Design and Efficient Implementation of Digital Non-integer Order Controllers for Electro-mechanical Systems

Journal of Vibration and Control
00(00):1–15
The Author(s) 2010
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI:
http://mms.sagepub.com
SAGE

Design and Efficient Implementation of Digital Non-integer Order Controllers for Electro-mechanical Systems

Riccardo Caponetto*, Vincenzo Tomasello
Department of Electrical, Electronic and Information Engineering, University of Catania, Italy

Paolo Lino, Guido Maione
Department of Electrical and Information Engineering, Politecnico di Bari, Italy

Abstract
Digital realization of non-integer order controllers is important to exploit the benefits provided by these controllers, in terms of flexibility, dynamic performance and robust stability, for applications in mechatronics, industrial and automotive systems. To realize infinite-dimensional fractional-order operators and controllers in the digital domain, a discrete-time approximation is necessary that must be characterized by stable and minimum-phase properties for control purpose. This paper provides a design method useful for a wide class of plants and applies a consolidated approximation technique. Moreover, the practical implementation problems of digital non-integer control algorithms are deeply analyzed by considering the effect of the sampling period, of the conversion between analog and digital domain (and vice versa) and the associated quantization. Results show benefits and limitations of the approach.

Keywords
Non-integer order controllers, fractional-order PI controllers, discrete fractional-order operators, approximation, realization of non-integer order systems, implementation, sampling, digital-analog converters

1. Introduction

It is well-known that proportional-integral-derivative (PID) controllers still dominate the process control loops in industrial contexts, over 95% of the times (Åström and Hägglund, 1995), in particular PI controllers are employed in the majority of the cases. However, fractional calculus contributed to investigate and develop new control solutions that extend the standard integer-order integral or derivative actions. Namely, much time after seminal applications and studies (Bode, Manabe, Tustin, etc.), non-integer order differentiation and integration were reconsidered. See, for example, the CRONE control method (Oustaloup, 1991), the $PI^{\lambda}D^{\mu}$-controller

* Corresponding author; e-mail: Riccardo.Caponetto@dieei.unict.it

(Podlubny, 1999a), and in general the different versions of the fractional-order PID or fractional-order controllers (Caponetto et al., 2010; Chen et al., 2009).

Currently, there exist several design methods or tuning techniques to determine the parameters of the fractional-order controllers, however there are still implementation issues to be addressed. Some techniques for analog implementation have been proposed by (Caponetto and Dongola, 2008; Caponetto et al., 2008; Petráš, 2012; Podlubny et al., 2002). See also (Caponetto et al., 2014) for a recent implementation by integrated circuits technology.

The problem addressed here is to control electro-mechanical systems by embedded non-integer order controllers. In particular, DC-motors are considered because they are widely employed in mechatronics, automotive systems, and other emerging areas. Since properly tuned PI controllers are frequently used for these plants, we consider the extension based on fractional-order PI controllers, FOPI controllers for short. Moreover, digital fractional-order control solutions are still to be analyzed and optimized, also because many applications deploy low-cost microcontrollers and digital signal processors. Then this paper is motivated to investigate digital implementation problems associated with discretized versions of FOPI controllers.

Section 2 describes how to synthesize the FOPI controller parameters for the considered electro-mechanical plants. Section 3 addresses the realization problem on the basis of a well-known approximation technique that is necessary to achieve rational transfer functions, i.e. to realize the fractional-order controller. Section 4 analyzes in details how to cope with existing limitations due to operations that are necessary to achieve an efficient digital implementation based on microcontrollers or similar devices. Namely, the sampling process, the digital-to-analog and analog-to-digital conversions, and quantization are often neglected whereas they may determine a serious decay in performance and even instability. Section 5 shows experimental results from a lab equipment. Section 6 gives some final remarks.

2. Controller design

This section illustrates how to design the FOPI controller by an efficient method that satisfies frequency-domain specifications on robustness and dynamic performance. Closed-form formulas allow to automatically set the controller parameters. However, other equivalent design methods can be also used.

Many industrial plants, processes, and electro-mechanical systems can be synthetically represented as a first-order system plus a time delay, a FOPTD system for short. In this case, the plant transfer function is given by $G_p(s) = \frac{K}{1+\tau s}\, e^{-Ls}$, where K is the static gain, τ is the dominant time constant, and L is the intrinsic deadtime (if present). If two (or more) time constants describe the process dynamics, they are combined into only one equivalent time constant.

Usually, an integer order PI controller is employed. However, the idea is to extend and give more flexibility to the controller by an integral action of non-integer order, say ν. Then the s-domain standard integral operator $1/s$ is replaced by the irrational non-integer integral operator $1/s^{\nu}$. The controller transfer function becomes $G_c(s) = K_P + \frac{K_I}{s^{\nu}}$. Obviously, $1/s^{\nu}$ must be approximated by a rational transfer function, which allows realization of the non-integer order controller. Moreover, if $1 < \nu < 2$ is used, then $1/s^{\nu} = (1/s)\,(1/s^{\xi})$, with $0 < \xi = \nu - 1 < 1$, meaning that the non-integer integrator is composed by two parts, namely an integer order integrator and a residual non-integer order integrator $1/s^{\xi}$. The first allows rejection of disturbances on the plant input, the latter improves robustness and is the part that must be approximated.

The controller parameters, namely K_P, K_I, and ν, can be designed by one of the available methods in the literature. This work employs an efficient method tailored to integrating plants with a first-order lag (Lino and Maione, 2013; Maione and Lino, 2007) and adapts it to FOPTD systems, which is useful for speed control

of electromechanical processes. The design objective is twofold: to achieve robustness and to approximate an optimal feedback system in a given bandwidth where input-output tracking is desired. On one side, the non-integer integration allows a constant non-integer slope of the Bode magnitude diagram and a nearly flat Bode phase diagram in a sufficiently wide frequency range around the gain crossover frequency (Maione and Lino, 2007; Monje et al., 2007). This allows robustness to gain variations. On the other side, optimality is pursued by shaping the open-loop frequency response so that gain is high at low frequencies and rolls off at high frequencies. Both robustness and performance are specified in the frequency domain by the phase margin and the closed-loop bandwidth, respectively.

To start with the design procedure, consider the open-loop frequency response $G(j\omega) = G_c(j\omega)\,G_p(j\omega)$ with the controller transfer function expressed by K_I and T_I, where $T_I = K_P/K_I$. Then $G(j\omega) = \frac{K\,K_I\,[1+T_I(j\omega)^\nu]}{(j\omega)^\nu\,[1+\tau(j\omega)]}\,e^{-j\omega L}$. Using a normalized frequency $u = \omega\,\tau$ and rearranging the frequency response yields:

$$G(ju) = \frac{K\,K_I\,\tau^\nu\,\left[1 + T_I\,(\frac{u}{\tau})^\nu\,(C + jS)\right]}{u^\nu\,(C + jS)\,(1 + j\,u)}\,e^{-j\,\frac{uL}{\tau}} \tag{1}$$

where $\theta = \frac{\pi}{2}\nu$, $C = \cos(\theta)$, and $S = \sin(\theta)$. Then the magnitude and phase angle of $G(ju)$ are:

$$|G(ju)| = \frac{K\,K_I\,\tau^\nu}{u^\nu}\,\sqrt{\frac{1 + 2\,T_I\,(\frac{u}{\tau})^\nu\,C + T_I^2\,(\frac{u}{\tau})^{2\nu}}{1 + u^2}} \tag{2}$$

$$\angle G(ju) = \arctan\left(\frac{T_I\,(\frac{u}{\tau})^\nu\,S}{1 + T_I\,(\frac{u}{\tau})^\nu\,C}\right) - \arctan(u) - \theta - \frac{uL}{\tau} \tag{3}$$

Now, an approximation of an optimal feedback system with unitary closed-loop gain is achieved in a specified bandwidth, say $u_B = \omega_B\,\tau$, which is determined according to the desired range for input-output tracking. u_B is chosen by a trade-off between the requirement of a prompt closed-loop response and the need to center the crossover in the region where the phase diagram is flat. Namely, a link between u_B and the normalized crossover frequency is given by $u_C \in [\frac{u_B}{1.7}, \frac{u_B}{1.3}]$ (Maciejowski, 1989), e.g. $u_C = \frac{u_B}{1.5}$, bearing in mind that $|G(ju)|$ is shaped around u_C to obtain a low-frequency high gain and a roll off at high frequencies.

