International Journal of Control
Vol. 84, No. 1, January 2011, 66–77

Taylor & Francis
Taylor & Francis Group

Generalised modal realisation as a practical and efficient tool for FWL implementation

Yu Feng[a]*, Philippe Chevrel[a] and Thibault Hilaire[b]

[a]*Institut de Recherche en Cybernétique et Communication de Nantes (IRCCyN UMR CNRS 6597) and Département d'Automatique-Productique, École des Mines de Nantes, France;* [b]*Laboratory of Computer Science (LIP6), University Pierre & Marie Curie of Paris, France*

(*Received 26 April 2010; final version received 12 November 2010*)

Finite word length (FWL) effects have been a critical issue in digital filter implementation for almost four decades. Although some optimisations may be attempted to get an optimal realisation with regards to a particular effect, for instance the parametric sensitivity or the round-off noise gain, the purpose of this article is to propose an effective one, i.e. taking into account all the aspects. Based on the specialised implicit form, a new effective and sparse structure, named $\boldsymbol{\rho}$-modal realisation, is proposed. This realisation meets simultaneously accuracy (low sensitivity, round-off noise gain and overflow risk), few and flexible computational efforts with a good readability (thanks to sparsity) and simplicity (no tricky optimisation is required to obtain it) as well. Two numerical examples are included to illustrate the $\boldsymbol{\rho}$-modal realisation's interest.

Keywords: coefficient sensitivity; dynamic scaling; FWL implementation; implicit framework; modal realisation; pole sensitivity; round-off noise gain

1. Introduction

It is well known that there exists an infinite set of realisations to represent a given filter. These realisations are equivalent in infinite precision since they yield the same input-output relationship. However, when digital filters are implemented, they are implemented with finite precision due to the finite word length (FWL) of the representation of numbers within computing devices, and the FWL effects lead to a deterioration of realisations' numerical properties. These effects can be normally classified into two categories: the round-off noise resulting from the rounding of variables before and after each arithmetic calculation; and the parameters modification resulting from the quantisation of coefficients. Hence, the so-called 'equivalent' realisations are no longer equivalent in finite precision, and one realisation may be better suited for implementation than another.

Generally speaking, the FWL effects depend naturally on the chosen word length and arithmetic format (floating-point, fixed-point, etc.). They, however, also depend strongly upon the type of realisation. For example, δ-operator, introduced by Middleton and Goodwin (1990), normally has much better numerical properties than the usual delay operator q for fast sampling. Optimal filter implementation problem consists in finding a realisation with which the digital deterioration imposed by the FWL effects is minimised. Diverse structures and different digital operators have been investigated in the literature with this aim since the late 1970s (Mullis and Roberts 1976; Rao 1986; Gevers and Li 1993; Madievski, Anderson, and Gevers 1995; Li, Gevers, and Sun 2000; Yan and Teo 2002). In Chevrel (2002), rational operators suitable for discretisation of both LTI and LPV systems were introduced taking potentially into account the frequency bandwidth of each sub-system. Moreover, the authors proposed a direct-form II transposed structure in ρ-operator (ρDFIIt) in Li and Zhao (2004) and Hao and Li (2007). This form not only yields good performance against round-off noise, but also has sparse structure.

On the other hand, most of the significant results have expressed the filter in the state-space form. Although most realisations can be transformed into the state-space form, this form is not completely general and has several limitations. For instance, many realisation forms require the computation of intermediate variables that cannot be represented within the state-space form. The framework of the specialised implicit form (SIF), not subject to these restrictions, was given by Hilaire, Chevrel, and Whidborne (2007a). It provides a generalised description of any realisation in a form allowing a straightforward analysis of the FWL effects.

In this article, motivated by the use of the multivariable ρ-operator in Li and Zhao (2004) and based on a

*Corresponding author. Email: yu.feng@mines-nantes.fr

ISSN 0020–7179 print/ISSN 1366–5820 online

DOI: 10.1080/00207179.2010.540714
http://www.informaworld.com

modal representation, a new structure, named ρ-modal realisation, is constructed within the framework of the SIF for implementation of the filters/controllers whose poles are distinct. Associated with a filter/controller of order n, this sparse and scaled realisation contains few inexactly-implemented[1] parameters, and is resilient to numerical errors. The present form achieves a good trade-off among sensitivity, round-off noises and computation efforts. Moreover, no optimisation process is required to obtain such a realisation.

This article is briefly outlined as follows. After recalling the SIF and the related analysis criteria in Section 2 and Section 3 as preliminaries, a new dynamic-range scaling, named the relaxed L_2-scaling is presented in Section 4. Then in Section 5, the particular ρ-modal realisation is proposed, while the optimisation on parameters under this structure with regard to different criteria is deduced in Section 6. Numerical illustrations are given in Section 7.

2. A unifying framework

Many useful realisations, such as δ-based realisations, require intermediate computational variables that cannot be expressed in the state-space form. The SIF proposed in Hilaire et al. (2007a) provides an explicit description of the parameters and variables involved during implementation. The SIF representation is given by

$$\begin{pmatrix} J & 0 & 0 \\ -K & I_n & 0 \\ -L & 0 & I_p \end{pmatrix} \begin{pmatrix} T(k+1) \\ X(k+1) \\ Y(k) \end{pmatrix} = \begin{pmatrix} 0 & M & N \\ 0 & P & Q \\ 0 & R & S \end{pmatrix} \begin{pmatrix} T(k) \\ X(k) \\ U(k) \end{pmatrix}, \tag{1}$$

in which

- $J \in \mathbb{R}^{l\times l}$, $K \in \mathbb{R}^{n\times l}$, $L \in \mathbb{R}^{p\times l}$, $M \in \mathbb{R}^{l\times n}$, $N \in \mathbb{R}^{l\times m}$, $P \in \mathbb{R}^{n\times n}$, $Q \in \mathbb{R}^{n\times m}$, $R \in \mathbb{R}^{p\times n}$ and $S \in \mathbb{R}^{p\times m}$;
- $U(k)$ is the vector of the m current inputs, $Y(k)$ is the p current outputs; $T(k+1)$ is the vector for the l intermediate variables used in the calculations of step k, while $X(k+1)$ is the vector of n new state variables stored till the next sampling time; $X(k)$ and $T(k)$ form the generalised variables;
- J is a lower triangular matrix with 1s in the diagonal;
- The computations associated with the realisation (1) are executed in row order

$$\begin{aligned} &\text{(i)} \quad JT(k+1) \leftarrow MX(k) + NU(k), \\ &\text{(ii)} \quad X(k+1) \leftarrow KT(k+1) + PX(k) + QU(k), \\ &\text{(iii)} \quad Y(k) \leftarrow LT(k+1) + RX(k) + SU(k). \end{aligned} \tag{2}$$

The related transfer function is defined by

$$H : z \mapsto C_Z(zI_n - A_Z)^{-1}B_Z + D_Z, \tag{3}$$

with

$$A_Z \triangleq KJ^{-1}M + P, \quad B_Z \triangleq KJ^{-1}N + Q, \tag{4}$$

$$C_Z \triangleq LJ^{-1}M + R, \quad D_Z \triangleq LJ^{-1}N + S. \tag{5}$$

Definition 1 (Hilaire et al. 2007a)**:** A realisation $\mathcal{R}$ is defined by the specific set of matrices J, K, L, M, N, P, Q, R and S in (1) as

$$\mathcal{R} \triangleq (J, K, L, M, N, P, Q, R, S). \tag{6}$$

The coefficients can also be regrouped into one matrix Z as

$$Z \triangleq \left(\begin{array}{c|c|c} -J & M & N \\ \hline K & P & Q \\ \hline L & R & S \end{array}\right), \tag{7}$$

and $\mathcal{R}$ can be defined by $\mathcal{R} \triangleq (Z, l, m, n, p)$ where l, m, n and p are the dimensions of the underlying matrices.

