Low Parametric Sensitivity Realizations with relaxed L2-dynamic-range-scaling constraints

Thibault Hilaire

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION INTERNE

N° 1924

LOW PARAMETRIC SENSITIVITY REALIZATIONS WITH *RELAXED* L_2-DYNAMIC-RANGE-SCALING CONSTRAINTS

THIBAULT HILAIRE

ISSN 1166-8687

IRISA

CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

Institut de Recherche en Informatique et Systèmes Aléatoires
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

Low Parametric Sensitivity Realizations with *relaxed* L_2-dynamic-range-scaling constraints

Thibault HILAIRE*

Systèmes numériques
Projet CAIRN

Publication interne n ° 1924 — February 2009 — 14 pages

Abstract: This paper presents a new dynamic-range scaling for the implementation of filters/controllers in state-space form. Relaxing the classical L_2-scaling constraints by specific fixed-point considerations allows for a higher degree of freedom for the optimal L_2-parametric sensitivity problem. However, overflows in the implementation are still prevented. The underlying constrained problem is converted into an unconstrained problem for which a solution can be provided. This leads to realizations which are still scaled but less sensitive.

Key-words: Digital filter implementation, coefficient sensitivity, scaling, fixed-point implementation

(Résumé : tsvp)

* CAIRN Project, INRIA/IRISA

Centre National de la Recherche Scientifique (UMR 6074) Université de Rennes 1 – Insa de Rennes

Institut National de Recherche en Informatique et en Automatique – unité de recherche de Rennes

Réalisations à faible sensibilité sous contrainte de mise à l'échelle L_2

Résumé : Ce papier présente une nouvelle mise à l'échelle pour l'implémentation de filtres ou controleurs sous forme d'état. Des considérations sur l'implémentation en virgule fixe nous permettent de relâcher la contrainte de mise à l'échelle L_2 et d'obtenir des degrés de libertés supplémentaires pour le problème de sensibilité paramétrique L_2. De plus, les dépassements durant l'exécutions sont toujours évités. Le problème sous contraintes qui en résulte est transformé en un problème d'optimisation non contraint, pour lequel une solution peut être trouvée. Cela nous permet d'obtenir des réalisations mises à l'échelle et moins sensibles à la quantification des coefficients.

Mots clés : implémentation de filtre, sensibilité des coefficient, implémentation virgule fixe, mise à l'échelle

1 Introduction

The majority of control (or signal processing) systems is implemented in digital general purpose processors, DSPs[1], FPGAs[2], etc. 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 roundoff errors occurring during the computations. They can be formalized as parametric errors and numerical noises, respectively. The focus of this paper are parametric errors, but one can refer to [3, 6, 10, 14] for roundoff noises.
It is also well known that these Finite Word Length (FWL) effects depend on the structure of the realization. This motivates to investigate the coefficient sensitivity minimization problem. It has been widely studied since Thiele published [16, 17] and the definition of a tractable input-output sensitivity norm (the L_1/L_2-sensitivity). This work has been extended with a more natural and reasonable measure, the L_2-sensitivity ([3, 8]).
The dynamic-range-scaling constraints have been introduced in [11] and [9] to prevent overflow and underflow during the evaluation of the state-vector, and as well as the state and criteria normalization. These constraints have to be considered in the L_2-sensitivity minimization problem, for which [7] proposes an efficient quasi-Newton algorithm to solve it.

This paper investigates the L_2-dynamic-range-scaling problem by considering concrete fixed-point implementation of state-space realizations. It reveals that the classical L_2-scaling is only a sufficient condition to prevent overflows and thus it can be slightly relaxed in order to extend the degrees of freedom for the optimization process. New *relaxed*-L_2-dynamic-range-scalings are then presented with respect to the described computational scheme. Finally, the L_2-sensitivity minimization problem with relaxed L_2-scaling constraints is solved. A numerical example illustrates that the proposed constraints can offer reduced L_2-sensitivity with overflow protection.

2 L_2-sensitivity analysis

Let $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$ be a stable, controllable and observable linear discrete time SISO[3] state-space system, i.e.

$$\left\{ \begin{array}{rcl} \boldsymbol{x}(k+1) & = & \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \\ y(k) & = & \boldsymbol{c}\boldsymbol{x}(k) + du(k) \end{array} \right. \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{b} \in \mathbb{R}^{n\times 1}$, $\boldsymbol{c} \in \mathbb{R}^{1\times n}$ and $d \in \mathbb{R}$. $u(k)$ is the scalar input, $y(k)$ is the scalar output and $\boldsymbol{x}(k) \in \mathbb{R}^{n\times 1}$ is the state vector.
Its input-output relationship is given by the scalar transfer function $h : \mathbb{C} \to \mathbb{C}$ defined by:

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

[1] Digital Signal Processors
[2] Field Programmable Gate-Array
[3] Single Input Single Output

The quantization of the coefficients introduces some uncertainly to $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and d leading to $\boldsymbol{A}+\boldsymbol{\Delta A}$, $\boldsymbol{b}+\boldsymbol{\Delta b}$, $\boldsymbol{c}+\boldsymbol{\Delta c}$ and $d+\Delta d$ respectively. It is of interest to consider the sensitivity of the transfer function with respect to the coefficients, based on the following definitions.

