Finite-precision implementation issues in narrowband active control

Martina Maggio, Alberto Leva, Luigi Piroddi

***Abstract*— The domain of active control of narrowband disturbances, of either vibrational or acoustic noise source, poses several methodological and implementational issues. Besides all the problems that normally arise in digital control, some specific issues are related to the fact that narrowband control addresses the attenuation of harmonic or quasi-harmonic signals, and consequently implies the usage of control blocks that approach, or even touch, the stability limit, such as notch filters or dynamic oscillators. The finite-precision implementation of these blocks can evidence several problems related to nonlinear effects such as signal quantization, numerical representation, and approximate computation. This paper focuses on the effects related to finite-precision computations and numerical representation, and, after a brief literature review, shows the necessity of extending conventional methodologies for the analysis of finite-precision controller implementation. Some illustrative examples are also provided to illustrate the implementation problems discussed.**

I. Introduction

In any field of automatic control, there is a gap between the performances forecast based on the models of the controlled system and the controller, and those actually obtained when the designed controller is applied to the physical system. Several facts, both methodological and technological, can be held responsible for such behavior. Some, such as the presence of unmodeled dynamics, appear in all types of digital control systems, and are consequently very well known and studied. In active control of noise and vibrations, however, specific issues arise that make it harder to obtain in practice the expected performances, and the mentioned gap tends to be wider than in other fields. Such issues are mainly related to the controller implementation than to the modeling or design of the controlled system, and specifically to the criticality of implementing systems that are near to the stability limit with finite precision arithmetic.

Active control deals with techniques and methods for noise or vibration attenuation that use artificially generated secondary acoustic or mechanical force sources to contrast the offending disturbances. Both narrowband and broadband noise control methods are reported in the literature, as well as feedback and (adaptive) feedforward approaches (see [1]). In this work we concentrate on narrowband noise control methods, more precisely on purely harmonic noise reduction techniques. In closed-loop schemes the frequency response of the sensitivity function, i.e. of the disturbance-to-output transfer function, must be shaped as a notch filter, to attenuate the offending frequencies, and correspondingly, the controller must present oscillatory dynamics, characterized in frequency by a pronounced magnitude peak of its frequency response function at the same frequencies. Clearly, the controller must contain some blocks at - or at least in the proximity of - the stability limit. In adaptive open-loop schemes, several methods can be used for harmonic noise rejection, such as waveform synthesis and adaptive notch filtering. A periodic reference signal correlated with the noise signal is passed through an adaptively adjusted filter to obtain the cancellation signal. Such reference signal is either directly obtained by measurement of a suitable noise-dependent signal, or artificially generated. In the latter case, the simplest solution for harmonic noise rejection is to generate a sinusoidal signal of the proper frequency and adaptively modulate it in magnitude and phase. A critical element of such schemes is the sinusoidal signal generator, which must be able to synthesize high-purity, sometimes variable-frequency sinusoidal signals. The high precision required - and the memory limitations - typically prevents the use of lookup tables, which are known to determine several types of signal distortion [1], and more accurate - and apparently computationally efficient - dynamical oscillators are preferred. Again, the latter are realized by dynamical systems at the limit of stability [2].

The design of control systems containing blocks of the type described above is critical, due to the unavoidable presence of nearly unstable dynamics. Moreover, even if satisfactory performance can be obtained in principle, and verified using 'infinite' precision computational means, the implementation of such dynamics on a finite precision digital controller is impaired by relevant numerical stability problems. In [3] a simulation analysis is carried out on a benchmark active vibration control problem to show the relevance of numerical issues in limiting the attainable performance in practice, both in feedback and feedforward methods. Apparently, the numerical and computational issues that are present in any controller realization come to play here far more relevant a role than in other situations, and the analysis techniques traditionally employed to address those issues prove not completely adequate. This work conjectures that the root of the problem resides in the implementation of nearly-unstable blocks with finite precision arithmetics, and is therefore devoted to the analysis and formalization of that problem, focusing on the (general enough) class of second order discrete-time linear, time invariant (LTI) systems with complex conjugate poles with radius near to 1. The usefulness of the reported analysis is then witnessed by some examples pertaining to vibration control.

The manuscript is organized as follows. Section II re-

M. Maggio is a former student at the Dipartimento di elettronica e Informazione, A. Leva (corresponding author) and L. Piroddi with Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy {leva,piroddi}@elet.polimi.it; .

ports a literature review that, albeit without any exhaustivity claims, evidences the actual relevance of the addressed problem. Section III states the problem and Section IV introduces some practical indices designed to evaluate the 'correctness' of a finite-precision controller implementation. Section V shows some examples, to point out the critical aspects of the finite implementation of second order nearly unstable oscillating systems. Finally, in section VI some conclusions are drawn.

