Finite Wordlength Controller Realizations using the Specialized Implicit Form

Thibault Hilaire, Philippe Chevrel, James F. Whidborne

Finite wordlength controller realisations using the specialised implicit form

Thibault Hilaire[a]*, Philippe Chevrel[b] and James F. Whidborne[c]

[a]*LIP6, Université Pierre et Marie Curie (Paris 6), CNRS, Paris, France;* [b]*IRCCyN, UMR CNRS 6597, 1 rue de la Noë, 44321 Nantes, France;* [c]*Department of Aerospace Science, Cranfield University, Bedfordshire, MK43 0AL, UK*

A specialised implicit state-space representation is introduced to deal with finite wordlength effects in controller implementations. This specialised implicit form provides a macroscopic description of the algorithm to be implemented. So, it constitutes a unifying framework, allowing to encompass various implementation forms, such as the δ-operator, the ρDirect Form II transposed, observer-based and many other realisations usually considered separately in the literature. Different measures quantifying the finite wordlength effects on the overall closed-loop behaviour are defined in this new context. They concern both stability and performance. The gap with the infinite precision case is evaluated classically through the coefficient sensitivity and roundoff noise analysis. The problem of determining a realisation with minimum finite wordlength effects can subsequently be solved using appropriate numerical methods. The approach is illustrated with an example.

Keywords: digital control; finite wordlength effects; digital controller implementation; optimal realisation

1. Introduction

When implemented in digital computing devices, controllers are subjected to numerical degradations due to the rounding and quantisation that occurs on the variables and constants used to define the controller. There are two main effects of this finite-precision (often known as the *Finite Word Length* (*FWL*) *effects*):

- *Roundoff noise* is the addition of noise into the system resulting from the rounding of variables before and after each arithmetic operation;
- *Parameter errors* are the quantisation of the controller coefficients/parameters. They degrade the performance and/or stability of the controller.

For most low-order controllers, the FWL effects are minor, but for higher order controllers, particularly when fast sampling is used, the FWL effects can become significant. For example, the stability of the system can be compromised even by a small quantisation of the coefficients (Whidborne, Wu, and Istepanian 2000).

However, it is well known that the FWL effects are dependent upon the controller realisation. Hence many papers deal with the problem of finding a realisation that minimises the FWL effects in some sense (see, for example, Gevers and Li (1993), Istepanian and Whidborne (2001), Whidborne, Wu, and Istepanian (2001) and references therein). It is also well known that the FWL effects are dependent on the operator used. The δ-operator, for example, generally has much better numerical properties than the usual delay operator, q^{-1}, for control systems with fast sampling (Goodall 2001).

The problem of addressing the optimal realisation for minimal FWL effects is usually addressed in the state space (e.g. Thiele 1984; Gevers and Li 1993; Whidborne et al. 2001). Briefly, if the controller is

$$K(\sigma) = C(\sigma I - A)^{-1}B + D, \tag{1}$$

where σ is the transform of the chosen operator (e.g. δ or q-operator), the problem is to search over the set

$$\left\{CT(\sigma I - T^{-1}AT)^{-1}TB + D : T \text{ a non-singular matrix}\right\}$$

to find a matrix T and corresponding controller realisation with small FWL effects. The limitations of this approach are that

- there are many realisations that cannot be expressed in such a standard state space form;
- the search is restricted to a single operator.

The δ-operator is more complex to implement than the q-operator, so in some circumstances, it may be better to have a mix of operators. These limitations may be

*Corresponding author. Email: thibault.hilaire@lip6.fr

overcome by using the specialised implicit form (SIF) (Hilaire, Chevrel, and Trinquet 2005b) for the controller. The SIF allows a formal and faithful macroscopic description of the numerical algorithm used to implement the controller.

In order to determine the optimal realisation, some measures of the roundoff noise and the closed-loop coefficient sensitivity are required. A fair number of these have been proposed over the years. The roundoff noise is generally measured by the output noise variance (e.g. Mullis and Roberts 1976; Hwang 1977; Gevers and Li 1993). Measures of the input–output performance (IO-performance) deterioration have been proposed by Gevers and Li (1993). Stability can be assessed using a probabilistic measure (Fialho and Georgiou 1994), a measure based on a small-gain theorem (Whidborne et al. 2000), μ-analysis (Wu, Li, Chen, and Chu 2008) or closed-loop pole sensitivity measures (Li 1998; Whidborne et al. 2001; Wu, Chen, Li, Istepanian, and Chu 2001; Ko and Yu 2003). Ideally, the chosen measures should be computationally tractable but reasonably representative of the actual perturbations that occur in implementation.

The SIF was originally proposed in Hilaire et al. (2005b). In Hilaire, Chevrel, and Whidbarne (2007b) the FWL filter problem (the open-loop case) is considered. In this article, some of the results of Hilaire et al. (2007b) and Hilaire, Ménard, and Sentieys (2007c) are extended to the FWL controller problem, that is the closed-loop case. A closed-loop IO sensitivity measure which extends that of Gevers and Li (1993) and a pole sensitivity stability related measure (PSSM) are proposed along with a closed-loop roundoff noise gain (RNG) measure. All are suitable for use with the SIF and are similar to those proposed for the FWL filter realisation problem (Hilaire et al. 2007b). Note that some preliminary results on FWL controller with the SIF appeared in Hilaire, Chevrel, and Trinquet (2005a).

This article is organised as follows. In the next section, the SIF is recalled, and a number of definitions are given. The recently proposed ρDFIIt realisation (Li and Zhao 2004) is shown to be a particular case of the SIF. In Section 2.2, the concept of equivalent classes (potentially structured) of realisations is introduced and illustrated with an example. Section 3 details, in a closed-loop context, the two sensitivity measures and the roundoff noise measure. In Section 4, an optimal design problem is introduced and it is illustrated with an example in Section 5.

2. The SIF

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.

The model takes the form of an implicit state-space realisation (Aplevich 1991) specialised according to

$$\begin{pmatrix} J & 0 & 0 \\ -K & I_n & 0 \\ -L & 0 & I_p \end{pmatrix} \begin{pmatrix} T(k+1) \\ X(k+1) \\ Y(k) \end{pmatrix} = \begin{pmatrix} 0 & M & N \\ 0 & P & Q \\ 0 & R & S \end{pmatrix} \begin{pmatrix} T(k) \\ X(k) \\ U(k) \end{pmatrix}, \tag{2}$$

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

It is implicitly assumed throughout this article that the computations associated with the realisation (2) are executed in row order, giving the following algorithm:

$$\begin{aligned} &\text{[i]} \quad J.T(k+1) \leftarrow M.X(k) + N.U(k) \\ &\text{[ii]} \quad X(k+1) \leftarrow K.T(k+1) + P.X(k) + Q.U(k) \\ &\text{[iii]} \quad Y(k) \leftarrow L.T(k+1) + R.X(k) + S.U(k). \end{aligned} \tag{3}$$

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

Equation (2) is equivalent in infinite precision to the classical state-space form

$$\left(\begin{array}{c} T(k+1) \\ \hline X(k+1) \\ Y(k) \end{array}\right) = \left(\begin{array}{cc|c} 0 & J^{-1}M & J^{-1}N \\ \hline 0 & A_Z & B_Z \\ 0 & C_Z & D_Z \end{array}\right) \left(\begin{array}{c} T(k) \\ \hline X(k) \\ U(k) \end{array}\right) \tag{4}$$

with $A_Z \in \mathbb{R}^{n\times n}$, $B_Z \in \mathbb{R}^{n\times m}$, $C_Z \in \mathbb{R}^{p\times n}$ and $D_Z \in \mathbb{R}^{p\times m}$ where

$$A_Z = KJ^{-1}M + P, \quad B_Z = KJ^{-1}N + Q, \tag{5}$$

$$C_Z = LJ^{-1}M + R, \quad D_Z = LJ^{-1}N + S. \tag{6}$$

Note that (4) corresponds to a different parametrisation than (2) (the finite-precision implementation of (4) will cause different numerical deterioration to

that of (2)). The associated system transfer function is given by

$$H : z \mapsto C_Z(zI_n - A_Z)^{-1}B_Z + D_Z. \tag{7}$$

A complete framework for the description of all digital controller implementations can be developed by using the following definitions. For further details, see Hilaire et al. (2007b).

Definition 2.1: A realisation $\mathcal{R}$ of a transfer matrix H is entirely defined by the data Z, l, m, n and p. $Z \in \mathbb{R}^{(l+n+p)\times(l+n+m)}$ is partitioned according to

$$Z \triangleq \begin{pmatrix} -J & M & N \\ K & P & Q \\ L & R & S \end{pmatrix} \tag{8}$$

and l, m, n and p are the matrix dimensions given previously. The notation used will be $\mathcal{R} := (Z, l, m, n, p)$.

The notation Z is introduced to make the further developments more compact ((41), (57), etc.).

Definition 2.2: $\mathcal{R}_H$ denotes the set of realisations described by (2) equivalent to the transfer function H, that is to say with the same IO relationship. These realisations are said to be IO-equivalent and IO-equivalent to the transfer function H.

In order to encompass realisations with some special structure (q or δ state-space, direct forms, cascades, lattice, etc.), a subset of realisations sharing the same structure is defined.

Definition 2.3: A structuration $\mathcal{S}$ is a set of structured realisations. That is realisations that share a common structure with some coefficients and/or some dimensions having been fixed a priori.

Some examples of structurations are given in the next subsection.

Definition 2.4: $\mathcal{R}_H^{\mathcal{S}}$ is the set of equivalent structured realisations. Realisations from $\mathcal{R}_H^{\mathcal{S}}$ are structured according to $\mathcal{S}$ and are IO-equivalent to H:

$$\mathcal{R}_H^{\mathcal{S}} \triangleq \mathcal{R}_H \cap \mathcal{S}. \tag{9}$$

2.1 *Some examples*

2.1.1 *δ-realisations*

Consider the δ-state-space form

$$\begin{cases} \delta[X(k)] = A_\delta X(k) + B_\delta U(k) \\ Y(k) = C_\delta X(k) + D_\delta U(k) \end{cases} \tag{10}$$

with $\delta = \frac{q-1}{\Delta}$, $\Delta \in \mathbb{R}_{+*}$ and q is the shift operator (Gevers and Li 1993).

This realisation should be implemented with the following algorithm:

$$\begin{array}{ll} \text{[i]} & T \leftarrow A_\delta.X(k) + B_\delta.U(k) \\ \text{[ii]} & X(k+1) \leftarrow X(k) + \Delta.T \\ \text{[iii]} & Y(k) \quad C_\delta.X(k) + D_\delta.U(k), \end{array} \tag{11}$$

where T is an intermediate variable. This could be modelled with the SIF as

$$\begin{pmatrix} I_n & 0 & 0 \\ -\Delta I_n & I_n & 0 \\ 0 & 0 & I_p \end{pmatrix} \begin{pmatrix} T(k+1) \\ X(k+1) \\ Y(k) \end{pmatrix} = \begin{pmatrix} 0 & A_\delta & B_\delta \\ 0 & I_n & 0 \\ 0 & C_\delta & D_\delta \end{pmatrix} \begin{pmatrix} T(k) \\ X(k) \\ U(k) \end{pmatrix} \tag{12}$$

2.1.2 *Cascade decomposition*

The cascade form is a common realisation for filter/controller implementations. It generally has good FWL properties compared to the direct forms and requires less operations than fully parametrised state-space realisations. The system is decomposed into a number of lower order (usually first and second order) subsystems connected in series.