Definition 1 (Transfer function sensitivity) *Consider $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ a matrix and $f : \mathbb{R}^{m\times n} \to \mathbb{C}$ a scalar complex function, differentiable with respect to all the entries of $\boldsymbol{X}$. The sensitivity of f with respect to $\boldsymbol{X}$ is defined by the matrix $\boldsymbol{S_X} \in \mathbb{R}^{m\times n}$:*

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

Definition 2 (L_p-Norm) *Let $\boldsymbol{H} : \mathbb{C} \to \mathbb{C}^{k\times l}$ be a function of the scalar complex variable z. $\left\|\boldsymbol{H}\right\|_p$ is the L_p-norm of $\boldsymbol{H}$, defined by:*

$$\left\|\boldsymbol{H}\right\|_p \triangleq \left(\frac{1}{2\pi}\int_0^{2\pi} \left\|\boldsymbol{H}\left(e^{j\omega}\right)\right\|_F^p \, d\omega\right)^{\frac{1}{p}} \tag{4}$$

where $\left\|.\right\|_F$ is the Froebenius norm.

Gevers and Li [3] have proposed the L_2-sensitivity measure to evaluate the coefficient roundoff errors. It is defined by

$$M_{L_2} \triangleq \left\|\frac{\partial h}{\partial \boldsymbol{A}}\right\|_2^2 + \left\|\frac{\partial h}{\partial \boldsymbol{b}}\right\|_2^2 + \left\|\frac{\partial h}{\partial \boldsymbol{c}}\right\|_2^2 + \left\|\frac{\partial h}{\partial d}\right\|_2^2 \tag{5}$$

and can be computed by $\frac{\partial h}{\partial \boldsymbol{A}}(z) = \boldsymbol{G}^\top(z)\boldsymbol{F}^\top(z)$, $\frac{\partial h}{\partial \boldsymbol{b}}(z) = \boldsymbol{G}^\top(z)$, $\frac{\partial h}{\partial \boldsymbol{c}}(z) = \boldsymbol{F}(z)$ and $\frac{\partial h}{\partial d}(z) = 1$, with

$$\boldsymbol{F}(z) \triangleq (z\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b}, \qquad \boldsymbol{G}(z) \triangleq \boldsymbol{c}(zI_n - \boldsymbol{A})^{-1}. \tag{6}$$

This measure is an extension of the more tractable but less natural L_1/L_2 sensitivity measure proposed by V. Tavşanoğlu and L. Thiele [16] ($\left\|\frac{\partial h}{\partial \boldsymbol{A}}\right\|_1^2$ instead of $\left\|\frac{\partial h}{\partial \boldsymbol{A}}\right\|_2^2$ in (5)).

Remark 1 It is also possible to regroup all the coefficients in one unique matrix

$$\boldsymbol{Z} \triangleq \begin{pmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{c} & d \end{pmatrix}. \tag{7}$$

Then, with L_2-norm property, $M_{L_2} = \left\|\frac{\partial h}{\partial \boldsymbol{Z}}\right\|_2^2$. From (6) and the associated state-spaces, the sensitivity transfer function $\frac{\partial h}{\partial \boldsymbol{Z}}$ can be described by the MIMO[4] state-space system $(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}, \tilde{\boldsymbol{D}})$ with

$$\begin{aligned} \tilde{\boldsymbol{A}} &\triangleq \begin{pmatrix} \boldsymbol{A} & \boldsymbol{bc} \\ \boldsymbol{0} & \boldsymbol{A} \end{pmatrix}, \tilde{\boldsymbol{B}} \triangleq \begin{pmatrix} \boldsymbol{0} & \boldsymbol{b} \\ \boldsymbol{I}_n & \boldsymbol{0} \end{pmatrix}, \\ \tilde{\boldsymbol{C}} &\triangleq \begin{pmatrix} \boldsymbol{I}_n & 0 \\ \boldsymbol{0} & \boldsymbol{c} \end{pmatrix}, \tilde{\boldsymbol{D}} \triangleq \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{pmatrix}. \end{aligned} \tag{8}$$

[4] Multiple Inputs Multiple Outputs

See [3] and [4] for more details.

The following proposition allows to compute M_{L_2}.

