

Article

Event-Based Implementation of Fractional Order IMC Controllers for Simple FOPDT Processes

Cristina I. Muresan ¹, Isabela R. Birs ^{1,2,*} and Eva H. Dulf ^{1,3,*}

¹ Automation Department, Technical University of Cluj-Napoca, Memorandumului 28, 400114 Cluj-Napoca, Romania; Cristina.Muresan@aut.utcluj.ro

² Research Group of Dynamical Systems and Control, Faculty of Engineering and Architecture, Ghent University, EEDT Decision & Control, Flanders Make Consortium, Tech Lane Science Park 125, B-9052 Ghent, Belgium

³ Physiological Controls Research Center, Óbuda University, H-1034 Budapest, Hungary

* Correspondence: Isabela.Birs@aut.utcluj.ro (I.R.B.); Eva.Dulf@aut.utcluj.ro (E.H.D.)

Received: 2 August 2020; Accepted: 14 August 2020; Published: 17 August 2020



Abstract: Fractional order calculus has been used to generalize various types of controllers, including internal model controllers (IMC). The focus of this manuscript is towards fractional order IMCs for first order plus dead-time (FOPDT) processes, including delay and lag dominant ones. The design is novel as it is based on a new approximation approach, the non-rational transfer function method. This allows for a more accurate approximation of the process dead-time and ensures an improved closed loop response. The main problem with fractional order controllers is concerned with their implementation as higher order transfer functions. In cases where central processing unit CPU, bandwidth allocation, and energy usage are limited, resources need to be efficiently managed. This can be achieved using an event-based implementation. The novelty of this paper resides in such an event-based algorithm for fractional order IMC (FO-IMC) controllers. Numerical results are provided for lag and delay dominant FOPDT processes. For comparison purposes, an integer order PI controller, tuned according to the same performance specifications as the FO-IMC, is also implemented as an event-based control strategy. The numerical results show that the proposed event-based implementation for the FO-IMC controller is suitable and provides for a smaller computational effort, thus being more suitable in various industrial applications.

Keywords: fractional order IMC; first order plus dead-time processes; event-based implementation; numerical simulations; comparative closed loop results

1. Introduction

Fractional calculus has been reaching a larger part of the research community due to the numerous advantages it has. The increasing interest is mainly due to the ability to capture essential dynamics in physical phenomena. This is seconded by the demonstrated ability of fractional order controllers to meet more design specifications and provide for overall increased robustness and performance. Several researchers have used fractional order tools to model more accurately viscoelastic phenomena [1], aerodynamics [2], structural engineering [3], non-Newtonian characteristics in blood [4,5], type 1 diabetes [6], diffusion phenomena in magnetic resonance imaging [7], post-exposure prophylaxis model in HIV [8], epidemic models for infectious diseases [9], biochemical phenomena [10], etc.

In terms of fractional order control, the starting point is the generalization to arbitrary orders of the proportional-integral-derivative (PID) controller, as proposed in [11]. Ever since then, a manifold of papers have been published, presenting various modifications of the original fractional order PID (FO-PID) controller, various tuning methods and improvements. The key idea is that the

generalization of the PID to a fractional order provides more flexibility in improving the system control performance [12,13]. Several enhancements for FO-PID controllers were proposed. An optimal FO-PID controller was proposed and tuned based on particle swarm optimization [14]. Designs based on phase and gain margin specifications are quite abundant [12,15,16], and quite frequently the design is based on ensuring the iso-damping property [12,16–18]. Tuning is usually performed in the frequency domain, but time domain approaches were also considered [19,20]. Autotuning methods for fractional order PID controllers were also proposed [21–23]. Some rather recent review papers on fractional order controllers can be found in [24–26] and provide an insight into fractional order control of different types. An excellent review paper on fractional order controllers, including their most widely used continuous and discrete approximation methods, as well as their digital and analogue implementation methods. The paper also presents the Matlab toolboxes that facilitate the use of fractional order calculus in modeling and control. At the same time, it clearly pinpoints the advantages and disadvantages of using fractional order calculus in control engineering [27].

For time delay systems, including first order plus dead time (FOPDT) processes, several approaches have been introduced and developed over the years. A recent review paper regarding the approaches for these types of system is given in [28]. Alternative control strategies based on fractional order calculus for variable time delay systems are proposed in [29]. For significant delays, a Smith predictor (SP) scheme can be useful. The fractional order controller design in this SP control scheme is based on several approaches. One method proposes a modified SP structure, where the tuning procedure is based on Bode's ideal transfer function and the internal model control (IMC) principle. The resulting control system is robust to changes in the process parameters [30]. A similar design for a fractional order PI controller in a SP control structure, also based on Bode's ideal transfer function, is presented in [31]. The analytical tuning rules are derived in the frequency domain and applied to various types of processes. The advantages of the method rely on a simple design scheme and a straightforward method, which can be easily implemented in the process industry. The SP control structure is also used as a means for comparing various fractional order controllers for a heat diffusion system in [32]. The research offers valuable insight into the performance of the proposed fractional order control algorithms. In [33], a time domain approach is considered for the design of fractional order controllers in a SP structure. Only two parameters need to be tuned, which simplifies considerably the design procedure. The tuning rules are derived based on an ideal closed-loop transfer function, with performance specifications imposed as overshoot and settling time.

One of the simplest tuning rules for integer order PID controllers, as well as for FO-PID controllers, highly suitable for time delay systems, is the IMC methodology. This consists in the simple inversion of the invertible part of the process model and in the addition of a properly selected filter. This method has also been tackled by researchers. For the design of a fractional order IMC controller (FO-IMC), the most widely used approach is based on using a modified fractional order filter [34]. Some tuning methods are based on the Ziegler–Nichols approach [35], Taylor series [36], dominant pole placement method [37]. Other approaches are based on frequency domain specifications, such as phase and gain margins [38–42]. Such an approach is also preferred in this research.

