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 $G_p(s) = \frac{K}{1+Ts}\, e^{-\tau s}$, where K is the static gain, T is the dominant time constant, and τ 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. 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 $G(j\omega) = \frac{K\,K_I\,[1+T_I(j\omega)^\nu]}{(j\omega)^\nu\,[1+T(j\omega)]}\,e^{-j\omega\tau}$. Using a normalized frequency $u = \omega\,T$ and rearranging the frequency response yields:

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

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

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

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

Now, an approximation of an optimal feedback system with unitary closed-loop gain is achieved in a specified bandwidth, say $u_B = \omega_B\,T$, which is determined according to the desired range for input-output tracking. u_B is chosen higher than the plant bandwidth 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 $u_C \in [\frac{u_B}{1.7}, \frac{u_B}{1.3}]$, 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 $PM = \arctan\left(\frac{T_I\,(\frac{u_C}{T})^\nu\,S}{1+T_I\,(\frac{u_C}{T})^\nu\,C}\right) - \arctan(u_C) - \theta - \frac{u_C\tau}{T} + \pi$, the controller parameter T_I is selected to exactly obtain $PM = \pi - \theta$. The closed-form solution is:

$$T_I = \left(\frac{T}{u_C}\right)^\nu\,\frac{u_C + \tan(\frac{u_C\tau}{T})}{S - u_C\,C - (C + u_C\,S)\,\tan(\frac{u_C\tau}{T})}. \tag{4}$$

Moreover, imposing $PM = \pi - \theta \equiv PM_s$ provides a strict relation between phase margin and fractional order, i.e. $PM_s = (2-\nu)\,\pi/2$, then it follows:

$$\nu = 2 - 2\,PM_s/\pi \tag{5}$$

which gives another closed-form formula to obtain ν from the specified phase margin. Since the plant does not include an integer order integrator, (5) provides $\nu > 1$ to include it in $G(s)$.

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

$$K_I = \frac{1}{K}\,\left(\frac{u_C}{T}\right)^\nu\,\sqrt{\frac{1+u_C^2}{1+2\,T_I\,(\frac{u_C}{T})^\nu\,C+T_I^2\,(\frac{u_C}{T})^{2\nu}}} \tag{6}$$

3. Controller realization: The approximation problem

The non-integer order controllers are often misnamed fractional order controllers because the differential or integral action could be, in a particular case, of fractional order. However, the control actions can be in general of any non-integer order (real or even complex). Then the key feature is that controllers are characterized by irrational transfer functions, as shown by the general structure of the CRONE controller (Oustaloup, 1991) or by the $PI^{\lambda}D^{\mu}$ controller (Podlubny, 1999a). Namely, the basic operators are non-integer powers of the s variable. 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, it is very important to approximate the fractional operator s^{ν}, with $\nu \in R$ and $0 < |\nu| < 1$, by an almost equivalent rational transfer function that exhibits the lowest possible approximation error with respect to s^{ν}, for example by considering the frequency response. More in details, 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. Moreover, the 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 peculiar choice is guaranteed by the approximation methods shown in (Maione, 2011b, 2013). The infinite-dimensional operator s^{ν} is then approximated by a ratio of two polynomials, which is called *approximant*, by applying one of the methods 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 methods (e.g. see (Oustaloup, 1995; Oustaloup et al., 2000)) and fitting techniques, one may use methods derived from 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 method from all aspects. To synthesize, the approximation methods provide analog or digital filters, many times in the form of ratios of two polynomials in the frequency variable s or z. In all cases, the coefficients of the filters depend on (or are related to) the non-integer exponent ν. The coefficients of the numerator and denominator polynomials in an approximant, then the zeros and poles of this function, depend on 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 continued fraction expansion (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):

$$ {}_aD_t^{\nu} f(t) = \lim_{h \to 0} \frac{1}{h^{\nu}} \sum_{j=0}^{[\frac{t-a}{h}]} (-1)^j \binom{\nu}{j} f(t - j\,h) \tag{7} $$

where h is the time step, $[\cdot]$ stands for the integer part and $\binom{\nu}{j} = \frac{\Gamma(\nu+1)}{\Gamma(j+1)\,\Gamma(\nu-j+1)}$, with $\Gamma(\cdot)$ defining the Euler's *Gamma* function. This definition reflects the limitations due to finite memory lengths in real digital signal