Proposition 1 *Let us consider $\boldsymbol{H}$ the MIMO state-space system $(\boldsymbol{K}, \boldsymbol{L}, \boldsymbol{M}, \boldsymbol{N})$. Its L_2-norm can be computed by*

$$\|\boldsymbol{H}\|_2^2 = tr(\boldsymbol{N}\boldsymbol{N}^\top + \boldsymbol{M}\boldsymbol{W}_c\boldsymbol{M}^\top) \tag{9}$$

$$= tr(\boldsymbol{N}^\top\boldsymbol{N} + \boldsymbol{L}^\top\boldsymbol{W}_o\boldsymbol{L}) \tag{10}$$

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

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

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

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 $\boldsymbol{T}$, and is denoted $M_{L_2}(\boldsymbol{T})$.
In this case, it is natural to define the following problem:

Problem 1 (optimal L_2-sensitivity problem) *Considering a state-space realization $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$, the optimal L_2-sensitivity problem consists of finding the coordinate transformation $\boldsymbol{T}_{opt}$ that minimizes M_{L_2}:*

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

[3] shows that the problem has one unique solution. Hence, for example, a gradient method can be used to solve it.

3 L_p-dynamic-range scaling

The L_p-dynamic-range-scaling constraints have been introduced by Jackson in [11] and Hwang in [9]. It consists in scaling the state-variable vector such that overflows or underflows during its evaluation are prevented.

Definition 3 (L_p-scaling) *A state-space realization $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$ is said to be L_p-**scaled** if the L_p-norms of the transfer functions from the input to each state are set to 1,* i.e.*:*

$$\left\| \boldsymbol{e}_i^\top (z\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b}) \right\|_p = 1, \quad \forall 1 \leqslant i \leqslant n \tag{13}$$

where $\boldsymbol{e}_i$ is the column vector of appropriate dimension and with all elements being 0 except from the i^{th} element which is 1.

Let $\overset{\max}{u}$ denote the maximum value of the input u:

$$\overset{\max}{u} \triangleq \max_{k \in \mathbb{N}} |u(k)| \tag{14}$$

The L_1-scaling guarantees that the dynamic of each state $\boldsymbol{x}_i$ is lower than $\overset{\max}{u}$, whereas the L_2-scaling guarantees that the variance of each state is unitary for a unit-variance centered white noise input. L_2-scaling doesn't completely prevent overflow as does L_1 , but it is less conservative and more realistic, so it is widely used [15].

With proposition 1 applied to the system $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{e}_i^\top, \boldsymbol{0})$, the L_2-scaling constraints (13) can be expressed as:

$$(\boldsymbol{W}_c)_{i,i} = 1, \quad \forall 1 \leqslant i \leqslant n \tag{15}$$

where $\boldsymbol{W}_c$ is the controllability Gramian of the state-space system $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$.

Problem 2 (sensitivity problem with L_2-scaling constraints) *The optimal L_2-sensitivity problem with L_2-scaling constraints can be formulated as the optimization problem 1, subject to the constraints in* (15).

Moreover, it is possible to L_2-scale a realization with the following proposition.

Proposition 2 (*a posteriori* L_2-scaling) *Considering a state-space realization $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$, it is also possible to* a posteriori *L_2-scale it with a diagonal coordinate transformation $\boldsymbol{T}$ such that:*

$$\boldsymbol{T}_{i,i} = \sqrt{(\boldsymbol{W}_c)_{i,i}}, \quad \forall 1 \leqslant i \leqslant n \tag{16}$$

Then, there exist infinite transformation matrices $\boldsymbol{T}$ (not necessary diagonal) that produces L_2-scaled realizations: Let us consider the invertible matrix $\boldsymbol{U} \in \mathbb{R}^{n \times n}$, then also the transformation matrix $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{V}$ produces L_2-scaling with $\boldsymbol{V}$ diagonal such that:

$$\boldsymbol{V}_{i,i} = \sqrt{(\boldsymbol{U}^{-1}\boldsymbol{W}_c\boldsymbol{U}^{-\top})_{i,i}}, \quad \forall 1 \leqslant i \leqslant n \tag{17}$$

Proof:
A transformation matrix $\boldsymbol{T}$ that transforms $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$ into $(\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}, \boldsymbol{T}^{-1}\boldsymbol{b}, \boldsymbol{c}\boldsymbol{T}, d)$ changes the controllability Gramian $\boldsymbol{W}_c$ into $\boldsymbol{T}^{-1}\boldsymbol{W}_c\boldsymbol{T}^{-\top}$.
Since $\boldsymbol{T}$ is diagonal, the constraints $(\boldsymbol{W}_c)_{i,i} = 1$ imply $\boldsymbol{T}_{i,i} = \sqrt{(\boldsymbol{W}_c)_{i,i}}$.
Moreover, it is also possible to successively apply two transformation matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ on $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, d)$. If $\boldsymbol{V}$ is composed according to (17), then the transformation $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{V}$ performs the L_2-scaling. ■
This proposition can be used to transform the constrained problem 2 into an unconstrained problem. Then an optimization algorithm, like quasi-Newton, can be used to solve it. Other analytical algorithms for this problem can be found in [7] and [10].