The FO-IMC control strategy proposed in this paper is based on specifying a certain gain crossover frequency, to ensure a specific closed loop settling time, as well as phase margin criteria to ensure a certain closed loop overshoot. The tuning rules are exemplified for first order plus dead-time processes. To implement the resulting fractional order controllers, an efficient method is used, namely the non-rational transfer function (NRTF) approximation method [43]. This allows for a more accurate approximation of the process dead-time and ensures an improved closed loop response [38]. The design is suitable for all types of FOPDT processes, but can be easily extended to higher order processes or even fractional order transfer functions [44]. The research is focused on controlling processes where CPU, bandwidth allocation, and energy usage are limited [45]. In this context, the idea of event-based control is a natural solution for controller implementation [46,47]. Such an approach has been only recently introduced to fractional order PID controllers [46,47]. The novelty of this paper resides in

introducing an event-based methodology for FO-IMC controllers. The method is entirely original compared to [46,47], where a standard fractional order PID type controller is tuned according to some frequency domain specifications. The IMC methodology presented in the current manuscript is not used in the actual tuning. Then, the event-based implementation of the fractional order PID controller in [46] is based on direct discretization methods that use direct fractional order mappers of the fractional integrator and differentiator of the fractional order PID controller. The event-based algorithm relies then on a generalization of the standard direct discretization methods for fractional order elements, where the sampling period is considered as a variable parameter (it depends on an event being triggered). The study implements the proposed strategy entirely in the control signal generator, using a single function to compute the fractional order control value. The novelty of the current manuscript, apart from a different tuning procedure and a different fractional order controller type, is based on a two-step implementation of the event-based fractional order controller. Firstly, the fractional order controller determined based on the FO-IMC methodology is decomposed into an integer order PI and a fractional order filter. Then, the NRTF approach is used to determine a standard discrete-time approximation for the fractional order filter. The remaining PI controller is implemented in an event-based approach. Numerical results are provided for lag and delay dominant FOPDT processes. For comparison purposes, an integer order PI controller, tuned according to the same performance specifications as the FO-IMC, is also implemented as an event-based control strategy. The numerical results show that the proposed event-based implementation for the FO-IMC controller is suitable and provides for a reduction in the resources used for computing the control signal.

The paper is structured as follows. In Section 2, the proposed tuning procedure for the FO-IMC controllers is detailed. This is a novel approach, being based on the NRTF approximation method. Then, the event-based algorithm for the FO-IMC controller is described and the NRTF approach is briefly presented. Section 3 presents the results obtained for lag and delay dominant FOPDT processes, in terms of reference tracking, disturbance rejection, and robustness. Comparative results are also given. Section 4 includes a brief discussion of the previously presented results, while the last section concludes the research.

2. Materials and Methods

2.1. Tuning the FO-IMC Controller

The following mathematical representation of the processes is considered:

$$H_p(s) = \frac{k}{Ts + 1} e^{-\tau s}, \quad (1)$$

where k is the process gain, T is the time constant, τ is the time delay, and s is the Laplace variable. The transfer function in (1) is generally used to model various types of processes, such as thermal, chemical, biomedicine systems. For these types of processes, a FO-IMC control strategy is proposed, as indicated in Figure 1, where $H_c(s)$ is the equivalent controller in a standard feedback structure, $H_m(s)$ is the process model, d —is the disturbance signal, y —the output signal, r —the reference signal.

The design of the FO-IMC controller is based on the inversion of the process model. For time delay processes, the time delay cannot be inverted and needs to be approximated, using either first Taylor series or Padé approximations [39]. Current research has shown that using these widely used approximation methods leads to poorer closed loop results [38], compared to a new approach based on the non-rational transfer function (NRTF) approximation method [43]. The new approach to tuning FO-IMC controllers for FOPDT processes in (1) is detailed next.

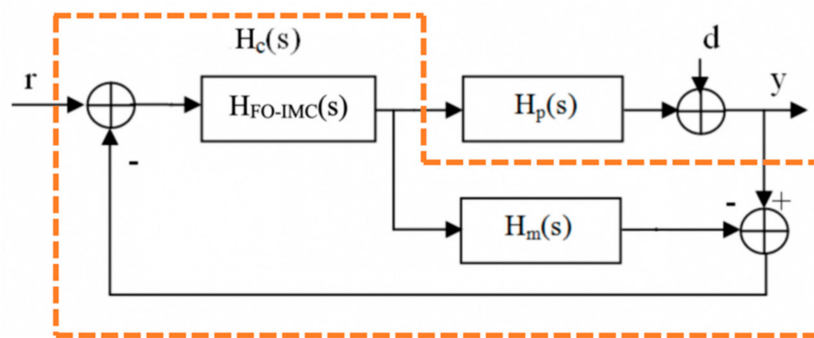


Figure 1. FO-IMC closed loop control scheme.

The proposed fractional order IMC (FO-IMC) controller is given by:

$$H_{FO-IMC}(s) = \frac{Ts + 1}{k} \frac{1}{\lambda s^\alpha + 1}, \quad (2)$$

where $\alpha \in (0, 2)$ is the fractional order and λ is the FO-IMC filter time constant. For $\alpha = 1$, the classical IMC controller is obtained. The limiting interval of the fractional order α is chosen such that the fractional order operation has a physical relevance from the control action point of view, as presented in [11].

Simple computations based on the diagram in Figure 1 lead to the following transfer function for the equivalent controller:

$$H_c(s) = \frac{Ts + 1}{k(\lambda s^\alpha + 1 - e^{-\tau s})}. \quad (3)$$

Notice the direct occurrence of the time delay in the denominator of (3). This can be further written as an integer order PI controller in series with a fractional order filter:

$$H_f(s) = \frac{s}{\lambda s^\alpha + 1 - e^{-\tau s}}. \quad (4)$$

To tune the parameters of the FO-IMC controller, simple tuning methods can be used. In this particular approach, the phase margin and gain crossover frequency specifications are employed. These two performance specifications refer to the loop transfer function:

$$H_l(s) = H_p(s) \cdot H_c(s) = \frac{1}{\lambda s^\alpha + 1 - e^{-\tau s}} e^{-\tau s}. \quad (5)$$

