omega \\ &= \sum_{ij} \left\| \frac{\partial\mathbf{H}}{\partial\mathbf{Z}_{ij}} \right\|^2_2 \sigma^2_{\Delta\mathbf{Z}_{ij}}. \end{aligned} \tag{50}$$

Finally, considering (40) and (42) for nonnull coefficients, we get

$$\sigma^2_{\Delta\mathbf{Z}_{ij}} = \frac{4}{3} 2^{-2\beta_{\mathbf{Z}_{ij}}} \left\lfloor \mathbf{Z}_{ij} \right\rfloor^2_2 (\delta_{\mathbf{Z}})_{ij}. \tag{51}$$

□

Remark 19. This proposition is the extension of Proposition 2 in [10] to the SIF and MIMO transfer function.

Proposition 20. *The transfer function sensitivity* $\partial\mathbf{H}/\partial\mathbf{Z}$ *can be explicited by*

$$\frac{\partial\mathbf{H}}{\partial\mathbf{Z}} = \mathbf{H}_1 \circledast \mathbf{H}_2, \tag{52}$$

where $\circledast$ *is the operator defined by*

$$\mathbf{A} \circledast \mathbf{B} \triangleq \mathrm{Vec}(\mathbf{A}) \cdot \left[\mathrm{Vec}\,(\mathbf{B})^\top \right]^\top, \tag{53}$$

$\mathrm{Vec}(\cdot)$ *is the classical operator that vectorizes a matrix, and* $\mathbf{H}_1$ *and* $\mathbf{H}_2$ *are defined by*

$$\begin{aligned} \mathbf{H}_1 &: z \longmapsto \mathbf{C}_{\mathbf{Z}}(z\mathbf{I}_n - \mathbf{A}_{\mathbf{Z}})^{-1}\mathbf{M}_1 + \mathbf{M}_2, \\ \mathbf{H}_2 &: z \longmapsto \mathbf{N}_1(z\mathbf{I}_n - \mathbf{A}_{\mathbf{Z}})^{-1}\mathbf{B}_{\mathbf{Z}} + \mathbf{N}_2, \end{aligned} \tag{54}$$

with

$$\begin{aligned} \mathbf{M}_1 &\triangleq \begin{pmatrix} \mathbf{K}\mathbf{J}^{-1} & \mathbf{I}_n & \mathbf{0} \end{pmatrix}, \qquad \mathbf{M}_2 \triangleq \begin{pmatrix} \mathbf{L}\mathbf{J}^{-1} & \mathbf{0} & \mathbf{I}_p \end{pmatrix}, \\ \mathbf{N}_1 &\triangleq \begin{pmatrix} \mathbf{J}^{-1}\mathbf{M} \\ \mathbf{I}_n \\ \mathbf{0} \end{pmatrix}, \qquad \mathbf{N}_2 \triangleq \begin{pmatrix} \mathbf{J}^{-1}\mathbf{N} \\ \mathbf{0} \\ \mathbf{I}_m \end{pmatrix}. \end{aligned} \tag{55}$$

The dimensions of $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{N}_1$, *and* $\mathbf{N}_2$ *are, respectively,* $n \times (l+n+p)$, $m \times (l+n+p)$, $(l+n+m) \times n$, *and* $(l+n+m) \times p$.

The transfer function sensitivity matrix $\delta\mathbf{H}/\delta\mathbf{Z}$ *can be computed by*

$$\left(\frac{\delta\mathbf{H}}{\delta\mathbf{Z}}\right)_{i,j} = \left\| \mathbf{H}_1\mathbf{E}_{i,j}\mathbf{H}_2 \right\|_2, \tag{56}$$

where $\mathbf{E}_{i,j}$ *is the matrix of appropriate size with all elements being 0 except the* (i, j)*th element which is unity.*

The system $\mathbf{H}_1\mathbf{E}_{i,j}\mathbf{H}_2$ *can be seen as the following state-space system, so that Proposition 4 can be used in order to compute the* L_2*-norm:*

$$\left(\begin{array}{cc|c} \mathbf{A}_{\mathbf{Z}} & \mathbf{0} & \mathbf{B}_{\mathbf{Z}} \\ \mathbf{M}_1\mathbf{E}_{i,j}\mathbf{N}_1 & \mathbf{A}_{\mathbf{Z}} & \mathbf{M}_1\mathbf{E}_{i,j}\mathbf{N}_2 \\ \hline \mathbf{M}_2\mathbf{E}_{i,j}\mathbf{N}_1 & \mathbf{C}_{\mathbf{Z}} & \mathbf{M}_2\mathbf{E}_{i,j}\mathbf{N}_2 \end{array}\right). \tag{57}$$

Proof. The proof is based on the following lemma and can be found in [29].

Lemma 21. *Let* $\mathbf{X}$ *be a matrix in* $\mathbb{R}^{p\times l}$ *while* $\mathbf{G}$ *and* $\mathbf{H}$ *are two transfer matrices independent of* $\mathbf{X}$ *with values in* $\mathbb{C}^{m\times p}$ *and* $\mathbb{C}^{l\times n}$*, respectively. Then,*

$$\begin{aligned}\frac{\partial(\mathbf{GXH})}{\partial\mathbf{X}} &= \mathbf{G}\circledast\mathbf{H},\\ \frac{\partial(\mathbf{GX}^{-1}\mathbf{H})}{\partial\mathbf{X}} &= (\mathbf{GX}^{-1})\circledast(\mathbf{X}^{-1}\mathbf{H}).\end{aligned}\tag{58}$$

By expanding (23) in (24), and using Lemma 21, all the derivative $\partial\mathbf{H}/\partial\mathbf{X}$ with $\mathbf{X}\in\{\mathbf{J},\mathbf{K},\ldots,\mathbf{S}\}$ can be obtained and then gathered using

$$\frac{\partial}{\partial\mathbf{Z}} = \begin{pmatrix} -\dfrac{\partial}{\partial\mathbf{J}} & \dfrac{\partial}{\partial\mathbf{M}} & \dfrac{\partial}{\partial\mathbf{N}}\\ \dfrac{\partial}{\partial\mathbf{K}} & \dfrac{\partial}{\partial\mathbf{P}} & \dfrac{\partial}{\partial\mathbf{Q}}\\ \dfrac{\partial}{\partial\mathbf{L}} & \dfrac{\partial}{\partial\mathbf{R}} & \dfrac{\partial}{\partial\mathbf{S}}\end{pmatrix}.\tag{59}$$

Equation (56) is quite straightforward and comes from the definition of the operator $\circledast$. □

Remark 22. In order to simplify the expressions, matrix extensions of $\log_2$, floor operator $\lfloor\cdot\rfloor$, and power of 2 can be used. For example, if $\mathbf{M}\in\mathbb{R}^{p\times q}$, then $\log_2(\mathbf{M})\in\mathbb{R}^{p\times q}$ such as $(\log_2(\mathbf{M}))_{i,j}\triangleq\log_2(\mathbf{M}_{i,j})$.