4 Fixed-point implementation

4.1 Fixed-point representation

In this paper, the notation (β, γ) is used for the fixed-point representation of a variable or coefficient (2's complement scheme), according to Figure 1. 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.

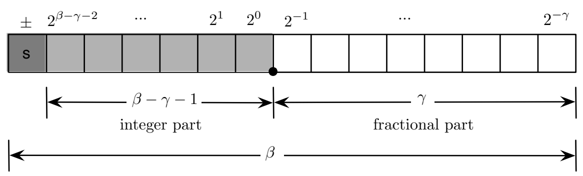

Figure 1: Fixed-point representation

To represent a value x without overflow, a fixed-point representation (β_x, γ_x) may satisfy:

$$\beta_x - \gamma_x - 1 \geqslant \lfloor \log_2 |x| \rfloor + 1 \tag{18}$$

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).

An important fixed-point issue is to find a valid fixed-point representation, such that (18) is satisfied for all values which x can assume during the execution of the algorithm.

4.2 State-overflow

Definition 4 (State-overflow) *The overflow of the state variables $(\boldsymbol{x}_i)_{1\leqslant i\leqslant n}$ can be strictly avoided iff ($1 \leqslant i \leqslant n$)*

$$\forall k, \quad -2^{\beta_{\boldsymbol{x}_i}-\gamma_{\boldsymbol{x}_i}-1} \leqslant \boldsymbol{x}_i(k) < 2^{\beta_{\boldsymbol{x}_i}-\gamma_{\boldsymbol{x}_i}-1}. \tag{19}$$

The overflows are avoided if the binary-point position of each state is carefully chosen, such that

$$\gamma_{\boldsymbol{x}_i} = \beta_{\boldsymbol{x}_i} - 2 - \left\lfloor \log_2 \overset{\max}{\boldsymbol{x}_i} \right\rfloor, \tag{20}$$

where $\overset{\max}{\boldsymbol{x}_i}$ is the maximum magnitude for the i^{th} state:

$$\overset{\max}{\boldsymbol{x}_i} \triangleq \max_{k\in\mathbb{N}} |\boldsymbol{x}_i(k)|. \tag{21}$$

However only upper bounds can be computed. A first upper bound $\overset{\text{up}}{\boldsymbol{x}_i}$ can be obtained by an L_1-norm:

$$\overset{\text{up}}{\boldsymbol{x}_i} = \left\| \left((z\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} \right)_i \right\|_1 \overset{\max}{u}, \tag{22}$$

and a second one can be estimated by an L_2-norm [15]:

$$\overset{\text{up}}{\boldsymbol{x}_i} \simeq \delta \left\| \left((z\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{b} \right)_i \right\|_2 \overset{\max}{u}. \tag{23}$$

Here, the parameter δ can be interpreted as a representation of the number of standard deviations of $\boldsymbol{x}_i$, if the input is unit-variance white centered noise ($\delta \geqslant 1$). Since the L_2-norm in (23) doesn't give a strict bound (contrary to (22)), δ can be seen as a *safety* parameter [15].
Finally, these upper bounds are used to define the binary-point positions:

$$\gamma_{\boldsymbol{x}_i} = \beta_{\boldsymbol{x}_i} - 2 - \left\lfloor \log_2 \overset{\text{up}}{\boldsymbol{x}_i} \right\rfloor. \tag{24}$$

In general, the L_1 and L_2 estimations of $\overset{\text{up}}{\boldsymbol{x}_i}$ approximately leads to the same binary-point position, with 1 or 2 bits deviation. However, since the L_2-norm is more tractable (with proposition 1) and the L_1-norm too conservative ($\overset{\max}{\boldsymbol{x}_i} \ll \overset{\text{up}}{\boldsymbol{x}_i}$), in practice (23) is used, with $\delta = 1$. After implementation, a simulation-based estimation like in [1] or [12] can also be used to verify *in situ* the peak values and the binary point positions, according to the inputs.

4.3 Computational scheme

In order to implement a realization without overflows, two equivalent choices are possible:

- set the binary-point position for each state, according to (24), to make sure that the fixed-point representation of the states avoids state-overflows;
- or define a binary-point position for each state, and apply a scaling to them in order to adapt the peak values of each state to the chosen binary-point position.

Here, we here focus on the 2^{nd} choice, referring to dynamic-range-scaling constraints.