See Hilaire et al. (2007a) and Hilaire, Chevrel, and Whidborne (2010) for more details on how to transform classical structures (cascade/parallel decomposition, state-space realisation, δ-realisation, lattice, ...) into the SIF.

Equivalent structured realisations can be defined through block diagonal similarity transform as

$$Z_1 = \begin{pmatrix} \mathcal{Y} & & \\ & \mathcal{U}^{-1} & \\ & & I_p \end{pmatrix} Z_0 \begin{pmatrix} \mathcal{W} & & \\ & \mathcal{U} & \\ & & I_m \end{pmatrix}, \tag{8}$$

with $\mathcal{Y}$, $\mathcal{U}$ and $\mathcal{W}$ invertible matrices.

3. Criterion analysis

3.1 *Input-output sensitivity*

In order to evaluate how much the digital implementation modifies filters' characteristics, the input-output (I/O) sensitivity measure is introduced. Consider a state-space system, denoted by (A, B, C, D). Measure of the transfer function sensitivity through its L_2-norm is defined as (Gevers and Li 1993)

$$M_{L_2} \triangleq \left\|\frac{\partial H}{\partial A}\right\|_2^2 + \left\|\frac{\partial H}{\partial B}\right\|_2^2 + \left\|\frac{\partial H}{\partial C}\right\|_2^2 + \left\|\frac{\partial H}{\partial D}\right\|_2^2. \tag{9}$$

Considering that the coefficients quantised without error make no contribution to the overall I/O

sensitivity, a weighting matrix W_Z associated with Z is then introduced

$$(W_Z)_{i,j} \triangleq \begin{cases} 0, & \text{if } Z_{i,j} \in \{0, \pm 1\}; \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

Given a realisation $\mathcal{R} \triangleq (Z, l, m, n, p)$ with the weighting matrix W_Z, the I/O transfer function sensitivity is defined in the single-input-single-output (SISO) case by

$$M_{L_2}^W \triangleq \left\| \frac{\partial H}{\partial Z} \times W_Z \right\|_2^2, \tag{11}$$

where $\times$ is the Schur product.

Then, this L_2-norm sensitivity can be computed by the following lemmas.

Lemma 1 (Gevers and Li 1993)**:** *Consider a state-space system $G := (\Phi, \Psi, \Omega, \Upsilon)$. Its L_2-norm can be computed by*

$$\begin{aligned} \|G\|_2^2 &= \operatorname{trace}(\Upsilon\Upsilon^\top + \Omega W_c \Omega^\top) \\ &= \operatorname{trace}(\Upsilon^\top \Upsilon + \Psi^\top W_o \Psi), \end{aligned} \tag{12}$$

where W_c and W_o are, respectively, the controllability and observability Gramian. They are solutions to the following Lyapunov equations:

$$W_c = \Phi W_c \Phi^\top + \Psi\Psi^\top, \quad W_o = \Phi^\top W_o \Phi + \Omega^\top \Omega. \tag{13}$$

Lemma 2 (Hilaire et al. 2007a)**:** *The sensitivity with regard to each matrix in the SIF can be written as*

$$\frac{\partial H}{\partial Z} = H_1^\top H_2^\top, \tag{14}$$

with

$$\begin{cases} H_1 : z \mapsto C_Z (zI_n - A_Z)^{-1} M_1 + M_2, \\ H_2 : z \mapsto N_1 (zI_n - A_Z)^{-1} B_Z + N_2, \\ M_1 \triangleq \begin{pmatrix} KJ^{-1} & I_n & 0 \end{pmatrix}, \quad M_2 \triangleq \begin{pmatrix} LJ^{-1} & 0 & 1 \end{pmatrix}, \\ N_1 \triangleq \begin{pmatrix} M^\top J^{-\top} & I_n & 0 \end{pmatrix}^\top, \quad N_2 \triangleq \begin{pmatrix} N^\top J^{-\top} & 0 & 1 \end{pmatrix}^\top. \end{cases} \tag{15}$$

3.2 *Pole sensitivity measure*

Some pole sensitivity measures are also commonly used to inform about the robust stability of FWL implementation. During quantisation process, the Z matrix is perturbed to $Z + \varepsilon \times W_Z$, where ε represents digital perturbations. Hence, the poles of the implemented realisation may be shifted outside the unit circle even if the initial realisation is stable. Based on this consideration, a stability measure is proposed (Fialho and Georgiou 1994)

$$\mu_0(Z) = \inf_{\varepsilon} \{ \|\varepsilon\|_{\max} / Z + \varepsilon \times W_Z \text{ instable} \}. \tag{16}$$

Since this measure is numerically difficult to evaluate, the following measure is most often used:

$$\mu(Z) \triangleq \min_{1 \le k \le n} \frac{1 - |\lambda_k|}{\|W_Z\|_F \left\| \frac{\partial |\lambda_k|}{\partial Z} \times W_Z \right\|_F}, \tag{17}$$

where λ_k denotes the kth pole of the system and $\|\cdot\|_F$ stands for the Frobenius norm.

This measure can be evaluated by the following lemma.

Lemma 3 (Hilaire et al. 2007a)**:**

$$\frac{\partial |\lambda_k|}{\partial Z} = M_1^\top \frac{\partial |\lambda_k|}{\partial A} M_2^\top, \tag{18}$$

where M_1 and M_2 are the matrices defined in Lemma 2. *Moreover $\frac{\partial |\lambda_k|}{\partial A}$ can be easily computed from the underlying right eigenvectors of A.*

3.3 *Round-off noise gain*

The round-off noise gain (RNG) is another criterion for the analysis of a realisation. Considering the quantisation noises after each multiplication, the algorithm in (2) becomes

(i) $JT(k+1) \leftarrow MX(k) + NU(k) + \xi_T(k)$,
(ii) $X(k+1) \leftarrow KT(k+1) + PX(k) + QU(k) + \xi_X(k)$,
(iii) $Y(k) \leftarrow LT(k+1) + RX(k) + SU(k) + \xi_Y(k)$,

where ξ_T, ξ_X and ξ_Y are the noise sources corrupting T, X and Y. These noises are usually modelled as independent white sequences.

Denote ξ the vector formed by all the noises: $\xi(k)^\top = (\xi_T(k)^\top \ \xi_X(k)^\top \ \xi_Y(k)^\top)$. It is possible to aggregate all of them as an additive noise $\xi'(k)$ on the output. This (coloured) noise results from the filtering of $\xi(k)$ via the transfer function H_1 defined in (15) (Figure 1; Hilaire, Ménard and Sentieys 2007b).

Definition 2 (Mullis and Roberts 1976)**:** Suppose that the round-off noises ξ have the same power σ_o^2.

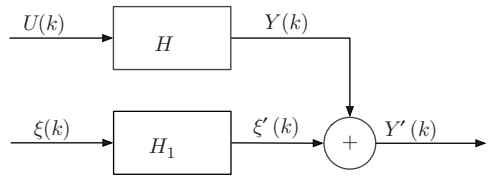

Figure 1. Equivalent noised model.

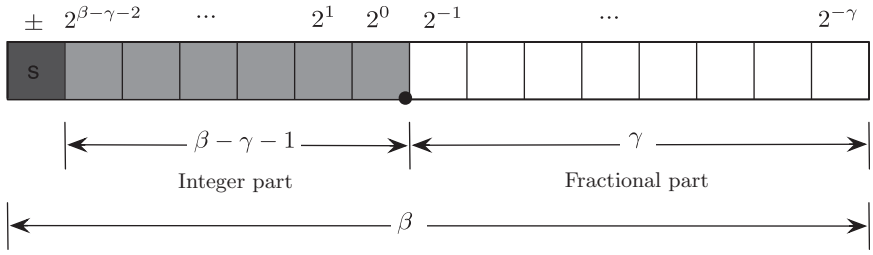

Figure 2. Fixed-point representation.