The binary-point positions of the coefficients can then be computed by

$$\gamma_{\mathbf{Z}} = \beta_{\mathbf{Z}} - 2\cdot\mathbb{1}_{\mathbf{Z}} - \left\lfloor\log_2|\mathbf{Z}|\right\rfloor,\tag{60}$$

where $\mathbb{1}_{\mathbf{Z}}$ represents the matrix with all coefficients set to 1 and with the same size than $\mathbf{Z}$.

Also, the $\Xi_{\mathbf{Z}}$ matrix is expressed by

$$\Xi_{\mathbf{Z}}\triangleq\frac{2}{\sqrt{3}}2^{-\beta_{\mathbf{Z}}}\times\lfloor\mathbf{Z}\rfloor_2\times\delta_{\mathbf{Z}}.\tag{61}$$

Remark 23. In the classical case where the wordlengths of the coefficients are all the same (equal to β), we can define a normalized transfer function error $\widetilde{\sigma}^2_{\Delta\mathbf{H}}$ by

$$\widetilde{\sigma}^2_{\Delta\mathbf{H}}\triangleq\frac{3\sigma^2_{\Delta\mathbf{H}}}{2^{-2\beta+2}}.\tag{62}$$

This measure is now independent of the wordlength and can be used for some comparisons. It can be computed by

$$\widetilde{\sigma}^2_{\Delta\mathbf{H}} = \left\|\frac{\delta\mathbf{H}}{\delta\mathbf{Z}}\times\lfloor\mathbf{Z}\rfloor_2\times\delta_{\mathbf{Z}}\right\|_F^2.\tag{63}$$

4.3. Comparison with the Classical M_{L_2} Measure. It is of interest to remark the relationship with the classical M_{L_2} measure. In [6] where the transfer function error appears for the first time (applied on a SISO state-space system), the coefficients are supposed to have the same fixed-point representation, so their second-order moments $(\sigma^2_{\mathbf{Z}_{ij}})$ are all equal and denoted σ_0^2. So, in that case, the M_{L_2} satisfies

$$M_{L_2} = \frac{\sigma^2_{\Delta\mathbf{H}}}{\sigma_0^2}.\tag{64}$$

Here, the transfer function error $\sigma^2_{\Delta\mathbf{H}}$ can be seen as an extension of the M_{L_2} measure with fixed-point considerations. The sensitivity is weighted according to the variance of the quantization noise of each coefficient. More details in that comparison can be found in [8].

5. Sensitivity-Based Pole Error

The same considerations applies to the poles. It is interesting to evaluate how the pole moduli are changed from $|\lambda_k|$ to $|\lambda_k|^\dagger\triangleq|\lambda_k|+\Delta|\lambda_k|$ by the coefficient quantization.

In the same way as in Definition 16, the degradation can be evaluated in a stochastic way.

Definition 24 (Sensitivity-Based Pole Error). The sensitivity-based pole error is defined by

$$\sigma^2_{\Delta|\lambda|}\triangleq\sum_{k=1}^{n}\sigma^2_{\Delta|\lambda_k|}\omega_k,\tag{65}$$

where $\sigma^2_{\Delta|\lambda_k|}$ is the second-order moment of the random variable $\Delta|\lambda_k|$

$$\sigma^2_{\Delta|\lambda_k|}\triangleq E\left\{(\Delta|\lambda_k|)^2\right\}.\tag{66}$$

This measure is tractable thanks to the two following propositions.

Proposition 25. *It can be computed with*

$$\sigma^2_{\Delta|\lambda_k|} = \left\|\frac{\partial|\lambda_k|}{\partial\mathbf{Z}}\times\Xi_{\mathbf{Z}}\right\|_F^2,\tag{67}$$

where $\Xi_{\mathbf{Z}}$ *is the matrix already defined in* (46).

Proof. A first-order approximation gives

$$\Delta|\lambda_k| = \sum_{i,j}\frac{\partial|\lambda_k|}{\partial\mathbf{Z}_{ij}}\Delta\mathbf{Z}_{ij}.\tag{68}$$

So,

$$\begin{aligned}\sigma^2_{\Delta|\lambda_k|} &= \sum_{i,j}\sum_{r,s}\frac{\partial|\lambda_k|}{\partial\mathbf{Z}_{ij}}\frac{\partial|\lambda_k|}{\partial\mathbf{Z}_{rs}}E\left\{\Delta\mathbf{Z}_{ij}\Delta\mathbf{Z}_{rs}\right\}\\ &= \sum_{ij}\left(\frac{\partial|\lambda_k|}{\partial\mathbf{Z}_{ij}}\right)^2\sigma^2_{\Delta\mathbf{Z}_{ij}}\end{aligned}\tag{69}$$

since the $(\Delta\mathbf{Z}_{ij})$ are indepedent centered random variables. □

Proposition 26. *The pole sensitivity, with respect to the coefficients, can be computed by*

$$\frac{\partial|\lambda_k|}{\partial\mathbf{Z}} = \frac{1}{|\lambda_k|}\mathrm{Re}\left(\mathbf{M}_1^\top\lambda_k^\star\mathbf{y}_k^\star\mathbf{x}_k^\top\mathbf{N}_1^\top\right),\quad\forall 1\leqslant k\leqslant n,\tag{70}$$

where $(\mathbf{x}_k)_{1\leqslant k\leqslant n}$ *are the right eigenvectors corresponding to the eigenvalues* $(\lambda_k)_{1\leqslant k\leqslant n}$ *and* $(\mathbf{y}_k)_{1\leqslant k\leqslant n}$ *the column vector of the matrix* $\mathbf{M}_\mathbf{y} = (\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_n)$ *defined by* $\mathbf{M}_\mathbf{y} \triangleq \mathbf{M}_\mathbf{x}^{-\top}$, *with* $\mathbf{M}_\mathbf{x} \triangleq (\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_n)$. $\mathbf{M}_1$ *and* $\mathbf{N}_1$ *are the matrices previously defined in* (55).

Proof. The proof is based on the following lemmas, proved in [1, 14].

Lemma 27. *Let* $\mathbf{V}_0$, $\mathbf{V}_1$, *and* $\mathbf{V}_2$ *be constant matrices of appropriate dimension.*

(i) *If* $\mathbf{A} = \mathbf{V}_0 + \mathbf{V}_1\mathbf{X}\mathbf{V}_2$, *then*

$$\frac{\partial\lambda_k}{\partial\mathbf{X}} = \mathbf{V}_1^\top \frac{\partial\lambda_k}{\partial\mathbf{A}}\mathbf{V}_2^\top. \tag{71}$$

(ii) *If* $\mathbf{A} = \mathbf{V}_0 + \mathbf{V}_1\mathbf{X}^{-1}\mathbf{V}_2$, *then*

$$\frac{\partial\lambda_k}{\partial\mathbf{X}} = -\left(\mathbf{V}_1\mathbf{X}^{-1}\right)^\top \frac{\partial\lambda_k}{\partial\mathbf{A}}\left(\mathbf{X}^{-1}\mathbf{V}_2\right)^\top. \tag{72}$$

This lemma can be applied to $\mathbf{J}$, $\mathbf{K}$, $\mathbf{L}$, …, $\mathbf{S}$, and gives

$$\frac{\partial\lambda_k}{\partial\mathbf{Z}} = \mathbf{M}_1^\top \frac{\partial\lambda_k}{\partial\mathbf{A}}\mathbf{N}_1^\top. \tag{73}$$

Then, the pole sensitivity matrix $\partial|\lambda_k|/\partial\mathbf{A}$ can be finally computed with the following lemma.