Let us consider in detail the fixed-point implementation of the system given in (1). It leads to $(n+1)$ scalar products to be evaluated, of the form:

$$S = \sum_{i=1}^{N} \boldsymbol{p}_i \boldsymbol{q}_i \tag{25}$$

where the $(\boldsymbol{p}_i)$ are given coefficients and $(\boldsymbol{q}_i)$ are bounded variables.
To avoid bit-shift operations between each addition in the evaluation of eq. (25), the binary-point positions of each partial product of the sum should be equal.
Then, two computational schemes are possible: the *Roundoff After Multiplication* scheme,

where shifts are added after each product to align the operands of the sum ($\boldsymbol{p}_i\boldsymbol{q}_i$ is implemented as $(\boldsymbol{p}'_i * \boldsymbol{q}'_i) >> d_i$) and the *Roundoff Before Multiplication* scheme, where the required shifts are reported into the coefficients ($\boldsymbol{p}_i\boldsymbol{q}_i$ is implemented as $(\boldsymbol{p}'_i >> d_i) * \boldsymbol{q}'_i$).

The main idea of the scaling is to scale each variable $(\boldsymbol{q}_i)$ such that the shifts ($d_i = 0,\ \forall i$) are prevented. In fixed-point representation, the scaling only implies that all the $(\boldsymbol{q}_i)$ have a common format, and so have the $(\boldsymbol{p}_i)$. See [2,6] for more details on implementation schemes.

Applied to the state-space realization (1), this yields that all the states must have the same binary-point position as the input and the coefficients $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and d.
Besides, since they have the same fractional part, their quantization's errors $\boldsymbol{\Delta A}$, $\boldsymbol{\Delta b}$, $\boldsymbol{\Delta c}$ and Δd have the same magnitude $2^{-\gamma_{\boldsymbol{z}}-1}$, and the L_2-sensitivity measure represents a meaningful bound on the transfer function error Δh:

$$\begin{aligned}
\|\Delta h\|_2^2 &\leqslant \left\|\frac{\partial h}{\partial \boldsymbol{A}} \times \boldsymbol{\Delta A}\right\|_2^2 + \left\|\frac{\partial h}{\partial \boldsymbol{b}} \times \boldsymbol{\Delta b}\right\|_2^2 \\
&\quad + \left\|\frac{\partial h}{\partial \boldsymbol{c}} \times \boldsymbol{\Delta c}\right\|_2^2 + \left\|\frac{\partial h}{\partial d} \times \Delta d\right\|_2^2 && (26) \\
&\leqslant 2^{-2(\gamma_{\boldsymbol{z}}+1)} M_{L_2} && (27)
\end{aligned}$$

4.4 New L_2-scaling constraints

Taken this into consideration, the overflows will be avoided by setting the same binary-point position for the states and the input, and by applying an appropriate scaling on the states such that the constraints (20) are satisfied.
Compared to strict L_2-scaling where the states must satisfy $\overset{\max}{\boldsymbol{x}_i} = \overset{\max}{u}$, here, the constraints are relaxed (but still restrictive enough to guarantee the protection against overflow) and replaced by $\gamma_{\boldsymbol{x}_i} = \gamma_u$.

Proposition 3 (*relaxed*-L_2-scaling constraints) *Since the input and the states may have the same binary-point position, the L_2-scaling constraints* (15) *are now transformed into*

$$\frac{2^{2\boldsymbol{\alpha}_i}}{\delta^2} \leqslant (\boldsymbol{W}_c)_{i,i} < 4\frac{2^{2\boldsymbol{\alpha}_i}}{\delta^2}, \qquad \forall 1 \leqslant i \leqslant n \tag{28}$$

where

$$\boldsymbol{\alpha}_i \triangleq \beta_{\boldsymbol{x}_i} - \beta_u - \mathscr{F}_2\left(\overset{\max}{u}\right) \tag{29}$$

and $\mathscr{F}_2(x)$ is defined as the fractional value of $\log_2(x)$*:*

$$\mathscr{F}_2(x) \triangleq \log_2(x) - \lfloor \log_2(x) \rfloor \tag{30}$$

For microcontroller or DSP implementations (contrary to FPGA or some ASIC implementations), the wordlength of all variables is equal, i.e. $\beta_u = \beta_{\boldsymbol{x}_i} \quad (1 \leqslant i \leqslant n)$*. Also* $\overset{\max}{u}$ *could* *be set to a power of* 2*. Then, if* δ *is set to unity (as for classical* L_2*-scaling constraints), the relaxed-*L_2*-scaling constraints* (28) *become:*

$$1 \leqslant (\boldsymbol{W}_c)_{i,i} < 4, \quad \forall 1 \leqslant i \leqslant n \tag{31}$$