Let us consider two realisations $\mathcal{R}_1$ and $\mathcal{R}_2$ connected in series as shown in Figure 1.

Assuming $\mathcal{R}_1$ and $\mathcal{R}_2$ to be defined by SIF matrices (J_1, K_1, L_1, M_1, N_1, P_1, Q_1, R_1, S_1) and (J_2, K_2, L_2, M_2, N_2, P_2, Q_2, R_2, S_2), and cascading them leads to the realisation $\mathcal{R} := (Z, m_1, p_1 + l_1 + l_2, n_1 + n_2, p_2)$ with

$$Z = \left(\begin{array}{ccc:ccc} -J_1 & 0 & 0 & M_1 & 0 & N_1 \\ L_1 & -I & 0 & R_1 & 0 & S_1 \\ 0 & N_2 & -J_2 & 0 & M_2 & 0 \\ \hdashline K_1 & 0 & 0 & P_1 & 0 & Q_1 \\ 0 & Q_2 & K_2 & 0 & P_2 & 0 \\ \hdashline 0 & S_2 & L_2 & 0 & R_2 & 0 \end{array}\right) \tag{13}$$

from which definition of the corresponding structuration $\mathcal{S}$ immediately follows. The outputs of $\mathcal{R}_1$ are computed in the intermediate variable and then used as the inputs of $\mathcal{R}_2$.

The main point is that this construction can represent cascade systems without changing the parametrisation.

Remark 1: The cascade structuration can be applied to realisations that are structured differently (q and δ-state-space realisations, for example) and easily extended to multiple cascaded systems.

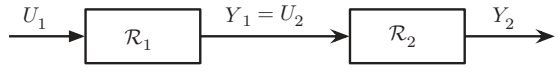

Figure 1. Cascade form.

2.1.3 *ρ transposed direct-form II*

Li and Hao (Li 2004; Li and Zhao 2004; Hao and Li 2005) have presented a new sparse structure called ρDFIIt. This is a generalisation of the transposed direct-form II structure with the conventional shift and the δ-operator and is similar to that of Palaniswami and Feng (1991). It is a sparse realisation (with $3n+1$ parameters when n is the order of the controller), leading to an economic (few computations) implementation that could be very numerically efficient. As we will see later, this realisation has n extra degrees of freedom that can be used to find an *optimal* realisation within its particular structuration.

Let us define

$$\rho_i : z \mapsto \frac{z-\gamma_i}{\Delta_i}, \quad 1 \leqslant i \leqslant n \tag{14}$$

and

$$\varrho_i : z \mapsto \prod_{j=1}^{i} \rho_j(z), \quad 1 \leqslant i \leqslant n, \tag{15}$$

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

$$H : z \mapsto \frac{b_0 + b_1 z^{-1} + \cdots + b_{n-1} z^{-n+1} + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_{n-1} z^{-n+1} + a_n z^{-n}}. \tag{16}$$

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

$$H(z) = \frac{\beta_0 + \beta_1 \varrho_1^{-1}(z) + \cdots + \beta_{n-1}\varrho_{n-1}^{-1}(z) + \beta_n \varrho_n^{-1}(z)}{1 + \alpha_1 \varrho_1^{-1}(z) + \cdots + \alpha_{n-1}\varrho_{n-1}^{-1}(z) + \alpha_n \varrho_n^{-1}(z)}. \tag{17}$$

Denoting

$$V_a \triangleq \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}, \quad V_b \triangleq \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}, \quad V_\alpha \triangleq \begin{pmatrix} 1 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}, \quad V_\beta \triangleq \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \tag{18}$$

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 (Hao and Li 2005) according to

$$\begin{cases} V_a = \kappa \Omega V_\alpha \\ V_b = \kappa \Omega V_\beta, \end{cases} \tag{19}$$

where $\kappa \triangleq \prod_{i=1}^{n} \Delta_i$ and $\Omega \in \mathbb{R}^{n+1\times n+1}$ is a lower triangular matrix whose i-th 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 $\Omega_{n+1,n+1} = 1$.

Equation (17) can be, for example, implemented with a transposed direct form II (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 $\left(\rho_i^{-1}\right)_{1\leqslant i\leqslant k}$). 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 extra degrees of freedom given by the choice of the parameters $(\gamma_i)_{1\leqslant i\leqslant n}$.

The corresponding algorithm is:

$$\begin{array}{lll} \text{[i]} & Y(k) & \beta_0 U(k) + W_1(k) \\ \text{[ii]} & W_i(k) & \rho_i^{-1}\left[\beta_i U(k) - \alpha_i Y(k) + W_{i+1}(k)\right] \\ \text{[iii]} & W_n(k) & \rho_n^{-1}\left[\beta_n U(k) - \alpha_n Y(k)\right]. \end{array} \tag{20}$$

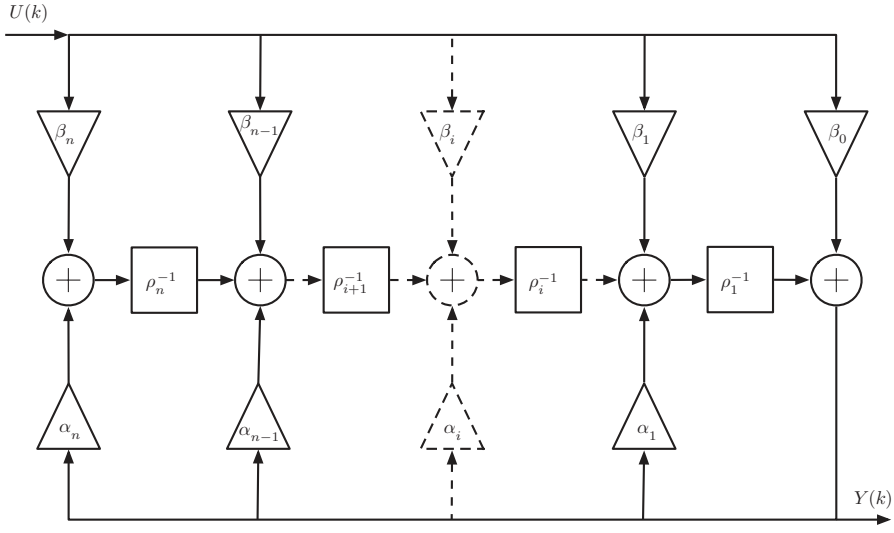

Figure 2. Generalised ρ direct form II.

By introducing the intermediate variables needed to realise the ρ_i^{-1} operator (according to $\rho_i^{-1} = \frac{1}{q^{-1}-\gamma_i}\Delta_i$, with the multiplication by Δ_i done last, see Figure 3), Equations (21)–(23) become

$$T = \begin{pmatrix} \Delta_1 & & & \\ & \Delta_2 & & \\ & & \ddots & \\ & & & \Delta_n \end{pmatrix} X(k) + \begin{pmatrix} \beta_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} U(k) \quad (21)$$

$$X(k+1) = \begin{pmatrix} -\alpha_1 & 1 & & \\ -\alpha_2 & 0 & \ddots & \\ \vdots & & \ddots & 1 \\ -\alpha_n & & & 0 \end{pmatrix} T + \begin{pmatrix} \gamma_1 & & & \\ & \gamma_2 & & \\ & & \ddots & \\ & & & \gamma_n \end{pmatrix} X(n) + \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix} U(k) \quad (22)$$

$$Y(k) = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} T \quad (23)$$

Within the SIF framework, the ρDFIIt form is described by

$$Z = \left(\begin{array}{cccc:ccc:c} -1 & & & & \Delta_1 & & & \beta_0 \\ & \ddots & & & & \Delta_2 & & 0 \\ & & \ddots & & & & \ddots & \vdots \\ & & & -1 & & & \Delta_n & 0 \\ \hdashline \alpha_1 & 1 & & & \gamma_1 & & & \beta_1 \\ -\alpha_2 & 0 & \ddots & & & \gamma_2 & & \beta_2 \\ \vdots & & \ddots & 1 & & & \ddots & \vdots \\ -\alpha_n & & & 0 & & & \gamma_n & \beta_n \\ \hdashline 1 & 0 & & 0 & 0 & & 0 & 0 \end{array}\right). \quad (24)$$

Remark 2: Thanks to the SIF, there is no need to use another operator unlike the shift operator.

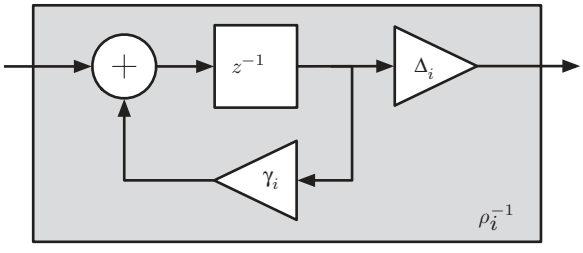

Figure 3. Realisation of operator ρ_i^{-1}.

A number of other examples of structurations are given in Hilaire (2006). They illustrate the generality of the SIF framework.

2.2 *Equivalent classes*

In order to exploit the potential offered by the SIF in improving implementations, it is necessary to characterise further the sets of equivalent system realisations. We first note that the non-minimal realisations may provide better implementations (the δ-form can be seen as a non-minimal realisation when written in the implicit state-space form – with the shift operator). Hence the notion of equivalence needs to be extended by considering that the system state dimension does not have to be invariant. The *inclusion principle*, introduced by Šiljak and Ikeda (Ikeda, Šiljak, and White 1984; Šiljak 1991) in the context of decentralised control, is useful here as it allows the formalisation of the *equivalence* and *inclusion* relations between two system realisations.

These two notions have been extended to the SIF in Hilaire et al. (2007b) in order to give a formal description of equivalent classes. Although it may be of practical interest to only consider realisations of the same dimensions, where transformations from one realisation to another is only a similarity transformation.

This could be achieved with the following proposition.

Proposition 2.5: *Consider a realisation* $\mathcal{R} := (Z, l, m, n, p)$. *All the realisations* $\tilde{\mathcal{R}} := (\tilde{Z}, l, m, n, p)$ *with*

$$\tilde{Z} = \begin{pmatrix} \mathcal{Y} & & \\ & \mathcal{U}^{-1} & \\ & & I_p \end{pmatrix} Z \begin{pmatrix} \mathcal{W} & & \\ & \mathcal{U} & \\ & & I_m \end{pmatrix} \quad (25)$$

and $\mathcal{U}$, $\mathcal{W}$, $\mathcal{Y}$ *are non-singular matrices, are equivalent to* $\mathcal{R}$, *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, say cascade or delta. For example, if an initial δ-structured realisation $\mathcal{R} := (Z_0, n, m, n, p)$ is given, the subset of equivalent δ-structured realisation is defined by

$$\mathcal{R}_H^{S_\delta} = \left\{ \begin{array}{l} \mathcal{R} := (Z, n, m, n, p) \backslash \\ Z = \begin{pmatrix} \mathcal{U}^{-1} & & \\ & \mathcal{U}^{-1} & \\ & & I_p \end{pmatrix} Z_0 \begin{pmatrix} \mathcal{U} & & \\ & \mathcal{U} & \\ & & I_m \end{pmatrix} \\ \forall \mathcal{U} \in \mathbb{R}^{n \times n} \text{ non-singular} \end{array} \right\}. \quad (26)$$

This compact algebraic characterisation of equivalent classes is particularly efficient when used to search for an optimal structured realisation (Section 4).