Lemma 28. *The derivative of the eigenvalues (and their moduli) of a given matrix with respect to that matrix is given by*

$$\begin{aligned}\frac{\partial\lambda_k}{\partial\mathbf{A}} &= \mathbf{y}_k^\star\mathbf{x}_k^\top,\\ \frac{\partial|\lambda_k|}{\partial\mathbf{A}} &= \frac{1}{|\lambda_k|}\operatorname{Re}\left(\lambda_k^\star\frac{\partial\lambda_k}{\partial\mathbf{A}}\right).\end{aligned} \tag{74}$$

□

Remark 29. Roughly similar to Remark 23, it is also possible to normalize the sensitivity-based pole error in the common case where the coefficients have all the same wordlength (equal to β). We can define a *normalized pole error* $\widetilde{\sigma}^2_{\Delta|\lambda|}$ by

$$\widetilde{\sigma}^2_{\Delta|\lambda|} \triangleq \frac{\sigma^2_{\Delta|\lambda|}}{2^{-2\beta+2}}. \tag{75}$$

This measure is now independent of the wordlength and can be used for some comparisons. It could be computed by

$$\widetilde{\sigma}^2_{\Delta|\lambda|} = \sum_{k=1}^{n}\omega_k\left\|\frac{\partial|\lambda_k|}{\partial\mathbf{Z}} \times \lfloor\mathbf{Z}\rfloor_2 \times \delta_\mathbf{Z}\right\|_F^2. \tag{76}$$

6. Extension to the Closed-Loop Control

In previous sections, the filtering problems were considered, and the open-loop contexts were implicitly taken into account. In this section, we extend previous results to closed-loop case, where a filter (denoted here as *controller*) is controlling a plant in a feedback scheme. The problem has an important practical interest in the context of robust control theory [30], when considering the model uncertainties of the process or even of the controller in the sense of FWL implementation [1].

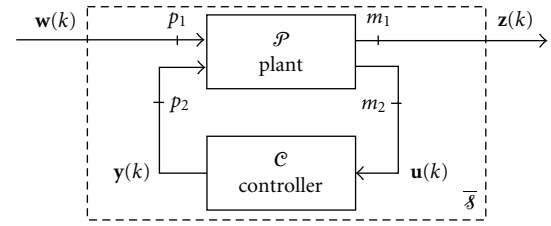

Figure 5: Closed-loop system considered.

Let us consider a plant $\mathcal{P}$ (defined by its transfer function or equivalently by a state-space relationship) controlled by a controller $\mathcal{C}$ in a standard form [30], as shown in Figure 5. $\mathbf{w}(k) \in \mathbb{R}^{p_1}$ and $\mathbf{z}(k) \in \mathbb{R}^{m_1}$ are the exogenous p_1 inputs and m_1 outputs (to control), whereas $\mathbf{u}(k) \in \mathbb{R}^{p_2}$ and $\mathbf{y}(k) \in \mathbb{R}^{m_2}$ are the p_2 control and m_2 measure signals, respectively.

The plant $\mathcal{P}$ is defined by the following state-space relation:

$$\begin{aligned}\mathbf{x}_\mathcal{P}(k+1) &= \mathbf{A}\mathbf{x}_\mathcal{P}(k) + \mathbf{B}_1\mathbf{w}(k) + \mathbf{B}_2\mathbf{u}(k),\\ \mathbf{z}(k) &= \mathbf{C}_1\mathbf{x}_\mathcal{P}(k) + \mathbf{D}_{11}\mathbf{w}(k) + \mathbf{D}_{12}\mathbf{u}(k),\\ \mathbf{y}(k) &= \mathbf{C}_2\mathbf{x}_\mathcal{P}(k) + \mathbf{D}_{21}\mathbf{w}(k),\end{aligned} \tag{77}$$

where $\mathbf{A} \in \mathbb{R}^{n_\mathcal{P}\times n_\mathcal{P}}$, $\mathbf{B}_1 \in \mathbb{R}^{n_\mathcal{P}\times p_1}$, $\mathbf{B}_2 \in \mathbb{R}^{n_\mathcal{P}\times p_2}$, $\mathbf{C}_1 \in \mathbb{R}^{m_1\times n_\mathcal{P}}$, $\mathbf{C}_2 \in \mathbb{R}^{m_2\times n_\mathcal{P}}$, $\mathbf{D}_{11} \in \mathbb{R}^{m_1\times p_1}$, $\mathbf{D}_{12} \in \mathbb{R}^{m_1\times p_2}$, and $\mathbf{D}_{21} \in \mathbb{R}^{m_2\times p_1}$. Note that the $\mathbf{D}_{22}$ term is null.

The controller is realized in the SIF form (see (18)), with l, m_2, n, and p_2 as intermediate variable, input, state and output dimensions, respectively.

Unlike open-loop context, the whole system $\overline{\mathcal{S}}$ is here considered, with $\mathbf{w}(k)$ and $\mathbf{z}(k)$ as inputs and outputs, respectively. Its transfer function is given by

$$\overline{\mathbf{H}} : z \longmapsto \overline{\mathbf{C}_\mathbf{Z}}\left(z\mathbf{I}_{n_\mathcal{P}+n} - \overline{\mathbf{A}_\mathbf{Z}}\right)^{-1}\overline{\mathbf{B}_\mathbf{Z}} + \overline{\mathbf{D}_\mathbf{Z}} \tag{78}$$

with $\overline{\mathbf{A}_\mathbf{Z}} \in \mathbb{R}^{n_\mathcal{P}+n\times n_\mathcal{P}+n}$, $\overline{\mathbf{B}_\mathbf{Z}} \in \mathbb{R}^{n_\mathcal{P}+n\times p_1}$, $\overline{\mathbf{C}_\mathbf{Z}} \in \mathbb{R}^{m_1\times n_\mathcal{P}+n}$, $\overline{\mathbf{D}_\mathbf{Z}} \in \mathbb{R}^{m_1\times p_1}$ and