Proof:
The binary-point position of the input is set to $\gamma_u = \beta_u - 2 - \left\lfloor \log_2 \overset{\max}{u} \right\rfloor$. Hence, with (24), the constraints $\gamma_u = \gamma_{\boldsymbol{x}_i}$ lead to

$$\beta_u - \left\lfloor \log_2 \overset{\max}{u} \right\rfloor = \beta_{\boldsymbol{x}_i} - \left\lfloor \log_2 \left(\delta \left\| \left((z\boldsymbol{I}_n - \boldsymbol{A})^{-1} \boldsymbol{b} \right)_i \right\|_2 \overset{\max}{u} \right) \right\rfloor$$

and

$$\left\lfloor \log_2 \left(\delta \sqrt{(\boldsymbol{W}_c)_{i,i}} \right) + \mathscr{F}_2 \left(\overset{\max}{u} \right) \right\rfloor = \beta_{\boldsymbol{x}_i} - \beta_u \tag{32}$$

And finally

$$2^{\boldsymbol{\alpha}_i} \leqslant \delta \sqrt{(\boldsymbol{W}_c)_{i,i}} < 2^{\boldsymbol{\alpha}_i + 1} \tag{33}$$

■

It is important to remark that these new constraints allow more freedom for the scaling and introduce a new degree of freedom for the search for optimal realizations. Even though not considered in this paper, moreover it could give more freedom for the minimization of the roundoff noise power.

5 Optimal L_2-sensitivity realization with relaxed L_2-norm dynamic-range-scaling constraints

Then, these relaxed constraints can be applied to a new sensitivity problem:

Problem 3 (*relaxed* sensitivity problem) *The optimal* L_2*-sensitivity problem with* relaxed L_2*-norm dynamic-range-scaling constraints can be expressed in the form of the constrained problem 2 subject to constraints in* (28)*.*

This constrained problem can be solved by two different means.
First, in addition to the n^2 free parameters of the transformation matrix U applied to the system, n extra parameters $(\boldsymbol{\gamma}_i)_{1 \leqslant i \leqslant n}$ can be considered. These $(\boldsymbol{\gamma}_i)$ represent the desired L_2-scaling and will be constrained by

$$\frac{2^{2\boldsymbol{\alpha}_i}}{\delta^2} \leqslant \boldsymbol{\gamma}_i < 4 \frac{2^{2\boldsymbol{\alpha}_i}}{\delta^2}, \quad \forall 1 \leqslant i \leqslant n. \tag{34}$$

Then a diagonal transformation matrix $\boldsymbol{V}_{\boldsymbol{\gamma}}$ is applied, with

$$(\boldsymbol{V}_{\boldsymbol{\gamma}})_{i,i} = \sqrt{\frac{(\boldsymbol{W}_c)_{i,i}}{\boldsymbol{\gamma}_i}}. \tag{35}$$

In this case, a constrained optimization algorithm (like quasi-Newton one, implemented in `fmincon` with Matlab) can then be used to solve the following problem:

$$(\boldsymbol{U}_{opt}, \gamma_{opt}) = \underset{\substack{\boldsymbol{U} \text{ invertible} \\ \boldsymbol{\gamma} \text{ satisfying (34)}}}{\arg\min} \; M_{L_2}(\boldsymbol{U}\boldsymbol{V}_{\boldsymbol{\gamma}}) \tag{36}$$

The optimal realization satisfying the relaxed-L_2 constraints is then obtained by applying the transformation matrix $\boldsymbol{T}_{opt} = \boldsymbol{U}_{opt}\boldsymbol{V}_{\boldsymbol{\gamma}_{opt}}$.

The other approach is to scale the system after each transformation to ensure that the constraints are met:

Proposition 4 (*a posteriori* relaxed scaling) *Considering a state-space realization, it is possible to* a posteriori *scale it with a diagonal transformation matrix $\boldsymbol{T}$ given by*

$$\boldsymbol{T}_{i,i} = \delta\sqrt{(\boldsymbol{W}_c)_{i,i}}\, 2^{-\mathscr{F}_2(\delta\sqrt{(\boldsymbol{W}_c)_{i,i}}) - \boldsymbol{\alpha}_i}, \tag{37}$$

such that the constraints (28) *are satisfied. Moreover, it is possible to build all the transformation matrices that meet the constraints* (28)*: Let us consider an invertible matrix $\boldsymbol{U} \in \mathbb{R}^{n\times n}$, then the transformation matrix $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{V}$ with $\boldsymbol{V}$ diagonal such that:*

$$\boldsymbol{V}_{i,i} = \delta\sqrt{(\boldsymbol{U}^{-1}\boldsymbol{W}_c\boldsymbol{U}^{-\top})_{i,i}}\, 2^{-\mathscr{F}_2(\delta\sqrt{(\boldsymbol{U}^{-1}\boldsymbol{W}_c\boldsymbol{U}^{-\top})_{i,i}}) - \boldsymbol{\alpha}_i} \tag{38}$$

produces the relaxed-L_2-scaling.
Proof:
$\mathscr{F}_2$ acts as a modulo operator. For $x \in \mathbb{R}$, $\bar{x} \triangleq 2^{\mathscr{F}_2(x)+a}$ is such that $2^a \leqslant \bar{x} < 2^{a+1}$.
Since the constraints (28) are equal to