The next specification is the phase margin, say PM_s. Given that, for each value of ν, the phase margin can be computed by

$$PM = \arctan\left(\frac{T_I\,(\frac{u_C}{\tau})^\nu\,S}{1 + T_I\,(\frac{u_C}{\tau})^\nu\,C}\right) - \arctan(u_C) - \theta - \frac{u_C L}{\tau} + \pi, \tag{4}$$

the relation $PM = \pi - \theta = (2 - \nu)\,\pi/2$ is enforced in order to obtain a direct and simple design formula between any specification PM_s on the phase margin and the parameter ν (see also (Lino and Maione, 2013; Maione and Lino, 2007)). To exactly obtain the mentioned relation between PM and ν, the unique solution for the controller parameter T_I is

$$T_I = \left(\frac{\tau}{u_C}\right)^\nu\,\frac{u_C + \tan(\frac{u_C L}{\tau})}{S - u_C\,C - (C + u_C\,S)\,\tan(\frac{u_C L}{\tau})}. \tag{5}$$

which is a closed-form design formula.

Moreover, given that (5) and $PM = (2 - \nu)\,\pi/2$ hold true, imposing the specification provides a strict relation between the phase margin and fractional order, i.e. $PM_s = (2 - \nu)\,\pi/2$, then it follows:

$$\nu = 2 - 2\,PM_s/\pi \tag{6}$$

which gives another closed-form formula to obtain ν from the specified phase margin. Since the plant does not include an integer order integrator, (6) provides $\nu > 1$ to include it in $G(s)$.

Finally, the crossover specification is enforced to obtain K_I by another closed-form relation:

$$K_I = \frac{1}{K} \left(\frac{u_C}{\tau}\right)^{\nu} \sqrt{\frac{1+u_C^2}{1+2\,T_I\,(\frac{u_C}{\tau})^{\nu}\,C + T_I^2\,(\frac{u_C}{\tau})^{2\nu}}} \tag{7}$$

The design method guarantees a good trade-off between robustness and performance and is based on simple analytical formulas that relate performance and robustness specifications to the controller parameters. Namely, (5) establishes the integral time constant to obtain the specified bandwidth, (6) establishes the required fractional order to obtain the specified phase-margin. Finally, simplicity of the formulas allows easy implementation.

3. Controller realization: The approximation problem

The key feature of a non-integer order controller is its irrational transfer function (Oustaloup, 1991; Podlubny, 1999a). Namely, the basic operators are non-integer powers of the s variable and correspond to control actions of any non-integer order. Then, synthesis of the controller structure and parameters must be followed by a realization step for a real implementation in control applications.

To this aim, the infinite-dimensional fractional operator s^{ν}, with $\nu \in R$ and $0 < |\nu| < 1$, must be approximated by an almost equivalent rational transfer function, for example by considering the frequency response. From the control point of view, the transfer function providing the approximation must be characterized by stable poles and minimum-phase zeros, both in the analog and in the discrete domain. An usual choice is to alternate zeros and poles along the negative real half-axis of the s-plane, for an analog realization, or inside the real segment $(-1, 1)$ of the z-plane, for a digital realization. This property is guaranteed by the methods in (Maione, 2011b, 2013).

Several methods are available for continuous or discrete approximation (Chen and Moore, 2002; Chen et al., 2004; Maione, 2006, 2008, 2013; Oustaloup, 1991, 1995; Oustaloup et al., 2000; Podlubny et al., 2002; Vinagre et al., 2000). Besides classical interpolation (Oustaloup, 1995; Oustaloup et al., 2000) and fitting techniques, one may use truncation of continued fraction expansions (CFEs) or other interpolation methods (Chen et al., 2004; Maione, 2008; Vinagre et al., 2000), or methods coming from the signal processing field (Barbosa et al., 2006). Some methods exhibit better frequency behavior, some a better time response, but it is difficult to establish the best one from all aspects. In all cases, the approximation provides analog or digital filters, many times in the form of ratios of two polynomials in the variable s or z. The coefficients of the filters, then the zeros and poles of this function, depend on the non-integer order ν.

Herein, the focus is on digital realization of fractional operators and controllers. First, we remark a distinction between discrete approximation methods that design finite impulse response (FIR) filters (Samadi et al., 2004; Tseng, 2001) or infinite impulse response (IIR) filters (Chen and Vinagre, 2003; Vinagre et al., 2003). IIR filters are preferable because the associated approximations are defined by rational transfer functions that provide faster convergence and wider domain of convergence in the complex plane than polynomial approximations (Stoer and Bulirsch, 2002). Moreover, direct or indirect discretization can be considered (Chen and Moore, 2002). Direct methods use a generating function $s = \psi(z^{-1})$ for conversion from the continuous-time to discrete-time domain, then truncate a power series expansion (PSE) or a CFE. The generating function depends on the sampling period T.

The most referenced method is the direct discretization based on the Grünwald-Letnikov (GL) definition of fractional derivative of order ν with lower and upper limits a and t (Oldham and Spanier, 1974):

$$ {}_aD_t^{\nu} f(t) = \lim_{h \to 0} \frac{1}{h^{\nu}} \sum_{j=0}^{[\frac{t-a}{h}]} w_j^{(\nu)} \, f(t - j\,h) \tag{8} $$

where h is the time step, $[\cdot]$ stands for the integer part, $w_j^{(\nu)} = (-1)^j \binom{\nu}{j}$, with $w_0^{(\nu)} = 1$ and $w_j^{(\nu)} = \left(1 - \frac{\nu+1}{j}\right) w_{j-1}^{(\nu)}$ for $j > 0$, to reflect the limitations imposed by the finite memory word lengths in digital signal processors. Now consider T in place of h. This method corresponds to applying a PSE to the generating function $s = \left(\frac{1}{T}\right)(1 - z^{-1})$ defining the backward difference rule by means of the Euler's operator (Tenreiro Machado, 1997; Vinagre et al., 2000). The yielded discrete transfer function is: $G_{GL}(z) = \left(\frac{1}{T}\right)^{\nu} \text{PSE}\{(1-z^{-1})^{\nu}\}$.

Another possible generating function is $s = \left(\frac{2}{T}\right) \frac{1-z^{-1}}{1+z^{-1}}$, which defines the trapezoidal rule or the Tustin's operator (Chen and Moore, 2002; Chen et al., 2004; Vinagre et al., 2000, 2003). Other conversion rules are given by the Al-Alaoui's operator, i.e. a weighted combination of Euler's and Tustin's rules, or the Simpson's operator. However, all discretization schemes based on PSEs generate polynomials, then FIR filters, that show convergence with slower speed and in a narrower domain. This problem is overcome by combining a transformation rule with CFEs. Namely, the CFE is applied to the s-to-z operator defined by the selected generating function: $s^{\nu} = \left(\frac{2}{T}\right)^{\nu} \text{CFE}\left\{\left(\frac{1-z^1}{1+z^1}\right)^{\nu}\right\}$. Then the CFE is truncated to obtain a rational transfer function: $s^{\nu} \approx \left(\frac{2}{T}\right)^{\nu} \frac{A(z^{-1})}{B(z^{-1})}$, where the *order of the approximation* is specified by the degree of polynomials A and B and determines the accuracy, then the required memory space to store coefficients of the function A/B. Other direct discretization methods are based on the Muir's recursion (Chen et al., 2009), or on a Taylor series expansion of the Euler's operator followed by truncation (Tenreiro Machado, 2001), or on numerical integration schemes performing linear or quadratic interpolation (Tenreiro Machado, 1997).

Indirect discretization is based on a frequency-domain fitting in the s-domain followed by a discretization of the fit transfer function by one of the available operators like Tustin, Euler, Al-Alaoui, Simpson, etc. (Oustaloup et al., 2000; Podlubny et al., 2002). The indirect discretization method proposed in (Maione, 2011a) shows robustness to round-off and truncation errors occurring because of a finite word length in memory. In what follows, we focus on indirect discretization methods that lead to IIR filters. The non-integer order controllers in this form can be easily and directly implemented in microprocessor systems.