To meet the phase margin constraint, the phase equation is used:

$$\angle H_l(j\omega_c) = -\pi + PM, \quad (6)$$

where ω_c is the desired gain crossover frequency and PM is the desired phase margin. The phase margin is a direct measure for the stability and robustness of a system. The larger the PM is, the more robust the overall closed loop system becomes. The selection of the gain crossover frequency is based on the maximization of the delay margin associated to the closed loop system, according to:

$$\omega_c = \frac{PM}{\tau_m - \tau}, \quad (7)$$

where τ_m is the maximum time delay that would make the process in (1) unstable. To meet the gain crossover frequency constraint, the modulus equation is used:

$$|H_l(j\omega_c)| = 1. \quad (8)$$

Then, replacing (1) and (3) in (6) and (8), leads to the following system of equations:

$$\begin{cases} \lambda \frac{\tan(\pi-PM-\tau\omega_c) - \tan(\pi-PM-\tau\omega_c) \cos(\tau\omega_c) - \sin(\tau\omega_c)}{\omega_c^\alpha \sin(\frac{\alpha\pi}{2}) - \omega_c^\alpha \tan(\pi-PM-\tau\omega_c) \cos(\frac{\alpha\pi}{2})}, \\ \lambda^2 \omega_c^{2\alpha} + 2\lambda \omega_c^\alpha \left[\cos(\frac{\alpha\pi}{2}) - \cos(\frac{\alpha\pi}{2} + \tau\omega_c) \right] - 2\cos(\tau\omega_c) + 1 = 0. \end{cases} \quad (9)$$

The tuning of the FO-IMC controller is completed when the system of nonlinear equations in (9) is solved [12,17]. To implement the equivalent controller, an event-based algorithm is preferred. Such an approach leads to a smaller computational effort [46] and is more suitable in various industrial applications [46].

2.2. Event-Based Algorithm for FO-IMC Controllers

The equivalent controller for a standard feedback loop, as usually encountered in industrial applications, obtained based on the IMC methodology is given in (3). To implement this controller, an alternative form is preferred, as mentioned previously, with an integer order PI controller in series with the fractional order filter in (4). The new mathematical model for this fractional order equivalent controller is given as:

$$H_c(s) = C(s)H_f(s) = \frac{Ts+1}{ks}H_f(s) = \frac{T}{k}\left(1 + \frac{1}{Ts}\right)H_f(s), \quad (10)$$

where $C(s)$ is the PI controller and $H_f(s)$ is the fractional order filter in (4).

Figure 2 presents the event-based paradigm, consisting of three components: process data measurement (data acquisition), event detector, and control input generator.

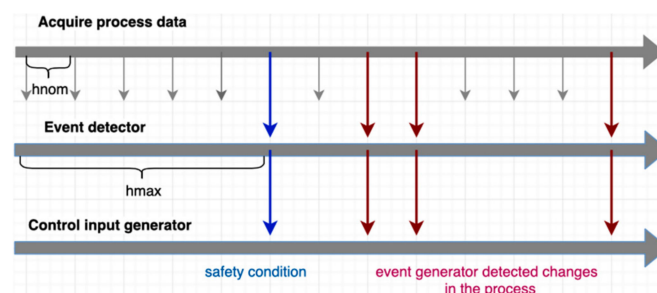


Figure 2. Basic paradigm of an event-based controller implementation [46].

The process output is measured at each sampling period h_{nom} , chosen according to standard discretization rules. The measured output data is transferred into the event detector. The main task of the detector is to decide whether an event has occurred, and in this case to trigger the control input generator. The event detector implements a function that optimizes the control process [47–49]. One of the most widely used event detection rules is based on computing the error signal and verifying whether it lies within a predefined range $[-\Delta_e, \Delta_e]$:

$$|e(t) - e(t - h_{act})| \geq \Delta_e, \quad (11)$$

where h_{act} denotes the elapsed time since the triggering of the previous event, $e(t)$ is the current error signal, and $e(t - h_{act})$ is the error at the previous event. Apart from the event triggering condition in (11), a safety condition is also used such that:

$$h_{act} \geq h_{max}, \quad (12)$$

where h_{max} is the maximum time between two consecutive events [50].

Once an event has been triggered, either when (11) or (12) occurs, the control input generator computes a new value for the control signal, according to a predefined algorithm. Since the computation of the control signal value occurs at variable sampling instants, the control law is represented by a discrete-time control algorithm where the sampling time parameter is considered as a variable parameter. The algorithm proposed in this paper is detailed below.

The overall Simulink implementation of such an event-based algorithm is given in Figure 3, where the blocks stand for: *nrtf_fo_filter* implements the NRTF approximation of the fractional order filter in (4), *event-detector* implements both the event detection part, as well as the control input generator for the integer order PI controller. The fractional order filter in (4) is used to filter the error signal. This occurs at every sampling period h_{nom} . The filtered error signal is then fed to the event-detector which implements the function that triggers of the control input generator. The latter is based on the standard PI controller, $C(s)$ in (10). Thus, in the proposed approach, the PI controller is implemented in an event-based manner.

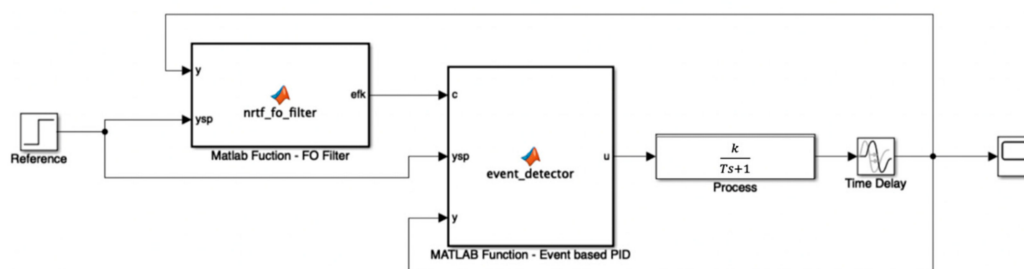


Figure 3. Simulink implementation of an event-based fractional order control algorithm.