$$\begin{aligned}\overline{\mathbf{A}_\mathbf{Z}} &= \begin{pmatrix}\mathbf{A}+\mathbf{B}_2\mathbf{D}_\mathbf{Z}\mathbf{C}_2 & \mathbf{B}_2\mathbf{C}_\mathbf{Z}\\ \mathbf{B}_\mathbf{Z}\mathbf{C}_2 & \mathbf{A}_\mathbf{Z}\end{pmatrix},\\ \overline{\mathbf{B}_\mathbf{Z}} &= \begin{pmatrix}\mathbf{B}_1+\mathbf{B}_2\mathbf{D}_\mathbf{Z}\mathbf{D}_{21}\\ \mathbf{B}_\mathbf{Z}\mathbf{D}_{21}\end{pmatrix},\\ \overline{\mathbf{C}_\mathbf{Z}} &= \begin{pmatrix}\mathbf{C}_1+\mathbf{D}_{12}\mathbf{D}_\mathbf{Z}\mathbf{C}_2 & \mathbf{D}_{12}\mathbf{C}_\mathbf{Z}\end{pmatrix},\\ \overline{\mathbf{D}_\mathbf{Z}} &= \mathbf{D}_{11}+\mathbf{D}_{12}\mathbf{D}_\mathbf{Z}\mathbf{D}_{21}.\end{aligned} \tag{79}$$

The closed-loop poles of the system, denoted $(\overline{\lambda}_k)_{1\leqslant k\leqslant n+n_\mathcal{P}}$, are the eigenvalues of the matrix $\overline{\mathbf{A}_\mathbf{Z}}$. Their moduli indicate directly the stability of the closed-loop system.

In order to evaluate the closed-loop transfer function degradation or the pole moduli deviation, the two closed-loop measures are used, as a natural extension to the open-loop case.

Definition 30 (Closed-Loop Sensitivity-Based Error). A measure of the closed-loop sensitivity-based transfer function error can be statistically defined by

$$\sigma^2_{\Delta\overline{\mathbf{H}}} \triangleq \frac{1}{2\pi}\int_0^{2\pi} E\left\{\left\|\Delta\overline{\mathbf{H}}\left(e^{j\omega}\right)\right\|_F^2\right\} d\omega. \tag{80}$$

The closed-loop sensitivity-based pole error is defined by

$$\sigma^2_{\Delta|\overline{\lambda}|} \triangleq \sum_{k=1}^{n} \sigma^2_{\Delta|\overline{\lambda}_k|}\,\omega_k. \tag{81}$$

They can be computed with Proposition 31.

Proposition 31. *The closed-loop transfer function error is given by*

$$\sigma^2_{\Delta\overline{\mathbf{H}}} = \left\|\frac{\delta\overline{\mathbf{H}}}{\delta\mathbf{Z}} \times \Xi_{\mathbf{Z}}\right\|_F^2, \tag{82}$$

where $\delta\overline{\mathbf{H}}/\delta\mathbf{Z}$ is obtained from the closed-loop transfer function sensitivity $\partial\overline{\mathbf{H}}/\partial\mathbf{Z}$ given by

$$\frac{\partial\overline{\mathbf{H}}}{\partial\mathbf{Z}} = \overline{\mathbf{H}}_1 \circledast \overline{\mathbf{H}}_2 \tag{83}$$

with

$$\begin{aligned}
&\overline{\mathbf{H}}_1 : z \longmapsto \overline{\mathbf{C}_\mathbf{Z}}\left(z\mathbf{I}_{n+n_\mathcal{P}} - \overline{\mathbf{A}_\mathbf{Z}}\right)^{-1}\overline{\mathbf{M}}_1 + \overline{\mathbf{M}}_2,\\
&\overline{\mathbf{H}}_2 : z \longmapsto \overline{\mathbf{N}}_1\left(z\mathbf{I}_{n+n_\mathcal{P}} - \overline{\mathbf{A}_\mathbf{Z}}\right)^{-1}\overline{\mathbf{B}_\mathbf{Z}} + \overline{\mathbf{N}}_2,\\
&\overline{\mathbf{M}}_1 = \begin{pmatrix} \mathbf{B}_2\mathbf{L}\mathbf{J}^{-1} & \mathbf{0} & \mathbf{B}_2 \\ \mathbf{K}\mathbf{J}^{-1} & \mathbf{I}_n & \mathbf{0} \end{pmatrix},\\
&\overline{\mathbf{N}}_1 = \begin{pmatrix} \mathbf{J}^{-1}\mathbf{N}\mathbf{C}_2 & \mathbf{J}^{-1}\mathbf{M} \\ \mathbf{0} & \mathbf{I}_n \\ \mathbf{C}_2 & \mathbf{0} \end{pmatrix}, \qquad \overline{\mathbf{N}}_2 = \begin{pmatrix} \mathbf{J}^{-1}\mathbf{N}\mathbf{D}_{21} \\ \mathbf{0} \\ \mathbf{D}_{21} \end{pmatrix},\\
&\overline{\mathbf{M}}_2 = \begin{pmatrix} \mathbf{D}_{12}\mathbf{L}\mathbf{J}^{-1} & \mathbf{0} & \mathbf{D}_{12} \end{pmatrix}.
\end{aligned} \tag{84}$$

In the same way, the sensitivity-based closed-loop pole error $\partial|\overline{\lambda}_k|/\partial\mathbf{Z}$ is given by

$$\frac{\partial\left|\overline{\lambda}_k\right|}{\partial\mathbf{Z}} = \frac{1}{\left|\overline{\lambda}_k\right|}\,\mathrm{Re}\left(\overline{\mathbf{M}}_1^\top\overline{\lambda}_k^\star\overline{\mathbf{y}}_k^\star\overline{\mathbf{x}}_k^\top\overline{\mathbf{N}}_1^\top\right), \quad \forall 1 \leqslant k \leqslant n, \tag{85}$$

where $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{y}}_k$ are associated to $\overline{\mathbf{A}_\mathbf{Z}}$ as in Proposition 26.

Proof. Lemmas 21 and 27 can be used in the same way they are used to compute the derivative $\partial\mathbf{H}/\partial\mathbf{Z}$ and $\partial|\overline{\lambda}_k|/\partial\mathbf{Z}$ in Propositions 20 and 26. See [31] for more details. □

7. Optimal Realization

7.1. Invariance with respect to Scaling. Let us consider a scaling of the intermediate variables and the states. The realization $\mathbf{Z}_0$ is changed into $\mathbf{Z}_1 = \mathcal{T}_1\mathbf{Z}_0\mathcal{T}_2$ with

$$\mathcal{T}_1 \triangleq \begin{pmatrix} \mathcal{Y} & & \\ & \mathcal{U}^{-1} & \\ & & I_p \end{pmatrix}, \qquad \mathcal{T}_2 \triangleq \begin{pmatrix} \mathcal{W} & & \\ & \mathcal{U} & \\ & & I_m \end{pmatrix} \tag{86}$$

with $\mathcal{U}$, $\mathcal{Y}$, and $\mathcal{W}$ some invertible diagonal matrices. So $\mathbf{x}(k)$ is changed in $\mathcal{U}^{-1}\mathbf{x}(k)$ and $\mathbf{t}(k)$ is changed in $\mathcal{W}^{-1}\mathbf{t}(k)$.