3.1. Approximation by the discretized Oustaloup's interpolation

For the purpose of this paper, one may consider the discrete-time implementation of the well-known and widely employed Oustaloup's recursive approximation technique at the basis of the CRONE control (Oustaloup, 1991, 1995; Oustaloup et al., 2000). It is a frequency-domain interpolation method that specifies the number n of zero-pole couples (i.e. the order of the approximation) in advance and is based on two parameters called recursive factors

$$ \alpha = \left(\frac{\omega_H}{\omega_L}\right)^{\frac{\nu}{n}} \qquad \eta = \left(\frac{\omega_H}{\omega_L}\right)^{\frac{1-\nu}{n}} \tag{9} $$

where ω_H and ω_L are the high- and low-transitional angular frequencies of the range $[\omega_L, \omega_H]$, in which the approximation is built by distributing zeros and poles around the unit gain frequency $\omega_u = \sqrt{\omega_L\,\omega_H}$ (e.g. $\omega_u = 1$ rad/s for sake of simplicity and $\omega_L = 0.01$ rad/s, $\omega_H = 100$ rad/s). The Oustaloup's s-domain approximant is

defined by

$$G_{Ous}(\nu, s) = \left(\frac{\omega_u}{\omega_H}\right)^{\nu} \prod_{k=-N}^{N} \frac{1 + s/\omega_{z_k}}{1 + s/\omega_{p_k}} \tag{10}$$

with $n = 2N + 1$, $N = \log(\omega_{p_N}/\omega_0)/\log(\alpha\,\eta)$, $\omega_{z_0} = \omega_u\,\alpha^{-0.5}$, $\omega_{p_0} = \omega_u\,\alpha^{0.5}$. The $2n$ singularities of the filter required to approximate a differentiator s^ν can be determined by the following formulas:

$$\omega_{z_1} = \omega_L\,\sqrt{\eta} \tag{11}$$

$$\omega_{p_i} = \omega_{z_i}\,\alpha \quad \text{for } i = 1, ..., n \tag{12}$$

$$\omega_{z_{i+1}} = \omega_{p_i}\,\eta \quad \text{for } i = 1, ..., n-1 \tag{13}$$

Then, it is immediate to obtain the relations $\omega_{z_{i+1}}/\omega_{z_i} = \omega_{p_{i+1}}/\omega_{p_i} = \alpha\,\eta > 1$. The final operation is the discretization by one of the available s-to-z transformation rules that provides the discrete transfer function $G_{Ous}(\nu, z)$. In particular, the discrete-time realization of s^ν can be easily obtained by applying the Tustin's bilinear transformation. This discretization rule may exhibit large errors in high frequency range. Then other discretization rules can be also considered, but they do not provide significant improvements if the sampling rate $1/T$ is high. However, Tustin's rule is better for low values of $1/T$.

An example. Consider three representative values of $\nu = 0.3, 0.5, 0.7$, with $n = 3$ zero-pole pairs only both to obtain an acceptable approximation and to reduce the complexity of the implementation. Let $T = 0.01$ s, which is a common value in real applications. The discrete transfer functions are the following:

$$G_{Ous}(0.3, z) = \frac{3.6137z^3 - 10.3572z^2 + 9.8765z - 3.1329}{z^3 - 2.6919z^2 + 2.3886z - 0.6967} \tag{14a}$$

$$G_{Ous}(0.5, z) = \frac{8.4476z^3 - 24.4973z^2 + 23.6558z - 7.6060}{z^3 - 2.6010z^2 + 2.2103z - 0.6094} \tag{14b}$$

$$G_{Ous}(0.7, z) = \frac{19.5331z^3 - 57.1436z^2 + 55.6929z - 18.0824}{z^3 - 2.4901z^2 + 1.9948z - 0.5047} \tag{14c}$$

where the form of transfer functions is useful to derive the input-output difference equation ready for a direct implementation of the fractional differentiator. Table 1 and Figure 1 show the zeros and poles of all the obtained approximations. All singularities of the Oustaloup's method are in the unit circle but some are very close to the point $(1, 0)$, namely the ones with the highest module. It is well-known that this result combined with memory limitation due to finite word length could provide problems for digital coding of the filter coefficients. The problem could be reduced by other approximation techniques based on partial fraction expansions or continued fraction expansions (e.g. (Charef, 2006; Maione, 2011a)) but a high number n of zeros and poles could be necessary ($n = 16$ in the examples shown in (Charef, 2006)), then high costs and computational requirements would arise. Moreover, a special care must be paid to sensitivity of the coefficients to truncation and round-off errors that affect efficiency of the real implementation. To this aim, using a partial fraction is a totally different solution and could allow a better discrete-time implementation code (less sensitive to the truncation of input/output signals and of the parameters of the controller) resulting from a continuous-time domain Oustaloup's approximation, as already presented in the literature. A dedicated and deeper investigation is postponed to future papers.

4. Controller implementation problems

Several digital control applications use hardware-in-the-loop devices, in which the control law is an algorithm that is pre-computed and stored in the device memory (Chen et al., 2009). The algorithm is defined by the coefficients

Table 1. Zeros and poles of rational approximations with $n = 3$ and $T = 0.01$ s

	ν	Zeros	Poles
$G_{Ous}(\nu, z)$	0.3	{0.9997, 0.9937, 0.8727}	{0.9993, 0.9843, 0.7083}
	0.5	{0.9998, 0.9954, 0.9048}	{0.9990, 0.9787, 0.6233}
	0.7	{0.9998, 0.9966, 0.9290}	{0.9986, 0.9711, 0.5204}

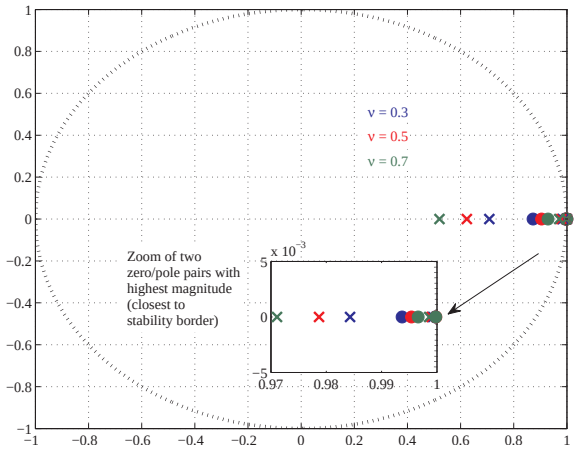

Fig. 1. Pole-zero maps of the discrete transfer function $G_{Ous}(\nu, z)$

in the IIR filter associated to the controller. The finite available memory restricts the number and accuracy of the coefficients in the real implementation. Moreover, the microprocessor speed must be taken into account. Then a good trade-off in realization must be achieved to guarantee a robust and efficient implementation. The lower is the number of coefficients, the faster is the computation and the less memory is occupied, but the higher is the approximation error and the higher are the errors due to sampling, quantization, and data representation.

It is important to stress the different complexity of the digital non-integer (fractional) and integer order control algorithms. They are both implemented by recurrent difference equations. But a digital non-integer order PI controller is realized by an equation with more coefficients (14 in the experiments of section 5). Then the difference equation in the non-integer case requires more computation time and more memory. Moreover, more coefficients must be numerically approximated by truncation or round-off. Then the implementation is more sensitive to relevant aspects like the selection of T and the quantization effects.

More in details, the implementation of the control law on a microcontroller involves the development of a source code including all the required operations. The first are the sampling and the analog-to-digital (AD) conversion, that includes the quantization of the error signal. Then the current time control signal is computed by linear combination of the digitalized errors in past time instants. Finally, there is a digital-to-analog (DA) conversion ending with the transmission of the control signal to the actuators.

The AD and DA conversions and the implementation of the control algorithm on a dedicated device introduce approximations determined by the finite resolution of the converters. In general, these approximations produce errors due to quantization, overflow, and round-off noise effects. Instead, the errors are negligible in higher-level digital signal processors employing double precision floating point representations.

A first analysis can be on the effect of the sampling period T. The selection of T depends on many factors and requirements: verifying the Shannon's theorem, keeping the closed-loop performance of the continuous-time

controlled behavior, finding a good trade-off between cost of implementation and the performance of the closed-loop system. In its turn, the cost of implementation depends on the computational load and the consequent required processing capacity, the speed and precision of converters, the amount of memory that is used to save parameters, coefficients, and variables with finite precision. The available hardware obviously influences the final choice. Another aspect regards the desired servo performance of the feedback system, but also robustness, limitation of the control effort, disturbance rejection, etc. To this aim, the selection of T must always be verified to avoid that sampling and quantization effects degrade the performance with respect to specifications.