The round-off noise gain measure is then defined by the ratio of the power of global noise $\xi'(k)$ and σ_o^2

$$G \triangleq \frac{E\{\xi'(k)^\top \xi'(k)\}}{\sigma_o^2}, \tag{19}$$

where $E\{\cdot\}$ is the mean operator.

Lemma 4:

$$G = \mathrm{trace}\big(d_Z(M_1^\top W_o M_1 + M_2^\top M_2)\big), \tag{20}$$

in which d_Z is a diagonal matrix with $(d_Z)_{i,i}$ denoting the number of non-trivial parameters in the ith row of Z (except 0, ±1 and powers of 2), and W_o is the observability Gramian of the state-space system (A_Z, B_Z, C_Z, D_Z).

4. L_2-scaling

The L_2-dynamic-range scaling constraints were introduced by Jackson (1970) and Hwang (1975). They consist in scaling the state variables in a way to prevent overflows or underflows. The L_2-scaling also contributes to normalise the format of different state variables and the aforementioned sensitivity criteria.

Definition 3: A SISO state-space system (A, B, C, D) is said to be L_2-scaled if the transfer functions from input to each state have a unitary L_2-norm $(1 \le i \le n)$

$$\left\| e_i^\top (zI - A)^{-1} B \right\|_2 = 1, \tag{21}$$

in which e_i is the vector with a 1 in the ith position and 0s elsewhere.

This definition can be extended to the SIF via unitary scaling of state and intermediate variables, i.e. $(1 \le i \le n, 1 \le j \le l)$

$$\begin{aligned} \|e_i^\top (zI - A_Z)^{-1} B_Z\|_2 &= 1, \\ \|e_j^\top \left(J^{-1} M (zI - A_Z)^{-1} B_Z + J^{-1} N\right)\|_2 &= 1. \end{aligned} \tag{22}$$

According to Lemma 1, (22) can be expressed as

$$\begin{aligned} (W_{cX})_{i,i} &= 1 \quad \forall 1 \le i \le n, \\ (W_{cT})_{j,j} &= 1 \quad \forall 1 \le j \le l, \end{aligned} \tag{23}$$

where W_{cX} and W_{cT} are the controllability Gramians with regard to the state and intermediate variables, respectively

$$\begin{cases} W_{cX} = A_Z W_{cX} A_Z^\top + B_Z B_Z^\top, \\ W_{cT} = J^{-1} M W_{cX} M^\top J^{-\top} + J^{-1} N N^\top J^{-\top}. \end{cases} \tag{24}$$

Recently in Hilaire (2009), new dynamic-range-scaling constraints have been proposed. We extend these constraints here to the SIF.

It appears that, in fixed-point format, the classical L_2-scaling constraints may be uselessly too strict. It is enough, for preventing overflows, to force all state and intermediate variables to possess the same binary-point position as the input.

Figure 2 illustrates the representation of a fixed-point position, in which β is the total word length in bits of the representation, while γ denotes the fractional part word length (it gives the binary-point position of the representation). Unlike floating-point representation, they are implicitly fixed for each variable and coefficient. In this section, β and γ will be suffixed by the variable/state/coefficient they refer to.

To represent a value x without overflows, a fixed-point representation (β_x, γ_x) should satisfy

$$\beta_x - \gamma_x - 1 \ge \lfloor \log_2 |x| \rfloor + 1, \tag{25}$$

where the $\lfloor a \rfloor$ operator rounds a to the nearest integer lower or equal to a.

Proposition 1 (Overflows)**:** *Let $\overset{\max}{X_i}$ and $\overset{\max}{T_j}$ be the maximum magnitude for the ith state and the jth intermediate variable, respectively. Then, the best binary-point positions $\{\gamma_{X_i}\}$ and $\{\gamma_{T_j}\}$ avoiding overflows are given by*

$$\gamma_{X_i} = \beta_{X_i} - 2 - \left\lfloor \log_2 \overset{\max}{X_i} \right\rfloor, \tag{26}$$

$$\gamma_{T_j} = \beta_{T_j} - 2 - \left\lfloor \log_2 \overset{\max}{T_j} \right\rfloor. \tag{27}$$

Proof: Applying (25) to the ith state gives $\beta_{X_i} - \gamma_{X_i} - 2 \ge \lfloor \log_2 |X_i(k)| \rfloor \ \forall k$. So γ_{X_i} satisfies $\gamma_{X_i} \le \beta_{X_i} - 2 - \lfloor \log_2 \overset{\max}{X_i} \rfloor$, and has the greatest possible value to increase the precision of the represented value. □

Let E_i be a state or an intermediate variable, and G_i the transfer function from the input to this variable. Although it is impossible to evaluate the maximum magnitude of E_i, some upper bounds can be calculated. Denote $\overset{\text{up}}{E}_i$ the estimation of E_i by L_1-norm,

$$\overset{\text{up}}{E}_i = \|G_i\|_1 \overset{\max}{U}, \tag{28}$$

or by L_2-norm:

$$\overset{\text{up}}{E}_i \simeq \kappa \|G_i\|_2 \overset{\max}{U}, \tag{29}$$

where the parameter κ can be interpreted as a representation of the value of the standard deviation of E_i, if the input is unit-variance white centred noise ($\kappa \geq 1$). Since the L_2-norm estimation in (29) does not give a strict bound, contrary to the L_1-norm case, κ can be viewed as a safety parameter.

In general, the L_1 and L_2-estimation of $\overset{\text{up}}{E}_i$ approximatively lead to the same binary-point position,within one or two bits. However, compared to the L_1-norm, the L_2-one is much less conservative and more tractable, hence (29) is practically used, with $\kappa = 1$. A simulation-based estimation, (see e.g. Belanovic and Rupp (2005) or Kim, Kum, and Sung (1998)), can be adapted after implementation to verify *in situ* the peak values and binary-point positions, accordingly to the inputs. Finally, these upper bounds are used to set the binary-point positions with

$$\gamma_{E_i} = \beta_{E_i} - 2 - \left\lfloor \log_2 \overset{\text{up}}{E}_i \right\rfloor. \tag{30}$$

Generally, two treatments are possible to prevent overflows:

- set the binary-point positions for each state and intermediate variable according to (26), (27) and (30), to make sure that the fixed-point representation can be dealt with their maximum peak values;
- or choose binary-point positions for each state and intermediate variable, and apply a scaling on them for adapting the peak value of each state or intermediate variable to the chosen binary-point positions.

The classical L_2-scaling corresponds to the second option, by imposing $\overset{\max}{E}_i = \overset{\max}{U}$, i.e. $\|G_i\|_2 = 1$. However, it has been shown in Hilaire (2009) that overflows can be avoided by setting the same binary-point position for the state, intermediate variables and inputs. This fact results in the constraints $\gamma_{E_i} = \gamma_U$ for the scaling, instead of $\overset{\max}{E}_i = \overset{\max}{U}$.

The following proposition exhibits these new constraints.

Proposition 2 (Relaxed L_2-scaling constraints)**:** *Assume that it is possible to get a good estimation of the upper bound of the generalised* (*state and intermediate*) *variables through the L_2-measure. In order to implement the input and the generalised variables with the same binary-point position, the L_2-scaling constraints* (23) *are transformed into* ($1 \leq i \leq n$, $1 \leq j \leq l$):

$$\begin{cases} \dfrac{2^{2\alpha_{X_i}}}{\kappa^2} \leq (W_{cX})_{i,i} < 4\dfrac{2^{2\alpha_{X_i}}}{\kappa^2}, \\ \dfrac{2^{2\alpha_{T_j}}}{\kappa^2} \leq (W_{cT})_{j,j} < 4\dfrac{2^{2\alpha_{T_j}}}{\kappa^2}, \end{cases} \tag{31}$$

where

$$\begin{cases} \alpha_{X_i} = \beta_{X_i} - \beta_U - \mathscr{F}_2\left(\overset{\max}{U}\right), \\ \alpha_{T_j} = \beta_{T_j} - \beta_U - \mathscr{F}_2\left(\overset{\max}{U}\right), \end{cases} \tag{32}$$

and $\mathscr{F}_2(x)$ is defined as the fractional value of $\log_2(x)$: $\mathscr{F}_2(x) \triangleq \log_2(x) - \lfloor \log_2(x) \rfloor$.