II. Literature review

The topic of finite precision implementation of digital controllers has been the object of plenty research activity, so that the brief review that follows cannot certainly claim to be exhaustive, and simply aims at better delimiting the problem addressed herein. In the sequel, an 'infinite precision' (IP) system denotes a system designed over the field of real numbers while its implementation in finite precision (FP) is referred to as FP system.

Many works aim at deriving conditions to guarantee that some property of a control loop including a digital IP controller carry over to the system where that controller is implemented in FP. For example, [4], [5], [6], [7], [8] address the problem of assessing closed-loop stability in the presence of FP controllers and several related issues. In particular, [7] studies the robust stability of FP digital systems, comparing approaches based on pole sensitivity and stability radius, and notes (which is interesting in the context of this study) that the 'linear' framework, inherent to the two compared approaches, leads to bounds on the stability measure, the exact determination of which is in that framework 'intractable'. On a similar front, [9] studies the effects of FP controllers on the stability and performance of sampled-data systems by adopting a statistical point of view.

Other works are devoted to the optimization of some property of loops with FP controllers: for example, [10] proposes a method to improve the robustness of closed-loop stability in the presence of FP digital controllers, resorting to eigenvalue sensitivity analysis, [11] designs finite precision digital controllers by optimizing a closed-loop stability measure, [12] addresses the problem of designing a digital controller in the presence of roundoff errors, based on a LQG technique, [13] applies genetic algorithms to the design of digital FP controllers, concentrating essentially on the 'rounding' of parameters.

There are also several works that explicitly refer to vibration or noise control and suppression: for example, [14] analyzes the FP implementation of the LMS algorithm (also addressed in this work) with respect to roundoff and quantization errors, [15] analyzes the effects of roundoff errors on LMS-based echo cancelers, [16] presents a variant of the LMS algorithm based on a transformation that evidences the principal components of the transfer matrix between the sensors and actuators at a single frequency, [17] introduces state space models (as opposed to FIR ones) in vibration control methods, claiming in particular that they 'provide a minimal parameterization which has the potential to be made robust to FP errors', and presents an experimental validation.

Some facts are worth noting from the review above. First, a common perspective when analyzing general purpose control *loops* including an FP controller, sets the focus mainly on the 'optimality' of FP realizations with respect to structural properties of the closed loop system, especially stability. The 'proximity' of the FP controller to its conceptual IP counterpart is seldom mentioned, as it is not usually perceived as a critical problem. Such a proximity concept is more often treated in *open-loop* contexts such as signal filtering (see, e.g., [18], [19], [20], [21] and many others), but that matter somehow strays from our study—it suffices to mention the recent work [22], that applies genetic algorithms to the optimization of FP implementation of digital systems, in order to minimize the difference with respect to the 'ideal' IP system in the frequency domain.

Second, and crucial for this study, the analysis of FP systems is typically conducted assuming that FP numbers are approximations of real numbers, *and* that arithmetic operations can be modeled either as exact, or at most as the computation of the *exact* result, *preceded* and/or *followed* by roundoffs, overflows and so forth. Doing so ultimately conceals the inherently nonlinear character *of the FP operators themselves*, and may in some cases lead to erratic analysis results.led either as exact, or at most as the computation of the *exact* result, *preceded* and/or *followed* by roundoffs, overflows and so forth. Doing so ultimately conceals the inherently nonlinear character *of the FP operators themselves*, and may in some cases lead to erratic analysis results.

A notable example in this respect is [23], where it is recognized that FP arithmetics may induce limit cycles when implementing LTI digital filters, and the amplitude of such cycles is analyzed with respect to overflow and quantization, the former being modeled as an operator applied to the result of exact multiplications and sums, the latter as an additive error. Also, many works in this area adopt a probabilistic framework to model the effects of FP arithmetics, whereas they are perfectly deterministic, though intricately nonlinear [24], [25], [26]. An exception is somehow given by [27], that addresses the problem of guaranteeing closed-loop stability for systems with digital FP controllers. Detailed consideration of arithmetic aspects is introduced, but the focus is on the minimization of the system's stability sensitivity to numerical errors as a whole, rather than on the 'proximity' of the FP controller to the IP one. In general, FP systems are mostly evaluated in terms of the preservation of some property of the control loop, but not also in terms of their 'proximity' to an IP reference system, and it is seldom taken into account that FP arithmetic operators do not behave as their IP counterparts. Some works touch part of the mentioned aspects, but none addresses them all together.

Actually, the behavior of FP operators is quite different from that of their IP counterparts, and is not generally acknowledged in the literature that quantization, roundoff and overflow are not always enough to describe such operators in such a way that the 'proximity' of an FP system to its IP counterpart can be assessed. A notable case where this happens is precisely the case of controller blocks approaching the stability limit. The FP implementation of such systems can be quite critical and is is seldom mentioned in the literature [28], although there are application frameworks where their use is natural.