Usually, rules of thumb suggest to start by choosing T in relation with the dominant open-loop time constant τ, for example by $\tau/10 \leq T \leq \tau/2$, and to adjust T by considering the desired performance. Anyway, the value of τ ensues from a model identification, while the real process behavior is a bit different. Then T is selected and tested by the real behavior and by taking into account the limitations of the employed hardware, in particular the encoder, and the computational load. More precisely, a too low sampling period implies several issues to be addressed.

a) *Cost and speed issues.* Fast sampling requires fast computations, then high-cost analog-to-digital (ADC) and digital-to-analog (DAC) converters and processing units. Then T must take into account the limitations posed by the available computing devices and must include all operations required by the digital non-integer order control algorithm. Moreover, the more complex is the realization of such algorithm, i.e. the more coefficients define the associated rational transfer function, the more time is necessary to compute the controller output, given its error input. (Higher complexity also means more required memory, while computing devices typically have a finite limited number of bits at disposal.) A trade-off is required between the speed of computation and the complexity of the control law that is actually implemented. In particular, note that if T is comparable to the required computation time, then the output produced by the controller and the DAC converter could be not available at the expected discrete time instant, and a delay approximately equal to T would affect the control loop and degrade performance.
b) *Interaction with converters and memory.* An unnecessary reduction of T does not imply performance improvements. Namely, if T is too small, the variations of the input signal between two sampling times may be smaller than the resolution of the converters or than the precision used to represent data in the microcontroller memory. In such case, the signal variations are not resolved and there is a loss of information. Then, in general, the reduction of T should be accompanied by an increase of the memory word length, as well as by an increase of the converters resolution, so that the difference between two successive values of the signal can be resolved and properly represented.
c) *Numerical issues.* These may occur because of the finite precision that is used for representing floating point numbers and for computing the control law: fast sampling would increase too much the truncation and round-off errors and the controller provides an output too different from the required one.
d) *Stability issues.* Increasing the sampling rate moves the zeros and poles of the discrete transfer function close to each other and gathers them around the critical point $(1, 0)$. Then, the zeros and poles are highly sensitive to changes in coefficients and could even make the implementation unstable.

As an example, the effect of changes of T is analyzed by considering the values 0.001, 0.002, 0.005, 0.02 and 0.04 s with respect to the previous value $T = 0.01$ s. The effect on the positions of zeros and poles is illustrated in Table 2. $\Delta_z\%$ and $\Delta_p\%$ are defined as the percentage relative variations of zeros and poles with respect to the values obtained for $T = 0.01$ s. For $T = 0.001$ s (or 0.002 s), the singularities of $G_{Ous}(\nu, z)$ get very close to or even coincide with $(1, 0)$. Moreover, a large variation of the third zero and, above all, of the third pole is observed when T increases up to 0.04.

Table 2. Variations of zeros and poles of $G_{Ous}(\nu, z)$ with T

ν	T	Zeros			$\Delta_z\%$			Poles			$\Delta_p\%$		
	0.001	1.0000	0.9994	0.9865	0.03	0.57	13.04	0.9999	0.9984	0.9664	0.06	1.43	36.44
	0.002	0.9999	0.9987	0.9732	0.02	0.51	11.51	0.9999	0.9968	0.9340	0.06	1.28	31.85
0.3	0.005	0.9999	0.9969	0.9343	0.01	0.32	7.05	0.9996	0.9921	0.8427	0.04	0.80	18.97
	0.02	0.9994	0.9875	0.7607	-0.03	-0.63	-12.94	0.9985	0.9688	0.4909	-0.07	-1.57	-30.69
	0.04	0.9988	0.9751	0.5725	-0.09	-1.88	-34.40	0.9971	0.9386	0.1884	-0.22	-4.65	-73.40
	0.001	1.0000	0.9995	0.9900	0.02	0.41	9.42	0.9999	0.9978	0.9546	0.09	1.95	53.15
	0.002	1.0000	0.9991	0.9802	0.02	0.37	8.34	0.9998	0.9957	0.9113	0.08	1.74	46.21
0.5	0.005	0.9999	0.9977	0.9512	0.01	0.23	5.13	0.9995	0.9893	0.7921	0.05	1.08	27.08
	0.02	0.9996	0.9908	0.8182	-0.02	-0.46	-9.57	0.9980	0.9578	0.3660	-0.10	-2.13	-41.28
	0.04	0.9991	0.9816	0.6667	-0.06	-1.38	-26.32	0.9960	0.9174	0.0372	-0.30	-6.26	-94.04
	0.001	1.0000	0.9997	0.9927	0.02	0.31	6.86	0.9999	0.9971	0.9388	0.13	2.68	80.40
	0.002	1.0000	0.9993	0.9854	0.01	0.27	6.07	0.9997	0.9942	0.8813	0.11	2.37	69.36
0.7	0.005	0.9999	0.9983	0.9639	0.01	0.17	3.75	0.9993	0.9855	0.7275	0.07	1.48	39.81
	0.02	0.9997	0.9932	0.8630	-0.02	-0.34	-7.11	0.9973	0.9431	0.2263	-0.14	-2.89	-56.52
	0.04	0.9994	0.9864	0.7435	-0.05	-1.02	-19.97	0.9946	0.8893	-0.1158	-0.41	-8.42	-122.25

It is also well-known that real implementation of fractional-order controllers on digital processing units pose some practical problems (Chen et al., 2009). Here it is remarked that an important, yet obvious, difference between the non-integer-order and the integer-order case is the need to approximate irrational fractional operators s^ν in the discrete domain, whereas a standard integer-order digital control algorithm does not require this step. Then it is expected that the implementation is more sensitive to changes in T in the non-integer than in the integer case (the same can be expected for variations of the converters resolution).

In synthesis, the highest possible values of T that verify the specifications and allow an acceptable approximation of the fractional operator should be used. To this aim, it is common to consider the desired servo performance of the closed-loop system, i.e. the bandwidth specification ω_B, and use a rule of thumb like $10\,\omega_B \leq 2\pi/T \leq 30\,\omega_B$ (Åström and Wittenmark, 1997).

Another analysis can focus on the AD-DA conversion. The AD conversion associates a finite number of levels of equal amplitude to the input signal by means of quantization, and then converts the quantized signal into a numerical coded value that is suitable for the computing device. The DAC associates a finite number of levels of equal amplitude to the numerical coded value that is computed by the control algorithm, and then generates a continuous signal to drive actuators.

The converter resolution, i.e. the lowest representable value other than 0, depends on the number of available quantization levels during conversion. This value is determined by the number of bits used in the conversion. Therefore, since the quantization error is defined by the difference between the input signal and the corresponding converted signal, it comes out that the converter resolution directly affects the control system performance, by determining a steady-state error and limit cycles with permanent oscillations.

It is expected that the effect of changes in the resolution of DAC and ADC converters is very significant when using a digital non-integer order controller. Reducing the number of bits devoted to conversion should determine more negative effects in this case than what could be observed with an integer-order controller. Namely, just like for sampling, the quantization error and the approximation error (which is not present in the integer case) combine with one another.

5. Experimental results

This section reports some experimental tests that were performed on a real laboratory equipment employing a DC-motor. Figure 2 shows all the devices required for the experiments. In details, the experimental set-up is

composed by: an EMG49 dc gear motor 24v with an intrinsic built-in encoder of 245 PPR (pulses per revolution); a rapid prototyping dSPACE 1103 board to acquire the converted reference and feedback signals, to store and compute the control law and to generate a duty-cycle command; a PWM generator; an L298 dual full-bridge bidirectional driver.