Once the tuning of the FO-IMC has been performed and the two parameters, λ and α , determined, the fractional order filter in (4) is implemented in a discrete-time approximation based on the NRTF method, while the PI controller is implemented as an event-based algorithm. The control signal $U(s)$ of the PI controller $C(s)$ in (10) is computed based on:

$$U(s) = k_p \left(1 + \frac{1}{T_i s} \right) + E_f, \quad (13)$$

where k_p and T_i are the proportional gain and integral time constant and $E_f(s)$ is the Laplace transform of the filtered error signal, at the output of the fractional order filter in (4):

$$H_f(s) = \frac{E_f(s)}{E(s)}, \quad (14)$$

where $E(s)$ is the Laplace transform of the error signal $e(t)$ defined as $E(s) = Y_{sp}(s) - Y(s)$, $Y_{sp}(s)$ is the Laplace transform of the reference signal, $Y(s)$ is the Laplace transform of the measured output signal. For the particular case of FOPDT processes, the PI controller parameters are given as: $k_p = \frac{T}{k}$ and $T_i = T$, as resulting from (10) and (13). The event-based implementation of the PI controller can be achieved based on (13).

2.3. Comparisons with an Event-Based PI Controller

To compare the results, a classical integer order PI controller is designed for the same process in (1). The choice of the PI controller is based on the same number of parameters, as in the case of

the FO-IMC controller, which allows for a similar tuning approach based on ensuring a certain gain crossover frequency and phase margin. The PI controller transfer function is given as:

$$C_{PI}(s) = k_p \left(1 + \frac{1}{T_i s} \right), \quad (15)$$

where k_p and k_i are the proportional and integral gains. The same performance specifications are used, as in the design of the FO-IMC controller. In this case, the loop transfer function is given by:

$$H_l(s) = H_p(s) \cdot C_{PI}(s) = k_p \left(1 + \frac{1}{T_i s} \right) \frac{k}{Ts + 1} e^{-\tau s}. \quad (16)$$

To meet the phase margin constraint, the phase Equation in (6) is used, leading to:

$$T_i = \frac{\tan\left(-\frac{\pi}{2} + PM - \angle H_p(j\omega_c)\right)}{\omega_c}, \quad (17)$$

where ω_c is the desired gain crossover frequency and PM is the desired phase margin. To meet the gain crossover frequency constraint, the modulus equation in (8) is used, leading to:

$$k_p = \frac{T_i \omega_c}{|H_p(j\omega_c)| \sqrt{T_i^2 \omega_c^2 + 1}}. \quad (18)$$

Thus, the tuning of the PI controller is complete, with the k_p and k_i parameters uniquely determined based on (17) and (18). To implement the PI controller, the event-based algorithm as proposed in [46] will be used. For a fair comparison, the parameters of the event-based algorithm will be similar to those used in the event-based implementation of the FO-IMC controller.

2.4. A Brief Overview of the NRTF Approximation Approach

Various discrete-time approximation methods for fractional order systems have been proposed over the years, including direct and indirect approaches [51]. One of the advantages of direct methods lies in the expedite approximation of fractional order systems as discrete-time higher order transfer functions. Most of the existing methods deal with the direct approximation of simple fractional order elements, such as the fractional order integrator or first order filter [52]. The NRTF method has been proposed as a means to offer a discrete-time approximation of low order for any type of non-rational transfer function, including complex fractional order elements and time delays [43]. The method consists of four steps, as detailed briefly below. A more detailed analysis and comparisons of the NRTF approach with other methods can be found in [43].

Step 1: The following generating function is used to replace the Laplace variable s in the fractional order system:

$$w(z^{-1}) = \frac{1 + \delta}{T_s} \frac{1 - z^{-1}}{1 + \delta z^{-1}}, \quad (19)$$

where $\delta \in [0, 1]$ is a shaping knob and T_s is the sampling period. To decrease the phase error between the approximation and the actual fractional order system, the parameter δ should be selected to be large, while a smaller value of δ decreases the magnitude error [43]. As fractional order systems have unlimited memory, the approximation is only possible within a certain limited frequency range. During this step, the maximum frequency boundary ω_h has to be specified, according to the Nyquist sampling theorem, with $\omega_h = \frac{\pi}{T_s}$. The approximation of the fractional order system will then be valid in an interval defined as $(0, \omega_h)$.

Step 2: The frequency response of the discrete-time fractional order system obtained in Step 1 is computed. To achieve this, the discrete-time operator z is replaced with $e^{j\omega T_s}$,

where $\omega = \frac{2\pi}{T_s N_s} \left[0 \ 1 \ 2 \ \dots \ \frac{N_s}{2} \right]$ is a vector of equally spaced frequencies and N_s is also a tuning knob. For a good approximation in the low frequency range, N_s should be large. The result of this second step consists in a vector of frequency response values of the fractional order discrete time transfer function.

Step 3: The inverse fast Fourier transform (FFT) algorithm is used to calculate the impulse response of the discrete-time fractional order system:

$$g[n] = \frac{1}{N_s} \sum_{k=0}^{N_s-1} G[k] e^{+j \frac{2\pi}{N_s} nk}, \quad n = 0, 1, 2, \dots, N_s - 1, \quad (20)$$

with $G[k]$ denoting the frequency response of the original fractional order system. The result of this step is a vector (20) containing N_s impulse response values.

Step 4: The Steiglitz–McBride approach [53] is used to determine a rational discrete-time transfer function with a similar impulse response as obtained from the inverse FFT in Step 3. The order N of the approximation has to be specified. The larger N is, the better the approximation. This also results in a higher order discrete-time transfer function. A compromise should be considered. The result of this step is the final discrete-time integer order transfer function of the form:

$$G(z^{-1}) = \frac{c_0 + c_1 z^{-1} + \dots + c_N z^{-N}}{d_0 + d_1 z^{-1} + \dots + d_N z^{-N}}, \quad (21)$$

where $c_0, c_1 \dots c_N$ and $d_0, d_1 \dots d_N$ are coefficients computed according to the SteiglitzMcBride approach.

The step-by-step design procedure for a FO-IMC controller, as proposed in this paper, is detailed below, along with the event-based implementation.

Step 1: Select the desired PM and τ_m . Compute ω_c based on (7).

Step 2: Solve (9) to determine the FO-IMC controller parameters, λ and α .

Step 3: Compute the PI controller parameters according to $k_p = \frac{T}{k}$ and $T_i = T$ and the fractional order filter as indicated in (4), for the equivalent fractional order controller as described in (3).