Remark 32. This is similar to (26), but here $\mathcal{U}$, $\mathcal{Y}$, and $\mathcal{W}$ are diagonal. This only implies scaling.

Proposition 33 (Invariance to scaling). *A scaling with powers of 2 ($\mathcal{U}$, $\mathcal{Y}$, and $\mathcal{W}$ diagonal with $\mathcal{U}_{ii} = 2^{u_i}$, $\mathcal{Y}_{ii} = 2^{y_i}$, $\mathcal{W}_{ii} = 2^{w_i}$ with u_i, y_i and $w_i \in \mathbb{Z}$) does not change the transfer function error $\sigma^2_{\Delta\mathbf{H}}$ nor the pole error $\sigma^2_{\Delta|\lambda|}$.*

Proof. Let $\mathcal{F}_2(x)$ denotes the fractional value of $\log_2|x|$

$$\mathcal{F}_2(x) \triangleq \log_2|x| - \left\lfloor\log_2|x|\right\rfloor. \tag{87}$$

Then, the operator $\lfloor\cdot\rfloor_2$ satisfies

$$\lfloor ab\rfloor_2 = \lfloor a\rfloor_2\lfloor b\rfloor_2 2^{\lfloor\mathcal{F}_2(a)+\mathcal{F}_2(b)\rfloor}, \tag{88}$$

and hence

$$\left\lfloor(\mathbf{Z}_1)_{ij}\right\rfloor_2 = \lfloor(\mathcal{T}_1)_{ii}\rfloor_2\left\lfloor(\mathbf{Z}_0)_{ij}\right\rfloor_2\left\lfloor(\mathcal{T}_2)_{jj}\right\rfloor_2\Phi_{ij} \tag{89}$$

with $\Phi_{ij} \triangleq 2^{\lfloor\mathcal{F}_2((\mathcal{T}_1)_{ii})+\mathcal{F}_2((\mathcal{T}_2)_{jj})+\mathcal{F}_2((\mathbf{Z}_0)_{ij})\rfloor}$. So, $\Xi_{\mathbf{Z}_1}$ is deduced from $\Xi_{\mathbf{Z}_0}$ by

$$(\Xi_{\mathbf{Z}_1})_{ij} = (\Xi_{\mathbf{Z}_0})_{ij}\lfloor(\mathcal{T}_1)_{ii}\rfloor_2\left\lfloor(\mathcal{T}_2)_{jj}\right\rfloor_2\Phi_{ij}. \tag{90}$$

By remarking that the similarity on $\mathbf{Z}_0$ changes the transfer function $\mathbf{H}_1$ and $\mathbf{H}_2$ in

$$\mathbf{H}_1|_{\mathbf{Z}_1} = \mathbf{H}_1|_{\mathbf{Z}_0}\mathcal{T}_1^{-1}, \qquad \mathbf{H}_2|_{\mathbf{Z}_1} = \mathcal{T}_2^{-1}\mathbf{H}_2\,|_{\mathbf{Z}_0} \tag{91}$$

it comes that the sensitivity transfer function is changed in

$$\left.\frac{\partial\mathbf{H}}{\partial\mathbf{Z}}\right|_{\mathbf{Z}_1} = \mathcal{T}_1^{-\top}\left.\frac{\partial\mathbf{H}}{\partial\mathbf{Z}}\right|_{\mathbf{Z}_0}\mathcal{T}_2^{-\top}, \tag{92}$$

and then

$$\left.\left(\frac{\partial\mathbf{H}}{\partial\mathbf{Z}_{ij}}(\Xi_\mathbf{Z})_{ij}\right)\right|_{\mathbf{Z}_1} = \left.\frac{\partial\mathbf{H}}{\partial\mathbf{Z}_{ij}}(\Xi_\mathbf{Z})_{ij}\right|_{\mathbf{Z}_0} \times \frac{\lfloor(\mathcal{T}_1)_{ii}\rfloor_2}{(\mathcal{T}_1)_{ii}}\frac{\left\lfloor(\mathcal{T}_2)_{jj}\right\rfloor_2}{(\mathcal{T}_2)_{jj}}\Phi_{ij}. \tag{93}$$

Now we can remark that $\Phi_{ij} \in \{1, 2, 4\}$ and $\Phi_{ij} = 1$ if the power of 2 are used for the scaling. Also $\lfloor a\rfloor_2/a = 1$ if a is a power of 2.

The same proof can be applied on the pole error since

$$\left.\frac{\partial|\lambda_k|}{\partial\mathbf{Z}}\right|_{\mathbf{Z}_1} = \mathcal{T}_1^{-\top}\left.\frac{\partial|\lambda_k|}{\partial\mathbf{Z}}\right|_{\mathbf{Z}_0}\mathcal{T}_2^{-\top}. \tag{94}$$

□

7.2. Optimal Problem. Even if it is not the main goal of this paper, it is now possible to consider optimal realization, according to a FWL criterion. Let $\mathcal{J}$ be a given criterion (it could be sensitivity-based transfer function error, pole error, or a combination of these two criteria), then the problem consists of finding the optimal realization that minimizes $\mathcal{J}$ or equivalently finding the optimal coordinate transform $(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}})$ that transform a given realization, that is,

$$\left(\boldsymbol{\mathcal{U}}_{\text{opt}}, \boldsymbol{\mathcal{Y}}_{\text{opt}}, \boldsymbol{\mathcal{W}}_{\text{opt}}\right) = \underset{\boldsymbol{\mathcal{U}},\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{W}} \text{ invertible}}{\arg\min} \mathcal{J}(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}}). \quad (95)$$

According to Proposition 33, $\mathcal{J}$ is invariant to power-of-2 scaling, and this optimization problem has an infinite number of solutions. Thus, it could be of interest to *normalize* all the coordinate transforms with regards to an extra consideration. For example, this could be a L_2-scaling constraint, even if it is not necessary here.

The idea is to define and set the binary-point position of the states and the intermediate variables [8]. This gives us a bound on the L_2-gain of the transfer functions from the input $\mathbf{u}$ to the states $\mathbf{x}$ and intermediate variables $\mathbf{t}$, respectively. One possible constraint is to ensure that

$$1 \leqslant \left\| \mathbf{e}_i^\top \left(z\mathbf{I}_n - \mathbf{A}_\mathbf{Z}\right)^{-1} \mathbf{B}_\mathbf{Z} \right) \Big\|_2 \leqslant 2, \quad (96)$$

$$1 \leqslant \left\| \mathbf{e}_i^\top \mathbf{J}^{-1} \mathbf{M} \left(z\mathbf{I}_n - \mathbf{A}_\mathbf{Z}\right)^{-1} \mathbf{B}_\mathbf{Z} + \mathbf{J}^{-1}\mathbf{N} \right) \Big\|_2 \leqslant 2. \quad (97)$$

This relaxed L_2-constraints were proposed in [32] as an extension of the strict L_2-scaling, that still prevents the implementation from overflow. Any other successive power of 2 can be used for the boundaries.