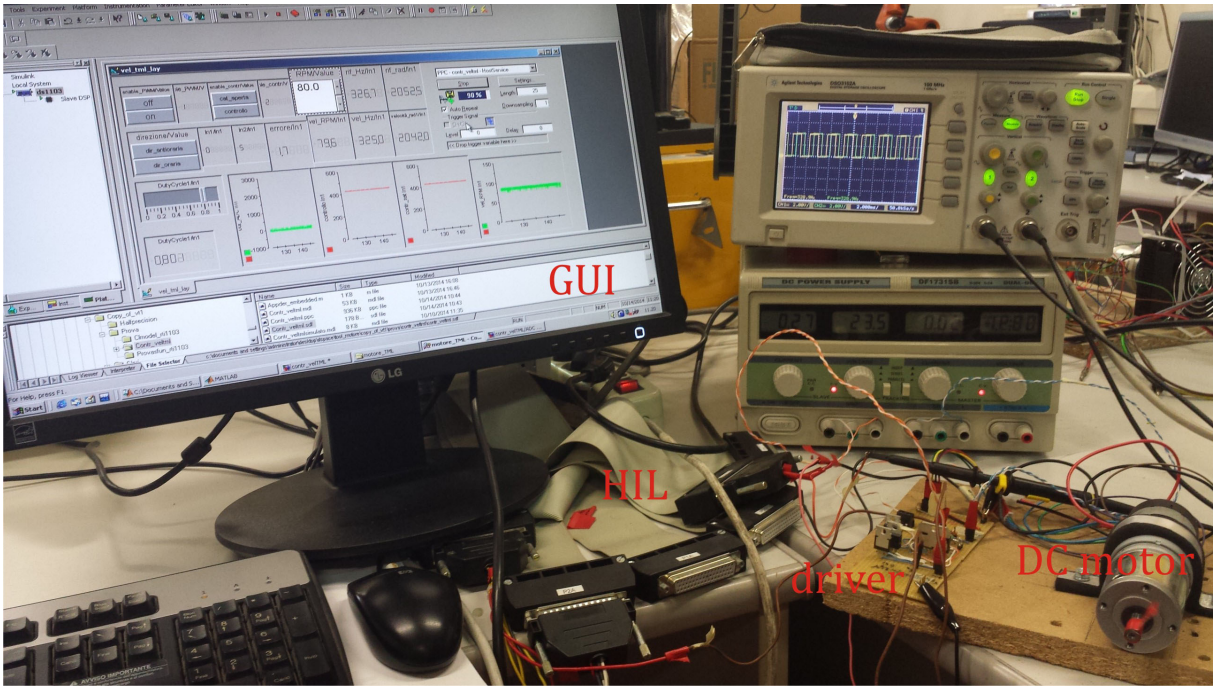

Fig. 2. Experimental platform

Figure 3 shows the block diagram representing the control system. Note that a Graphical User Interface (GUI) and the Real-Time Interface (RTI) control system are implemented in the *ControlDesk* software environment. In details, the RTI control system is the highlighted part which communicates by ADC and DAC converters. The last allow to change the resolution then the number of quantized levels to represent the input signal. The set-point is provided in RPM units to the RTI, which uses the sign of the set-point to establish the motor (clockwise or counterclockwise) direction of rotation by the two driver inputs. The allowed configurations at the direction input pins are always opposite, except for the shutdown condition when the inputs are both 0. Therefore, the upper branch on the block diagram only sets the direction of the motor. The system automatically detects the sign of the RPM set-point and provides the two associated voltage drive signals (0V-5V or 5V-0V) to set the correct rotor direction. The control error is the difference between the set-point and the measured speed in RPM units. The `Transmitter` block makes the RPM measured feedback signal available and shows that this speed depends on the number of impulses per second measured by the incremental encoder. Finally note that the controller output is normalized with respect to the maximum motor speed, then a duty-cycle signal between 0 and 1 feeds a `PWM generator` block.

The motor speed is controlled by the FOPI controllers designed by the method proposed in section 2.

The identified plant model is a FOPTD system characterized by the parameters $K = 1.6862$, $\tau = 0.0583$ s, and $L = 0.025$ s. The performance specifications are the gain crossover angular frequency $\omega_C = 15$ rad/s (then $u_C = \omega_C \tau = 0.8745$), i.e. a bandwidth $19.5 \leq \omega_B \leq 25.5$ rad/s (then $1.1368 \leq u_B \leq 1.4867$) by the Maciejowski's relation cited in section 2, and a phase margin $PM_s = \pi/3$. Then the design procedure leads to

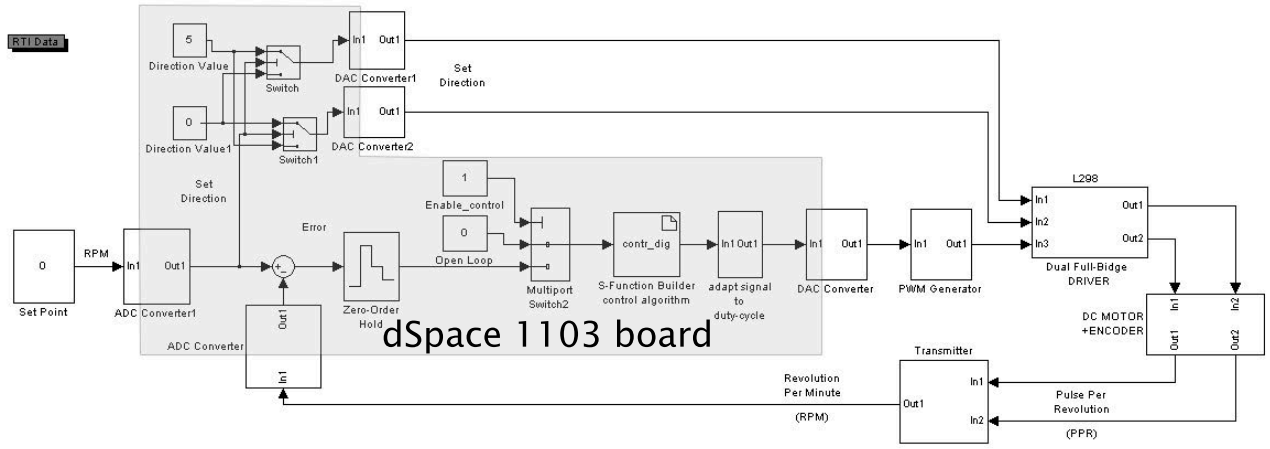

Fig. 3. The control system block diagram

the FOPI controller transfer function that takes the following form:

$$G_c(s) = 0.8081 + \frac{28.3334}{s^{1.3333}}.$$

As previously recalled, the approximation method used to implement the digital controller is based on the Oustaloup's recursive technique. In particular, an approximation order $n = 5$ is used.

Table 3 indicates the coefficients, zeros, and poles of the discrete transfer function representing the approximation of the fractional operator $s^{1/3}$, which is the non-integer component of $s^{4/3}$. The table also specifies the same information obtained for the discretization G_{cz} of the FOPI controller $G_c(s)$ specified above. Note that, for increasing values of T, the zeros and poles move close to each other and gather around the critical point $(1, 0)$.

Table 3. Variations with T of zeros and poles of the transfer functions G_{Oz} and G_{cz} respectively representing the discretized Oustaloup's approximation of the fractional operator $s^{1/3}$ and the digital controller ($n = 5$, $\nu = 4/3$)

	T	Coefficients of numerator (top) & denominator (bottom)						Zeros (top) & poles (bottom)					
G_{Oz}	0.01	4.0940	-19.2027	35.9294	-33.5112	15.5751	-2.8846	0.9998	0.9988	0.9927	0.9546	0.7445	
		1.0000	-4.4758	7.9466	-6.9840	3.0318	-0.5185	0.9997	0.9978	0.9865	0.9178	0.5741	
	0.02	3.7253	-16.5437	29.1069	-25.3085	10.8447	-1.8247	0.9996	0.9977	0.9854	0.9113	0.5470	
		1.0000	-4.1079	6.5701	-5.0592	1.8398	-0.2427	0.9993	0.9957	0.9732	0.8420	0.2977	
	0.04	3.2497	-13.1839	20.7486	-15.6248	5.4911	-0.6806	0.9993	0.9953	0.9710	0.8301	0.2612	
		1.0000	-3.6047	4.8077	-2.7748	0.5456	0.0262	0.9986	0.9914	0.9471	0.7072	-0.0395	
G_{cz}	0.01	0.8427	-4.7185	11.0020	-13.6728	9.5518	-3.5566	0.5514	0.9997	0.9978	0.9862	$0.8994 \pm 0.0722i$	0.8171
		1.0000	-5.6905	13.4667	-16.9617	11.9899	-4.5090	0.7046	1.0000	0.9998	0.9988	0.9927	0.9546	0.7445
	0.02	0.8841	-4.6330	10.0894	-11.6884	7.5972	-2.6267	0.3773	0.9993	0.9957	0.9726	$0.8039 \pm 0.1308i$	0.6648
		1.0000	-5.4409	12.2543	-14.6071	9.7048	-3.4010	0.4898	1.0000	0.9996	0.9977	0.9854	0.9113	0.5470
	0.04	0.9824	-4.5405	8.6473	-8.6900	4.8619	-1.4349	0.1738	0.9986	0.9914	0.9459	$0.6300 \pm 0.2163i$	0.4259
		1.0000	-5.0570	10.4418	-11.1929	6.4978	-1.8992	0.2094	1.0000	0.9993	0.9953	0.9710	0.8301	0.2612