Step 4: Select the parameters of the NRTF approximation method for the discrete-time approximation of the fractional order filter in (4): N , ω_h , and δ . The sampling period T_s is indirectly obtained as $\omega_h = \frac{\pi}{T_s}$.

Step 5: Select the parameters for the event-based implementation of the PI controller: $h_{nom} = T_s$, Δ_e , and h_{max} . The event-based algorithm is implemented based on two functions: an event detector (Figure 4) and a control signal generator (Figure 5), as follows.


```

function u = event_detector(ysp, y)

% define PI parameters Kp, Ti

global hact hnom Delta_e hmax es e u_last
e = ysp - y;
hact = hact + hnom;

    if (abs(e - es) > Delta_e || hact >= hmax)
        u = control_input_generator(ysp, y, Kp, Ti);
        u_last = u;
    else
        u = u_last;
    end

end

```

Figure 4. Event detector function.

```

function u = control_input_generator(ysp, y, Kp, Ti)

global hact y_old ui es e

up = Kp * (ysp - y);
u = up + ui;

ui = ui + Kp/Ti*hact*(ysp-y);

y_old = y;
hact = 0;
es = e;

end

```

Figure 5. Control input generator function.

3. Results

This section presents the main results obtained. The design is specific for FOPDT processes. Two different types will be discussed, the lag dominant and the delay dominant process. In both cases, reference tracking, disturbance rejection, and robustness to gain variations are considered as simulation tests. Reference tracking tests have been included in order to show the efficiency of the event-based controller in terms of setpoint trailing, while disturbance rejection results are considered in order to demonstrate the ability of the event-based controller to cope with external disturbances. Only step disturbance signals have been considered. As the results show, the event-based FO-IMC controller ensures better closed loop results compared to the event-based integer order PI controller, despite both controller being tuned and implemented in a similar fashion. Robustness tests are included to demonstrate that a fractional order IMC controller is intrinsically more robust to gain uncertainties, compared to a traditional integer order PI controller, even though robustness is not directly tackled in the design.

3.1. The Lag Dominant FOPDT Process

The following process model is considered for the first case study:

$$H_p(s) = \frac{1}{4s + 1} e^{-s}. \quad (22)$$

To design the FO-IMC controller based on the tuning method described in Section 2.1, a phase margin $PM = 85^\circ$ and a gain crossover frequency $\omega_c = 0.3$ rad/s are imposed. The solution of (9) yields $\lambda = 1.95$ and $\alpha = 0.85$, with the FO-IMC controller given as:

$$H_{FO-IMC-NRTF}(s) = \frac{4s+1}{1} \frac{1}{1.95s^{0.85}+1}, \quad (23)$$

while the equivalent controller is computed as:

$$H_c(s) = \frac{4s+1}{1.95s^{0.85}+1-e^{-s}}. \quad (24)$$

In this case, the parameters of the PI controller are $k_p = 4$ and $T_i = 4$. The fractional order filter:

$$H_f(s) = \frac{s}{1.95s^{0.8475}+1-e^{-s}} = \frac{E_f(s)}{E(s)}, \quad (25)$$

is implemented as a discrete-time transfer function based on the NRTF approach with the order $N = 7$, $\delta = 1$ and sampling period $T_s = 0.1$ s. For the event-based PI control algorithm, the following parameters are used: $h_{nom} = T_s = 0.1$ s, $h_{max} = 0.5$, and $\Delta_e = 0.1$.

For comparison purposes, a PI controller is also designed for the same performance specifications. The parameters are obtained according to (17) and (18), leading to:

$$C_{PI}(s) = 1.3841 \left(1 + \frac{1}{6.3715s} \right). \quad (26)$$

The controller in (26) is then implemented in an event-based algorithm with the same parameters as in the case of the FO-IMC controller.

The reference tracking, disturbance rejection and robustness results are given in Figures 6–9.

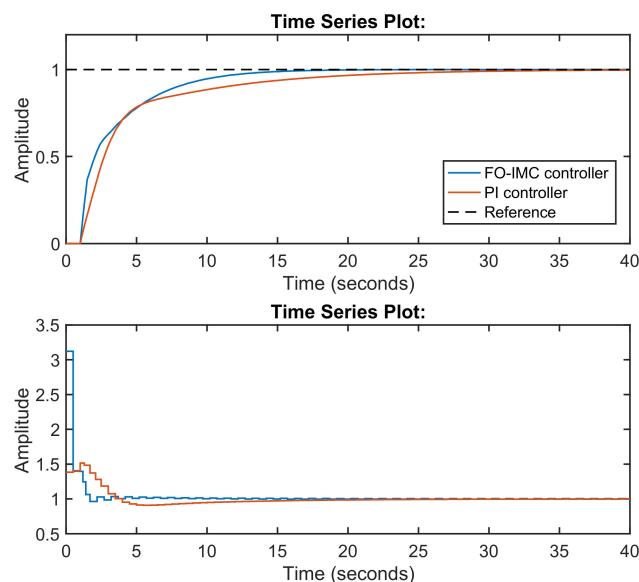


Figure 6. Comparison between the closed loop systems with the event-based FO-IMC and the classical proportional integral PI controllers for reference tracking (output signal in the upper plot, input signal in the lower plot).

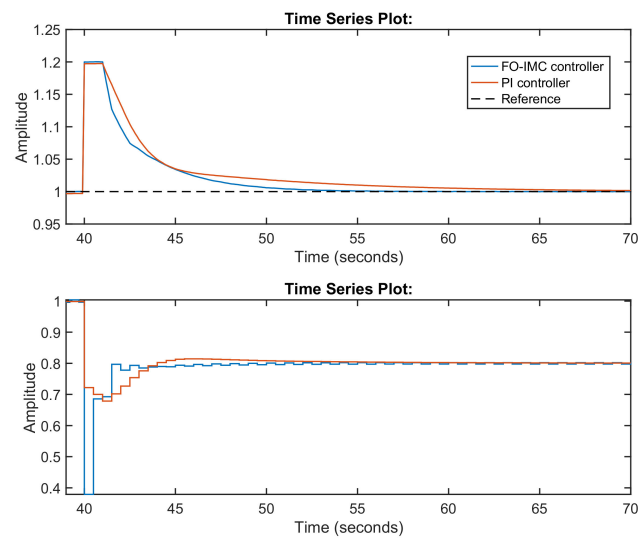


Figure 7. Comparison between the closed loop systems with the event-based FO-IMC and the PI controllers for a 0.2 disturbance rejection (output signal in the upper plot, input signal in the lower plot).