Proof: γ_U is given by $\gamma_U = \beta_U - 2 - \left\lfloor \log_2 \overset{\max}{U} \right\rfloor$, so $\gamma_U = \gamma_{X_i}$ leads to

$$\beta_U - \left\lfloor \log_2 \overset{\max}{U} \right\rfloor = \beta_{X_i} - \left\lfloor \log_2 \left(\kappa \left\| e_i^\top (zI_n - A)^{-1} B \right\|_2 \overset{\max}{U} \right) \right\rfloor,$$

and

$$\left\lfloor \log_2 \left(\kappa \sqrt{(W_{cX})_{i,i}} \right) + \mathscr{F}_2 \left(\overset{\max}{U} \right) \right\rfloor = \beta_{X_i} - \beta_U.$$

Hence, $2^{\alpha_{X_i}} \leq \kappa \sqrt{(W_{cX})_{i,i}} < 2^{\alpha_{X_i}+1}$. □

Corollary 1: *For micro-controller or DSP implementation, the word length of all variables are equal* ($\beta_{X_i} = \beta_{T_j} = \beta_U$), *and* $\overset{\max}{U}$ *can be set to a power of* 2. *Then, if* $\kappa = 1$ (*as for classical L_2-scaling constraints*), *the relaxed L_2-scaling constraints become*

$$1 \leq (W_{cX})_{i,i} < 4, \quad 1 \leq (W_{cT})_{j,j} < 4. \tag{33}$$

Proposition 3 (*a posteriori* relaxed L_2-scaling)**:** *Given an SIF realisation. Relaxed L_2-scaling constraints* (31) *can be satisfied by applying the similarity transform* (8) *with diagonal matrices $\mathcal{U}$ and $\mathcal{W}$ such that*

$$\mathcal{U}_{i,i} = \kappa \sqrt{(W_{cX})_{i,i}} 2^{-\mathscr{F}_2\left(\sqrt{(W_{cX})_{i,i}}\right) - \alpha_{X_i}}, \tag{34}$$

$$\mathcal{W}_{j,j} = \kappa \sqrt{(W_{cT})_{j,j}} 2^{-\mathscr{F}_2\left(\sqrt{(W_{cT})_{j,j}}\right) - \alpha_{T_j}}. \tag{35}$$

Not specified, yet $\mathcal{Y}$ should be chosen to preserve the structure of the realisation, for instance $\mathcal{Y} = I_n$ or $\mathcal{Y} = \mathcal{W}^{-1}$.

Proof: $\mathscr{F}_2$ acts as a modulo operator. For $x \in \mathbb{R}$, $\overline{x} \triangleq 2^{\mathscr{F}_2(x)+\alpha}$ is so that $2^\alpha \le \overline{x} < 2^{\alpha+1}$. The constraints (31) are equal to

$$2^{\alpha_{X_i}} \le \kappa\sqrt{(W_{cX})_{i,i}} < 2^{\alpha_{X_i}+1},$$
$$2^{\alpha_{T_j}} \le \kappa\sqrt{(W_{cT})_{j,j}} < 2^{\alpha_{T_j}+1}.$$

Moreover, $\mathcal{U}$, $\mathcal{W}$ transform W_{cX} and W_{cT} into $\mathcal{U}^{-1}W_{cX}\mathcal{U}^{-\top}$ and $\mathcal{W}^{-1}W_{cT}\mathcal{W}^{-\top}$, respectively. Hence, $\mathcal{U}_{i,i}$ and $\mathcal{W}_{i,i}$ should be

$$\kappa\mathcal{U}_{i,i}^{-1}\sqrt{(W_{cX})_{i,i}} = 2^{\mathscr{F}_2\left(\sqrt{(W_{cX})_{i,i}}\right)+\alpha_{X_i}},$$
$$\kappa\mathcal{W}_{j,j}^{-1}\sqrt{(W_{cT})_{j,j}} = 2^{\mathscr{F}_2\left(\sqrt{(W_{cT})_{j,j}}\right)+\alpha_{T_j}}.$$

□

5. Special ρ-based modal realisation

In the parts to follow, the poles of the considered transfer function are supposed to be distinct. This assumption is not restrictive, as transfer functions with multiple poles are often avoided due to their ill-conditioning. Moreover, the present result of this article can be equally extended to the case of multiple poles by using the Jordan form and changing slightly the development below.

Consider the transfer function $H(z)$ and its related modal realisation (Λ, B, C, D):

$$H(z) = D + C(zI - \Lambda)^{-1}B = D + \sum_{i=1}^{n}\frac{c_i b_i}{1 - \lambda_i z^{-1}}, \quad (36)$$

with $\lambda_i \neq \lambda_j$ for all $i \neq j$ so that Λ can be chosen as a diagonal matrix.

Rather to diagonalise the A-matrix, it is preferred in the sequel to combine the complex-conjugate pole pairs to form a real 'block-diagonal' section in which Λ has two-by-two real matrices along its diagonal as follows:

$$\Lambda = \begin{pmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_2 & \alpha_2 & & & & \\ & \alpha_3 & \beta_3 & & & \\ & \beta_4 & \alpha_4 & & & \\ & & & \ddots & & \\ & & & & \alpha_{n-1} & \beta_{n-1} \\ & & & & \beta_n & \alpha_n \end{pmatrix}, \quad (37)$$

where α_i and β_i are linked to the real and imaginary parts of the ith pole, respectively. If the ith pole is real, then $\beta_i = 0$; if the ith and $(i+1)$th poles are complex-conjugate, then $\alpha_i = \alpha_{i+1}$ and $\beta_i = -\beta_{i+1} = \mathrm{Im}(\lambda_i)$.

Remark 1: The modal representation is not unique since B and C may be scaled in compensation ways to produce the same transfer function, and the diagonal elements of Λ may also be permuted. One invariant however is that modal representation decouples the dynamic modes λ_i and is closely related to the partial-fraction expansion of $H(z)$. Some linear algebra libraries, like the LAPACK library (Anderson et al. 1999), can be used for the modal decomposition.

Such a modal form has intrinsically good features. First, it has low pole sensitivity: perturbation on coefficients of (37) directly affects the poles so that a slight deviation will not lead to a dramatic change on poles. While using other forms, it may not be the case. For example, with the companion form, a tiny perturbation on the coefficients of the characteristic equation may imply a huge pole displacement. Second, it performs quite well in terms of quantisation noises because the whole system is decomposed and realised by several parallel second- (or first-) order sections. As known, these low-order sections are less sensitive to quantisation noises (Oppenheim, Schafer, and Buck 1999). Besides, the sensitivity may be still reduced under the relaxed L_2-scaling constraints by simply choosing β_i potentially different from β_{i+1}, with an appropriately scaled B matrix.

With the above modal form, the whole system is 'decoupled' in terms of parallel cells. Each individual cell is realised by the real and imaginal part of the underlying pair of eigenvalues. An alternative realisation of each cell of Λ, denoted by Λ_i, resulting in the same eigenvalue, is to take the form

$$\Lambda_i = \begin{pmatrix} 0 & 1 \\ \omega_{1i} & \omega_{2i} \end{pmatrix}, \quad (38)$$

where $\omega_{1i} = \lambda_i + \lambda_{i+1}$, $\omega_{2i} = -\lambda_i\lambda_{i+1}$ and λ_i, λ_{i+1} are complex-conjugate.