The first test analyzes the effect of the sampling period T on the step response and on the corresponding time evolution of the control variable, i.e. the controller output. The considered values of T are 0.01, 0.02, and 0.04 s. (In this way, the relation $10\,\omega_B \leq 2\pi/T \leq 30\,\omega_B$ (Åström and Wittenmark, 1997) is satisfied.) Lower values of T are not considered for the reasons specified before in section 4. Moreover, other problems arise from the interaction between the sampling process and the updating of the speed measurement by the

incremental encoder. Since the speed is obtained by the pulse increment in a sampling period, the accuracy of speed measurement depends on the encoder inherent PPR resolution. Moreover, the accuracy depends on the speed, because the higher is the speed, the higher is the number of counted pulses. If T is too low, then the pulse increment can be too low to provide a reliable measurement, so that the speed oscillates around an average value in adjacent sampling periods. In synthesis, a better resolution would be required to decrease T. However, the inaccuracy in the speed measurement also affects the control action during rising transients. For low speed values, the measurement of speed error is quite inaccurate and the resulting control action is not optimal. This effect is reduced for high speed values, as it is evident by comparing Figures 4 (speed reference of 40 RPM) and 5 (speed reference of 98 RPM). Namely, the step response overshoot is much higher for the lower speed reference and is considerably decreased for the higher speed reference (see also the compared performance in Table 4).

The resolutions of the ADC and DAC converters, respectively indicated by ADCRes and DACRes, are both kept constant to the same value: ADCRes = DACRes = 16 bits (Figures 4-5, top) or ADCRes = DACRes = 8 bits (Figures 4-5, down). These values are usually found in commercial devices for industrial applications, then reflect real conditions. Resolutions of lower values are not realistic and would downgrade the converter performance too much.

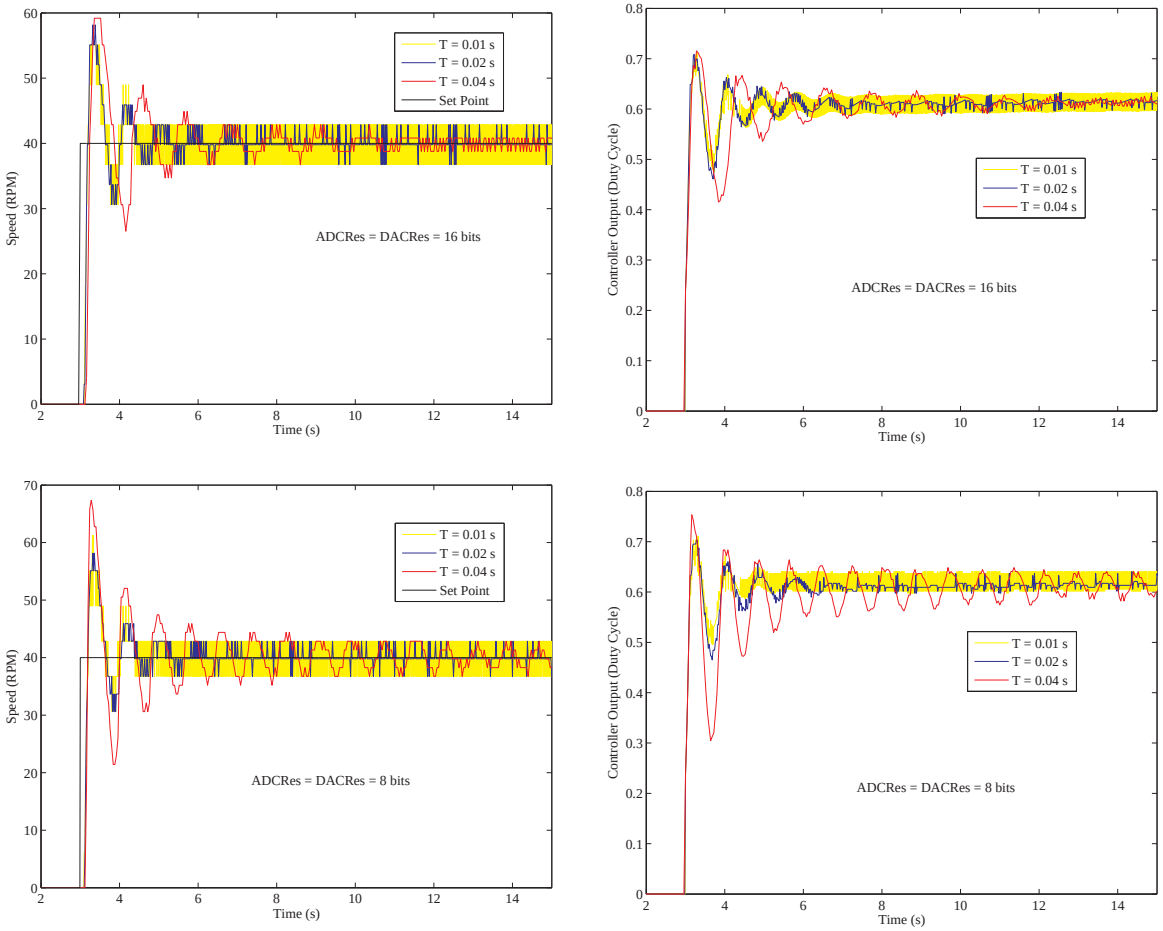

Fig. 4. Step responses (left) and control variables (right) for a 40 RPM set-point and for $T = 0.01, 0.02, 0.04$ s: with ADCRes = DACRes = 16 bits (top), with ADCRes = DACRes = 8 bits (down)

Figure 4 shows that, for both values of the converters resolution, increasing the sampling period implies a growth of the overshoot and settling time in the step response, whereas the rise time is comparable in all