The inequalities (96) can also be expressed with the controllability Gramian $\mathbf{W}_c$ of the realization.

With that normalization, the optimal problem is now a constrained optimization problem. One way to deal with it is to normalize each coordinate transform $(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{W}})$ before applying it. More details can be found in [8].

Since the sensitivity-based transfer function error $\sigma^2_{\Delta\mathbf{H}}$ and pole error $\sigma^2_{\Delta|\lambda|}$ measures are nonsmooth, this optimization problem can be solved with a global optimization method such as the Adaptive Simulated Algorithm (ASA) [33, 34]. A gradient-base method such as the quasi-Newton algorithm leads to local optima and are not used here.

The FWR Toolbox (sources available at http://fwrtoolbox.gforge.inria.fr) was used for the numerical examples, and few minutes of computation were here required on a desktop computer.

7.3. Numerical Examples. Let us consider the filter with coefficients given by the Matlab command `butter(4,0.125)`. We are considering, in order to compare them, some equivalent (in infinite precision) realizations described below. The values of the measures are shown in Table 2.

7.3.1. State-Space Realization

$\mathbf{Z}_1$: the canonical form (corresponds to the Direct Form II).

Table 2: $\widetilde{\sigma}^2_{\Delta\mathbf{H}}$, $\widetilde{\sigma}^2_{\Delta|\lambda|}$ and number of operations for the different realizations.

Realization	$\widetilde{\sigma}^2_{\Delta\mathbf{H}}$	$\widetilde{\sigma}^2_{\Delta\|\lambda\|}$	Nb $+\times$
$\mathbf{Z}_1$	6989.1918	28144.499	$8 + 12\times$
$\mathbf{Z}_2$	1.6782	2.5804	$20 + 25\times$
$\mathbf{Z}_3$	0.70122	1.749	$20 + 25\times$
$\mathbf{Z}_4$	1.9094	0.8868	$20 + 25\times$
$\mathbf{Z}_5$	0.79439	0.9441	$20 + 25\times$
$\mathbf{Z}_6$	0.90704	23.8916	$12 + 13\times$
$\mathbf{Z}_7$	0.66403	2.3766	$12 + 17\times$
$\mathbf{Z}_8$	3.0183	1.5589	$12 + 17\times$
$\mathbf{Z}_9$	0.67242	2.0486	$12 + 17\times$

$\mathbf{Z}_2$: the *balanced* realization (it is often considered as a good realization. The work in [1] shows that the balanced realizations minimizes the L_1/L_2 sensitivity measure).

$\mathbf{Z}_3$: the normalized $\widetilde{\sigma}^2_{\Delta\mathbf{H}}$-optimal realization. It is obtained with ASA and (63) as criterion.

$\mathbf{Z}_4$: the normalized $\widetilde{\sigma}^2_{\Delta|\lambda|}$-optimal realization (obtained with ASA and (75)).

Even if the goal of this paper is not multiobjective optimal realization, it is interesting to look for a realization that is *good enough* for the two measures. One possibility is to consider the following tradeoff criterion:

$$\mathcal{J}_1 \triangleq \frac{\widetilde{\sigma}^2_{\Delta\mathbf{H}}}{\left(\widetilde{\sigma}^2_{\Delta\mathbf{H}}\right)^{\text{opt}}} + \frac{\widetilde{\sigma}^2_{\Delta|\lambda|}}{\left(\widetilde{\sigma}^2_{\Delta|\lambda|}\right)^{\text{opt}}}, \quad (98)$$

where $\left(\widetilde{\sigma}^2_{\Delta\mathbf{H}}\right)^{\text{opt}}$ and $\left(\widetilde{\sigma}^2_{\Delta|\lambda|}\right)^{\text{opt}}$ are the optimum values obtained for $\widetilde{\sigma}^2_{\Delta\mathbf{H}}$ and $\widetilde{\sigma}^2_{\Delta|\lambda|}$ in realization $\mathbf{Z}_3$ and $\mathbf{Z}_4$, respectively.

$\mathbf{Z}_5$: the $\mathcal{J}_1$-optimal realization. With this measure, we aim to have a realization that simultaneously has low transfer function error and low pole error.

7.3.2. ρDirect Form II Transposed

$\mathbf{Z}_6$: the δ-Direct Form II transposed ($\gamma_i = 1$).

$\mathbf{Z}_7$: the normalized $\widetilde{\sigma}^2_{\Delta\mathbf{H}}$-optimal ρDFIIt realization. The optimal $(\gamma_i)_{1\leqslant i\leqslant 4}$ are

$$\gamma = \begin{pmatrix} 0.49984 & 0.73389 & 0.69192 & 0.70086 \end{pmatrix}^\top. \quad (99)$$

$\mathbf{Z}_8$: the normalized $\widetilde{\sigma}^2_{\Delta|\lambda|}$-optimal ρDFIIt realization. Here the optimal $(\gamma_i)_{1\leqslant i\leqslant 4}$ values are

$$\gamma = \begin{pmatrix} 0.98699 & 0.17365 & 0.68805 & 0.68582 \end{pmatrix}^\top. \quad (100)$$

Table 3: Transfer function and pole errors of the quantized realizations.

Realization	$\|h - h^{\dagger}\|_2$			$\max_k \frac{\|\lambda_k\| - \|\lambda_k^{\dagger}\|}{1 - \|\lambda_k\|}$		
	16 bits	12 bits	8 bits	16 bits	12 bits	8 bits
$\mathbf{Z}_1$	$1.49e-3$	$6.9896e-3$	N.A.	$4.0735e-3$	$1.5805e-2$	$8.0122e-1$
$\mathbf{Z}_2$	$1.7124e-5$	$5.4588e-4$	$6.4839e-3$	$2.93e-5$	$6.544e-4$	$1.2095e-2$
$\mathbf{Z}_3$	$7.2454e-6$	$1.1821e-4$	$5.7031e-3$	$3.1825e-5$	$9.9173e-4$	$1.8286e-2$
$\mathbf{Z}_4$	$2.0669e-5$	$3.9455e-4$	$4.4698e-3$	$5.2194e-5$	$6.2182e-4$	$6.907e-3$
$\mathbf{Z}_5$	$1.2535e-5$	$2.2808e-4$	$2.9784e-3$	$6.2296e-5$	$5.4436e-4$	$1.9987e-3$
$\mathbf{Z}_6$	$2.9412e-5$	$4.5313e-4$	$8.9759e-3$	$1.1577e-4$	$3.0793e-3$	$5.5694e-2$
$\mathbf{Z}_7$	$1.1615e-5$	$1.4539e-4$	$5.5738e-3$	$2.3205e-5$	$7.8623e-4$	$2.1418e-2$
$\mathbf{Z}_8$	$2.3421e-5$	$4.4123e-4$	$8.9101e-3$	$1.7631e-5$	$7.5066e-4$	$7.0628e-3$
$\mathbf{Z}_9$	$1.2353e-5$	$1.8973e-4$	$6.9613e-3$	$2.2346e-5$	$1.0337e-3$	$1.3509e-2$