Mantey (1968) noted that cells of the form of (38) require fewer multiplicative operations than those with modal form. In fact, using modal form, each pair of eigenvalues needs two addition multiplications for computation of each output. However, for a given specification, about half the number of bits is required for the parameters of each cell with modal representation as for those of (38). This indicates that modal decomposition outperforms the form of (38) in terms of pole sensitivity.

Now we are in a position to propose a spare realisation, named ρ-modal realisation. It will be clear that the transformation to get this realisation preserves the dynamic scaling, while improving the implementation. Let us first define the following sequence of first order polynomial operators, named ρ-operators

$$\rho_i = \frac{q - \gamma_i}{\Delta_i}, \quad 1 \le i \le n, \quad (39)$$

$\{\gamma_i\}$ and $\{\Delta_i > 0\}$ are two sets of constants to determine. The particular choice $\gamma_i = 0$ and $\Delta_i = 1$ (resp., $\gamma_i = 1$) leads to the shift operator (resp. the δ-operator). The SIF related to the ρ-operator has the following structure:

$$\begin{pmatrix} I & 0 & 0 \\ -\Delta & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} T(k+1) \\ X(k+1) \\ Y(k) \end{pmatrix} = \begin{pmatrix} 0 & A_\rho & B_\rho \\ 0 & \gamma & 0 \\ 0 & C_\rho & D_\rho \end{pmatrix} \begin{pmatrix} T(k) \\ X(k) \\ U(k) \end{pmatrix}, \tag{40}$$

with

$$A_\rho = \Delta^{-1}(\Lambda - \gamma), \quad B_\rho = \Delta^{-1}B, \quad C_\rho = C, \quad D_\rho = D, \tag{41}$$

$$\Delta = \mathrm{diag}(\Delta_1, \ldots, \Delta_n), \quad \gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_n). \tag{42}$$

Besides, the matrix Z in (7) in this case is parameterised as:

$$Z \triangleq \left(\begin{array}{c|c|c} -I & A_\rho & B_\rho \\ \hline \Delta & \gamma & 0 \\ \hline 0 & C_\rho & D_\rho \end{array}\right). \tag{43}$$

Moreover, according to the number of complex poles, the number of inexactly-implemented coefficients is between $3n+1$ and $4n+1$, plus $2n$ free parameters $\{\Delta_i\}$ and $\{\gamma_i\}$.

Proposition 4 (ρ-modal realisation)**:** *The algorithm to establish the ρ-modal realisation is given as follows*:

(1) *Start with a q-based modal realisation*;
(2) *Relaxed-L_2-scale it according to Proposition* 3;
(3) *Build the equivalent ρ-realisation described in* (40), *then choose $\{\gamma_i\}$ to minimise the FWL effects*;
(4) *Deduce $\{\Delta_i\}$ to realise the relaxed L_2-scaling of the intermediate variables*.

Steps (3) and (4) are detailed later, and the results are given in Proposition 6 and Equation (46), respectively.

To make more precise the way to perform the relaxed L_2-scaling on both the state and intermediate variables, let us reconsider the Gramian's equation (24) which is described for the ρ-modal realisation as

$$\begin{cases} W_{cX} = \Lambda W_{cX} \Lambda^\top + BB^\top, \\ W_{cT} = \Delta^{-2}[(\Lambda - \gamma) W_{cX} (\Lambda - \gamma)^\top + BB^\top]. \end{cases} \tag{44}$$

It observes that W_{cX} is independent of the choice of $\{\gamma_i\}$ and $\{\Delta_i\}$. This fact indicates that the relaxed L_2-scaling of $T(k)$ can be fulfilled by simply choosing appropriate $\{\gamma_i\}$ and $\{\Delta_i\}$ sets. In order to explain the way to get a scaled $T(k)$, we denote $\widetilde{W}_{cT}$ as the solution of W_{cT} in (44) with $\Delta = I_n$. Consequently, the condition $1 \le (W_{cT})_{i,i} < 4$ is equal to

$$1 \le \Delta_i^{-2} (\widetilde{W}_{cT})_{i,i} < 4. \tag{45}$$

Obviously any

$$\Delta_i \in \left] \frac{1}{2} \sqrt{(\widetilde{W}_{cT})_{i,i}}, \sqrt{(\widetilde{W}_{cT})_{i,i}} \right]$$

achieves the relaxed L_2-scaling constraints. Hence, giving the set of $\{\gamma_i\}$, it is always possible to choose the set of $\{\Delta_i\}$ as powers of 2 according to the following expression, while keeping the relaxed L_2-scaling as well.

$$\Delta_i = 2^{\left\lfloor \sqrt{(\widetilde{W}_{cT})_{i,i}} \right\rfloor}. \tag{46}$$

Remark 2: The same trick can be used to create a ρDFIIt realisation with which the state satisfies the relaxed L_2-scaling constraints. Instead of setting the $\{\Delta_i\}$ with (Li and Zhao 2004, Equation 25)

$$\Delta_1 = \sqrt{(W_c)_{1,1}}, \quad \Delta_k = \sqrt{\frac{(W_c)_{k,k}}{(W_c)_{k-1,k-1}}}, \quad k \le 2, \tag{47}$$

the following expression is used:

$$\Delta_1 = 2^{\left\lfloor \sqrt{(W_c)_{1,1}} \right\rfloor}, \quad \Delta_k = 2^{\left\lfloor \sqrt{\frac{(W_c)_{k,k}}{(W_c)_{k-1,k-1}}} \right\rfloor}, \quad k \le 2. \tag{48}$$

The resulting realisation requires fewer operations, since $\{\Delta_i\}$ are now powers of 2. This however is not yet enough to L_2-scale the intermediate variables for the ρDFIIt realisation.

6. Optimisation of ρ-modal realisation

The parameters optimisation of the ρ-modal realisation with respect to different criteria is derived in this section, and an analytical solution to the optimised ρ-modal realisation is exhibited.

Consider a given state-space realisation $\mathcal{R}_0 := (A, B, C, D)$ of $H(z)$. Let $\mathscr{C}_{\rho L_2}$ be the set of relaxed-L_2-scaled ρ-modal realisations equivalent to $\mathcal{R}_0$, that is the realisations described by (40) and satisfying (33).

In what follows, our objective is to find the best realisation within this relaxed-L_2-scaled ρ-based equivalent class $\mathscr{C}_{\rho L_2}$.

Proposition 5: *The solution (Δ^*, γ^*) such as*

$$Z(\mathcal{R}_0, \Delta^*, \gamma^*) = \underset{\mathcal{R} \in \mathscr{C}_{\rho L_2}}{\arg\min}\ \mathcal{J}(Z), \tag{49}$$

where $\mathcal{J}$ is one of the criteria defined in Section 3, *can be obtained in a simple way through minimising Δ_i under the constraint* (45).

It is noted that Proposition 5 is important since it makes the minimisation problem much simpler than in the general case. Its proof is detailed in the following three sections.

Corollary 2: (Δ^*, γ^*) *can be obtained explicitly according to Equations* (46) *and* (54).

6.1 *I/O sensitivity minimisation*

With the proposed ρ-modal form, from the preceding discussion, Δ_i can be implemented exactly, hence the modification of γ_i only relates to the sensitivity of the matrices A_ρ, B_ρ in (40).