III. Problem Statement

In this research we restrict the scope to the implementation of LTI systems. Such systems are defined over the set $\mathbb{R}$ of real numbers, which forms a field when coupled to the $+$ and $\times$ operators. FP implementations of LTI systems, on the other hand, are defined over the set $\mathbb{F}_{\mathbb{P}}$ of FP numbers with precision P, which coupled to the 'FP sum' $\oplus$ and 'FP product' $\otimes$ operators is *not* a field.

Rigorously speaking no property of LTI systems can be *a priori* transferred to their FP implementations, as not only FP numbers approximate real ones, but the FP operators *do not* behave as their IP counterparts. The importance of that is pointed out by the following example. Let a generic binary IP operator be

$$z = f(x, y), \quad x, y, z \in \mathbb{R} \tag{1}$$

Let $\overline{x}$, $\overline{y}$ be the FP representation of x, y, i. e.,

$$\overline{x} = x(1 + \epsilon_x), \quad \overline{y} = y(1 + \epsilon_y), \tag{2}$$

and $\overline{f}$ be the FP implementation of f, i. e.,

$$\overline{z} = \overline{f}(\overline{x}, \overline{y}), \quad \overline{x}, \overline{y}, \overline{z} \in \mathbb{F}. \tag{3}$$

Suppose that the function $f(\cdot,\cdot):\ \mathbb{R}^2 \mapsto \mathbb{R}$ is suitably differentiable, but observe immediately that no similar assumption is possible (nor makes sense) for $\overline{f}(\cdot,\cdot):\ \mathbb{F}^2 \mapsto \mathbb{F}$. One can therefore write

$$\begin{aligned} \overline{z} &= \overline{f}(\overline{x}, \overline{y}) = \overline{f}(x(1+\epsilon_x), y(1+\epsilon_y)) \\ &= f(x(1+\epsilon_x), y(1+\epsilon_y)) + \delta f(x, y, \epsilon_x, \epsilon_y) \end{aligned} \tag{4}$$

where δf accounts for the different behaviors of f and $\overline{f}$. Expanding to the first order yields

$$\begin{aligned} \overline{z} &= z + f_x(x,y)x\epsilon_x + f_y(x,y)y\epsilon_y \\ &\quad + \delta f(x, y, \epsilon_x, \epsilon_y) + O\left(|x\epsilon_x|^2, |y\epsilon_y|^2\right). \end{aligned} \tag{5}$$

Now, second order terms can be safely discarded and, defining the multiplicative error $\epsilon_z:\ \overline{z} = z(1+\epsilon_z)$, it comes immediately that

$$\epsilon_z = \frac{f_x(x,y)}{f(x,y)}x\epsilon_x + \frac{f_y(x,y)}{f(x,y)}y\epsilon_y + \frac{\delta f(x,y,\epsilon_x,\epsilon_y)}{f(x,y)}. \tag{6}$$

The first two terms of ϵ_z in (6) are real numbers, and (as well known) quantify the *conditioning* of the operator, i.e., the 'difficulty' of evaluating it with certain values of the operands, *irrespective of any FP representation of them*. Under the hypothesis that the last term can be neglected, on the first two is based the vast literature concerning operator and algorithm stability analysis, where problems are dealt with *entirely* in $\mathbb{R}$, and plenty of well established results exist. The last term can be easily interpreted as a 'relative operator behavior error', and brings the particular FP implementation (from the number format to the coding of the operator's algorithm) into play.

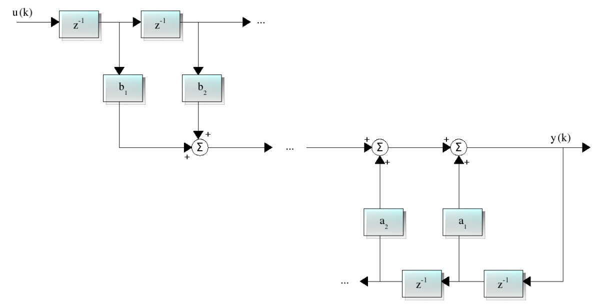

Fig. 1. Block diagram of an LTI system using the IP + and * operators and delays.

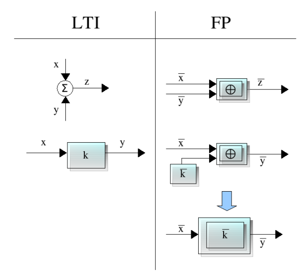

Fig. 2. IP and FP operators as blocks.