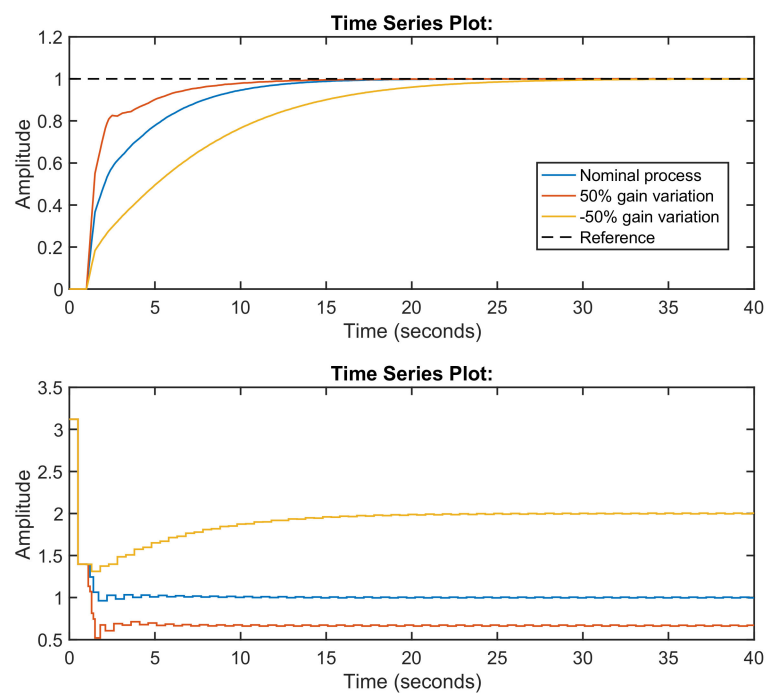


Figure 8. Robustness validation of the event-based FO-IMC controller (output signal in the upper plot, input signal in the lower plot).

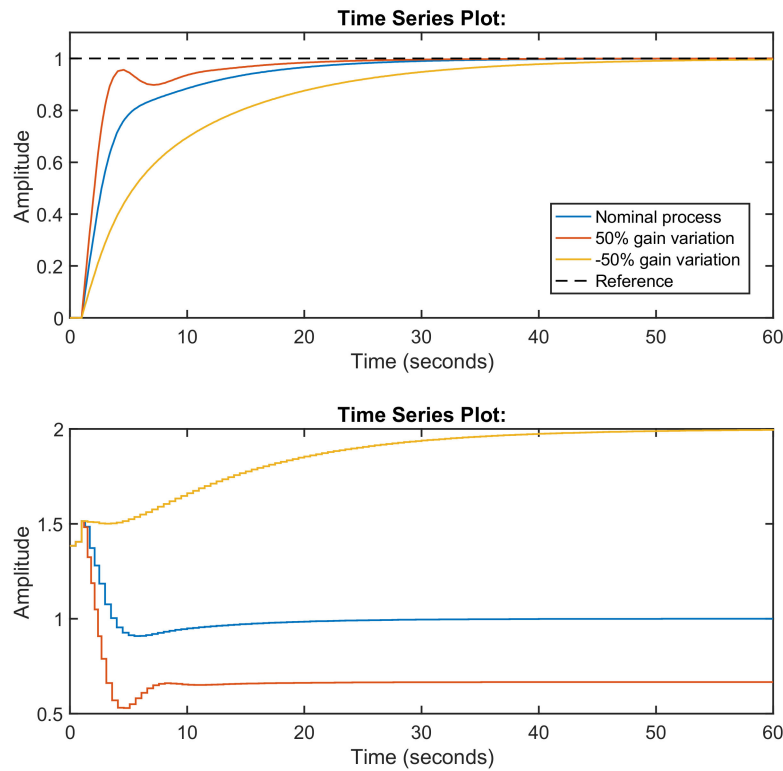


Figure 9. Robustness validation of the event-based PI controller (output signal in the upper plot, input signal in the lower plot).

3.2. The Delay Dominant FOPDT Process

The following process model is considered for the second case study:

$$H_p(s) = \frac{2}{s+1} e^{-2s}. \quad (27)$$

To design the FO-IMC controller based on the tuning method described in Section 2.1, a phase margin $PM = 80^\circ$ and a gain crossover frequency $\omega_c = 0.3$ rad/s are imposed. The solution of (9) yields $\lambda = 0.88$ and $\alpha = 0.62$, with the FO-IMC controller given as:

$$H_{FO-IMC-NRTF}(s) = \frac{s+1}{2} \frac{1}{0.88s^{0.62} + 1}, \quad (28)$$

while the equivalent controller is computed as:

$$H_c(s) = \frac{s+1}{2(0.88s^{0.62} + 1 - e^{-2s})}. \quad (29)$$

In this case, the parameters of the PI controller are $k_p = 0.5$ and $T_i = 1$. The fractional order filter:

$$H_f(s) = \frac{s}{0.88s^{0.62} + 1 - e^{-2s}} = \frac{E_f(s)}{E(s)}, \quad (30)$$

is implemented as a discrete-time transfer function based on the NRTF approach with the order $N = 5$, $\delta = 0.5$ and sampling period $T_s = 0.1$ s. For the event-based PI control algorithm, the following parameters are used: $h_{nom} = T_s = 0.1$ s, $h_{max} = 0.5$ and $\Delta_e = 0.1$.

For comparison purposes, a PI controller is also designed for the same performance specifications. The parameters are obtained according to (17) and (18), leading to:

$$C_{PI}(s) = 0.3430 \left(1 + \frac{1}{2.9055s} \right). \quad (31)$$

The controller in (31) is then implemented in an event-based algorithm with the same parameters as in the case of the FO-IMC controller.