3. Closed-loop measures

The quantisation of the coefficients and the roundoff noise may have a negative impact on the closed-loop system behaviour. Three measures that may be used to evaluate this impact are described in this section.

3.1 *Problem statement*

Consider the plant $\mathcal{P}$ together with controller $\mathcal{C}$ according to the standard form shown in Figure 4, where $W(k) \in \mathbb{R}^{p_1}$ is the exogenous input, $Y(k) \in \mathbb{R}^{p_2}$ the control input, $Z(k) \in \mathbb{R}^{m_1}$ the controlled output and $U(k) \in \mathbb{R}^{m_2}$ the measured output.

The controller is defined as $\mathcal{C} := (Z, l, m_2, n, p_2)$ and the plant $\mathcal{P}$ as

$$\mathcal{P} := \left(\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right), \tag{27}$$

where $A \in \mathbb{R}^{n_\mathcal{P} \times n_\mathcal{P}}$, $B_1 \in \mathbb{R}^{n_\mathcal{P} \times p_1}$, $B_2 \in \mathbb{R}^{n_\mathcal{P} \times p_2}$, $C_1 \in \mathbb{R}^{m_1 \times n_\mathcal{P}}$, $C_2 \in \mathbb{R}^{m_2 \times n_\mathcal{P}}$, $D_{11} \in \mathbb{R}^{m_1 \times p_1}$, $D_{12} \in \mathbb{R}^{m_1 \times p_2}$, $D_{21} \in \mathbb{R}^{m_2 \times p_1}$ and $D_{22} \in \mathbb{R}^{m_2 \times p_2}$ is assumed to be zero only to simplify the mathematical expressions.

Note that open-loop results (filter modelling) may be obtained as a particular case, with:

$$\mathcal{P} := \left(\begin{array}{c|cc} & & \\ \hline & 0 & I \\ & I & 0 \end{array}\right). \tag{28}$$

The closed-loop system $\bar{\mathcal{S}}$ is then given by

$$\bar{\mathcal{S}} = F_l(\mathcal{P}, \mathcal{C}) := \left(\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array}\right), \tag{29}$$

where $F_l(\cdot,\cdot)$ is the well-known lower linear fractional transform (Zhou, Doyle, and Gloyer 1996) and

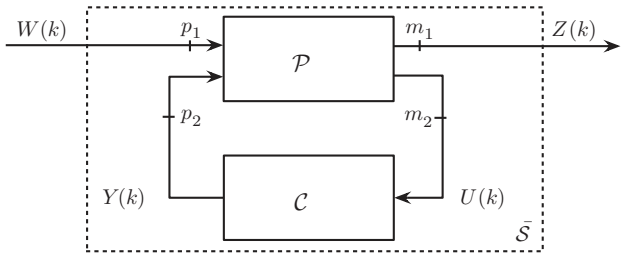

Figure 4. Closed-loop control system.

where $\bar{A} \in \mathbb{R}^{n_\mathcal{P}+n \times n_\mathcal{P}+n}$, $\bar{B} \in \mathbb{R}^{n_\mathcal{P}+n \times p_1}$, $\bar{C} \in \mathbb{R}^{m_1 \times n_\mathcal{P}+n}$ and $\bar{D} \in \mathbb{R}^{m_1 \times p_1}$ are such that

$$\bar{A} = \begin{pmatrix} A + B_2 D_Z C_2 & B_2 C_Z \\ B_Z C_2 & A_Z \end{pmatrix}, \quad \bar{B} = \begin{pmatrix} B_1 + B_2 D_Z D_{21} \\ B_Z D_{21} \end{pmatrix}, \tag{30}$$

$$\bar{C} = \begin{pmatrix} C_1 + D_{12} D_Z C_2 & D_{12} C_Z \end{pmatrix}, \quad \bar{D} = D_{11} + D_{12} D_Z D_{21}. \tag{31}$$

The closed-loop transfer function is

$$\bar{H} : z \mapsto \bar{C}\left(zI - \bar{A}\right)^{-1} \bar{B} + \bar{D}. \tag{32}$$

3.2 *Input–output sensitivity*

In order to evaluate how much the quantisation of the controller's coefficients (due to FWL implementation) affects the closed-loop transfer function, the sensitivity $\frac{\partial \bar{H}}{\partial Z}$ can be used. Before that, the nature of the perturbation on each coefficient must made precise.

A coefficient's quantisation depends both on its value and its representation. First, if the value of a coefficient is such that it will be quantised without error (like 0, ± 1 or a power of 2), then, that parameter makes no contribution to the overall coefficient sensitivity and is called a *trivial* parameter. Hence we introduce the weighting matrices W_Z associated with Z such that

$$(W_Z)_{i,j} \triangleq \begin{cases} 0 & \text{if } X_{i,j} \text{ is exactly implemented,} \\ 1 & \text{otherwise.} \end{cases} \tag{33}$$

For a fixed-point representation, Z is perturbed to $Z^\dagger = Z + W_Z \times \Delta$, where Δ represents the quantisation error.

Remark 1: For floating-point representations, Z is perturbed to $Z^\dagger = Z + W_Z \times Z \times \Delta$ (Wu, Chen, Whidborne, and Chu 2003; Hilaire, Chevrel, and Whidborne 2007a). The following measures can then be easily extended to the floating-point (and block-floating-point) case.

The closed-loop transfer function resulting from the quantisation process is denoted by $\bar{H}^\dagger \triangleq \bar{H}\big|_{Z+W_Z \times \Delta}$. For the single input single output (SISO) case, the following is true $\forall z \in \mathbb{C}$

$$\bar{H}^\dagger(z) - \bar{H}(z) = \sum_{i,j} \Delta_{i,j} \frac{\partial \bar{H}^\dagger(z)}{\partial \Delta}\bigg|_{\Delta=0} + o\left(\|\Delta\|_{\max}^2\right) \tag{34}$$

and

$$\left\|\bar{H}^\dagger - \bar{H}\right\|_2 \leq \|\Delta\|_{\max} \left\|\frac{\partial \bar{H}^\dagger}{\partial \Delta}\bigg|_{\Delta=0}\right\|_2 + \left|o\left(\|\Delta\|_{\max}^2\right)\right|, \tag{35}$$

where $\|\cdot\|_2$ denotes the H_2-norm. The wordlength can be chosen so that $\|\Delta\|_{\max}$ is sufficiently small, but if the $\|\frac{\partial \bar{H}^\dagger}{\partial \Delta}|_{\Delta=0}\|_2$ term is made small by an appropriate choice of realisation, then it is clear that a lower wordlength can be used. The actual performance degradation can be checked a posteriori.

It is easy to show that

$$\left.\frac{\partial \bar{H}^\dagger}{\partial \Delta}\right|_{\Delta=0} = \frac{\partial \bar{H}}{\partial Z} \times W_Z. \tag{36}$$

From (35) and (36), we define an IO-sensitivity measure as follows:

Definition 3.1: Consider a realisation $\mathcal{C} := (Z, l, m_2, n, p_2)$. For the SISO case, the closed-loop transfer function sensitivity, with respect to all the non-trivial coefficients of $\mathcal{C}$, is defined by

$$\bar{M}^W_{L_2} \triangleq \left\| \frac{\partial \bar{H}}{\partial Z} \times W_Z \right\|_2^2. \tag{37}$$

Remark 2: It is possible to include a frequency weighting to emphasise certain frequency range (Gevers and Li 1993) to ensure that the closed-loop degradation is constrained over a given frequency range.

This measure can be extended to the multiple input multiple output (MIMO) case. It is also useful to consider the contribution of each coefficient to the overall sensitivity. The *closed-loop transfer function sensitivity matrix*, denoted by $\frac{\delta \bar{H}}{\delta Z}$, is the matrix of the H_2-norm of the IO-sensitivity of the transfer function $\bar{H}$ with respect to each coefficient $Z_{i,j}$. It is defined by

$$\left(\frac{\delta \bar{H}}{\delta Z}\right)_{i,j} \triangleq \left\| \frac{\partial \bar{H}}{\partial Z_{i,j}} \right\|_2. \tag{38}$$

It can be used to obtain a *map* of the sensitivity with respect to each coefficient and help to choose a specific fixed-point format for each coefficient. From the properties of H_2-norms, we get

$$\left\| \frac{\delta \bar{H}}{\delta Z} \right\|_F = \left\| \frac{\partial \bar{H}}{\partial Z} \right\|_2, \tag{39}$$

where $\|\cdot\|_F$ is the Frobenius norm. Definition 3.1 can now be stated for the general case.

Definition 3.2: The closed-loop IO-sensitivity measure is defined by

$$\bar{M}^W_{L_2} \triangleq \left\| \frac{\delta \bar{H}}{\delta Z} \times W_Z \right\|_F^2. \tag{40}$$

The IO-sensitivity $\frac{\partial \bar{H}}{\partial Z}$ can be evaluated by the following proposition.

Proposition 3.3:

$$\frac{\partial \bar{H}}{\partial Z} = \bar{H}_1 \circledast \bar{H}_2, \tag{41}$$

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

$$A \circledast B \triangleq \mathrm{Vec}(A) \cdot \left[\mathrm{Vec}\left(B^\top\right)\right]^\top, \tag{42}$$

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

$$\bar{H}_1 : z \mapsto \bar{C}\left(zI - \bar{A}\right)^{-1} \bar{M}_1 + \bar{M}_2 \tag{43}$$

$$\bar{H}_2 : z \mapsto \bar{N}_1 \left(zI - \bar{A}\right)^{-1} \bar{B} + \bar{N}_2 \tag{44}$$

and

$$\bar{M}_1 = \begin{pmatrix} B_2 L J^{-1} & 0 & B_2 \\ K J^{-1} & I_n & 0 \end{pmatrix}, \quad \bar{N}_1 = \begin{pmatrix} J^{-1} N C_2 & J^{-1} M \\ 0 & I_n \\ C_2 & 0 \end{pmatrix}, \tag{45}$$

$$\bar{M}_2 = \begin{pmatrix} D_{12} L J^{-1} & 0 & D_{12} \end{pmatrix}, \quad \bar{N}_2 = \begin{pmatrix} J^{-1} N D_{21} \\ 0 \\ D_{21} \end{pmatrix}. \tag{46}$$

The dimensions of $\bar{M}_1$, $\bar{M}_2$, $\bar{N}_1$ *and* $\bar{N}_2$ *are, respectively,* $(n + n_{\mathcal{P}}) \times (l + n + p_2)$, $m_1 \times (1 + n + p_2)$, $(l + n + m_2) \times (n + n_{\mathcal{P}})$ *and* $(l + n + m_2) \times p_1$.

Proof: The proof is based on the following lemma and can be found in Hilaire and Chevrel (2008) and Hilaire (2006).