processors. An alternative definition is

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

where $w_j^{(\alpha)} = (-1)^j \binom{\nu}{j}$, with $w_0^{(\alpha)} = 1$ and $w_j^{(\alpha)} = \left(1 - \frac{\alpha+1}{j}\right) w_{j-1}^{(\alpha)}$ for $j > 0$. 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: $G_{GL}(z) = \left(\frac{1}{T}\right)^{\nu}$ PSE$\{(1 - z^{-1})^{\nu}\}$. If the short-memory principle is taken into account (Podlubny, 1999b), then the finite memory length L determines the following discrete transfer function:

$$ G_{GLsh}(z) = \left(\frac{1}{T}\right)^{\nu} z^{-[L/T]} \sum_{j=0}^{[L/T]} w_j^{(\alpha)} \, z^{[L/T]-j} \tag{9} $$

which corresponds to the time-domain approximation

$$ {}_{(k-L/T)}D_{kT}^{\nu} f(t) \approx \left(\frac{1}{T}\right)^{\nu} \sum_{j=v}^{k} w_j^{(\alpha)} \, f_{k-j} \tag{10} $$

with $v = 0$ for $k < (L/T)$ and $v = k - (L/T)$ for $k > (L/T)$.

Another possible generating function is $s = \psi(z^{-1}) = \left(\frac{2}{T}\right) \frac{1 - z^{-1}}{1 + z^{-1}}$, 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: $s^{\nu} = \left(\frac{2}{T}\right)^{\nu}$ CFE $\left\{\left(\frac{1 - z^1}{1 + z^1}\right)^{\nu}\right\}$. The final step is to truncate the CFE in order 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 a particular 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 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

$$\alpha = \left(\frac{\omega_H}{\omega_L}\right)^{\frac{\nu}{n}} \quad \eta = \left(\frac{\omega_H}{\omega_L}\right)^{\frac{1-\nu}{n}} \tag{11}$$

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

$$G_{Ous}(\nu, s) = \left(\frac{\omega_u}{\omega_H}\right)^{\nu} \prod_{k=-N}^{N} \frac{1 + s/\omega_{z_k}}{1 + s/\omega_{p_k}} \tag{12}$$

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:

$$\omega_{z_1} = \omega_L\,\sqrt{\eta} \tag{13}$$

$$\omega_{p_i} = \omega_{z_i}\,\alpha \quad \text{for } i = 1, ..., n \tag{14}$$

$$\omega_{z_{i+1}} = \omega_{p_i}\,\eta \quad \text{for } i = 1, ..., n-1 \tag{15}$$

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 an approximation for 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 by the Oustaloup's method, $G_{Ous}(\nu, z)$ are the following:

$$G_{Ous}(0.3, z) = \frac{3.6137z^3 - 10.3572z^2 + 9.8765z - 3.1329}{z^3 - 2.6919z^2 + 2.3886z - 0.6967} \tag{16a}$$

$$G_{Ous}(0.5, z) = \frac{8.4476z^3 - 24.4973z^2 + 23.6558z - 7.6060}{z^3 - 2.6010z^2 + 2.2103z - 0.6094} \tag{16b}$$

$$G_{Ous}(0.7, z) = \frac{19.5331z^3 - 57.1436z^2 + 55.6929z - 18.0824}{z^3 - 2.4901z^2 + 1.9948z - 0.5047} \tag{16c}$$

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 highest module.

4. 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 in the IIR filter associated to the controller. The finite available memory restricts the number and accuracy of

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}

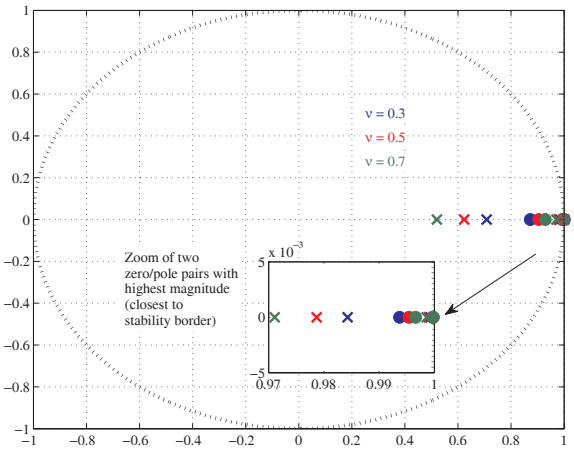


Fig. 1. Pole-zero maps of the discrete transfer function $G_{Ous}(\nu, z)$

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.