The first consequence of introducing δf in the analysis is that, as anticipated, there is no such thing as a 'linear FP system'. In the same way as a discrete-time LTI system can be implemented by a block diagram including gains (i.e., IP products), IP sums, and delays, any FP implementation of that system can be built by means of delays and the *nonlinear* blocks of figures 1 through 3.

In most cases, the facts above are totally irrelevant. However, it is intuitive that if a movement of an LTI system is taken as the reference one to evaluate the precision of an FP implementation of that system, then the effect of δf can be

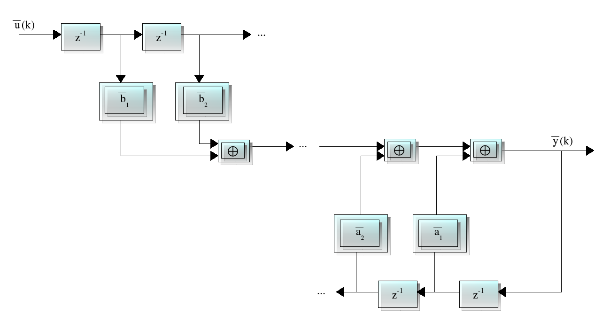

Fig. 3. Block diagram of an FP system using the FP $\oplus$ and $\otimes$ operators and delays.

interpreted as a perturbation of that movement, nontrivially related to the movement itself via the used FP arithmetics. Let us now synthetically formalize the concept.

Assuming, for consistency with current technology, that FP numbers are represented in an IEEE754-like format, and restricting the set of operators to sum and product, an FP arithmetic $\mathcal{A}$ is characterized by the numbers m and e of mantissa and exponent bits, which define the set $\mathbb{F}_{m,e}$ of FP numbers, and by the functions $\overline{f}_{\oplus}(\cdot,\cdot) : \mathbb{F}_{m,e} \times \mathbb{F}_{m,e} \mapsto \mathbb{F}_{m,e}$ and $\overline{f}_{\otimes}(\cdot,\cdot) : \mathbb{F}_{m,e} \times \mathbb{F}_{m,e} \mapsto \mathbb{F}_{m,e}$ that realize the FP sum and product in $\mathbb{F}_{m,e}$. We then term the arithmetic under question $\mathcal{A}(m,e,\overline{f}_{\oplus},\overline{f}_{\otimes})$, and define an '$\mathcal{A}$-FPLTI' dynamic system as the (nonlinear) dynamic system obtained from the LTI one

$$\left\{ \begin{array}{rcl} x(k+1) & = & Ax(k)+Bu(k) \\ y(k) & = & Cx(k)+Du(k) \end{array} \right. \tag{7}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, and (A,B,C,D) are real matrices with the convenient dimensions, by substituting the $(+,\times)$ arithmetics over $\mathbb{R}$ with the arithmetic $\mathcal{A}(m,e,\overline{f}_{\oplus},\overline{f}_{\otimes})$, i.e.

$$\left\{ \begin{array}{rcl} \overline{x}(k+1) & = & \overline{A}\otimes\overline{x}(k)\oplus\overline{B}\otimes\overline{u}(k) \\ \overline{y}(k) & = & \overline{C}\otimes\overline{x}(k)\oplus\overline{D}\otimes\overline{u}(k) \end{array} \right. \tag{8}$$

where $\overline{x} \in \mathbb{F}^n_{m,e}$, $\overline{u} \in \mathbb{F}^m_{m,e}$, $\overline{y} \in \mathbb{F}^p_{m,e}$, and $(\overline{A},\overline{B},\overline{C},\overline{D})$ are matrices of $\mathbb{F}_{m,e}$ numbers. We call (7) the 'LTI system', and (8) its '$\mathcal{A}$-FPLTI counterpart' With respect to similar definitions in the literature, we stress that the key point here is that the definition of $\mathcal{A}(m,e,\overline{f}_{\oplus},\overline{f}_{\otimes})$ includes all details on how the operators are implemented (rounding, truncation, guard bits, over/underflow management, and so forth), and that those details are formally accounted for by the term δf, introduced in (4) for the generic binary operator, and here applied to sum and product.

Suppose now that an LTI system $\mathcal{S}$ is implemented by realizing an $\mathcal{A}$-FPLTI counterpart $\overline{\mathcal{S}}$ of it, and consider the free motion of the state vector of both systems, *starting from an initial value that allows for an exact representation in* $\mathbb{F}_{m,e}$, i.e., from $\tilde{x}(0) \in \mathbb{F}_{m,e} \subset \mathbb{R}$. For $\mathcal{S}$ we apparently have $x(1) = A\tilde{x}(0)$, while for $\overline{\mathcal{S}}$ $\overline{x}(1) = \overline{A}\otimes\tilde{x}(0)$, which in general is not equal to $x(1)$. Hence, despite in the first movement step there is no error in the number representation, the different behaviors of the IP and FP operators can make the movements of $\mathcal{S}$ and $\overline{\mathcal{S}}$ differ right from the beginning, with a variety of possible dynamic effects (irrelevant in most cases but sometimes important, as the following examples will show).