Lemma 3.4: *Let* X *be a matrix in* $\mathbb{R}^{p \times l}$ *while* G *and* H *are two transfer matrices independent of* X *with values in* $\mathbb{C}^{m \times p}$ *and* $\mathbb{C}^{l \times n}$, *respectively, and that are independent of* X. *Then*

$$\frac{\partial (GXH)}{\partial X} = G \circledast H, \tag{47}$$

$$\frac{\partial (GX^{-1}H)}{\partial X} = (GX^{-1}) \circledast (X^{-1}H). \tag{48}$$

From (30), (5) and (6), it is possible to write

$$\bar{A} = \begin{pmatrix} A + B_2 L J^{-1} N C_2 & B_2 C_Z \\ B_Z C_2 & A_Z \end{pmatrix} + \begin{pmatrix} B_2 \\ 0 \end{pmatrix} S \begin{pmatrix} C_2 & 0 \end{pmatrix} \tag{49}$$

and finally with Lemma 3.4

$$\frac{\partial \bar{H}}{\partial S} = \begin{pmatrix} B_2 \\ 0 \end{pmatrix} \circledast \begin{pmatrix} C_2 & 0 \end{pmatrix}. \tag{50}$$

The other derivatives $\frac{\partial \bar{H}}{\partial R}, \frac{\partial \bar{H}}{\partial Q}, \ldots$ can be similarly obtained and then gathered using

$$\frac{\partial}{\partial Z} = \begin{pmatrix} -\frac{\partial}{\partial J} & \frac{\partial}{\partial M} & \frac{\partial}{\partial N} \\ \frac{\partial}{\partial K} & \frac{\partial}{\partial P} & \frac{\partial}{\partial Q} \\ \frac{\partial}{\partial L} & \frac{\partial}{\partial R} & \frac{\partial}{\partial S} \end{pmatrix}. \tag{51}$$

□

Proposition 3.5: *The closed-loop transfer function sensitivity matrix* $\frac{\delta \bar{H}}{\delta Z}$ *can be computed as*

$$\left(\frac{\delta \bar{H}}{\delta Z}\right)_{i,j} = \left\| \bar{H}_1 E_{i,j} \bar{H}_2 \right\|_2 \tag{52}$$

with

$$\bar{H}_1 E_{i,j} \bar{H}_2 := \left(\begin{array}{cc|c} \bar{A} & 0 & \bar{B} \\ \bar{M}_1 E_{i,j} \bar{N}_1 & \bar{A} & \bar{M}_1 E_{i,j} \bar{N}_2 \\ \hline \bar{M}_2 E_{i,j} \bar{N}_1 & \bar{C} & \bar{M}_2 E_{i,j} \bar{N}_2 \end{array}\right) \tag{53}$$

and $E_{i,j}$ *is the matrix of appropriate size with all elements being* 0 *except the* (i, j)*-th element which is unity.*

Proof: The proof is quite straightforward, and comes from the definition of operator $\circledast$ in Proposition 3.3. □

Remark 3: In the SISO case, the problem becomes simpler by noting that

$$\left(\frac{\delta \bar{H}}{\delta Z}\right)_{i,j} = \left\| (\bar{H}_2 \bar{H}_1)_{i,j} \right\|_2 \tag{54}$$

$$= \left\| \left(\begin{array}{cc|c} \bar{A} & 0 & \bar{B} \\ \bar{M}_1 \bar{N}_1 & \bar{A} & \bar{M}_1 \bar{N}_2 \\ \hline \bar{M}_2 \bar{N}_1 & \bar{C} & \bar{M}_2 \bar{N}_2 \end{array}\right)_{i,j} \right\|_2. \tag{55}$$

The $(l+n+1) \times (l+n+1)$ H_2-norm evaluations here require only $l+n+1$ Lyapunov equations to be solved (instead of the $(l+n+p) \times (l+n+m_2)$ equations in the MIMO case represented by (53)), so this expression is preferred.

3.3 *Pole sensitivity measures*

The IO-sensitivity does not explicitly consider the stability of the closed-loop system. To ensure that the implementation is stable, the sensitivity of the poles may be considered. We define the following pole sensitivity measure.

Definition 3.6: Consider a controller realisation $\mathcal{C} := (Z, l, m_2, n, p_2)$. The closed-loop pole sensitivity measure is defined by

$$\bar{\Psi} \triangleq \sum_{k=1}^{n_P+n} \left\| \frac{\partial |\bar{\lambda}_k|}{\partial Z} \times W_Z \right\|_F^2, \tag{56}$$

where $(\bar{\lambda}_k)_{1 \leqslant k \leqslant n_P+n}$ denotes the poles of the closed-loop system (the eigenvalues of $\bar{A}$).

The following lemma will be required next to evaluate $\bar{\Psi}$.

Lemma 3.7: *Consider a differentiable function* $f: \mathbb{R}^{m \times n} \to \mathbb{C}$*, and two matrices* $Y \in \mathbb{R}^{m \times n}$ *and* $X \in \mathbb{R}^{p \times q}$*. Let* Y_0*,* Y_1 *and* Y_2 *be constant matrices with appropriate dimensions. Then the following results hold:*

- *if* $Y = Y_0 + Y_1 X Y_2$*, then*

$$\frac{\partial f(Y)}{\partial X} = Y_1^\top \frac{\partial f(Y)}{\partial Y} Y_2^\top,$$

- *if* $Y = Y_0 + Y_1 X^{-1} Y_2$*, then*

$$\frac{\partial f(Y)}{\partial X} = -\left(Y_1 X^{-1}\right)^\top \frac{\partial f(Y)}{\partial Y} \left(X^{-1} Y_2\right)^\top.$$

Proof: See Li (1998). □

The measure $\bar{\Psi}$ can be evaluated, thanks to the following proposition and lemma.

Proposition 3.8:

$$\frac{\partial |\bar{\lambda}_k|}{\partial Z} = \bar{M}_1^\top \frac{\partial |\bar{\lambda}_k|}{\partial \bar{A}} \bar{N}_1^\top, \tag{57}$$

where $\bar{M}_1$ *and* $\bar{N}_1$ *are defined in Equations* (45) *and* (46)*.*

Proof: The proof is similar to the one used in Proposition 3.3, by applying Lemma 3.7, instead of Lemma 3.4. □

Lemma 3.9: *Let* $M \in \mathbb{R}^{n \times n}$ *be diagonalisable. Let* $(\lambda_k)_{1 \leqslant k \leqslant n}$ *be its eigenvalues and* $(x_k)_{1 \leqslant k \leqslant n}$ *the corresponding right eigenvectors. Denote* $M_x \triangleq (x_1, x_2, \ldots, x_n)$ *and* $M_y = (y_1, y_2, \ldots, y_n) \triangleq M_x^{-H}$*. Then*

$$\frac{\partial \lambda_k}{\partial M} = y_k^* x_k^\top \quad \forall k = 1, \ldots, n \tag{58}$$

and

$$\frac{\partial |\lambda_k|}{\partial M} = \frac{1}{|\lambda_k|} \mathrm{Re}\left(\lambda_k^* \frac{\partial \lambda_k}{\partial M}\right), \tag{59}$$

where $\cdot^*$ *denotes the conjugate operation,* $\mathrm{Re}(\cdot)$ *the real part and* $\cdot^H$ *the transpose conjugate operator.*

Proof: See Wu et al. (2001). □

Remark 4: Similarly to the IO-sensitivity matrix, (38), a *pole sensitivity matrix* can be constructed to evaluate the overall impact of each coefficient. Let $\frac{\delta |\bar{\lambda}|}{\delta Z}$ denote the pole sensitivity matrix defined by

$$\left(\frac{\delta |\bar{\lambda}|}{\delta Z}\right)_{i,j} \triangleq \sqrt{\sum_{k=1}^{n_P+n} \left(\frac{\partial |\bar{\lambda}_k|}{\partial Z_{i,j}}\right)^2}. \tag{60}$$

It can be computed from

$$\frac{\partial|\bar{\lambda}_k|}{\partial Z_{i,j}} = \left(\frac{\partial|\bar{\lambda}_k|}{\partial Z}\right)_{i,j}. \tag{61}$$

During the quantisation process, Z is perturbed to $Z^\dagger$ and the closed-loop eigenvalues $(\bar{\lambda}_k)_{1\leqslant k\leqslant n_\mathcal{P}+n}$ may be outside the open unit disc. Therefore, it is crucial to know when the FWL error will cause closed-loop instability. On the basis of this consideration, a stability related measure (Fialho and Georgiou 1994) is defined as

$$\mu_0(Z) \triangleq \inf_{\Delta}\big\{\|\Delta\|_{\max}/\text{ realisation } Z^\dagger \text{ makes the closed-loop system unstable}\big\}. \tag{62}$$

This measure is not directly tractable (Fialho and Georgiou 1994; Wu and Chen 2004), but can be approximated with the following measure.

Definition 3.10: Consider a realisation $\mathcal{C} := (Z, l, m_2, n, p_2)$. The PSSM of $\mathcal{C}$ is defined by

$$\mu_1(Z) \triangleq \min_{1\leqslant k\leqslant n_\mathcal{P}+n} \frac{1-|\bar{\lambda}_k|}{\|W_Z\|_F \left\|\frac{\partial|\bar{\lambda}_k|}{\partial Z}\times W_Z\right\|_F}. \tag{63}$$

This measure evaluates how a perturbation, Δ, of the parameters, Z, can cause instability. It is determined by how close the eigenvalues of $\bar{A}$ are to the unit circle and by how sensitive they are to the controller parameter perturbation.

This measure is an extension to the SIF framework of the sensitivity stability related measure originally defined in the classical state-space framework (Li 1998) and can be directly linked to an estimation of the smallest wordlength required for the controller realisation to be implemented while preserving the closed-loop stability (Wu et al. 2003).

3.4 *Closed-loop roundoff noise analysis*

Complementary to the other two measures, a measure of the roundoff noise in the generalised context of the SIF is presented next. It extends the measure proposed in Hilaire et al. (2007c) to the closed-loop case.

3.4.1 *Preliminaries*

The first (μ)- and second (σ, ψ)-order centred-moments of a noise vector $\xi(k)$ are denoted and defined by

$$\mu_\xi \triangleq E\big\{\xi(k)\big\}, \tag{64}$$

$$\psi_\xi \triangleq E\Big\{\big(\xi(k)-\mu_\xi\big)\big(\xi(k)-\mu_\xi\big)^\top\Big\}, \tag{65}$$

$$\sigma_\xi^2 \triangleq E\Big\{\big(\xi(k)-\mu_\xi\big)^\top\big(\xi(k)-\mu_\xi\big)\Big\} = \mathrm{tr}\big(\psi_\xi\big), \tag{66}$$

where $E\{\cdot\}$ and $\mathrm{tr}(\cdot)$ are, respectively, the *mean* and the *trace* operator.