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 (A/D) conversion, that includes the quantization of the error signal. Then the current control signal is computed by linear combination of the digitalized errors in past time instants. Finally, there is a digital-to-analog (D/A) conversion ending with the transmission of the control signal to the actuators.

The A/D and D/A 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.

4.1. The effect of the sampling period

It is well-known that real implementation of fractional-order controllers on digital processing units pose some practical problems (Chen et al., 2009). In particular, the sampling period T must take into account the limitations posed by the available computing devices. Moreover, the period T must include all operations required by the digital control algorithm. Then a trade-off is required between the speed of computation and the complexity of the control law that is actually implemented. On the other hand, the more complex is the controller and the more coefficients are required to define its rational transfer function, the more time is necessary to compute the controller output, given its input. Moreover, the required memory increases with the complexity of the controller, while computing devices typically have a finite limited number of bits at disposal. Finally, increasing

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

ν	T	Zeros	$\Delta_z\%$	Poles	$\Delta_p\%$
0.3	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
	0.1	{0.9971, 0.9388, 0.1907}	-0.26 -5.52 -78.15	{0.9927, 0.8531, -0.2612}	-0.66 -13.33 -136.88
0.5	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
	0.1	{0.9978, 0.9546, 0.3333}	-0.20 -4.10 -63.16	{0.9900, 0.8055, -0.3977}	-0.90 -17.70 -163.81
0.7	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
	0.1	{0.9984, 0.9664, 0.4622}	-0.14 -3.03 -50.25	{0.9865, 0.7445, -0.5186}	-1.21 -23.33 -199.65

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 and 0.1 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 the low value $T = 0.001$ s, the singularities of $G_{Ous}(\nu, z)$ get very close to or even coincide with $(1, 0)$. Moreover, a big variation of the third zero and, above all, of the third pole is observed when T increases to 0.1.

4.2. The A/D-D/A conversion

The A/D conversion is performed in two steps consisting of quantization and coding. The former operation associates a finite number of levels of equal amplitude to the input signal, the latter converts the quantized signal into a numerical code suitable for the computing device. The D/A converter transforms the control algorithm output into a signal with a format suitable to drive actuators. Conversion is in two steps. In the former, the converter associates a finite number of levels of equal amplitude to the numerical code computed by the control algorithm. In the latter, the hold circuit generates a continuous signal that is sent to the actuators. The converter resolution represents the lower representable value other than 0, determines the number of quantization levels and greatly affects the error due to A/D and D/A conversions. The resolution depends on the number of bits involved in the conversion because the quantized signal is obtained as an integer multiple of the resolution and can only assume a value corresponding to a quantization level. To synthesize, a quantization error can be defined as the difference between the input signal and the converted signal, represented by the finite number of levels.

Given the number n_b of bits used in conversion, the number of available quantization levels is 2^{n_b} and the converter resolution is $V_{res} = V_{fs}/(2^{n_b} - 1)$, where V_{fs} is the full scale value (i.e. the maximum representable value) of the input signal (typically a voltage). Usually, the quantized signal is expressed in the form $V = V_{res} \sum_{i=0}^{n_b-1} 2^i A_i$, with $A_i = \{0, 1\}$. As previously mentioned, the upper bound of the quantization error is $\pm V_{res}/2$ and can be reduced by increasing the converter resolution. Namely, by increasing n_b, the input signal is represented by a larger number of levels.

Finally note that the quantization error in the A/D and D/A conversions directly affects the control system performance, by causing a steady-state error and limit cycles with permanent oscillations. The largest steady-state error can be evaluated by assuming the quantization error upper bound as a constant input disturbance to the plant. The amplitude of oscillations of the limit cycle can be estimated by using the discrete describing function.

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 the following elements:

- motor: an EMG49 dc gear motor 24v with encoder 245 PPR (pulses per revolution);
- driver: an L298 dual full-bridge for bidirectional DC-motor control;
- HIL: a Hardware-In-the-Loop dSPACE 1103 board;
- GUI: a Graphical User Interface in the *ControlDesk* software environment.