Intuitively, (movements of $\mathcal{A}$-FPLTI counterparts of) LTI systems near the stability limit suffer more from such effects, which leads us to formulate a conjecture: the nearer the LTI system is to the stability boundary, the greater the effect of δf will be. To back up the conjecture, in the following we first define some suitable indicators, which we subsequently use to assess the proximity of an $\mathcal{A}$-FPLTI system to the LTI one it is aimed at implementing.

IV. The proposed indicators

The rationale of the proposed indicators is to evaluate the proximity of an $\mathcal{A}$-FPLTI system $\overline{\mathcal{S}}$ to the LTI system $\mathcal{S}$ it should realize. Given the linear nature of the latter, and therefore (with a slight abuse of notation) the 'almost' or 'ideally linear' character of the former, the indicators are based on the response of the $\mathcal{A}$-FPLTI system to the discrete impulse, or in some sense equivalently, on its free motion. For convenience, we distinguish the $\mathcal{A}$-FPLTI counterparts of asymptotically stable LTI systems from those of LTI systems at the stability boundary.

In the former case, the employed indicators are

- the sum η of squared differences between the responses of $\overline{\mathcal{S}}$ and $\mathcal{S}$, from the initial time up to a number N of samples sufficient to make the norm of the response of $\mathcal{S}$ become less then the machine epsilon as commonly defined in the literature (recall that the limit of the response for time going to infinity is apparently zero);
- the total number Φ of samples for which the responses of $\overline{\mathcal{S}}$ and $\mathcal{S}$ have different sign;
- the maximum number J of *subsequent* samples for which the responses of $\overline{\mathcal{S}}$ and $\mathcal{S}$ have different sign.

The proposed indicators account for differences in terms of both amplitude and time (or phase), therefore providing a preliminary portrait of the problem sufficient to evaluate an FP implementation, and also to evidence the need for a further theoretical study (see the examples later on).

For $\mathcal{A}$-FPLTI counterparts of LTI systems at the stability boundary, the indicators above are not adequate. The responses of such systems are typically oscillations without damping, and do not admit limit nor sum. Therefore, the employed indicators in this case are the time behavior of amplitude and phase of the produced oscillatory response (or of each component of it if the produced oscillation is not sinusoidal).

To compute the indicators, in principle one needs infinite precision computations. This is not possible, of course, and therefore a software library was created with an *ad hoc* number representation, capable of emulating any possible operator algorithm with any precision. The library allows to compute at precisions that are far higher than those typically employed, such as the IEEE standards, and from an operational point of view can be taken as 'infinite'. Notice that doing so is not a theoretical flaw, since to evaluate a precision versus a better one it is only necessary that the latter allows to represent numbers that do not have a representation in the former, irrespective of whether the 'better' precision is infinite or not.

V. Implementation of second order nearly unstable oscillating systems

The FP implementation of oscillating LTI systems such as

$$G(z) = \frac{1}{z^2 - 2\rho\cos(\theta)\,z + \rho^2}, \tag{9}$$

representative of the type of dynamical blocks that would be used to implement a closed-loop notch filter or a dynamical