The following lemma recalls the basic properties of noise transmission through a linear system:

Lemma 3.11: *Assume the input noise, $U(k)$, to be such that*

$$E\big\{(U(k)-\mu_U)(U(k-l)-\mu_U)^\top\big\} = \delta_{0,l}\psi_U \tag{67}$$

where $\delta_{i,j}$ represents the Kronecker delta. Denote by Y the resulting output of the transfer matrix G. If (A, B, C, D) is a state-space realisation of G, the first- and second-order moments of Y are given by

$$\mu_Y = G(1)\mu_U \tag{68}$$

$$\sigma_Y^2 = \mathrm{tr}\big(\psi_U(D^\top D + B^\top W_o B)\big), \tag{69}$$

where $G(1)$ is the steady-state gain of G, given by $G(1) = C(I-A)^{-1}B + D$ and W_o is the observability Gramian of G. W_o is the unique solution of the discrete Lyapunov equation

$$W_o = A^\top W_o A + C^\top C. \tag{70}$$

Proof: It is well known that $\sigma_Y^2 = \|G\varphi_U\|_2^2$, with φ_U the square root of ψ_U (Papoulis 1991). The classical formula linking the H_2-norm to the Gramians is then applied. □

3.4.2 *Roundoff noise analysis*

Consider the realisation $\mathcal{R} := (Z, l, m_2, n, p_2)$. By taking into account the quantisation noise after each multiplication, the algorithm given by (3) becomes

$$\begin{aligned}
&\text{[i]} \quad J.T^*(k+1) \quad M.X^*(k) + N.U(k) + \xi_T(k)\\
&\text{[ii]} \quad X^*(k+1) \quad K.T^*(k+1) + P.X^*(k)\\
&\qquad + Q.U(k) + \xi_X(k)\\
&\text{[iii]} \quad Y^*(k) \quad L.T^*(k+1) + R.X^*(k) + S.U(k) + \xi_Y(k),
\end{aligned} \tag{71}$$

where ξ_T, ξ_X and ξ_Y are, respectively, the noise sources corrupting T, X and Y (ξ_T is added on $JT(k+1)$, so $J^{-1}\xi_T$ is added on $T(k+1)$).

Noise sources ξ_T, ξ_X and ξ_Y depend on

- the way the computations are performed, the order of the arithmetic operations, etc.
- the fixed-point representation of the inputs,
- the fixed-point representation of the outputs,

- the fixed-point representation chosen for the states and the intermediate variables,
- the fixed-point representation chosen for the coefficients.

They are modelled as independent white noise, characterised by their first- and second-order moments.

Remark 5: The quantisation or roundoff process can be considered as the addition of a noise, ξ. If ε represents the quantisation step, then (Widrow 1960) $\mu_\xi = 0$ and $\sigma_\xi = \varepsilon^2/12$ for roundoff, and $\mu_\xi = \varepsilon/2$ and $\sigma_\xi = \varepsilon^2/12$ for truncation.

The noise is added through the controller and the plant to the output $Z(k)$ of the closed-loop system $\bar{\mathcal{S}}$. Denote the noise added to $Z(k)$ by $\xi'(k)$:

$$\xi'(k) \triangleq Z^*(k) - Z(k). \tag{72}$$

Definition 3.12: The output noise power $\bar{P}$ is defined as the power of $\xi'(k)$

$$\bar{P} \triangleq E\left\{\xi'^{\top}(k)\xi'(k)\right\} \tag{73}$$

Denote by ξ the vector stacking all the noise sources

$$\xi(k) \triangleq \begin{pmatrix} \xi_T(k) \\ \xi_X(k) \\ \xi_Y(k) \end{pmatrix}. \tag{74}$$

Proposition 3.13: *The noise $\xi'(k)$ corresponds to the noise $\xi(k)$ filtered through the transfer function $\bar{H}_1$ defined in Equation* (43) (*the closed-loop system is then equivalent to the system described in* Figure 5). *Hence, we get*

$$\bar{P} = \mathrm{tr}\left(\psi_\xi\left(\bar{M}_2^{\top}\bar{M}_2 + \bar{M}_1^{\top}\bar{W}_o\bar{M}_1\right)\right) + \mu_{\xi'}^{\top}\mu_{\xi'}, \tag{75}$$

where $\mu_{\xi'} = \left(C_Z(I - A_Z)^{-1}\bar{M}_1 + \bar{M}_2\right)\mu_\xi$.

Proof: If $X_{\mathcal{P}}$ denotes the state of the plant, Equation (71) combined with the state-space realisation of the plant leads to

$$\begin{cases} \begin{pmatrix} X_{\mathcal{P}} \\ X \end{pmatrix}(k+1) = \bar{A}\begin{pmatrix} X_{\mathcal{P}} \\ X \end{pmatrix}(k) + \bar{B}W(k) + \bar{M}_1\xi(k) \\ Z(k) = \bar{C}\begin{pmatrix} X_{\mathcal{P}} \\ X \end{pmatrix}(k) + \bar{D}W(k) + \bar{M}_2\xi(k). \end{cases} \tag{76}$$

So, $\bar{H}_1$ (cf. Equation (43)) appears explicitly as the transfer function relating $\xi(k)$ to $Z(k)$ as stated in the proposition. Therefore, $P = E\{\xi'^{\top}(k)\xi'(k)\} = \sigma_{\xi'}^2 + \mu_{\xi'}^{\top}\mu_{\xi'}$ and Lemma 3.11 gives the first- and second-order moments. □

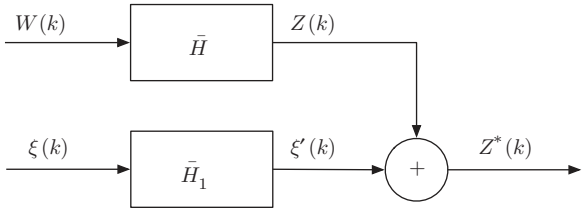

Figure 5. Equivalent system with noise extracted.

Remark 6: Equation (75) is a good illustration of the relationship between the work done in the *hardware/software* (HW/SW) community and that done in the *control* community. The former is based on the accurate evaluation of the noise for particular HW/SW fixed-point implementations on various targets (DSP, FPGA) whereas the latter is based on the search for *good* realisations with particular well-conditioned structures. In the first case, only the classical direct form is studied, whereas the actual HW/SW impact is neglected in the second case.

The moments ψ_ξ and μ_ξ depend only on the HW/SW implementation, whereas the other terms ($\bar{A}$, $\bar{C}$, $\bar{M}_1$, $\bar{M}_2$ and $\bar{W}_o$) depend only on the algorithm used.

3.4.3 *Roundoff noise gain*

The *closed-loop RNG* is the output noise power in a specific (and simplified) computational scheme: the noise is assumed to appear only after each multiplication (roundoff after multiplication scheme). It is modelled as a zero-mean centred, statistically independent, white noise. Each noise source has the same power σ_0^2 (determined by the wordlength chosen for all the variables and coefficients).

Definition 3.14: The closed-loop RNG is defined as

$$\bar{G} \triangleq \frac{\bar{P}}{\sigma_0^2}. \tag{77}$$

This measure has been studied for the open-loop case by Mullis and Roberts (1976), Hwang (1977) and Gevers and Li (1993) and has been established for classical state-space realisations and some other particular realisations. The particular computational scheme considered gives the moments of ξ_T, ξ_X and ξ_Y: here they depend only on the number of non-trivial parameters in the realisation.

Let us introduce the matrices d_J to d_S. They are diagonal matrices defined by

$$(d_X)_{i,i} \triangleq \left\{ \begin{array}{c} \text{number of non-trivial parameters} \\ \text{in the } i\text{-th row of X} \end{array} \right\}. \quad (78)$$

The trivial parameters considered are 0, 1 and -1 because they do not imply a multiplication.

Step [i] of algorithm (3) is realised as follows (for $i \in \{1, 2, \ldots, l\}$):

$$T_i(k+1) \quad \sum_{j=1}^{n} M_{ij} X_j(k) + \sum_{j=1}^{m} N_{ij} U_j(k) - \sum_{j<i} J_{ij} T_j(k+1). \quad (79)$$

Each multiplication by a non-trivial parameter implies a quantisation noise. Since they are independent centred white noise, ψ_{ξ_T} is given by

$$_{\xi_T} = (d_M + d_N + d_J)\sigma_0^2 \quad (80)$$

(J is a lower diagonal matrix with 1 on the diagonal. So the number of non-trivial parameters on the i-th row is equal to the number of non-trivial parameters of the i-th row restricted to its subdiagonal part).

In the same way (steps [ii] and [iii]),

$$_{\xi_Y} = (d_L + d_R + d_S)\sigma_0^2 \quad (81)$$

$$\psi_{\xi_X} = \big(d_K + d_P + d_Q\big)\sigma_0^2. \quad (82)$$

Proposition 3.15: *The RNG is given by*

$$\bar{G} = \mathrm{tr}\big(d_Z\big(\bar{M}_2^\top \bar{M}_2 + \bar{M}_1^\top \bar{W}_o \bar{M}_1\big)\big) \quad (83)$$

where

$$d_Z = \begin{pmatrix} d_J + d_M + d_N & & \\ & d_K + d_P + d_Q & \\ & & d_L + d_R + d_S \end{pmatrix} \quad (84)$$

(d_Z *is also defined by Equation* (78) *applied on* Z).

Proof: The noise sources ξ_T, ξ_X and ξ_Y are zero mean centred independent noises so μ_ξ is null and

$$\psi_\xi = \begin{pmatrix} _{\xi_T} & & \\ & _{\xi_X} & \\ & & _{\xi_Y} \end{pmatrix} \quad (85)$$

□

4. Optimal design

For the implementation of a digital controller, it is important to choose a realisation having low FWL effects. Hence it is of interest to find an optimal realisation in a sense to be defined.

Problem 4.1: The *global optimal realisation problem* is to find the best realisation $\mathcal{R}_{\text{opt}}$ associated with the transfer function H according to the criteria $\mathcal{J}$

$$\mathcal{R}_{\text{opt}} = \arg \min_{\mathcal{R} \in \mathcal{R}_H} \mathcal{J}(\mathcal{R}). \quad (86)$$

Due to the size of $\mathcal{R}_H$, this problem generally cannot be solved practically. Hence the following problem is introduced to restrict the search to some particular structurations.

Problem 4.2: Consider some structurations $(\mathcal{S}_i)_{1 \leqslant i \leqslant N}$. The *optimal structured realisation problem* is to find the optimal realisation $\mathcal{R}^{\mathcal{S}}_{\text{opt}}$:

$$\mathcal{R}^{\mathcal{S}}_{\text{opt}} = \arg \min_{\substack{\mathcal{R} \in \mathcal{R}_H^{\mathcal{S}_i} \\ 1 \leqslant i \leqslant N}} \mathcal{J}(\mathcal{R}). \quad (87)$$

Since the measure $\mathcal{J}$ could be non-smooth and/or non-convex, the adaptive simulated annealing (ASA) (Ingber 1996; Chen and Luk 1999) method has been chosen to solve Problem 4.2. This method has worked well for other optimal realisation problems (Wu et al. 2001).