According to Lemma 2, the sensitivities with respect to these two matrices can be written as follows:

$$\frac{\partial H}{\partial A_\rho} = \left(C(zI-\Lambda)^{-1}\Delta\right)^\top \left((zI-\Lambda)^{-1}B\right)^\top, \tag{50}$$

$$\frac{\partial H}{\partial B_\rho} = \left(C(zI-\Lambda)^{-1}\Delta\right)^\top, \tag{51}$$

which can be viewed as transfer functions of the following state-space systems, respectively:

$$\Gamma_1 := \left(\begin{pmatrix}\Lambda & BC\\ 0 & \Lambda\end{pmatrix}, \begin{pmatrix}0\\ \Delta\end{pmatrix}, \begin{pmatrix}I_n & 0\end{pmatrix}, 0\right), \tag{52}$$

$$\Gamma_2 := (\Lambda, \Delta, C, 0). \tag{53}$$

Lemma 5: *The minimisation of the L_2-norm of $\frac{\partial H}{\partial A_\rho}$ and $\frac{\partial H}{\partial B_\rho}$ is equivalent to the minimisation of Δ_i.*

Proof: Using Lemma 1, $\left\|\frac{\partial H}{\partial B_\rho}\right\|_2^2 = \mathrm{trace}(\Delta^\top W_o \Delta)$. By noting that the observability Gramian W_o is invariant to the choice of Δ_i, it is clear that the minimisation of $\left\|\frac{\partial H}{\partial B_\rho}\right\|_2^2$ is equivalent to the minimisation of Δ_i. The same result is obtained when considering minimising $\left\|\frac{\partial H}{\partial A_\rho}\right\|_2^2$. □

Due to (45), it appears that the minimisation of Δ_i is linked to the minimisation of the diagonal terms of $\widetilde{W}_{cT}$.

Proposition 6 (Optimal $\{\gamma_i\}$)**:** *The diagonal terms of $\widetilde{W}_{cT}$ are minimised by the choice of γ_i as follows*:

$$\gamma_i = \begin{cases} \alpha_i + \dfrac{\beta_i \left(W_{cX}\right)_{i+1,i}}{\left(W_{cX}\right)_{i,i}}, & i \text{ is odd},\\ \alpha_i + \dfrac{\beta_i \left(W_{cX}\right)_{i,i-1}}{\left(W_{cX}\right)_{i,i}}, & i \text{ is even}. \end{cases} \tag{54}$$

Proof: See Appendix A. □

Corollary 3: *The choice of $\{\gamma_i\}$ in* (54) *leads to the lowest I/O sensitivity*.

Remark 3: The ρ-modal realisation proposed here retains the best combination of $\{\Delta_i\}$ and $\{\gamma_i\}$. Once $\{\gamma_i\}$ are computed, $\{\Delta_i\}$ are consequently determined to meet the relaxed L_2-scaling constraints.

6.2 *Pole sensitivity minimisation*

Proposition 7: *With the ρ-modal realisation, the minimisation of the pole sensitivity is equivalent to the minimisation of Δ_i.*

Proof: Applying Lemma 3 to the ρ-modal realisation, the pole sensitivity is exclusively related to the matrices M and P with

$$\frac{\partial |\lambda_k|}{\partial M} = \Delta \frac{\partial |\lambda_k|}{\partial A_Z} I_n, \quad \frac{\partial |\lambda_k|}{\partial P} = \frac{\partial |\lambda_k|}{\partial A_Z}.$$

Since A_Z only depends on the relaxed scaled q-based modal realisation, $\frac{\partial |\lambda_k|}{\partial A_Z}$ is invariant to the choice of Δ_i. It is easy to see that the minimisation of $\left\|\frac{\partial |\lambda_k|}{\partial M}\right\|_F$ and $\left\|\frac{\partial |\lambda_k|}{\partial P}\right\|_F$ is equal to the minimisation of Δ_i. □

6.3 *Round-off noise gain minimisation*

Owing to its parallel structure and second-order/first-order sub-sections, the modal realisation is less sensitive to quantisation noises. According to (20), the RNG of the ρ-modal realisation is

$$G = \mathrm{trace}\left(d_Z \begin{pmatrix}\Delta & I_n & 0\end{pmatrix}^\top W_o \begin{pmatrix}\Delta I_n 0\end{pmatrix} + \begin{pmatrix}0 & 0 & 1\end{pmatrix}^\top \begin{pmatrix}0 & 0 & 1\end{pmatrix}\right). \tag{55}$$

As W_o is the observability Gramian of (Λ, B, C, D), it is constant with respect to Δ_i. The minimisation of G is hence equal to choosing Δ_i as small as possible. This converges to the same optimal conditions as previously deduced for I/O sensitivity and pole sensitivity minimisation.

So far the feature of ρ-modal realisation has been derived and minimisations of different criteria converge to the unique condition, whose solution can be obtained through the analytical formula (54) without requiring any tricky optimisation algorithm.

Remark 4: Once $\{\Delta_i\}$ is fixed and implemented exactly, the sensitivities and RNG do not depend anymore on the choice of $\{\gamma_i\}$. It is also possible to further operate $\{\gamma_i\}$, for instance, rounding them to the nearest exactly-implemented numbers while keeping the system under the relaxed L_2-scaled constraints.

7. Numerical examples

Two numerical examples (Li and Zhao 2004) are presented in this section to illustrate the performance of the proposed realisation, and the comparison with some existing methods is also given.

In these examples, the optimal $\{\gamma_i\}$ for the ρ-modal realisation is rounded to the nearest exactly-implemented number. Here, only five bits are used to represent $\{\gamma_i\}$. Besides, the numerical parameterisations of the ρ-modal realisation are exhibited in Appendix B.

Five different realisations are compared:

Z_1: Cascade form with q-based second-order companion canonical sections;
Z_2: Optimised ρDFIIt[2] (Li and Zhao 2004);
Z_3: Equivalent state-space realisation of Z_2 (Li and Zhao 2004);
Z_4: q-based balanced realisation;
Z_5: ρ-modal realisation.

Numerical simulations are launched by using the FWR Toolbox[3] developed with MATLAB, and the I/O transfer function sensitivity is chosen as the criterion to optimise Z_2. Note that, in Li and Zhao (2004), the authors did not take into account the L_2-scaling constraints of intermediate variables, which seems necessary when the ρ-operator is used.

Example 1: This is a fourth-order low-pass Butterworth filter with narrow bandwidth, generated by the MATLAB command `butter(4, 0.05)`. The ρ-modal realisation yields an I/O sensitivity of 7.1048. The corresponding pole sensitivity and round-off noise gain are 0.2221 and 6.8033, respectively. The set of γ is computed as

$$\begin{aligned}\gamma &= (0.9375, 0.9375, 0.9375, 0.8125)\\ &= (15, 15, 15, 13) \times 2^{-4}.\end{aligned}$$

In order to provide an illustrative case for showing how Z_5 can be implemented in real integer processor, the pseudocode algorithm associated with the realisation Z_5 is given by Algorithm 1 (it is assumed that this realisation is performed on a 16-bit processor, and the additions are 32-bit).

Example 2: The second example is a sixth-order passband Butterworth filter which is obtained by the MATLAB command `butter(3, [0.75 0.90])`. The optimal I/O sensitivity measure obtained with the proposed realisation is 17.299. The corresponding pole sensitivity and round-off noise gain are 1.588 and 11.523, respectively. The set of γ is computed as

$$\begin{aligned}\gamma &= (-0.5625, -0.6875, -0.8125,\\ &\quad -0.9375, -0.5625, -0.875)\\ &= (-9, -11, -13, -15, -19, -14) \times 2^{-4}.\end{aligned}$$

Table 1. Performance comparison of five realisations of Example 1.

Realisation	$M_{L_2}^W(Z)$	$\mu(Z)$	$G(Z)$	$N.+$	$N.\times$
Z_1	469.34	1.2326	11.277	8	12
Z_2	7.1600	0.5848	5.0231	12	16
Z_3	16.590	3.5100	4.6816	11	16
Z_4	28.695	4.3014	12.454	20	25
Z_5	7.1048	0.2221	6.8033	16	25

Table 2. Performance comparison of five realisations of Example 2.