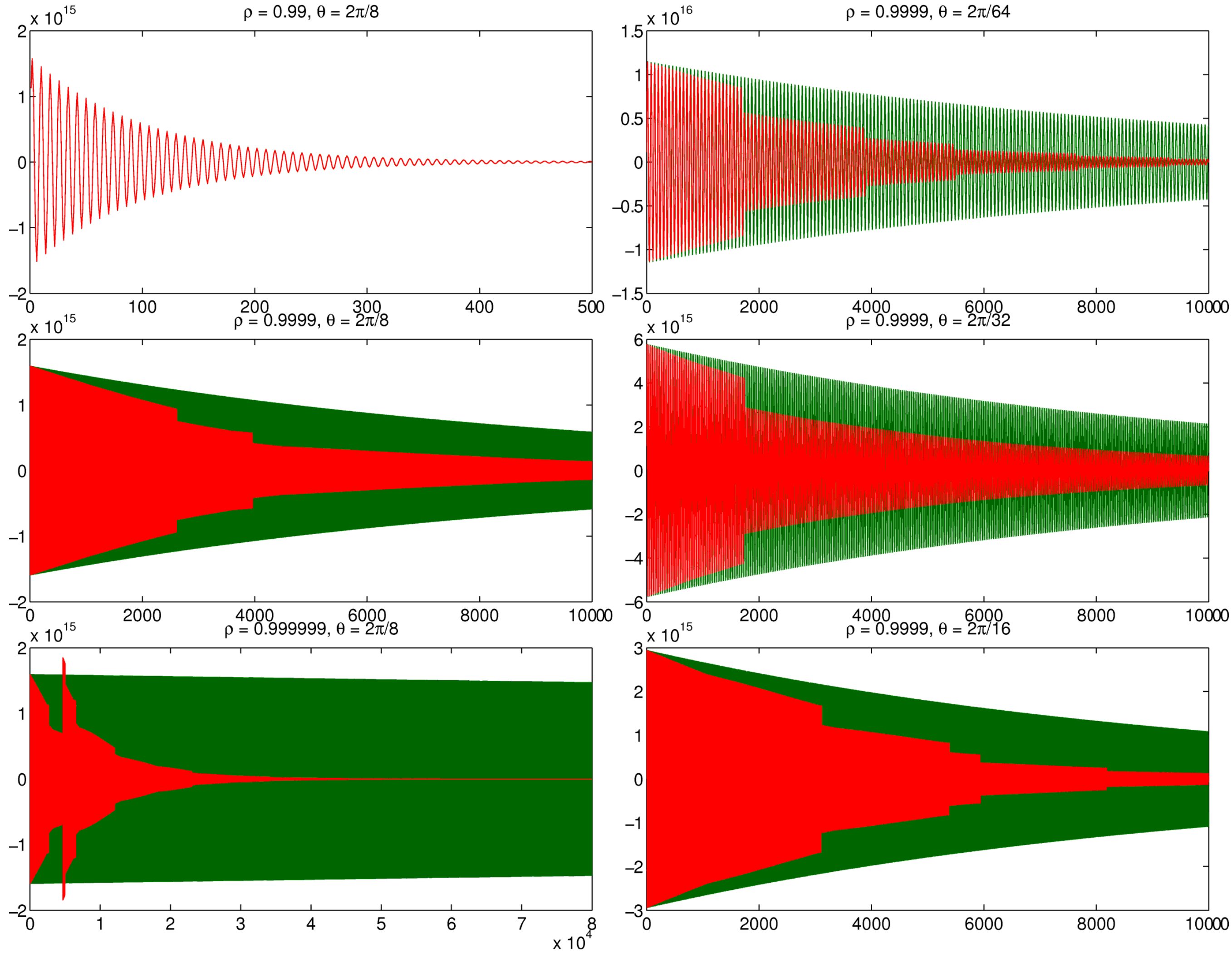

Fig. 4. Effects of ρ and θ: 16 bit precision (red), 32 bit precision (green).

oscillator, can reveal several interesting aspects for varying ρ (which determines the vicinity to the stability limit, i.e. the unit circle) and θ (which accounts for the oscillation period or, stated otherwise, for the time quantization of the response) at different precision levels. Such results can be interpreted in a two-fold manner: from one perspective they set the performance limitations for a given FP precision, in terms, e.g., of notch filter peak limitations and operating frequency ranges; secondly, they suggest a minimum FP precision level that must be used in the implementation for given performance requirements. For reasons of space the case $\rho = 1$ is left out to be documented in further works and examples with $\rho < 1$ will only be discussed.

The first example aims at showing that the effects of finite precision are more relevant when the stability limit is approached or significant time quantization is introduced. The example also shows that the mentioned effects depend on the precision in a nontrivial way, often hard to predict. Figure 4 compares the free motion of (9) to that of some $\mathcal{A}$-FPLTI counterpart of it, for different values of ρ and θ. In particular, the left column shows the effects of the pole radius ρ approaching the stability limit, while the right one shows the effects of variations of the angular frequency θ. It is worth noticing that lower precision implementations exhibit more significant degradations when $\rho \rightarrow 1$; for values close to one, the degradation is even *qualitative*, with evident points of discontinuity. As for the angular frequency, higher values of it introduce degradations that also are related to the employed precision.

The second example shows a qualitative summary of the three indicator values for different precision levels and values of θ, with $\rho = 0.9999$. The relevant facts is that some $\mathcal{A}$-FPLTI are better or worse than others, depending on which indicator is observed. From one side, this means that no ‘univocal quality measure’ is to be sought, and also indicates that apparently similar implementations may behave very differently from different standpoints—i.e., yield differently good results when employed (in control schemes) in different ways. Figure 5 shows that the IP and FP movements are almost indistinguishable only for very high precision and