If the equivalent structured realisations are linked through the similarity transformation of Proposition 2.5, the computation of the previously defined FWL measures can be improved thanks to the following proposition:

Proposition 4.3: *If we consider two realisations* Z_0 *and* Z_1 *such that*

$$Z_1 = \mathcal{T}_1 Z_0 \mathcal{T}_2, \quad (88)$$

where

$$\mathcal{T}_1 = \begin{pmatrix} \mathcal{Y} & & \\ & \mathcal{U}^{-1} & \\ & & I_p \end{pmatrix}, \quad \mathcal{T}_2 = \begin{pmatrix} \mathcal{W} & & \\ & \mathcal{U} & \\ & & I_m \end{pmatrix} \quad (89)$$

then the closed-loop measures of realisation Z_1 *can be computed from those of* Z_0 *according to*

$$\left(\frac{\delta \bar{H}}{\delta Z}\right)_{i,j}\Bigg|_{Z_1} = \left\| \bar{H}_1\big|_{Z_0} \mathcal{T}_1^{-1} E_{i,j} \mathcal{T}_2^{-1} \bar{H}|_2\big|_{Z_0} \right\|_2, \quad (90)$$

$$\frac{\partial |\bar{\lambda}_k|}{\partial Z}\Bigg|_{Z_1} = \mathcal{T}_1^{-\top} \frac{\partial |\bar{\lambda}_k|}{\partial Z}\Bigg|_{Z_0} \mathcal{T}_2^{-\top}. \quad (91)$$

Proof: The proof comes directly from

$$\bar{H}_1|_{Z_1} = \bar{H}_1|_{Z_0}\mathcal{T}_1^{-1}, \quad \bar{H}_2|_{Z_1} = \mathcal{T}_2^{-1}\bar{H}_2|_{Z_0}. \tag{92}$$

□

A Matlab toolbox (*FWR Toolbox*, available from `http://fwrtoolbox.gforge.inria.fr/`) has been specially developed to use the SIF and solve optimal structured realisation problems with the previously defined measures.

5. Example

The example is taken from Gevers and Li (1993, pp. 236–237). The discrete time system to be controlled is given by

$$A_p = \begin{pmatrix} 3.7156 & -5.4143 & 3.6525 & -0.9642 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{93}$$

$$B_p = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^\top, \tag{94}$$

$$C_p = \begin{pmatrix} 0.1116 & 0.0043 & 0.1088 & 0.0014 \end{pmatrix} \times 10^{-5}. \tag{95}$$

Remark 1: All the computations are performed with Matlab double floating-point precision, but the results are quoted only to 4 significants digits (which may be insufficient to characterise the considered system). For each different realisation, bold font is used to exhibit non-trivial parameters (the weighting matrice W_Z is built accordingly).

The plant corresponds to the following standard form (see (27))

$$\mathcal{P} := \left(\begin{array}{c|cc} A & B_p & B_p \\ \hline C_p & 0 & 0 \\ C_p & 0 & 0 \end{array}\right). \tag{96}$$

The initial realisation of the feedback controller is designed to place the closed-loop poles at

$$\lambda_{1,2} = 0.9844 \pm 0.0357j, \quad \lambda_{3,4} = 0.9643 \pm 0.0145j, \tag{97}$$

$$\lambda_{5,6} = 0.7152 \pm 0.6348j, \quad \lambda_{7,8} = 0.3522 \pm 0.2857j. \tag{98}$$

The controller has the following transfer function

$$H : z \mapsto \frac{38252z^3 - 101878z^2 + 91135z - 27230}{z^4 - 2.3166z^3 + 2.1662z^2 - 0.96455z + 0.17565}. \tag{99}$$

Let us consider different realisations for this controller. The realisations, Z_1–Z_{11}, are described below. The values of the measures are shown in Table 1. The realisations and corresponding sensitivity matrices, $\frac{\delta\bar{H}}{\delta Z}$ and $\frac{\delta|\bar{\lambda}|}{\delta Z}$, are given in the Appendix. Note that only the bold values shown in the realisations are considered, via the weighting matrix W_Z.

State-space realisations:

Z_1: Canonical form (corresponds to direct form II). This realisation has the following results:

$$\begin{aligned} \bar{M}_{L_2}^W &= 1.9046e+7, \quad \bar{\Psi} = 3.3562e+7, \\ \mu_1 &= 1.8065e-6, \quad \bar{G} = 1.186e+6 \end{aligned} \tag{100}$$

Z_2: The *internally balanced* state-space realisation is often considered as a low sensitivity realisation (Gevers and Li (1993) shows that the balanced realisations

Table 1. Example 1: FWL measures for different realisations.

	$\bar{M}_{L_2}^W$	$\bar{\Psi}$	μ_1	$\bar{G}$	$T\bar{O}$	Nb. op.
Z_1	1.9046*e*+7	3.3562*e*+7	1.8065*e*−6	1.186*e*+6	3.6764*e*+8	7 + 8×
Z_2	3.6427*e*+5	6.5007*e*+5	7.4933*e*−6	3.6582*e*+2	1.1387*e*+5	19 + 24×
Z_3	1.5267*e*+3	1.6689*e*+4	1.167*e*−4	1.7455*e*+2	5.4111*e*+4	19 + 24×
Z_4	1.6272*e*+3	2.7425*e*+3	1.189*e*−4	1.1778*e*+2	3.6512*e*+4	19 + 24×
Z_5	1.9474*e*+13	1.2294*e*+13	1.7244*e*−9	3.2261*e*−3	1.7239*e*+10	19 + 24×
Z_6	2.8696*e*+3	4.5371*e*+3	9.2351*e*−5	7.9809*e*−3	6.0078*e*+0	19 + 24×
Z_7	1.5342*e*−2	8.1051*e*−2	6.6047*e*−2	2.8082*e*−8	4.5466*e*+0	11 + 12×
Z_8	1.5341*e*−2	8.089*e*−2	6.6045*e*−2	4.217*e*−8	4.8783*e*+0	11 + 16×
Z_9	1.1388*e*−1	2.8203*e*−2	6.6159*e*−2	3.7783*e*−6	9.8937*e*+1	11 + 16×
Z_{10}	1.5342*e*−2	8.0015*e*−2	6.6052*e*−2	4.1742*e*−8	4.8371*e*+0	11 + 16×
Z_{11}	1.6065*e*−2	3.8802*e*−2	6.0413*e*−2	4.7451*e*−8	3.5597*e*+0	11 + 16×

minimises the L_1/L_2 sensitivity measure). It has the following measure values:

$$\begin{aligned} \bar{M}_{L_2}^W &= 3.6427e+5, \quad \bar{\Psi} = 6.5007e+5, \\ \mu_1 &= 7.4933e-6, \quad \bar{G} = 365.82. \end{aligned} \tag{101}$$

Despite it being fully parametrised (24 parameters), its overall sensitivity is lower than the canonical form.

Z_3: With the similarity

$$\mathcal{T}_1 = \begin{pmatrix} \cdot & & \\ & \mathcal{U}^{-1} & \\ & & 1 \end{pmatrix}, \quad \mathcal{T}_2 = \begin{pmatrix} \cdot & & \\ & \mathcal{U} & \\ & & 1 \end{pmatrix} \tag{102}$$

it is possible to consider all state-space equivalent realisations, and find the $\bar{M}_{L_2}^W$-optimal state-space realisation Z_3. Its closed-loop transfer function sensitivity measure is $\bar{M}_{L_2}^W = 1526.7$ and is much lower than other state-space realisations.

Z_4: It is also possible to consider the $\bar{\Psi}$-optimal state-space realisation. Then $\bar{\Psi} = 2742.5$.

Z_5: $\bar{G}$-optimal state-space Z_5. Here, $\bar{G}$ is very low: $\bar{G} = 0.0032261$, but the other measures are quite poor:

$$\begin{aligned} \bar{M}_{L_2}^W &= 1.9474e+13, \quad \bar{\Psi} = 1.2294e+13, \\ \mu_1 &= 1.7244e-9. \end{aligned} \tag{103}$$

Even if the goal of this article is not multi-objective optimal realisation, it is interesting to look for a realisation that is *good enough* for the three measures $\bar{M}_{L_2}^W$, $\bar{\Psi}$ and $\bar{G}$. Let us denote

$$\bar{TO}(Z) \triangleq \frac{\bar{M}_{L_2}^W(Z)}{\bar{M}_{L_2}^{W\,\mathrm{opt}}} + \frac{\bar{\Psi}(Z)}{\bar{\Psi}^{\mathrm{opt}}} + \frac{\bar{G}(Z)}{\bar{G}^{\mathrm{opt}}}, \tag{104}$$

where $\bar{M}_{L_2}^{W\,\mathrm{opt}}$ is the optimal transfer function sensitivity value ($\bar{M}_{L_2}^{W\,\mathrm{opt}} = \bar{M}_{L_2}^W(Z_3)$), $\bar{\Psi}^{\mathrm{opt}}$ the optimal value for the pole sensitivity ($\bar{\Psi}^{\mathrm{opt}} = \bar{\Psi}(Z_4)$) and $\bar{G}^{\mathrm{opt}}$ the optimal RNG value ($\bar{G}^{\mathrm{opt}} = \bar{G}(Z_5)$).

Remark 2: This *tradeoff* measure is defined for this example and this structuration (state-space). Clearly, it is lower bounded by 3.

Z_6: *Tradeoff*-optimal state-space Z_6. With this measure, we aim to have a realisation that simultaneously has low transfer function sensitivity, pole sensitivity and RNG. The *tradeoff* measure is quite low ($\bar{TO} = 6.0078$), and the corresponding measures are:

$$\begin{aligned} \bar{M}_{L_2}^W &= 2869.6, \quad \bar{\Psi} = 4537.1, \\ \mu_1 &= 9.2351e-5, \quad \bar{G} = 0.0079809. \end{aligned} \tag{105}$$

ρ direct forms II transposed: The realisation (24) is considered with various values for $(\gamma_i)_{1 \leqslant i \leqslant n}$. Δ is chosen to be 2^{-3}. Since there is no possibility here to use similarity on Z like that proposed in Proposition 2.5, the realisation matrix Z cannot be built from another Z matrix: for $(\gamma_i)_{1 \leqslant i \leqslant n}$ given, the parameters $(\alpha_i)_{1 \leqslant i \leqslant n}$ and $(\beta_i)_{0 \leqslant i \leqslant n}$ have to be rebuilt from (19).

Z_7: with $\gamma = (1\ 1\ 1\ 1)^\top$, the direct form II with the δ-operator is obtained.

Z_8: $M_{L_2^W}$-optimal ρDFIIt. The optimisation gives

$$\gamma = \begin{pmatrix} 0.29758 & 0.99939 & 0.99953 & 0.99977 \end{pmatrix}^\top. \tag{106}$$

Z_9: $\bar{\Psi}$-optimal ρDFIIt. The optimisation gives

$$\gamma = \begin{pmatrix} 0.35114 & 0.30858 & 0.66309 & 0.99856 \end{pmatrix}^\top. \tag{107}$$

Z_{10}: $\bar{G}$-optimal ρDFIIt. The optimisation gives

$$\gamma = \begin{pmatrix} 0.93207 & 0.99335 & 0.99863 & 0.99963 \end{pmatrix}^\top. \tag{108}$$

Z_{11}: It is here also possible to apply a new *tradeoff* measure, like the one in equation (104) (with new $\bar{M}_{L_2}^{W\,\mathrm{opt}}$, $\bar{\Psi}^{\mathrm{opt}}$ and $\bar{G}^{\mathrm{opt}}$ values). The $\bar{TO}$-optimal realisation (Equation (A5)) is obtained with

$$\gamma = \begin{pmatrix} 0.99744 & 0.41349 & 0.8646 & 0.99346 \end{pmatrix}^\top. \tag{109}$$

and $\bar{TO} = 3.5597$.