$\mathbf{Z}_9$: the tradeoff criterion used in (98) is here used (with the values obtained for $\mathbf{Z}_7$ and $\mathbf{Z}_8$ as $(\widetilde{\sigma}^2_{\Delta\mathbf{H}})^{\text{opt}}$ and $(\widetilde{\sigma}^2_{\Delta|\lambda|})^{\text{opt}}$) to obtain a *good enough* ρDFIIt realization. The γ_i obtained are

$$\gamma = \begin{pmatrix} 0.24998 & 0.80129 & 0.72471 & 0.70086 \end{pmatrix}^{\top}. \quad (101)$$

These different results could be compared to the a *posteriori* shift of the poles and transfer function, as presented in Table 3. It depends of course on how far the coefficients are from the closest fixed-point number, the round-off mode, the wordlengths, and the sensitivities. The wordlengths used are 16, 12, and 8 bits. However, 8 bits are not enough to preserve the stability of $\mathbf{Z}_1$.

The realizations $\mathbf{Z}_5$ and $\mathbf{Z}_9$ exhibit the lowest transfer function and pole error estimated from the sensitivities. Their 16-bit fixed-point implementations are given by Algorithms 1 and 2, respectively.

Table 3 confirms that minimizing the sensitivity-based transfer function and pole errors minimizes the probability to have the shift of the poles and transfer function to be greater than a given bound. The unpredictable part of the deterioration comes from the coefficient shift (how far the coefficients are from the closest fixed-point number), and only stochastic approach can be used to evaluate it. Since the direct shift of poles and transfer function ($\|h - h^{\dagger}\|_2$ and $\||\lambda_k| - |\lambda_k^{\dagger}|\|$) cannot be used in optimization (it is an *a posteriori* measure that requires the final hardware/software implementation to be evaluated), the sensitivity-based transfer function and pole errors $\sigma^2_{\Delta\mathbf{H}}$ and $\sigma^2_{\Delta|\lambda|}$ exhibited here are important measures to evaluate the FWL deterioration.

8. Conclusion

After presenting the classical sensitivity analysis for the finite precision implementation of linear filters or controllers, the paper has shown that its use sometimes leads to erroneous conclusion, as it does not take into consideration the exact fixed-point representation of the coefficients. So, poles and input-output errors are better indicators.

```
Input: u: 16 bits integer
Output: y: 16 bits integer
Data: xn, xnp: array [1 · · · 13] of 16 bits integers
Data: Acc: 32 bits integer
Begin
  // Intermediate variables
  Acc ← xn(1) ≪ 15;
  Acc ← Acc + (xn(2) ∗ −28337) ≫ 1;
  Acc ← Acc + (xn(3) ∗ −28385);
  Acc ← Acc + (xn(4) ∗ −23822) ≫ 1;
  Acc ← Acc + (u ∗ −22982) ≫ 3;
  xnp(1) ← Acc ≫ 16;
  Acc ← (xn(1) ∗ 23368) ≫ 3;
  Acc ← Acc + (xn(2) ∗ 26984);
  Acc ← Acc + (xn(3) ∗ 32601) ≫ 3;
  Acc ← Acc + (xn(4) ∗ 28648) ≫ 3;
  Acc ← Acc + (u ∗ 32078) ≫ 2;
  xnp(2) ← Acc ≫ 15;
  Acc ← (xn(1) ∗ 31391) ≫ 2;
  Acc ← Acc + (xn(2) ∗ 32755) ≫ 4;
  Acc ← Acc + (xn(3) ∗ 29692);
  Acc ← Acc + (xn(4) ∗ 32631) ≫ 3;
  Acc ← Acc + (u ∗ −20798) ≫ 3;
  xnp(3) ← Acc ≫ 15;
  Acc ← (xn(1) ∗ 32657) ≫ 3;
  Acc ← Acc + (xn(2) ∗ −24825) ≫ 1;
  Acc ← Acc + (xn(3) ∗ 17894) ≫ 1;
  Acc ← Acc + (xn(4) ∗ 24486);
  Acc ← Acc + (u ∗ 32733) ≫ 4;
  xnp(4) ← Acc ≫ 15;
  // Outputs
  Acc ← (xn(1) ∗ 20763);
  Acc ← Acc + (xn(2) ∗ 29635) ≫ 2;
  Acc ← Acc + (xn(3) ∗ 24740) ≫ 2;
  Acc ← Acc + (xn(4) ∗ −19580) ≫ 2;
  Acc ← Acc + (u ∗ 31323) ≫ 11;
  y ← Acc ≫ 14;
  // Permutations
  xn ← xnp;
end
```

Algorithm 1: $\mathbf{Z}_5$ implemented in 16-bit fixed point.

Input: u: 16 bits integer
Output: y: 16 bits integer
Data: xn: array $[1 \cdots 5]$ of 16 bits integers
Data: T: array $[1 \cdots 5]$ of 16 bits integers
Data: Acc: 32 bits integer
Begin
// *Intermediate variables*
$Acc \leftarrow xn(1) \ll 14;$
$Acc \leftarrow Acc + (u * 31323) \gg 11;$
$T_1 \leftarrow Acc \gg 14;$
$Acc \leftarrow xn(2);$
$T_2 \leftarrow Acc;$
$Acc \leftarrow xn(3);$
$T_3 \leftarrow Acc;$
$Acc \leftarrow xn(4);$
$T_4 \leftarrow Acc;$
// *States*
$Acc \leftarrow T_1 \ll 14;$
$Acc \leftarrow Acc + T_2 \ll 14;$
$Acc \leftarrow Acc + (xn(1) * 32766) \gg 2;$
$Acc \leftarrow Acc + (u * 25359) \gg 7;$
$xn(1) \leftarrow Acc \gg 15;$
$Acc \leftarrow (T_1 * -26735) \gg 2;$
$Acc \leftarrow Acc + T_3 \ll 13;$
$Acc \leftarrow Acc + (xn(2) * 26257);$
$Acc \leftarrow Acc + (u * 17831) \gg 4;$
$xn(2) \leftarrow Acc \gg 15;$
$Acc \leftarrow (T_1 * -32768) \gg 5;$
$Acc \leftarrow Acc + T_4 \ll 13;$
$Acc \leftarrow Acc + (xn(3) * 23747);$
$Acc \leftarrow Acc + (u * 19675) \gg 2;$
$xn(3) \leftarrow Acc \gg 15;$
$Acc \leftarrow (T_1 * -21440) \gg 4;$
$Acc \leftarrow Acc + (xn(4) * 22966);$
$Acc \leftarrow Acc + u \ll 13;$
$xn(4) \leftarrow Acc \gg 15;$
// *Outputs*
$Acc \leftarrow T_1;$
$y \leftarrow Acc;$
end

Algorithm 2: $\mathbf{Z}_9$ implemented in 16-bit fixed point.