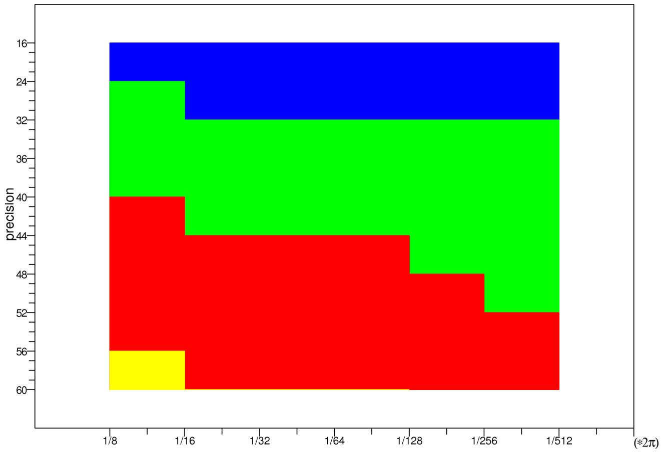

Fig. 5. Qualitative behavior of η ($\rho = 0.9999$): η negligible (yellow), small (red), medium (green), high (blue).

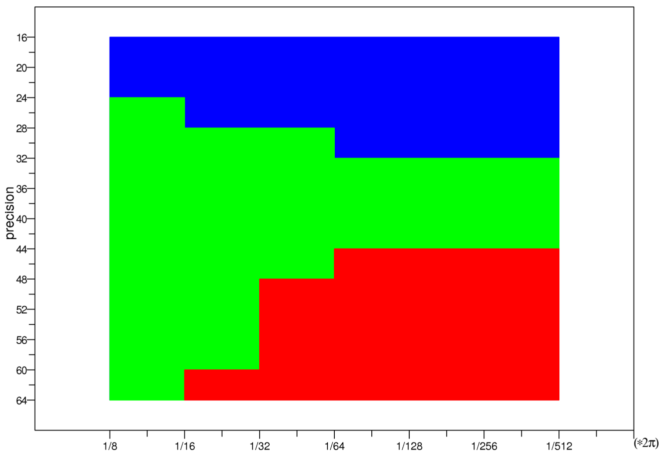

Fig. 6. Qualitative behavior of J and Φ ($\rho = 0.9999$): J low and Φ low (red), J low and Φ high (green), J high and Φ high (blue).

high oscillation frequency (yellow region). Excellent results are obtained for different (medium to high) precisions depending on the oscillation frequency (red zone). Acceptable accuracy may be obtained with a precision as low as 24 bits, but only for higher frequencies, while at least 32 bits are required in the other cases (green region). On the other hand, the performance drastically degrades in the blue region. To obtain a given accuracy in terms of η different precision levels are required depending on the oscillation frequency. Figure 6 mostly accounts for phase errors and is divided in three regions. The red region represents situations with almost negligible phase error, while the blue zone accounts for severe phase inaccuracy. In the green region, the J index is low but Φ is high so that a small but consistent phase error can be presumed. The latter behavior is typically observed as a blurring effect on the resonance peak of the frequency response function.

VI. Conclusions

The problem of evaluating FP implementations of discrete-time LTI controllers was addressed in a way suitable for cases where (part of) the controller approaches or touches the stability limit, as is typical in vibration control.

Classical approaches were reviewed, and the need was evidenced to complement them with the capability of accounting for the behavior of FP operators, the relevance of which was evidenced by suitable examples. In particular, it was pointed out that the properties of an LTI system and of an FP implementation of it are completely different sets of things, the latter often needing even to be defined.

Some indicators were proposed to assess the quality of an FP implementation of an LTI system in terms of 'vicinity' of the two, and it was evidenced how the proposed indicators depend on the precision used. A software tool was realized to evaluate FP implementations for arbitrary precision levels. The obtained results can be employed in the technological design of narrowband active control systems and, in general, to pre-analyze an FP implementation of an IP system. This can be of great help, for example, when planning to realize the designed controllers with technologies in which the computation machinery is part of the realization itself, as happens with FPGA devices.

References

[1] S. Kuo and D. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations*. New York (NY), USA: John Wiley & Sons, 1996.

[2] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice Hall, 1995.

[3] L. Piroddi, A. Leva, and F. Casaro, "Feedback and feedforward active vibration control schemes for a turbomolecular vacuum pump," in *European Control Conference 2007*, Kos, Greece, 2007.

[4] S. Chen, R. Istepanian, and J. Wu, "Optimizin stability bounds of finite precision pid controllers using adaptive simulated annealing," *Proceedings of the American Control Conference*, pp. 4316–4320, 1999.

[5] J. Wu, S. Chen, G. Li, and J. Chu, "Optimal finite precision state-estimate feedback controller realizations of discrete-time systems," *IEEE transactions on automatic control*, vol. 45/8, pp. 1550–1554, 2000.

[6] J. Wu, S. Chen, G. Li, R. Istepanian, and J. Chu, "Shift and delta operator realisations for digital controllers with finite word lenght considerations," *IEE Proceeding Control Theory Applications*, vol. 147/6, pp. 664–672, 2000.