The reference tracking, disturbance rejection, and robustness results are given in Figures 10–13.

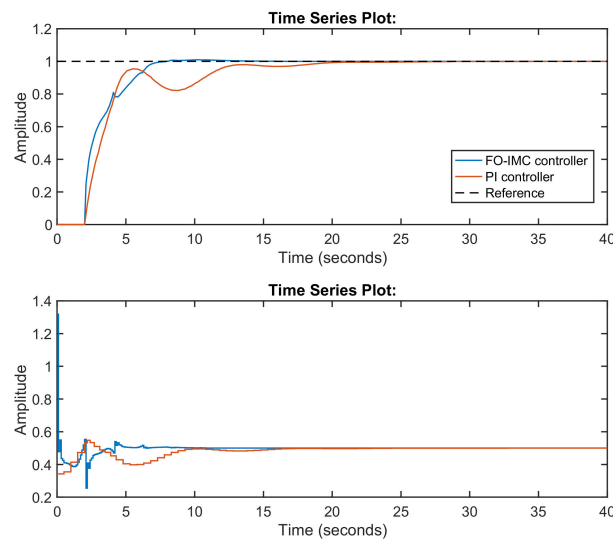


Figure 10. Comparison between the closed loop systems with the event-based FO-IMC and the PI controllers for reference tracking (output signal in the upper plot, input signal in the lower plot).

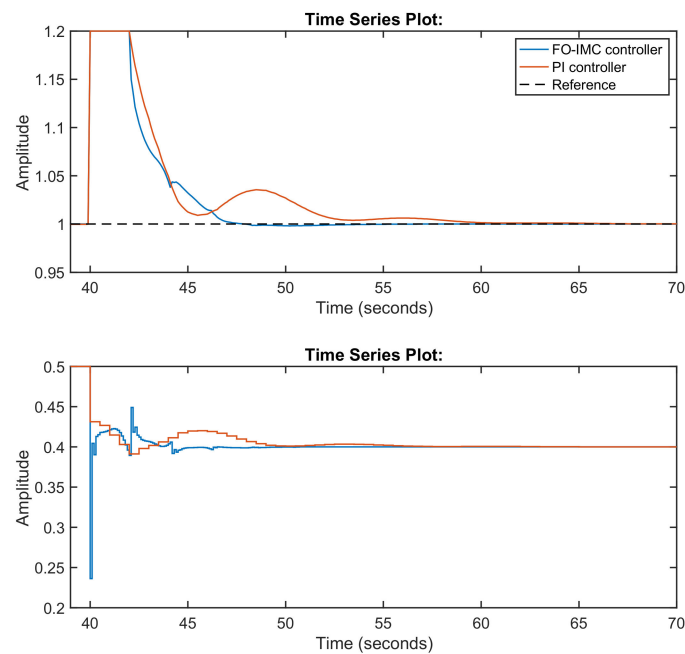


Figure 11. Comparison between the closed loop systems with the event-based FO-IMC and the PI controllers for a 0.2 disturbance rejection (output signal in the upper plot, input signal in the lower plot).

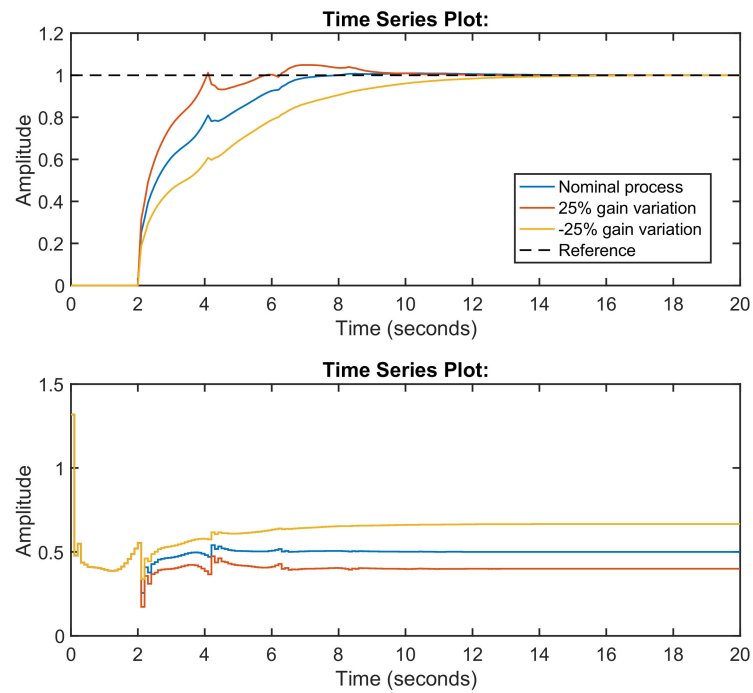


Figure 12. Robustness validation of the event-based FO-IMC controller (output signal in the upper plot, input signal in the lower plot).

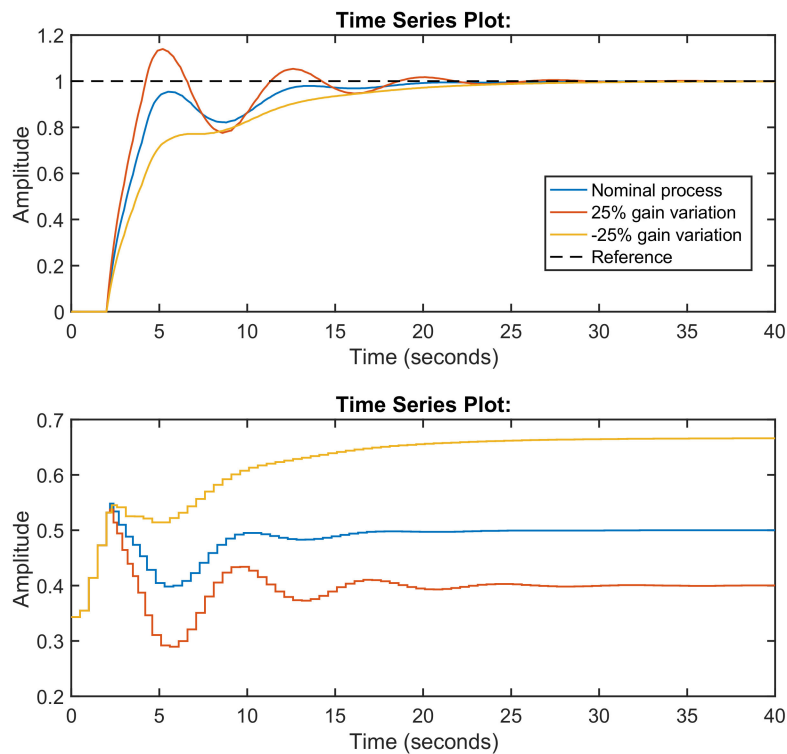


Figure 13. Robustness validation of the event-based PI controller (output signal in the upper plot, input signal in the lower plot).