Realisation	$M_{L_2}^W(Z)$	$\mu(Z)$	$G(Z)$	$N.+$	$N.\times$
Z_1	48.899	2.6938	5.4092	12	18
Z_2	26.369	9.1963	16.405	18	25
Z_3	29.955	13.487	12.375	17	25
Z_4	26.815	6.4235	23.633	42	49
Z_5	17.299	1.5880	11.523	24	34

From Tables 1 and 2, it is observed that the cascade form with companion canonical cellules (Z_1) is the best in terms of computational efforts (just $2n+1$ non-trivial coefficients); it however behaves relatively poor in terms of the I/O sensitivity and pole sensitivity. Besides, to construct this realisation, we have to choose a suitable configuration of poles and zeros among all possibilities, and determine a best cascade order of subsections. When the order of the given filter/controller is high, this task may be exhausting. For instance, in general, a system possessing N second-order sections has $N!$ configurations of poles and zeros and $N!$ possible cascade orders.

The ρDFIIt (Z_2) and its equivalent state-space realisation (Z_3) possess $3n+1$ non-trivial coefficients. In terms of sensitivity and noise gain, they perform quite well. One disadvantage of these two realisations is the requirement of optimisation algorithm. Indeed, 'exhaustive research' and 'genetic algorithm' were used in Li and Zhao (2004) and Zhao and Li (2006), respectively, for obtaining the optimal realisations. In addition, no scaling is used for intermediate variables, so overflows on intermediate variables may happen.

The proposed ρ-modal realisation holds $4n+1$ non-trivial coefficients, and achieves a good trade-off among different criteria. It is, in the authors' own experience, always better in terms of pole sensitivity, and comparable in terms of parametric sensitivity. It has especially the great advantage of not requiring a cumbersome and uncertain optimisation (non-convex problem, very large decision space for high-order filters or controllers, convergence not guaranteed within a

reasonable time). Moreover, thanks to its parallel structure, measurement of parametric sensitivity of the ρ-modal form is nothing else than the sum of parametric sensitivities of individual cells. This property allows a more detailed analysis of resilience depending on the considered frequency range, and also the use of pre-programmed cells making *a posteriori* modification easier. Furthermore, this form is applicable to multivariable filters (e.g. Kalman filters) or controllers, in contrast with Z_1 or ρDFIIt realisation. Finally, it is worth noting that compared with the other realisations, the ρ-modal realisation can be executed faster due to its parallel form by using several processing units simultaneously. In other realisations, on the contrary, downstream sections have to wait for the accomplishment of upstream sections during computation.

8. Concluding remarks

This article deals with the FWL implementation problem of digital LTI filters/controllers. Its principal contribution consists in the proposition of a systematic way to get a realisation well managing the compromise between the different aspects. The proposed structure is deduced based on a modal realisation and the use of a ρ-operator adapted to each mode. This realisation holds a sparse parameterisation and results in a low computational effort (the number of coefficients is less than $4n+1$ for an nth-order filter/controller), small pole and I/O parametric sensitivity, as well as a small round-off noise gain. All these are obtained under the relaxed L_2-scaling constraints which allow to normalise the intermediate computational variables, and to limit the risk of overflow as well. Moreover, contrary to other reported results asking some tricky non-linear optimisation, the present method requires the minimisation of parameters whose optimum can be attained analytically.

The present results are concerned with the open-loop case, and the considered filters/controllers are assumed to have no multiple poles. These facts may limit the application of the proposed method. In order to reduce this restriction, the generalisation of the current results to the closed-loop control problem and the diagonalisation of a system with multiple poles by using the Jordan form are under research.

Notes

1. Exactly-implemented parameters mentioned here are those that are not modified by the process of quantisation (those completely represented by a number of bits lower that the word length used for implementation).
2. ρDFIIt is evaluated by the methods proposed in Li and Zhao (2004). It is under the strict L_2-scaling constraints, without L_2-scaling on intermediate variables.
3. From a practical viewpoint, these measures are programmed in the toolbox developed by the authors and freely available at http://fwrtoolbox.gforge.inria.fr.

References

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J.D., Green-baum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999), *LAPACK Users' Guide* (3rd ed.), Philadelphia: SIAM, http://www.netlib.org/lapack/lug

Belanovic, P., and Rupp, M. (2005), 'Automated Floating-point to Fixed-point Conversion with the Fixify Environment', in *the 16th IEEE International Workshop on Rapid System Prototyping*, June, Montreal, Canada, pp. 172–178.

Chevrel, P. (2002), 'Discrétisation et Codage des régulateurs en vue de Leur Maintenance et de Leur réglage', in *Réunion du GT Commande et Robuste des Systèmes Multivariables*, December, Bordeaux, France, http://personnel.supaero.fr/alazard-daniel/gtmosar/decembre2002/TR_Chevrel.pdf.

Fialho, I., and Georgiou, T. (1994), 'On Stability and Performance of Sampled-data Systems Subject to Wordlength Constraint', *IEEE Transactions on Automatic Control*, 39, 2476–2481.

Gevers, M., and Li, G. (1993), *Parameterizations in Control, Estimation and Filtering Problems*, London: Springer-Verlag.

Hao, J., and Li, G. (2007), 'An Efficient Controller Structure with Minimum Round-off Noise Gain', *Automatica*, 43, 921–927.

Hilaire, T. (2009), 'Low Parametric Sensitivity Realizations with Relaxed-L_2-dynamic-range-scaling Constraints', *IEEE Transactions on Circuits and Systems, II*, 56, 590–594.

Hilaire, T., Chevrel, P., and Whidborne, J. (2007a), 'A Unifying Framework for Finite Wordlength Realizations', *IEEE Transactions on Circuits and Systems, I*, 54, 1765–1774.

Hilaire, T., Chevrel, P., and Whidborne, J. (2010), 'Finite Wordlength Controller Realizations using the Specialized Implicit Form', *International Journal of Control*, 83, 330–346.

Hilaire, T., Ménard, D., and Sentieys, O. (2007b), 'Roundoff Noise Analysis of Finite Wordlength Realizations with the Implicit State-space Framework', in *Proceedings of the 15th European Signal Processing Conference*, September, Poznan, Poland, pp. 1019–1023.

Hwang, S. (1975), 'Dynamic-range Constraint in State-space Digital Filtering', *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23, 591–593.

Jackson, L. (1970), 'Roundoff Noise Analysis for Fixed-point Digital Filters Realized in Cascade or Parralel Form', *IEEE Transactions on Audio and Electroacoustics*, 18, 107–122.

Kim, S., Kum, K., and Sung, W. (1998), 'Fixed-point Optimization Utility for C and C++ Based Digital

Signal Processing Programs', *IEEE Transactions on Circuits and Systems, I*, 45, 1455–1464.
Li, G., Gevers, M., and Sun, Y.X. (2000), 'Performance Analysis of a New Structure for Digital Filter Implementation', *IEEE Transactions on Circuits and Systems, I*, 47, 474–482.
Li, G., and Zhao, Z. (2004), 'On the Generalized DFIIt Structure and its State-space Realization in Digital Filter Implementation', *IEEE Transactions on Circuits and Systems, I*, 51, 769–778.
Madievski, A., Anderson, B., and Gevers, M. (1995), 'Optimum Realizations of Sampled-data Controllers for FWL Sensitivity Minimization', *Automatica*, 31, 367–379.
Mantey, P. (1968), 'Eigenvalue Sensitivity and State-variable Selection', *IEEE Transactions on Automatic Control*, 13, 263–269.
Middleton, R., and Goodwin, G. (1990), *Digital Control and Estimation: A Unified Approach*, Englewood Cliffs, NJ: Prentice-Hall.
Mullis, C., and Roberts, R. (1976), 'Synthesis of Minimum Roundoff Noise Fixed-point Digital Filters', *IEEE Transactions on Circuits and Systems*, CAS-23, 551–562.
Oppenheim, A., Schafer, R., and Buck, J. (1999), *Discrete-time Signal Processing*, Upper Saddle River, NJ: Prentice-Hall.
Rao, D.V.B. (1986), 'Analysis of Coefficient Quantization Errors in State-space Digital Filters', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34, 131–139.
Yan, W., and Teo, K. (2002), 'Optimal Finite-precision Approximation of FIR Filers', *Signal Processing*, 82, 1695–1705.
Zhao, Z., and Li, G. (2006), 'Roundoff Noise Analysis of Two Efficient Digital Filter Structures', *IEEE Transactions on Signal Processing*, 54, 790–795.