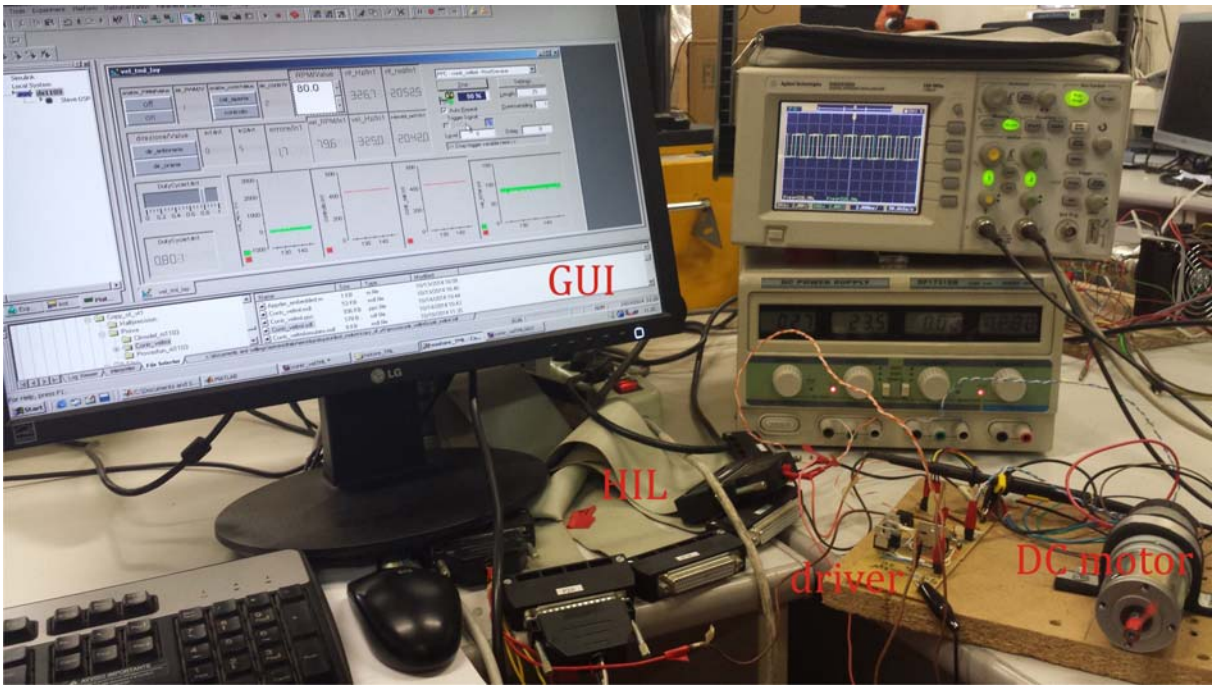


Fig. 2. Photo of the experimental platform

The motor speed is controlled by the FOPI controllers designed by the method proposed in section 2. The control lows have been downloaded into the HIL dSPACE 1103 board. The RTI (Real-Time Interface) control system and the GUI have been implemented in *ControlDesk* as shown in Figures 3 and 4. The control error depends on the number of impulses per second measured by the encoder, then the controller output is given in s^{-1}. This signal is converted in RPM units and normalized to the maximum motor speed, and finally a PWM command is obtained to drive the DC-motor.

The identified plant model is a FOPTD system characterized by the parameters $K = 1.6862$, $T = 0.0583$ s, and $\tau = 0.025$ s. Then the design procedure leads to the FOPI controller transfer function that takes the following form:

$$G_c(s) = 0.114 + \frac{1.6286}{s^{1.333}}.$$

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. The table also specifies the same information obtained for the FOPI controller.