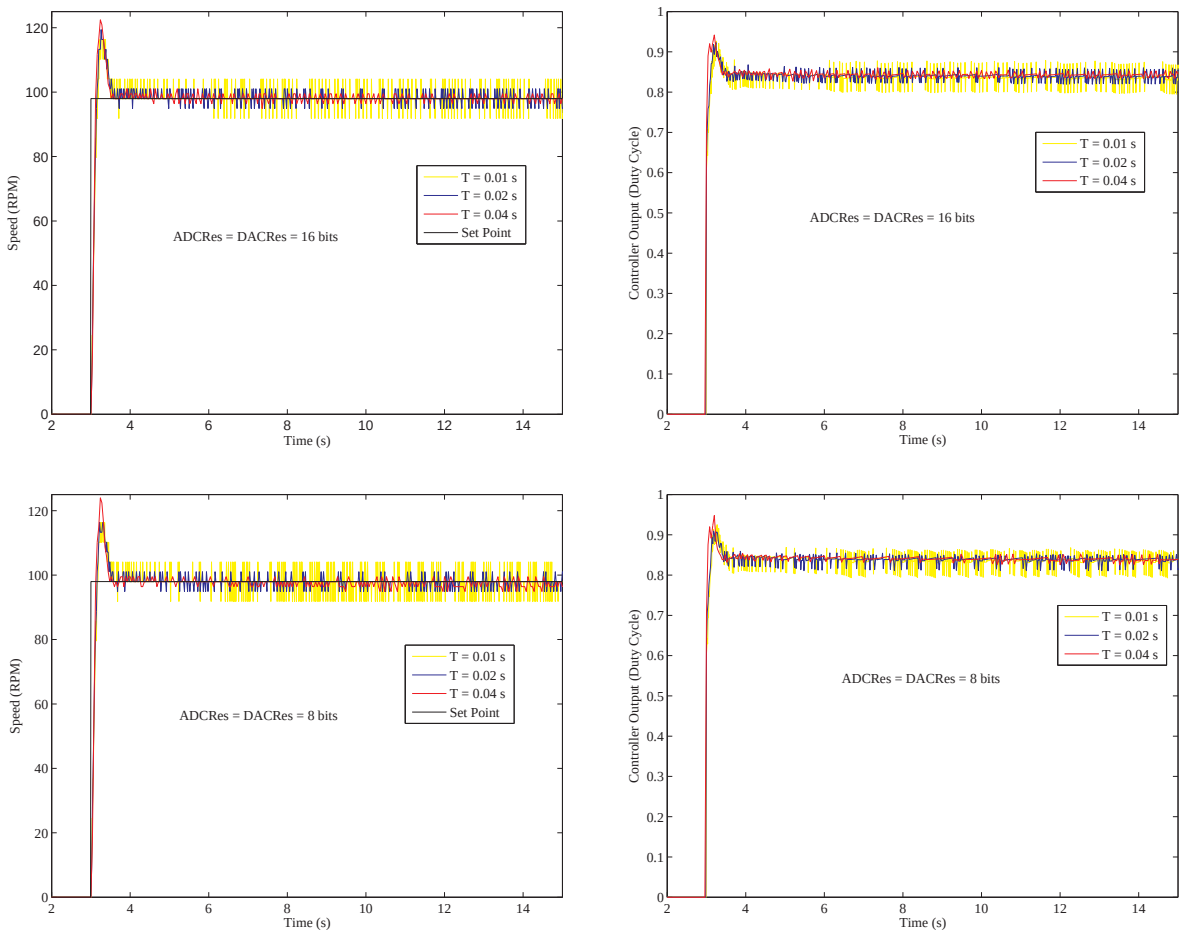

Fig. 5. Step responses (left) and control variables (right) for a 98 RPM set-point and for $T = 0.01$, 0.02, 0.04 s: with ADCRes = DACRes = 16 bits (top), with ADCRes = DACRes = 8 bits (down)

conditions (see Table 4). This worsening is more evident with a lower resolution of 8 bits (Figure 4, lower part). However, even if an improvement can be achieved by increasing the sampling rate, the control system can't afford a lower T due to hardware limitations (in particular of the built-in encoder, see the case $T = 0.01$ s).

Moreover, steady-state high frequency oscillations of limited amplitude and a steady-state offset occur because of quantization effects and of the previously mentioned problems with the encoder resolution. More or less, they are the same for all the resolutions even if they are a bit less significant with $T = 0.04$ s and 16 bits. In this last case, the steady-state response has an offset which is reduced with respect to other cases. However, the steady-state oscillations do not have a significant impact on the overall behavior of the controlled system. In any case, reducing T does not improve the response (see what happens for $T = 0.01$ s).

If the control variable is analyzed, then it is verified that a higher T implies initial oscillations of higher amplitude, so that a higher control effort is necessary during transients with high T. But a lower T induces more important steady-state oscillations. Then a trade-off can be given by $T = 0.02$ s.

The second analysis considers the effect of the variation of the converters resolution while maintaining a constant sampling period ($T = 0.01$ or 0.02 or 0.04 s). Again, the step responses and the control variables are shown in different cases for a 40 RPM set-point. The plots for the case of 98 RPM are not shown for sake of space but the results can be checked from Table 4.

Table 4. Characteristics of the step response in all the experimented cases

Case no.	Set-point (RPM)	T (s)	ADCRes (bit)	DACRes (bit)	Overshoot (%)	Rise Time (s)	Settling Time (s) (10% of set-point)
1	40	0.01	16	16	37.7	0.08	1.20
2	40	0.01	12	12	37.7	0.08	1.20
3	40	0.01	8	8	53.1	0.08	1.20
4	40	0.02	16	16	45.4	0.08	4.08
5	40	0.02	12	12	45.4	0.08	4.00
6	40	0.02	8	8	45.4	0.08	3.02
7	40	0.04	16	16	47.9	0.08	8.40
8	40	0.04	12	12	60.7	0.08	11.00
9	40	0.04	8	8	68.4	0.08	-
10	98	0.01	16	16	18.7	0.13	0.40
11	98	0.01	12	12	18.7	0.13	0.40
12	98	0.01	8	8	18.7	0.13	0.40
13	98	0.02	16	16	21.8	0.13	0.40
14	98	0.02	12	12	21.8	0.13	0.40
15	98	0.02	8	8	18.7	0.13	0.40
16	98	0.04	16	16	24.9	0.13	0.40
17	98	0.04	12	12	24.9	0.13	0.40
18	98	0.04	8	8	26.5	0.13	0.40

Figure 6 in the left top part shows that, with $T = 0.01$ s, there is no significant difference between using higher or lower resolution (apart an increase in the overshoot), especially in the steady-state oscillations associated to limit cycles. If T is increased, the benefits of higher resolution are evident (see Figure 6 for $T = 0.04$ s in the left lower part). Then increasing the resolution allows better performance, especially by looking at the control variable. Similar considerations hold for the responses to a 98 RPM reference input.

Finally, note that the computational cost of the fractional order digital control algorithm obviously depends on the number of operations required. The effect on the processing time required by the algorithm can be analytically determined by using the clock of the employed HIL.

6. Conclusion

In this work, the problem of implementing non-integer order digital control of electro-mechanical systems was addressed. The controlled plants are modeled as first-order plus time-delay systems of integer order, namely this model is widely used to represent electro-mechanical systems subject to speed regulation. Then fractional-order PI controllers are considered because they provide a configuration a very close to the standard PI controllers that are traditionally used for this type of plants. In this way, it is possible to explore the possibility to improve the control loops of many industrial, mechatronics or automotive feedback systems. Moreover, the FOPI controller is designed by a strategy that resembles loop-shaping techniques that are well-known in tuning of PI/PID controllers (e.g. the symmetrical optimum method). A robust FOPI controller is designed and optimality is pursued for the closed-loop system. The rational transfer function that realizes the FOPI controller is obtained by the well-known Oustaloup's recursive approximation. The digital version of the controller is obtained by the Tustin's discretization rule. In particular, the paper focuses on the operational issues in implementation of the control algorithms. Namely, sampling, quantization, and conversion must be taken into account. A detailed analysis of experimental results on a real platform show how performance gets worse if these limitations are taken into account with respect to the ideal conditions provided by the theoretical designed controller. However, the robustness and dynamic performance are satisfactory even if large variations are considered, owing to the benefits offered by fractional-order controllers.