Table 1 gives all the measure values for the realisation Z_1–Z_{11}. Realisations Z_6 and Z_{11} are interesting, low sensitivity, low roundoff noise realisations. Moreover Z_{11} requires fewer operations (11 additions and 16 multiplications) than Z_6. These results are case dependent and some controllers may be less sensitive in state-space forms than in ρDFIIt form.

The pseudocode algorithms associated with realisations Z_6 and Z_{11} are given by Algorithms 1 and 3 listed in the Appendix. It is assumed that these realisations are performed on a fixed-point 16-bit processor (the additions are 32 bits, without guard bits for the additions) and the input is in the interval $[-10, 10]$ (so 11 bits are given for the fractional part). Due to the gain of the controller, the output has -5 bits for the fractional part (the integer value coding for the output must be multiplied by 2^6 to obtain the real value). The binary point position is adjusted for each intermediate variable, state and coefficient. So the fixed-point algorithms of realisations Z_6 and Z_{11} are given by Algorithms 2 and 4.

6. Conclusions

The SIF is a powerful tool for controller implementation modelling. It provides a macroscopic description

of the algorithm to be implemented, in the context of embedded systems. Being more general than previous forms, it allows, in a unified framework, the analysis and design of particular realisations of linear controllers. Different measures can give insight on the quality of a given realisation: IO-sensitivity, pole sensitivity, RNG, amount of computation, etc. All have been defined in the new context of the SIF. Some of them are worked out in a efficient way through the use of Gramians and Lyapunov equations.

The SIF allows all possible linear realisations, not necessarily of the same order, to be compared. Some optimisations are computationally tractable, by restricting the class of equivalent realisations to specific subclasses or structures. This has been tested in the case of classical state-space realisations, with δ-structures, etc. The sparse realisation proposed recently in Li and Zhao (2004) has also been examined.

There are numerous areas for future work. First, it would be of practical interest to make use of the SIF to propose some practical realisations that are generically good (sparse and faithful) in a given context. Second is the modelling of internal delay, this being both computational delay and communication time delay, for example, when the controller algorithm has to be split on different processors. Third is to take more precisely into account HW/SW target, so linking the present work more deeply with what is done in the HW/SW community. Last, but not the least, improving the optimisation process (cheap evaluation of the measures, choice and tuning of the optimisation solver, distance evaluation to the optimal optimum) is still an important challenge, although the developed Matlab toolbox, the *FWR Toolbox*, has been able to provide interesting results in different situations.

References

Aplevich, J. (1991), *Implicit Linear Systems*, Heidelberg: Springer-Verlag.

Chen, S., and Luk, B. (1999), 'Adaptive Simulated Annealing for Optimization in Signal Processing Applications', *Signal Processing*, 79, 117–128.

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.

Goodall, R. (2001), 'Perspectives on Processing for Real-time Control', *Annual Reviews in Control*, 25, 123–131.

Hao, J., and Li, G. (2005), 'An Efficient Structure for Finite Precision Implementation of Digital Systems', in *Information, Communications and Signal Processing, 2005 Fifth International Conference on*, pp. 564–568.

Hilaire, T. (2006), 'Analyse et synthése de l'implémentation de lois de contrôle-commande en précision finie (Étude dans le cadre des applications automobiles sur calculateur embarquée)', Université de Nantes.

Hilaire, T., and Chevrel, P. (2008), 'On the Compact Formulation of the Derivation of a Transfer Matrix with Respect to Another Matrix', Technical Report RR-6760, INRIA.

Hilaire, T., Chevrel, P., and Trinquet, Y. (2005a), 'Designing Low Parametric Sensitivity FWL Realizations of LTI Controllers/Filters within the Implicit State-space Framework', in *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference (CDC-ECC'05)*, pp. 5192–5197.

Hilaire, T., Chevrel, P., and Trinquet, Y. (2005b), 'Implicit State-space Representation: A Unifying Framework for FWL Implementation of LTI Systems', *Proceedings of the 16th IFAC World Congress*, pp. 285–290.

Hilaire, T., Chevrel, P., and Whidborne, J. (2007a), 'Low Parametric Closed-loop Sensitivity Realizations using Fixed-point and Floating-point Arithmetic', in *Proceedings of the European Control Conference (ECC'07)*, pp. 4707–4714.

Hilaire, T., Chevrel, P., and Whidborne, J. (2007b), 'A Unifying Framework for Finite Wordlength Realizations', *IEEE Transactions on Circuits and Systems*, 8, 1765–1774.

Hilaire, T., Ménard, D., and Sentieys, O. (2007c), 'Roundoff Noise Analysis of FiniteWordlength Realizations with the Implicit State-space Framework', in *15th European Signal Processing Conference (EUSIPOC'07)*, pp. 1019–1023.

Hwang, S. (1977), 'Minimum Uncorrelated Unit Noise in State-space Digital Filtering', *IEEE Transactions on Acoustics Speech, and Signal Processing*, 25, 273–281.

Ikeda, M., Šiljak, D., and White, D. (1984), 'An Inclusion Principle for Dynamic Systems', *IEEE Transactions on Automatic Control*, 29, 244–249.

Ingber, L. (1996), 'Adaptive Simulated Annealing (ASA): Lessons Learned', *Control and Cybernetics*, 25, 33–54.

Istepanian, R., and Whidborne, J. (eds) (2001), *Digital Controller Implementation and Fragility: A Modern Perspective*, London, UK: Springer-Verlag.

Ko, H.J., and Yu, W.S. (2003), 'Improved Eigenvalue Sensitivity for Finite-precision Digital Controller Realisations via Orthogonal Hermitian Transform', *IEE Proceedings on Control Theory and Applications*, 150, 365–375.

Li, G. (1998), 'On the Structure of Digital Controllers with Finite Word Length Consideration', *IEEE Transactions on Automatic Control*, 43, 689–693.

Li, G. (2004), 'A Polynomial-operator-based DFIIt Structure for IIR Filters', *IEEE Transactions on Circuits and Systems-II*, 51, 147–151.

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*, 51, 769–778.

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.

Palaniswami, M., and Feng, G. (1991), 'Digital Estimation and Control with a New Discrete time Operator', in *Proceedings of the 30th IEEE Conference Decision Control*, Brighton, UK, pp. 1631–1632.

Papoulis, A. (1991), *Probability, Random Variables, and Stochastic Processes*, New York: Mc Graw Hill.

Šiljak, D. (1991), *Decentralized Control of Complex Systems*, Boston: Academic Press.

Thiele, L. (1984), 'Design of Sensitivity and Round-off Noise Optimal State-space Discrete Systems', *International Journal of Circuit Theory Applications*, 12, 39–46.

Whidborne, J., Istepanian, R., and Wu, J. (2001), 'Reduction of Controller Fragility by Pole Sensitivity Minimization', *IEEE Transactions on Automatic Control*, 46, 320–325.

Whidborne, J., Wu, J., and Istepanian, R. (2000), 'Finite Word Length Stability Issues in an ℓ_1 Framework', *International Journal of Control*, 73, 166–176.

Widrow, B. (1960), 'Statistical Analysis of Amplitude Quantized Sampled-data Systems', *Transactions of the AIEE*, 2, 555–568.

Wu, J., and Chen, S. (2004), 'Stable Controller Coefficient Perturbation in Floating Point Implementation', *Unsolved Problems in Mathematical Systems and Control Theory*, Princeton: Princeton University Press, pp. 280–284.

Wu, J., Chen, S., Li, G., Istepanian, R., and Chu, J. (2001), 'An Improved Closed-loop Stability Related Measure for Finite-precision Digital Controller Realizations', *IEEE Transactions on Automatic Control*, 46, 1162–1166.

Wu, J., Chen, S., Whidborne, J., and Chu, J. (2003), 'A Unified Closed-loop Stability Measure for Finite-precision Digital Controller Realizations Implemented in Different Representation Schemes', *IEEE Transactions on Automatic Control*, 48, 816–823.

Wu, J., Li, G., Chan, S., and Chu, J. (2008), 'A μ-based Optimal Finite-word-length Controller Design', *Automatica*, 44, 3093–3099.

Zhou, K., Doyle, J., and Glover, K. (1996), *Robust and Optimal Control*, Englewood Cliffs, NJ: Prentice-Hall.

Appendix. Algorithms and numerical values

Input: u : real
Output: y : real
Data: xn : array of four reals
Data: xnp : array of four reals
Data: Acc : real
begin
  // *compute* $xnp(1)$
  $Acc \leftarrow xn(1) * 1.0056699573;$
  $Acc \leftarrow Acc + xn(2) * -0.3855253273;$
  $Acc \leftarrow Acc + xn(3) * 0.7882084769;$
  $Acc \leftarrow Acc + xn(4) * -0.8602211557;$
  $xnp(1) \leftarrow Acc + u * -1991.2978135292;$
  // *compute* $xnp(2)$
  $Acc \leftarrow xn(1) * -1.7060282729;$
  $Acc \leftarrow Acc + xn(2) * 1.1129704773;$
  $Acc \leftarrow Acc + xn(3) * 0.6255751647;$
  $Acc \leftarrow Acc + xn(4) * -3.4333411367;$
  $xnp(1) \leftarrow Acc + u * 5980.9414091468;$
  // *compute* $xnp(3)$
  $Acc \leftarrow xn(1) * -0.8063580681;$
  $Acc \leftarrow Acc + xn(2) * 0.3468387941;$
  $Acc \leftarrow Acc + xn(3) * 0.5800952206;$
  $Acc \leftarrow Acc + xn(4) * -0.9426058134;$
  $xnp(3) \leftarrow Acc + u * 4482.5598405197;$
  // *compute* $xnp(4)$
  $Acc \leftarrow xn(1) * -2.5973181092;$
  $Acc \leftarrow Acc + xn(2) * 1.5009691911;$
  $Acc \leftarrow Acc + xn(3) * -1.9422913020;$
  $Acc \leftarrow Acc + xn(4) * -0.3821356552;$
  $xnp(4) \leftarrow Acc + u * 15599.2014809957;$
  // *compute the output*
  $Acc \leftarrow xn(1) * 1.3425518386;$
  $Acc \leftarrow Acc + xn(2) * -0.0635813666;$
  $Acc \leftarrow Acc + xn(3) * -0.5530485340;$
  $y \leftarrow Acc + xn(4) * 2.8068277711;$
  // *save the states*
  $xn \leftarrow xnp$
end

Algorithm 1: Realisation Z_6.