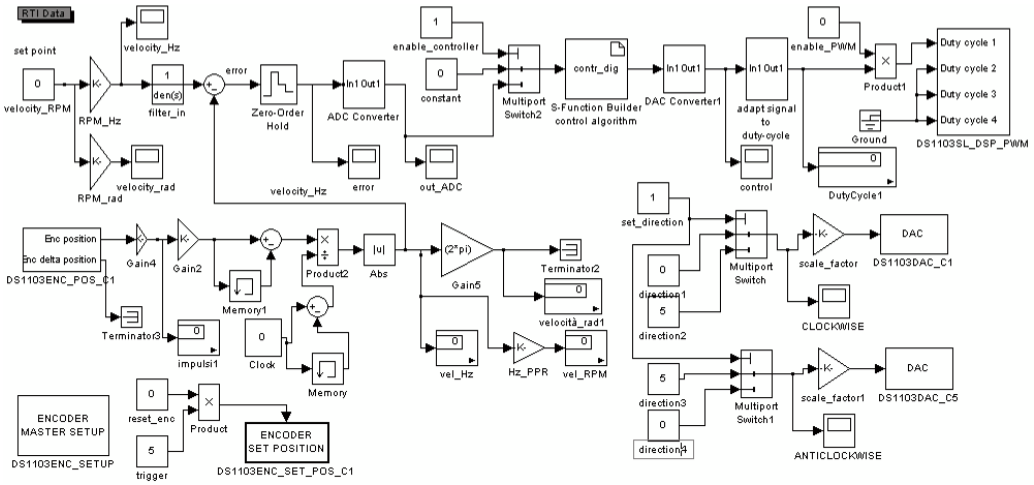


Fig. 3. RTI control system block diagram

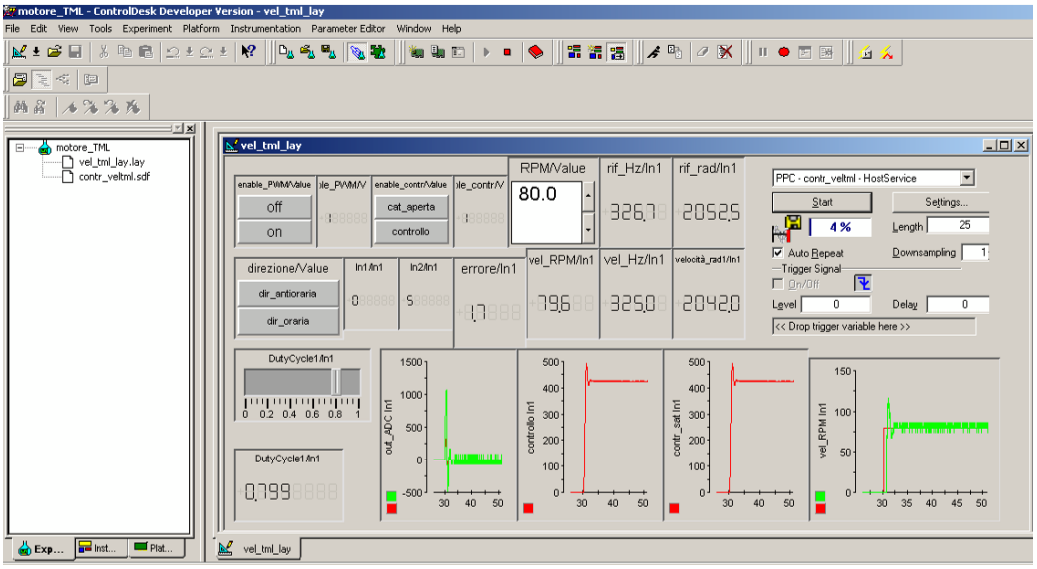


Fig. 4. Graphical User Interface in *ControlDesk*

Note that, for increasing values of T, the zeros and poles move close to each other and gather around the critical point $(1, 0)$.

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 0.05, 0.08, and 0.11 s. Lower values of T are not considered because in such cases the encoder provides a very noisy signal and because very high sampling rates are not frequent in the considered applications. The resolution of the analog/digital and digital/analog converters are both kept constant in the following way: ADCRes $=$ 16 bits whereas the DAC resolution can be DACRes $=$ 16, 12 or 8 bits. These values are usually found in commercial devices for industrial applications. Note that considering the case of equal but lower resolutions, i.e. ADCRes $=$ DACRes $=$ 12 or 8 bits, does not change results in a significant way with respect to ADCRes $=$ DACRes $=$ 16. Then, the specifically chosen combinations of ADCRes and DACRes reflect real practice conditions. Resolutions of lower values are not realistic and would downgrade the converter performance too much.