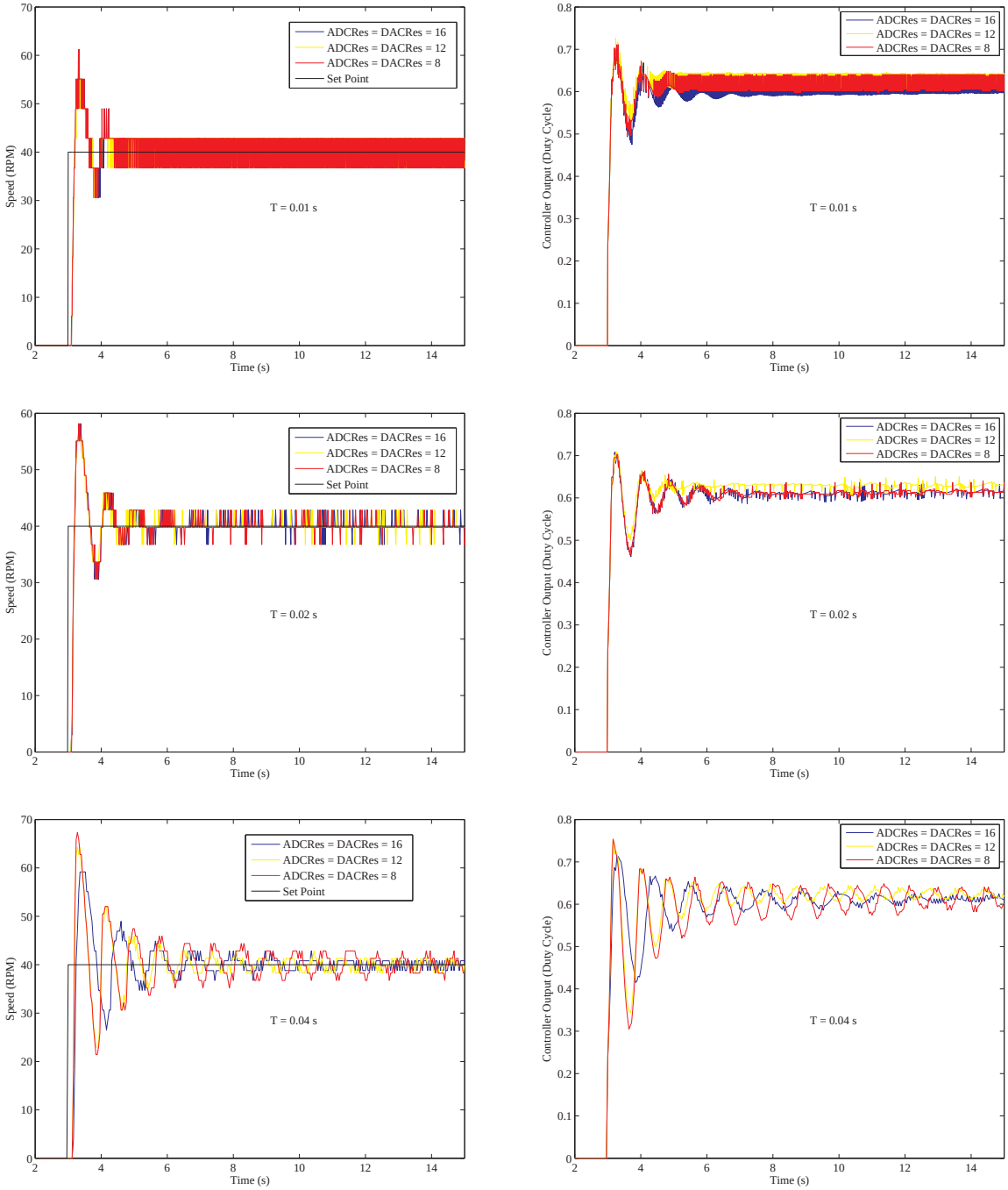

Fig. 6. Step responses (left) and control variables (right) with different resolutions and $T = 0.01$ s (top figures), $T = 0.02$ s (middle figures), $T = 0.04$ s (down figures)

References

Åström KJ and Hägglund T (1995) *PID Controllers: Theory, Design, and Tuning*. Research Triangle Park, NC: Instrument Society of America.

Åström KJ and Wittenmark B (1997) *Computer-Controlled Systems – Theory and Design*, 3rd ed. Upper Saddle River, NJ: Prentice Hall.

Barbosa RS, Tenreiro Machado JA and Silva MF (2006). Time domain design of fractional differintegrators using least squares. *Signal Processing* 86(10): 2567–2581.

Caponetto R and Dongola G (2008) Field programmable analog array implementation of non integer order $PI^{\lambda}D^{\mu}$ controller. *Journal of Computational and Nonlinear Dynamics* 3(2): 1–8.

Caponetto R, Dongola G and Fortuna L (2008) Switched capacitors implementation of fractional differ-integral operator. In: *3rd IFAC Workshop on Fractional Differentiation and its Applications, FDA 2008*, Ankara, Turkey, 5-7 November 2008.

Caponetto R, Dongola G, Fortuna L, et al. (2010) *Fractional Order Systems: Modeling and Control Applications*, World Scientific Series on Nonlinear Science. Series A, vol. 72, ch. 4. Singapore: World Scientific.

Caponetto R, Dongola G, Maione G, et al. (2014) Integrated technology fractional order proportional-integral-derivative design. *Journal of Vibration and Control* 20(7): 1066–1075.

Charef A (Nov. 2006) Analogue realisation of fractional-order integrator, differentiator and fractional $PI^{\lambda}D^{\mu}$ controller. *IET Control Theory & Applications* 153(6): 714–720.

Chen YQ and Moore KL (2002) Discretization schemes for fractional-order differentiators and integrators. *IEEE Transactions on Circuits and Systems I. Fundamental Theory and Applications* 49(3): 363–367.

Chen YQ, Petráš I and Xue D (2009) Fractional order control - A tutorial. In *Proc. American Control Conf*, Hyatt Regency Riverfront, St. Louis, MO, USA, June 10-12, 1397–1411.

Chen YQ and Vinagre BM (2003) A new IIR-type digital fractional order differentiator. *Signal Processing* 83: 2359–2365.

Chen YQ, Vinagre BM and Podlubny I (2004) Continued fraction expansion approaches to discretizing fractional order derivatives. An expository review. *Nonlinear Dynamics* 38(1-2): 155–170.

Lino P and Maione G (2013) Loop-shaping and easy tuning of fractional-order proportional integral controllers for position servo systems. *Asian Journal of Control* 5(3): 1–10.

Maciejowski, JM (1989) *Multivariable Feedback Design*. Wokingham, UK: Addison-Wesley.

Maione G (2006) Concerning continued fractions representation of noninteger order digital differentiators. *IEEE Signal Processing Letters* 13(12): 725–728.

Maione G (2008) Continued fractions approximation of the impulse response of fractional order dynamic systems. *IET Control Theory and Applications* 2(7): 564–572.

Maione G (2011a) High-speed digital realizations of fractional operators in the delta domain. *IEEE Transactions on Automatic Control* 56(3): 697–702.

Maione G (2011b) Conditions for a class of rational approximants of fractional differentiators/integrators to enjoy the interlacing property. In: Bittanti S, Cenedese A, Zampieri S (eds) *Proc 18th IFAC World Congress (IFAC WC 2011)*, Università Cattolica del Sacro Cuore, Milan, Italy, Aug. 28 - Sept. 2, IFAC Proceedings on line: Vol. 18, Part 1, 13984–13989.

Maione G (June 2013) On the Laguerre rational approximation to fractional discrete derivative and integral operators. *IEEE Transactions on Automatic Control* 58(6): 1579–1585.

Maione G and Lino P (2007) New tuning rules for fractional PI^{α} controllers. *Nonlinear Dynamics* 49(1-2): 251–257.

Monje CA, Vinagre BM, Feliu V, Chen YQ (2008) Tuning and autotuning of fractional order controllers for industry applications. *Control Engineering Practice* 16: 798–812.

Oldham KB and Spanier J (1974) *The Fractional Calculus*. New York: Academic Press.

Oustaloup A (1991) *La Commande CRONE. Commande Robuste d'Ordre Non Entière*. Paris: Editions Hermès.

Oustaloup A (1995) *La Dérivation Non Entière: Théorie, Synthèse et Applications*. Paris: Editions Hermès.

Oustaloup A, Levron F, Mathieu B, et al. (Jan. 2000) Frequency-band complex noninteger differentiator: Characterization and synthesis. *IEEE Transactions on Circuit and Systems - I: Fundamental Theory and Applications* 47(1): 25–39.

Petráš I (2012) Tuning and implementation methods for fractional-order controllers. *Fractional Calculus and Applied Analysis* 15(2): 282–30.

Podlubny I (1999a) Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers. *IEEE Transactions on Automatic Control* 44(1): 208–214.

Podlubny I (1999b) *Fractional Differential Equations*. San Diego, CA: Academic Press.

Podlubny I, Petráš I, Vinagre BM, et al. (2002) Analogue realizations of fractional-order controllers. *Nonlinear Dynamics* 29(1-4): 281–296.

Samadi S, Ahmad MO and Swamy MNS (2004) Exact fractional-order differentiators for polynomial signals. *IEEE Signal Processing Letters* 11(6): 529–532.

Stoer J and Bulirsch R (2002) *Introduction to Numerical Analysis*, third ed.. Berlin, New York: Springer-Verlag.

Tenreiro Machado JA (1997) Analysis and design of fractional-order digital control systems. *Journal of Systems Analysis Modeling Simulation* 27: 107–122.

Tenreiro Machado JA (2001) Discrete-time fractional-order controllers. *Fractional Calculus and Applied Analysis* 4(1): 47–66.

Tseng C-C (2001) Design of fractional order digital FIR differentiators. *IEEE Signal Processing Letters* 8(3): 77–79.

Vinagre BM, Podlubny I, Hernandez A, et al. (2000) Some approximations of fractional order operators used in control theory and applications. *Fractional Calculus and Applied Analysis* 13(3): 231–248.

Vinagre BM, Chen YQ and Petráš I (2003) Two direct Tustin discretization methods for fractional-order differentiator/integrator. *Journal of the Franklin Institute* 340(5): 349–362.