Input: u : 16 bits integer
Output: y : 16 bits integer
Data: xn : array of four 16 bits integers
Data: xnp : array of four 16 bits integers
Data: Acc : 32 bits integer
begin
  // *compute* $xnp(1)$
  $Acc \leftarrow xn(1) * 16477;$
  $Acc \leftarrow Acc + xn(2) * -12633;$
  $Acc \leftarrow Acc + xn(3) * 6457;$
  $Acc \leftarrow Acc + xn(4) * -7047;$
  $Acc \leftarrow Acc + u * -498;$
  $xnp(1) \leftarrow Acc >> 14;$
  // *compute* $xnp(2)$
  $Acc \leftarrow xn(1) * -13976;$
  $Acc \leftarrow Acc + xn(2) * 18235;$
  $Acc \leftarrow Acc + xn(3) * 2562;$
  $Acc \leftarrow Acc + xn(4) * -14063;$
  $Acc = Acc + u * 748;$
  $xnp(2) \leftarrow Acc >> 14;$
  // *compute* $xnp(3)$
  $Acc \leftarrow xn(1) * -26423;$
  $Acc \leftarrow Acc + xn(2) * 22730;$
  $Acc \leftarrow Acc + xn(3) * 9504;$
  $Acc \leftarrow Acc + xn(4) * -15444;$
  $Acc \leftarrow Acc + u * 2241;$
  $xnp(3) \leftarrow Acc >> 14;$
  // *compute* $xnp(4)$
  $Acc \leftarrow xn(1) * -21277;$
  $Acc \leftarrow Acc + xn(2) * 24592;$
  $Acc \leftarrow Acc + xn(3) * -7956;$
  $Acc \leftarrow Acc + xn(4) * -1565;$
  $Acc \leftarrow Acc + u * 1950;$
  $xnp(4) \leftarrow Acc >> 12;$
  // *compute the output*
  $Acc \leftarrow xn(1) * 21996;$
  $Acc \leftarrow Acc + xn(2) * -2083;$
  $Acc \leftarrow Acc + xn(3) * -4531;$
  $Acc \leftarrow Acc + xn(4) * 22994;$
  $y \leftarrow Acc >> 15;$
  // *save the states*
  $xn \leftarrow xnp$
end

Algorithm 2: Fixed-point algorithm of realisation Z_6.

Input: u : real
Output: y : real
Data: xn : array of four reals
Data: Acc : real
Data: T : array of four reals
begin
    // *Intermediate variables*
    $T(1) \leftarrow xn(1) * 0.125$;
    $T(2) \leftarrow xn(2) * 0.125$;
    $T(3) \leftarrow xn(3) * 0.125$;
    $T(4) \leftarrow xn(4) * 0.125$;
    // *compute* $xn(1)$
    $Acc \leftarrow T(1) * -8.5940609251$;
    $Acc \leftarrow Acc + T(2)$;
    $Acc \leftarrow Acc + xn(1) * 0.9974440349$;
    $xn(1) \leftarrow Acc + u * 306012.0144582504$;
    // *compute* $xn(2)$
    $Acc \leftarrow T(1) * -35.2839059945$;
    $Acc \leftarrow Acc + T(3)$;
    $Acc \leftarrow Acc + xn(2) * 0.4134893631$;
    $xn(2) \leftarrow Acc + u * -660870.6659178101$;
    // *compute* $xn(3)$
    $Acc \leftarrow T(1) * -201.7634931054$;
    $Acc \leftarrow Acc + T(4)$;
    $Acc \leftarrow Acc + xn(3) * 0.9864594697$;
    $xn(3) \leftarrow Acc + u * 966164.3351972550$;
    // *compute* $xn(4)$
    $Acc \leftarrow T(1) * -237.4643508571$;
    $Acc \leftarrow Acc + xn(4) * 0.9934647479$;
    $xn(4) \leftarrow Acc + u * 1086873.2436256856$;
    // *compute the output*
    $y \leftarrow T(1)$;
end

Algorithm 3: Realisation Z_{11}.

Input: u : 16 bits integer
Output: y : 16 bits integer
Data: xn : array of four 16 bits integers
Data: Acc : 32 bits integer
Data: T : array of four 16 bits integers
begin
    // *Intermediate variables*
    $T \leftarrow xn$;
    // *compute* $xn(1)$
    $Acc \leftarrow T(1) * -17601$;
    $Acc \leftarrow Acc + T(2) << 13$;
    $Acc \leftarrow Acc + xn(1) * 16342$;
    $Acc \leftarrow Acc + u * 4781$;
    $xn(1) \leftarrow Acc >> 14$;
    // *compute* $xn(2)$
    $Acc \leftarrow T(1) * -18065$;
    $Acc \leftarrow Acc + T(3) << 13$;
    $Acc \leftarrow Acc + xn(2) * 6775$;
    $Acc \leftarrow Acc + u * -2582$;
    $xn(2) \leftarrow Acc >> 14$;
    // *compute* $xn(3)$
    $Acc \leftarrow T(1) * -25826$;
    $Acc \leftarrow Acc + T(4) << 12$;
    $Acc \leftarrow Acc + xn(3) * 16162$;
    $Acc \leftarrow Acc + u * 944$;
    $xn(3) \leftarrow Acc >> 14$;
    // *compute* $xn(4)$
    $Acc \leftarrow T(1) * -30395$;
    $Acc \leftarrow Acc + xn(4) * 32554$;
    $Acc \leftarrow Acc + u * 1061$;
    $xn(4) \leftarrow Acc >> 15$;
    // *compute the output*
    $y \leftarrow T(1)$;
end

Algorithm 4: Fixed-point algorithm of realisation Z_{11}.

$$Z_1 = \left(\begin{array}{cccc:c} 0 & 0 & 0 & \mathbf{-0.17565} & 1 \\ 1 & 0 & 0 & \mathbf{0.96455} & 0 \\ 0 & 1 & 0 & \mathbf{-2.1662} & 0 \\ 0 & 0 & 1 & \mathbf{2.3166} & 0 \\ \hdashline \mathbf{38252} & \mathbf{-13264} & \mathbf{-22452} & \mathbf{-13615} & 0 \end{array}\right),$$

$$Z_2 = \left(\begin{array}{cccc:c} \mathbf{0.11188} & \mathbf{-0.54082} & \mathbf{0.19539} & \mathbf{-0.053116} & \mathbf{203.18} \\ \mathbf{0.54082} & \mathbf{0.72159} & \mathbf{0.1647} & \mathbf{-0.034978} & \mathbf{63.57} \\ \mathbf{0.19539} & \mathbf{-0.1647} & \mathbf{0.76428} & \mathbf{0.12977} & \mathbf{-32.042} \\ \mathbf{0.053116} & \mathbf{-0.034978} & \mathbf{-0.12977} & \mathbf{0.71885} & \mathbf{-4.1143} \\ \hdashline \mathbf{203.18} & \mathbf{-63.57} & \mathbf{-32.042} & \mathbf{4.1143} & 0 \end{array}\right) \tag{A1}$$

$$Z_3 = \left(\begin{array}{cccc:c} \mathbf{3.0771} & \mathbf{1.9943} & \mathbf{-3.5223} & \mathbf{-0.81099} & \mathbf{-8.6995} \\ \mathbf{19.018} & \mathbf{17.794} & \mathbf{-28.317} & \mathbf{-4.7792} & \mathbf{-14.709} \\ \mathbf{15.651} & \mathbf{13.987} & \mathbf{-22.86} & \mathbf{-4.4711} & \mathbf{-24.353} \\ \mathbf{-11.38} & \mathbf{-10.264} & \mathbf{17.463} & \mathbf{4.3055} & \mathbf{19.502} \\ \hdashline \mathbf{3953.9} & \mathbf{3517.5} & \mathbf{-5956.1} & \mathbf{-1059.4} & 0 \end{array}\right),$$

$$Z_4 = \left(\begin{array}{ccccc} & & & & \\ \hline \mathbf{2.1976} & \mathbf{2.225} & \mathbf{1.4698} & \mathbf{-0.6568} & \mathbf{-77.48} \\ \mathbf{0.18131} & \mathbf{-0.82788} & \mathbf{-1.5695} & \mathbf{-0.4138} & \mathbf{69.498} \\ \mathbf{-0.95285} & \mathbf{1.0322} & \mathbf{2.2218} & \mathbf{0.88142} & \mathbf{-45.666} \\ \mathbf{2.5862} & \mathbf{-0.54545} & \mathbf{-1.6235} & \mathbf{-1.2749} & \mathbf{42.167} \\ \hline \mathbf{-394.63} & \mathbf{40.523} & \mathbf{200.59} & \mathbf{332.48} & 0 \end{array}\right), \tag{A2}$$

$$Z_5 = \left(\begin{array}{ccccc} & & & & \\ \hline \mathbf{26860} & \mathbf{1.1171e+5} & \mathbf{-64054} & \mathbf{16716} & \mathbf{6.2454e+8} \\ \mathbf{3731.3} & \mathbf{15520} & \mathbf{-8898.5} & \mathbf{2322.2} & \mathbf{8.4763e+7} \\ \mathbf{23883} & \mathbf{99334} & \mathbf{-56955} & \mathbf{14864} & \mathbf{5.5625e+8} \\ \mathbf{23421} & \mathbf{97413} & \mathbf{-55854} & \mathbf{14577} & \mathbf{5.612e+8} \\ \hline \mathbf{-0.18915} & \mathbf{-0.78675} & \mathbf{0.4511} & \mathbf{-0.11772} & 0 \end{array}\right),$$

$$Z_6 = \left(\begin{array}{ccccc} & & & & \\ \hline \mathbf{1.0057} & \mathbf{-0.38553} & \mathbf{0.78821} & \mathbf{-0.86022} & \mathbf{-1991.3} \\ \mathbf{-1.706} & \mathbf{1.113} & \mathbf{0.62558} & \mathbf{-3.4333} & \mathbf{5980.9} \\ \mathbf{-0.80636} & \mathbf{0.34684} & \mathbf{0.5801} & \mathbf{-0.94261} & \mathbf{4482.6} \\ \mathbf{-2.5973} & \mathbf{1.501} & \mathbf{-1.9423} & \mathbf{-0.38214} & \mathbf{15599} \\ \hline \mathbf{1.3426} & \mathbf{-0.063581} & \mathbf{-0.55305} & \mathbf{2.8068} & 0 \end{array}\right), \tag{A3}$$

$$Z_7 = \left(\begin{array}{cccc:cccc:c} -1 & 0 & 0 & 0 & 0.125 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0.125 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0.125 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0.125 & 0 \\ \hdashline \mathbf{-13.467} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \mathbf{3.0601e+5} \\ \mathbf{-77.847} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & \mathbf{8.2411e+5} \\ \mathbf{-214} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \mathbf{1.0924e+6} \\ \mathbf{-248.44} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{1.1418e+6} \\ \hdashline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \tag{A4}$$

$$Z_{11} = \left(\begin{array}{cccc:cccc:c} -1 & 0 & 0 & 0 & 0.125 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0.125 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0.125 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0.125 & 0 \\ \hdashline \mathbf{-8.5941} & 1 & 0 & 0 & 0.99744 & 0 & 0 & 0 & \mathbf{3.0601e+5} \\ \mathbf{-35.284} & 0 & 1 & 0 & 0 & 0.41349 & 0 & 0 & \mathbf{-6.6087e+5} \\ \mathbf{-201.76} & 0 & 0 & 1 & 0 & 0 & 0.98646 & 0 & \mathbf{9.6616e+5} \\ \mathbf{-237.46} & 0 & 0 & 0 & 0 & 0 & 0 & 0.99346 & \mathbf{1.0869e+6} \\ \hdashline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \tag{A5}$$