Appendix A

In this part, the proof of Proposition 6 is given. To this end, we recall the expression of $\widetilde{W}_{cT}$

$$\widetilde{W}_{cT} = (\Lambda - \gamma) W_{cX} (\Lambda - \gamma)^{\top} + BB^{\top}.$$

Define $F = \Lambda - \gamma$, then

$$F = \begin{pmatrix} \alpha_1 - \gamma_1 & \beta_1 & & & \\ \beta_2 & \alpha_2 - \gamma_2 & & & \\ & & \ddots & & \\ & & & \alpha_{n-1} - \gamma_{n-1} & \beta_{n-1} \\ & & & \beta_n & \alpha_n - \gamma_n \end{pmatrix}.$$

The diagonal of $\widetilde{W}_{cT}$ can be described by $(\widetilde{W}_{cT})_{i,i} = (F_{i,*}) W_{cX} (F_{i,*})^{\top} + b_i^2$, where b_i is the ith element of the matrix B. If i is odd, then $(F_{i,*}) = (0 \cdots 0\ f_{i,i}, f_{i,i+1}\ 0 \cdots 0)$. Hence,

$$\begin{aligned}(\widetilde{W}_{cT})_{i,i} = {} & (W_{cX})_{i,i} (\alpha_i - \gamma_i)^2 + \beta_i^2 (W_{cX})_{i+1,i+1} \\ & + 2\beta_i (W_{cX})_{i+1,i} (\alpha_i - \gamma_i) + b_i^2,\end{aligned}$$

which can be written as a quadratic function of γ_i

$$\begin{aligned}(\widetilde{W}_{cT})_{i,i} = {} & (W_{cX})_{i,i} \left[\gamma_i - \alpha_i - \frac{\beta_i (W_{cX})_{i+1,i}}{(W_{cX})_{i,i}} \right]^2 + b_i^2 \\ & + \frac{\beta_i^2 \left[(W_{cX})_{i,i} (W_{cX})_{i+1,i+1} - (W_{cX})_{i+1,i}^2 \right]}{(W_{cX})_{i,i}}.\end{aligned}$$

As W_{cX} is both symmetrical and positive definite, the third term of the above equation is positive. Hence, the minimal value of diagonal of $\widetilde{W}_{cT}$ is

$$\widetilde{W}_{cT}\Big|_{\min} = \frac{\beta_i^2 \left[(W_{cX})_{i,i} (W_{cX})_{i+1,i+1} - (W_{cX})_{i+1,i}^2 \right]}{(W_{cX})_{i,i}} + b_i^2,$$

and the corresponding γ_i is

$$\gamma_i = \alpha_i + \frac{\beta_i (W_{cX})_{i+1,i}}{(W_{cX})_{i,i}}.$$

Similarly, by following the same mechanism, the corresponding optimal γ_i for i even is

$$\gamma_i = \alpha_i + \frac{\beta_i (W_{cX})_{i,i-1}}{(W_{cX})_{i,i}}.$$

Appendix B

The ρ-modal (Z_5) of Example 1 is (the non-exactly implemented values are shown in bold and rounded to 5 digits)

$$Z_5 = \left(\begin{array}{cccc|cccc|c} -1 & 0 & 0 & 0 & \mathbf{0.039247} & \mathbf{0.36998} & 0 & 0 & \mathbf{-1.5036} \\ 0 & -1 & 0 & 0 & \mathbf{-0.80414} & \mathbf{-0.085302} & 0 & 0 & \mathbf{-0.90686} \\ 0 & 0 & -1 & 0 & 0 & 0 & \mathbf{-0.28837} & \mathbf{0.31715} & \mathbf{-1.4118} \\ 0 & 0 & 0 & -1 & 0 & 0 & \mathbf{-0.06901} & \mathbf{0.062747} & \mathbf{-1.1104} \\ \hline 0.25 & 0 & 0 & 0 & 0.9375 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0.9375 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 & 0.9375 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.8125 & 0 \\ \hline 0 & 0 & 0 & 0 & \mathbf{0.36004} & \mathbf{0.033563} & \mathbf{-0.36108} & \mathbf{-0.028345} & \mathbf{3.1239e-05} \end{array}\right).$$

The ρ-modal realisation of Example 2 is

$$Z_5 = \left(\begin{array}{cccccc|cccccc|c}
-1 & 0 & 0 & 0 & 0 & 0 & \mathbf{-0.095031} & \mathbf{0.57471} & 0 & 0 & 0 & 0 & \mathbf{-0.60202} \\
0 & -1 & 0 & 0 & 0 & 0 & \mathbf{-0.57471} & \mathbf{0.065489} & 0 & 0 & 0 & 0 & \mathbf{-0.79499} \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & \mathbf{-0.13769} & \mathbf{0.83082} & 0 & 0 & \mathbf{1.3422} \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & \mathbf{-0.83078} & \mathbf{0.24842} & 0 & 0 & \mathbf{0.76651} \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & \mathbf{-0.11569} & \mathbf{0.94998} & \mathbf{1.1595} \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & \mathbf{-0.94992} & \mathbf{1.5488} & \mathbf{0.63535} \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & -0.5625 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -0.6875 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & -0.8125 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & -0.9375 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -0.5625 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.125 & 0 & 0 & 0 & 0 & 0 & -0.875 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{-0.29212} & \mathbf{0.12808} & \mathbf{0.21633} & \mathbf{0.08549} & \mathbf{-0.28447} & \mathbf{0.7058} & \mathbf{0.0085987}
\end{array}\right).$$

Algorithm 1 (16-bit pseudocode algorithm of Z_5 (Example 1)):
Input: u: 16-*bits integer*
Output: y: 16-*bits integer*
Data: xn: *array* [1..5] *of* 16-*bits integers*
Data: T: *array* [1..5] *of* 16-*bits integers*
Data: Acc: 32-*bits integer*
begin

// Intermediate variables
$Acc \leftarrow (xn(1) * 2572) + (xn(2) * 24247) + (u * -24634);$
$T_1 \leftarrow Acc >> 16;$
$Acc \leftarrow (xn(1) * -26350) + (xn(2) * -2795) + (u * -7429);$
$T_2 \leftarrow Acc >> 15;$
$Acc \leftarrow (xn(3) * -18899) + (xn(4) * 10392) + (u * -23131);$
$T_3 \leftarrow Acc >> 15;$
$Acc \leftarrow (xn(3) * -4523) + (xn(4) * 2056) + (u * -18193);$
$T_4 \leftarrow Acc >> 14;$
// States
$Acc \leftarrow T_1 << 13 + (xn(1) * 30720);$
$xn(1) \leftarrow Acc >> 15;$
$Acc \leftarrow T_2 << 13 + (xn(2) * 30720);$
$xn(2) \leftarrow Acc >> 15;$
$Acc \leftarrow T_3 << 12 + (xn(3) * 30720);$
$xn(3) \leftarrow Acc >> 15;$
$Acc \leftarrow T_4 << 13 + (xn(4) * 26624);$
$xn(4) \leftarrow Acc >> 15;$
// Outputs
$Acc \leftarrow (xn(1) * 23596) + (xn(2) * 2200) + (xn(3) * -23664);$
$Acc \leftarrow Acc + (xn(4) * -929) + u;$
$y \leftarrow Acc >> 14;$

end