$$2^{\boldsymbol{\alpha}_i} \leqslant \delta\sqrt{(\boldsymbol{W}_c)_{i,i}} < 2^{\boldsymbol{\alpha}_i+1} \tag{39}$$

and $\boldsymbol{T}$ transforms $(\boldsymbol{W}_c)_{i,i}$ into $\boldsymbol{T}_{i,i}^{-2}(\boldsymbol{W}_c)_{i,i}$, then $\boldsymbol{T}_{i,i}$ has to be of the form:

$$\delta\boldsymbol{T}_{i,i}^{-1}\sqrt{(\boldsymbol{W}_c)_{i,i}} = 2^{\mathscr{F}_2(\delta\sqrt{(\boldsymbol{W}_c)_{i,i}}) + \boldsymbol{\alpha}_i} \tag{40}$$

■

Thus, the optimization problem is given by

$$\boldsymbol{U}_{opt} = \underset{\substack{\boldsymbol{U} \text{ invertible} \\ \boldsymbol{V} \text{ defined by (38)}}}{\arg\min} \; M_{L_2}(\boldsymbol{U}\boldsymbol{V}). \tag{41}$$

These two ways of solving problem 3 are implemented in the FWR toolbox[5] for Matlab, with `fminsearch`, `fmincon` and `fminunc` functions, and they both give same results with similar numbers of iterations.
Of course, the use of matrices $\boldsymbol{V}_{\boldsymbol{\gamma}}$ and $\boldsymbol{V}$, that are merely used to eliminate the constraints and solve an unconstrained minimization problem, increases the degree of non-linearity for the objective function to minimize. However, this seems not to be a problem, since in our tests, the optimal realizations found seem to be global optima.

[5]sources available at `http://fwrtoolbox.gforge.inria.fr/`

6 Example

Let us consider the following state-space digital controller, given in modal form[6]:

$$\begin{aligned} \boldsymbol{A} &= \begin{pmatrix} 0.3820 & 0 & 0 \\ 0 & 0.7964 & 0.5598 \\ 0 & -0.5598 & 0.7964 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0.5391 \\ -0.8417 \\ 0.6232 \end{pmatrix}, \\ \boldsymbol{c} &= \begin{pmatrix} 0.1664 & 0.1639 & 0.2047 \end{pmatrix}, \quad d = 0.0159 \end{aligned} \tag{42}$$

and its multiple equivalent (in infinite precision) realizations:

- $\mathcal{R}_1$ is the original realization given by (42)
- $\mathcal{R}_2$ is the optimal L_2-scaled realization (solution of problem 2). It is obtained with proposition 2 and a quasi-newton algorithm. The numerical values are given by (43).
- $\mathcal{R}_3$ is the optimal relaxed-L_2-scaled realization (problem 3), with $\overset{\max}{u}$ a power of 2, and $\delta = 1$. It is obtained with proposition 4. The numerical values are given by (44).

The following table gives the M_{L_2} sensitivities of these different realizations:

realization	M_{L_2} **sensitivity**
$\mathcal{R}_1$	6355.5
$\mathcal{R}_2$	530.0964
$\mathcal{R}_3$	528.2532

In this example, the relaxed L_2-scaled realization $\mathcal{R}_3$ achieves lower sensitivity than the strict L_2-scaled optimal realization $\mathcal{R}_2$ while protecting implementation from overflows.
But it is not always the case : if we consider the example in [7], the optimal relaxed-L_2-scaled realization satisfies $(\boldsymbol{W}_c)_{i,i} = 1$ and is then also a strict L_2-scaled realization. This depends on the diagonal terms of the controllability Gramians of the (non scaled) optimal realization.
It is also interesting to notice that a good estimation of $\overset{\max}{u}$ (if it is not a power of 2) can allow to achieve lower sensitivity by moving the constraints (it could also be the case for the example in [7]).

$$\begin{aligned} \boldsymbol{A}_2 &= \begin{pmatrix} 0.65461 & -0.16286 & 0.48021 \\ 0.42726 & 0.64411 & 0.19385 \\ -0.19690 & -0.47595 & 0.67614 \end{pmatrix}, \quad \boldsymbol{b}_2 = \begin{pmatrix} 0.07080 \\ -0.41671 \\ 0.13160 \end{pmatrix}, \\ \boldsymbol{c}_2 &= \begin{pmatrix} 0.77215 & -0.01306 & 0.14582 \end{pmatrix}, \quad d_2 = 0.0159 \end{aligned} \tag{43}$$

$$\begin{aligned} \boldsymbol{A}_3 &= \begin{pmatrix} 0.68533 & 0.16894 & 0.49126 \\ -0.44207 & 0.62734 & -0.10822 \\ -0.27546 & 0.43413 & 0.66219 \end{pmatrix}, \quad \boldsymbol{b}_3 = \begin{pmatrix} -0.26649 \\ 0.50520 \\ -0.14719 \end{pmatrix}, \\ \boldsymbol{c}_3 &= \begin{pmatrix} 0.33283 & 0.40944 & 0.26399 \end{pmatrix}, \quad d_3 = 0.0159 \end{aligned} \tag{44}$$

[6] Due to a lack of space, only 4 digits are given, but more may be required to completely define the system.