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 and the digital controller ($n = 5$, $\nu = 1.333$)

	T	Coefficients of numerator and denominator	Zeros (ze) and Poles (po)
G_{Oz}	0.05	*num*: 3.081 -12.03 17.98 -12.56 3.89 -0.3608 *den*: 1.0000 -3.419 4.2 -2.054 0.1829 0.0894	ze: $\{0.9991, 0.9942, 0.9639, 0.7920, 0.1545\}$ po: $\{0.9983, 0.9893, 0.9343, 0.6470, -0.1498\}$
	0.08	*num*: 2.73 -9.663 12.54 -6.863 1.112 0.1383 *den*: 1.0000 -2.998 2.925 -0.6958 -0.389 0.1582	ze: $\{0.9985, 0.9907, 0.9428, 0.6867, -0.0791\}$ po: $\{0.9973, 0.9829, 0.8969, 0.4892, -0.3679\}$
	0.11	*num*: 2.498 -8.161 9.297 -3.788 -0.1618 0.3152 *den*: 1.0000 -2.696 2.102 0.0363 -0.5921 0.1495	ze: $\{0.9980, 0.9873, 0.9222, 0.5932, -0.2341\}$ po: $\{0.9933, 0.9766, 0.8609, 0.3591, -0.4969\}$
G_{cz}	0.05	*num*: 0.351 -1.641 3.142 -3.139 1.708 -0.470 0.048 *den*: 3.081 -15.106 30.002 -30.539 16.451 -4.251 0.361	ze: $\{0.998, 0.989, 0.930, 0.758 \pm 0.240i, 0.237\}$ po: $\{1.000, 0.999, 0.994, 0.964, 0.792, 0.154\}$
	0.08	*num*: 0.311 -1.282 2.140 -1.830 0.820 -0.163 0.005 *den*: 2.730 -12.392 22.198 -19.398 7.985 -0.984 -0.138	ze: $\{0.997, 0.983, 0.891, 0.607 \pm 0.362i, 0.036\}$ po: $\{1.000, 0.998, 0.991, 0.943, 0.687, -0.079\}$
	0.11	*num*: 0.285 -1.036 1.507 -1.115 0.420 -0.052 -0.009 *den*: 2.498 -10.659 17.458 -13.084 3.626 0.477 -0.315	ze: $\{0.996, 0.976, 0.853, 0.451 \pm 0.470i, -0.091\}$ po: $\{1.000, 0.998, 0.987, 0.922, 0.593, -0.234\}$

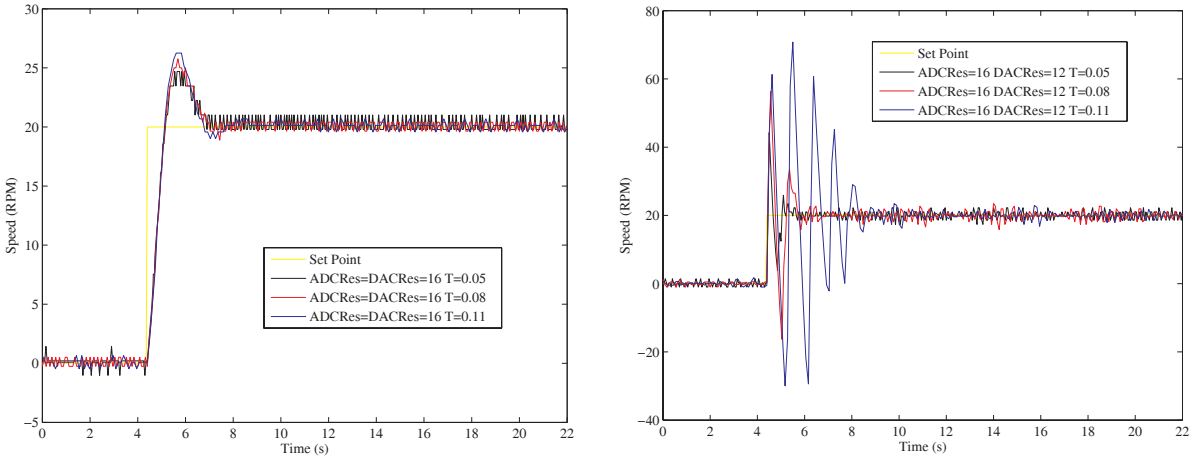


Fig. 5. Step responses for $T = 0.05, 0.08, 0.11$ s with: ADCRes = DACRes = 16 (left) and ADCRes = 16, DACRes = 12 (right)