It has been then discussed how to appreciate them *a priori*, from the sensitivity computation, leading to the sensitivity-based pole and transfer function errors. All the results are given in the general framework associated to the Specialized Implicit Form, that can encompass a great variety of realization, including general state-space ones, cascade decomposition, lattice filter, ρDFIIt, the use of different operators, and so forth.

Though the new measures exhibited do not require hardware and/or software implementation of the filter, they give a good approximation of the transfer function error and the pole error, under some standardizing assumptions (on the inputs and the coefficients roundoff).

Additional work includes methodological development to solve, by using these new indicators, the resilient realization synthesis. Specific structure and *ad-hoc* constrained optimization algorithms will be investigated.

Acknowledgment

This work has been partially funded by the CNRS (project PEPS "ReSyst").

References

[1] M. Gevers and G. Li, *Parametrizations in Control, Estimation and Filtering Problems*, Springer, Berlin, Germany, 1993.

[2] T. Hilaire, D. Ménard, and O. Sentieys, "Bit accurate roundoff noise analysis of fixed-point linear controllers," in *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design (CACSD '08)*, pp. 607–612, San Antonio, Tex, USA, September 2008.

[3] S. Y. Hwang, "Minimum uncorrelated unit noise in state-space digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 273–281, 1977.

[4] C. T. Mullis and R. A. Roberts, "Synthesis of minimum roundoff noise fixed point digital filters," *IEEE Transactions on Circuits and Systems*, vol. 23, no. 9, pp. 551–562, 1976.

[5] J. A. López, C. Carreras, and O. Nieto-Taladriz, "Improved interval-based characterization of fixed-point LTI systems with feedback loops," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1923–1933, 2007.

[6] T. Hinamoto, S. Yokoyama, T. Inoue, W. Zeng, and W.-S. Lu, "Analysis and minimization of L_2-sensitivity for linear systems and two-dimensional state-space filters using general controllability and observability Gramians," *IEEE Transactions on Circuits and Systems I*, vol. 49, no. 9, pp. 1279–1289, 2002.

[7] T. Hilaire, P. Chevrel, and J. F. Whidborne, "A unifying framework for finite wordlength realizations," *IEEE Transactions on Circuits and Systems I*, vol. 54, no. 8, pp. 1765–1774, 2007.

[8] T. Hilaire, "On the transfer function error of state-space filters in fixed-point context," *IEEE Transactions on Circuits and Systems II*, vol. 56, no. 12, pp. 936–940, 2009.

[9] L. Thiele, "On the sensitivity of linear state space systems," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 5, pp. 502–510, 1986.

[10] V. Tavşanoğlu and L. Thiele, "Optimal design of state-space digital filters by simultaneous minimization of sensibility and roundoff noise," *IEEE Transactions on Circuits and Systems*, vol. 31, no. 10, pp. 884–888, 1984.

[11] R. E. Skelton and D. A. Wagie, "Minimal root sensitivity in linear systems," *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 5, pp. 570–574, 1984.

[12] G. Li, "On pole and zero sensitivity of linear systems," *IEEE Transactions on Circuits and Systems I*, vol. 44, no. 7, pp. 583–590, 1997.

[13] J. F. Whidborne, J. Wu, and R. S. H. Istepanian, "Finite word length stability issues in an ℓ_1 framework," *International Journal of Control*, vol. 73, no. 2, pp. 166–176, 2000.

[14] R. Istepanian and J. Whidborne, Eds., *Digital Controller Implementation and Fragility*, Springer, Berlin, Germany, 2001.

[15] 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 Transactions on Automatic Control*, vol. 46, no. 7, pp. 1162–1166, 2001.

[16] J. Wu, S. Chen, and J. Chu, "Comparative study on finite-precision controller realizations in different representation schemes," in *Proceedings of the 9th Annual Conference Chinese Automation and Computing Society*, Luton, UK, September 2003.

[17] J. Aplevich, *Implicit Linear Systems*, Springer, Berlin, Germany, 1991.
[18] R. Roberts and C. Mullis, *Digital Signal Processing*, Kluwer Academic Publishers, Dodrecht, The Netherlands, 1987.
[19] M. Ikeda, D. D. Šiljak, and D. E. White, "An inclusion principle for dynamic systems," *IEEE Transactions on Automatic Control*, vol. 29, no. 3, pp. 244–249, 1984.
[20] R. H. Middleton and G. C. Goodwin, "Improved finite word length characteristics in digital control using delta operators," *IEEE Transactions on Automatic Control*, vol. 31, no. 11, pp. 1015–1021, 1986.
[21] R. H. Middleton and G. C. Goodwin, *Digital Control and Estimation, A Unified Approach*, Prentice-Hall International Editions, Upper Saddle River, NJ, USA, 1990.
[22] G. Li and Z. Zhao, "On the generalized DFIIt structure and its state-space realization in digital filter implementation," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 4, pp. 769–778, 2004.
[23] J. Hao and G. Li, "An efficient structure for finite precision implementation of digital systems," in *Proceedings of the 5th International Conference on Information, Communications and Signal Processing*, pp. 564–568, December 2005.
[24] G. Li, "A polynomial-operator-based DFIIt structure for IIR filters," *IEEE Transactions on Circuits and Systems II*, vol. 51, no. 3, pp. 147–151, 2004.
[25] M. Palaniswami and G. Feng, "Digital estimation and control with a new discrete time operator," in *Proceedings of the 30th IEEE Conference on Decision and Control*, pp. 1631–1632, Brighton, UK, December 1991.
[26] R. Rocher, D. Menard, N. Herve, and O. Sentieys, "Fixed-point configurable hardware components," *EURASIP Journal on Embedded Systems*, vol. 2006, Article ID 23197, 13 pages, 2006.
[27] B. Widrow and I. Kollár, *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*, Cambridge University Press, Cambridge, UK, 2008.
[28] A. B. Sripad and D. L. Snyder, "A necessary and sufficient condition for quantization error to be uniform and white," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 442–448, 1977.
[29] T. Hilaire and P. Chevrel, "On the compact formulation of the derivation of a transfer matrix with respect to another matrix," Tech. Rep. RR-6760, INRIA, 2008.
[30] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.
[31] T. Hilaire, P. Chevrel, and J. F. Whidborne, "Finite wordlength controller realisations using the specialised implicit form," *International Journal of Control*, vol. 83, no. 2, pp. 330–346, 2010.
[32] T. Hilaire, "Low-parametric-sensitivity realizations with relaxed l_2-dynamic-range-scaling constraints," *IEEE Transactions on Circuits and Systems II*, vol. 56, no. 7, pp. 590–594, 2009.
[33] L. Ingber, "Adaptive simulated annealing (ASA): lessons learned," *Control and Cybernetics*, vol. 25, no. 1, pp. 32–54, 1996.
[34] 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.