7 Conclusion

This paper has presented the L_2-sensitivity minimization problem and the associated L_2-scaling constraints. These constraints that prevent from overflows have been considered with concrete fixed-point implementation schemes. Novel L_2-dynamic-range constraints have been exhibited.

Even if the goal of this paper is not a detailed optimization algorithm like in [7], two different means to solve the constrained optimization problem have been exhibited and applied on a numerical example.

These relaxed constraints could also be very important for some other realizations, like δ-operator state-space or the ρ-Direct Form II transposed [13]. For these realizations where a parameter Δ should be used to achieve the L_2-scaling, a relaxed-L_2-scaling permits to fix this parameter as a power of 2, in order to decrease the amount of computations.
To apply this work to other classical structures, it will be soon extended to the Specialized Implicit Framework [5] that allows to encompass existing structures in an implicit state-space form.

References

[1] P. Belanovic and M. Rupp. Automated floating-point to fixed-point conversion with the fixify environment. *Rapid System Prototyping, 2005. (RSP 2005). The 16th IEEE International Workshop on*, pages 172–178, June 2005.

[2] G. Constantinides, P.Y.K. Cheung, and W. Luk. *Synthesis And Optimization Of DSP Algorithms*. Kluwer Academic Publishers, 2004.

[3] M. Gevers and G. Li. *Parametrizations in Control, Estimation and Filtering Probems.* Springer-Verlag, 1993.

[4] T. Hilaire and P. Chevrel. On the compact formulation of the derivation of a transfer matrix with respect to another matrix. Technical Report RR-6760, INRIA, 2008.

[5] T. Hilaire, P. Chevrel, and J.F. Whidborne. A unifying framework for finite wordlength realizations. *IEEE Trans. on Circuits and Systems*, 8(54), August 2007.

[6] T. Hilaire, D. Ménard, and O. Sentieys. Bit accurate roundoff noise analysis of fixed-point linear controllers. In *Proc.* IEEE International Symposium on Computer-Aided Control System Design *(CACSD'08)*, September 2008.

[7] T. Hinamoto, H. Ohnishi, and W.-S. Lu. Minimization of l2 sensitivity of one- and two dimensional state-space digital filters subject to l2-dynamic-range-scaling constraints. *IEEE Trans. on Circuits and Systems-II*, 52(10):641–645, October 2005.

[8] T. Hinamoto and Y. Sugie. L2-sensitivity analysis and minimization of 2-d separable-denominator state-space digital filters. *Signal Processing, IEEE Transactions on*, 50(12):3107–3114, Dec 2002.

[9] S.Y. Hwang. Dynamic range constraint in state-space digital filtering. In *IEEE Trans. on Acoustics, Speech and Signal Processing*, volume 23, pages 591–593, 1975.

[10] S.Y. Hwang. Minimum uncorrelated unit noise in state-space digital filtering. *IEEE Trans. on Acoust., Speech, and Signal Processing*, 25(4):273–281, August 1977.

[11] L. Jackson. Roundoff-noise analysis for fixed-point digital filters realized in cascade or parallel form. *Audio and Electroacoustics, IEEE Transactions on*, 18(2):107–122, June 1970.

[12] S. Kim, KI. Kum, and W. Sung. Fixed-point optimization utility for C and C++ based digital signal processing programs. *IEEE Transactions on Circuits and Systems*, 45(11):1455–1464, November 1998.

[13] G. Li and Z. Zhao. On the generalized DFIIt structure and its state-space realization in digital filter implementation. *IEEE Trans. on Circuits and Systems*, 51(4):769–778, April 2004.

[14] C. Mullis and R. Roberts. Synthesis of minimum roundoff noise fixed point digital filters. In *IEEE Transactions on Circuits and Systems*, volume CAS-23, September 1976.

[15] K.K. Parhi. *VLSI Digital Signal Processing Systems: Design and Implementation of Digital Controllers*. Number ISBN 0-471-24186-5. John Wiley & Sons, 1999.

[16] V. Tavşanoğlu and L. Thiele. Optimal design of state-space digital filters by simultaneous minimization of sensibility and roundoff noise. In *IEEE Trans. on Acoustics, Speech and Signal Processing*, volume CAS-31, October 1984.

[17] L. Thiele. Design of sensitivity and round-off noise optimal state-space discrete systems. *Int. J. Circuit Theory Appl.*, 12:39–46, 1984.