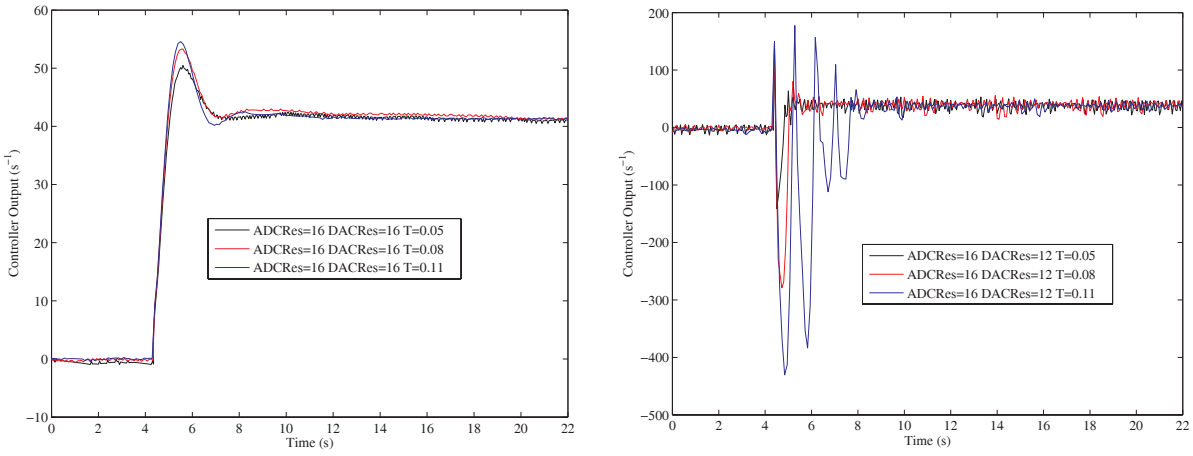


Fig. 6. Control variables for $T = 0.05, 0.08, 0.11$ s with: ADCRes = DACRes = 16 (left) and ADCRes = 16, DACRes = 12 (right)

Figure 5 shows that increasing the sampling period implies a corresponding growth of both the overshoot and the settling time (see Table 4). This phenomenon is emphasized when the DAC resolution is lower than the ADC one (e.g. see right part of figure 5). The case with ADCRes = 16 and DACRes = 8 makes the control system unstable for any value of T and is not shown in the figures. Conversely, the rise time is comparable in all conditions with reduced overshoot. Figure 6 shows the control variable in the respective conditions. An higher control effort is necessary during transients when increasing the sampling period. This is more evident when the DAC resolution decreases, which determines a sensible increment of the control variable. In synthesis, the benefits of assuming ADCRes = DACRes = 16 are clear, so is the worsening of the behavior when the difference between ADCRes and DACRes increases. Finally, note that high frequency oscillations of limited amplitude in steady-state conditions depend on the encoder resolution. However, these oscillations do not have a significant impact on the overall behavior of the controlled system.

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

Case no.	T (s)	ADCRes (bit)	DACRes (bit)	Overshoot (%)	Rise Time (s)	Settling Time (s) (10% of set-point)
1	0.05	16	16	23.4	0.55	2.25
2	0.05	12	12	23.4	0.55	2.25
3	0.05	8	8	23.4	0.55	2.20
4	0.05	16	14	17.4	0.25	2.25
5	0.05	16	12	121.4	0.036	1.55
6	0.05	16	8	unstable	—	—
7	0.08	16	16	28.8	0.55	2.25
8	0.08	12	12	28.8	0.55	2.32
9	0.08	8	8	28.8	0.55	2.40
10	0.08	16	14	25.0	0.25	2.14
11	0.08	16	12	181.9	0.036	2.32
12	0.08	16	8	unstable	—	—
13	0.11	16	16	31.2	0.55	2.31
14	0.11	12	12	31.2	0.55	2.31
15	0.11	8	8	31.2	0.55	2.31
16	0.11	16	14	34	0.25	2.20
17	0.11	16	12	206,6	0.036	6.27
18	0.11	16	8	unstable	—	—

The second test examines the effect of the variation of the converters resolution while maintaining a constant sampling period. Again, the step responses and the control variables are shown in different cases.

Figure 7 shows that if the DAC resolution decreases with respect to the ADC one, then the rise time decreases whereas the overshoot and the settling time increase up to make the system unstable. This phenomenon is accentuated by increasing the sampling period (see figure 7 top-down). Moreover, the evolution of the control variable gets worst (see amplitude and number of oscillations). In particular, the rise time is highly variable with the DAC resolution and, for a constant sampling period, it considerably reduces for lower DAC resolutions. However, there is a corresponding increase of the control effort and a worsening of the overall system dynamic behavior, that possibly lead the system to instability.