[7] S. Chen, J. Wu, and G. Li, "Two approaches based on pole sensitivity and sability radius measures for finite precision digital controller realizations," 2000.

[8] J. Wu, S. Chen, G. Li, R. Istepanian, and J. Chu, "An improved closed-loop stability related measure for finite-precision digital controller realizations," *IEEE transactions on automatic control*, vol. 46/7, pp. 1162–1166, 2001.

[9] J. Fialho and T. Georgiou, "On stability and performance of sampled-data systems subject to wordlenght constraint," *IEEE transactions on automatic control*, vol. 39/12, pp. 2476–2481, 1994.

[10] W. Yu and H. Ko, "Improved eigenvalue sensitivity for finite-precision diital controller realisations via orthogonal hermitian transform," *IEE Proceeding Control Theory Applications*, vol. 150/4, pp. 365–375, 2003.

[11] J. Wu, W. Xu, S. Chen, and G. Li, "A new approach for finite-precision controller realizations," *Proceedings of 6th World Congress on Intelligent Control and Automation*, pp. 461–465, 2006.

[12] K. Liu, R. Skelton, and K. Grigoriadis, "Optimal controllers for finite wordlenght implementation," *IEEE transactions on automatic control*, vol. 37/9, pp. 1294–1304, 1992.

[13] J. Whidborne and R. Istepanian, "Genetic algorithm approach to designing finite-precision controller structures," *IEE Proceeding Control Theory Applications*, vol. 14X/5, pp. 337–382, 2001.

[14] S. Alexander, “Transident weight misadjustment properties for the finite precision lms algorithm,” *IEEE transactions on acoustic, speech, and signal processing*, vol. 35/9, pp. 1250–1258, 1987.
[15] J. Cioffi, “A finite precision analysis of the block-gradient adaptive data-driven echo canceller,” *IEEE transactions on comunications*, vol. 40/5, pp. 940–946, 1992.
[16] R. Cabell, *A principal component algorithm for Feedforward Active Noise and Vibration Control*, 1998.
[17] G. Nijsse, M. Berhaegen, B. De Schutter, D. Westwick, and N. Doleman, “State space modeling in multichannel active control systems,” *Proceedings of the 1999 International Symposium on Active Control of Sound and Vibration*, pp. 909–920, 1999.
[18] S. Parker and S. Hess, “Canonic realizations of second-order digital filters due to finite precision arithmetic,” *IEEE transactions on circuit theory*, vol. 19, no. 4, pp. 410–413, 1972.
[19] B. Jaumard, M. Minoux, and P. Siohan, “Finite precision design of fir digital filters using a convexity property,” *IEEE transactions on acoustic, speech, and signal processing*, vol. 36/3, pp. 407–411, 1988.
[20] N. Cho and S. Lee, “Optima design of finite precision fir filters using linear programming with reduced constraints,” *IEEE transactions on signal processing*, vol. 46/1, pp. 195–199, 1995.
[21] B. Al-Hasimi, W. Somerset, and M. Moniri, “Constrained genetic algorithm design of finite precision fir linear phase raised cosine filters,” *IEEE transactions on circuits and systems*, vol. 5, pp. 445–448, 1998.
[22] M. Haseyama and D. Matsuura, “A ga-based realization method of optimal finite-precision system,” *EEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 413–416, 2004.
[23] K. Johnson and I. Sandberg, “A separation theorem for finite precision digital filters,” *IEEE transactions on circuits and systems*, vol. 42/9, pp. 541–545, 1995.
[24] E. I. Verriest, “Error analysis of linear recursions in floating point,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’85*, vol. 10, 1985, pp. 1711–1714.
[25] S. Nishimura, “Convergence analysis of adaptive notch filter realized with floating-point arithmetic,” in *IEEE International Conference on Systems Engineering*, 1992, pp. 552–555.
[26] L. Montgomery Smith, B. Bomar, R. Joseph, and G. Yang, “Floating-point roundoff noise analysis of second-order state-space digital filter structures,” *IEEE transactions on circuits and systems, Analog and Digital Signal Processing*, vol. 39, no. 2, pp. 90–98, 1992.
[27] J. Wu, S. Chen, J. Whidborne, and J. Chu, “A unified closed-loop stability measure for finite-precision digital controller realizations implemented in different representation schemes,” *IEEE transactions on automatic control*, vol. 48/5, pp. 816–822, 2003.
[28] F. Chaitin-Chatelin and S. Gratton, “Convergence in finite precision of successive iteration methods under high nonnormality,” *Journal BIT Numerical Mathematics*, vol. 36, no. 3, pp. 455–469, 1996.