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.

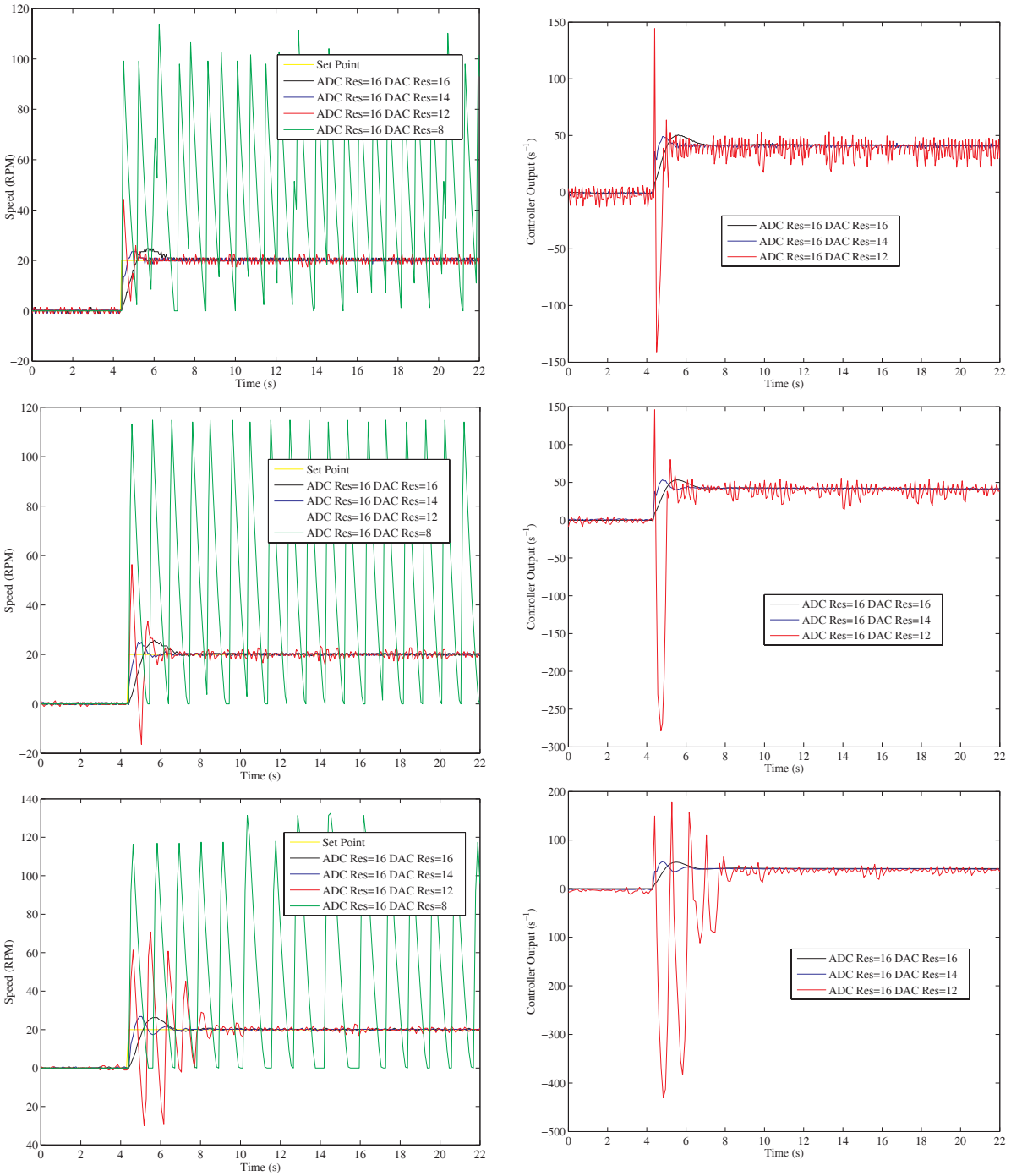


Fig. 7. Step responses (left) and control variables (right) with different resolutions and $T = 0.05$ s (top figures), $T = 0.08$ s (middle figures), $T = 0.11$ s (down figures)

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.

References

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

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.

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.

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.

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.