4. Discussion

A performance comparison regarding the two different event-based controllers is presented in Table 1, for the lag dominant process. Figure 6 depicts a comparison regarding reference tracking with

the event-based FO-IMC controller and the event-based PI controller. As indicated here, the FO-IMC achieves a faster settling time, compared to the PI: 13 s, compared to nearly 25 s. The drawback is that the FO-IMC control signal is twice as large, but solely during one event. Both controllers manage to achieve this without any overshooting. Over the 40 s seconds simulation time, the event-based FO-IMC controller requires 82 control signal computations; similarly for the event-based PI controller. In a classical discrete-time approximation, both type of controllers would have needed 400 computations. This leads to an overall reduction of the resources used of 79.5%. Improved disturbance rejection results are also visible in the case of the FO-IMC controller, compared to the PI controller, as indicated in Figure 7. The settling time for the FO-IMC is approximately 8 s, compared to 16 s for the PI. In terms of control signal computations, there are 61 for the event-based FO-IMC implementation, compared to 60 for the event-based PI implementation and 300 for the standard discrete-time implementation. This leads to 79–80% reduction in the resources used. The robustness validation of the event-based FO-IMC controller is given in Figure 8, whereas the robustness tests for the event-based PI controller are indicated in Figure 9. Although none of the control strategies were designed specifically to ensure the robustness to gain variations, the simulation results in Figures 8 and 9 show that for $\pm 50\%$ gain variations, the event-based controllers manage to maintain 0 overshoot. A comparison between the two shows that the event-based PI controller has better chances of turning the closed loop into an underdamped response for larger positive gain variations.

Table 1. Performance comparison between the event-based FO-IMC and event-based PI controller for the lag dominant FOPDT process.

Test Scenario	FO-IMC Control			PI Control		
	Overshoot (%)	Settling Time (s)	Control Computations	Overshoot (%)	Settling Time (s)	Control Computations
Reference tracking	0	13	82	0	25	82
Disturbance rejection		8	61		16	60
Robustness assessment						
50% gain variation	0	7		0	15	
−50% gain variation	0	23		0	40	

A performance comparison regarding the two different event-based controllers is presented in Table 2, for the delay dominant process. In this case, Figure 10 depicts a comparison regarding reference tracking with the event-based FO-IMC controller and the event-based PI controller. The event-based FO-IMC controller, proposed in this manuscript, achieves zero overshoot and a 7 s settling time. The event-based PI controller achieves no overshoot, but with an oscillatory response, as well as a larger settling time of 18 s. In terms of control effort, this is larger for the FO-IMC, compared to the PI. A comparison regarding resources used over 40 s simulation shows that the event-based FO-IMC requires 103 computations of the control signal, whereas the event-based PI controller needs solely 83. This accounts for 74% reduction in the resources used for the event-based FO-IMC and a slightly better 79% for the event-based PI controller, compared to the standard discrete-time implementation (with 400 computations over the 40 s simulation period). In terms of disturbance rejection, the comparative simulation results given in Figure 11 demonstrate the efficiency of the FO-IMC compared to the PI, with a settling time of 7 s, compared to 12 s. In this case also, the event-based FO-IMC requires 87 computations of the control signal, higher than the 61 for the event-based PI controller. The simulation period considered was 30 s, with a standard discrete-time implementation requiring 300 control signal computations overall. In this case, 71% improvement in the resources used is achieved with the event-based FO-IMC implementation and 80% with the event-based PI controller, compared to the standard discrete-time implementation. The robustness validation of the event-based FO-IMC controller is given in Figure 12, whereas the robustness tests for the event-based PI controller are indicated in Figure 13. Although none of the control strategies were designed specifically to ensure the robustness to gain variations, the simulation results in Figure 12 show that better robustness can

be achieved by using the event-based FO-IMC controller, compared to the event-based PI controller. The latter has oscillatory response and higher overshoot and nearly twice settling time.

The results obtained for the event-based FO-IMC controller show that this can be considered as a viable option for controlling processes where resources, bandwidth allocation, energy usage are limited. Furthermore, as the simulation results show, the event-based FO-IMC control strategy offers better results in terms of reference tracking, disturbance rejection, and robustness for both lag and delay dominant systems. The robustness of the event-based FO-IMC controller is significantly improved compared to the event-based PI controller, for delay dominant systems. This is an aspect intrinsic to the IMC methodology. The sole disadvantage of the event-based FO-IMC controller is that there is an increase in the control effort and in the number of events that require the computation of the control signal, compared to the event-based PI controller.

Table 2. Performance comparison between the event-based FO-IMC and event-based PI controller for the delay dominant FOPDT process.

Test Scenario	FO-IMC Control			PI Control		
	Overshoot (%)	Settling Time (s)	Control Computations	Overshoot (%)	Settling Time (s)	Control Computations
Reference tracking	0	7	103	0	18	83
Disturbance rejection		7	87		12	61
Robustness assessment						
25% gain variation	10	9		15	20	
−25% gain variation	0	12		0	20	

Further research includes the modification of the FO-IMC tuning procedure to an optimization routine where the control effort is also directly tackled, as well as the robustness property. Additionally, an experimental validation is to be considered.

5. Conclusions

Fractional order calculus has been used to provide for a generalization of the IMC. Such an approach is usually considered for the control of dead time processes. In this paper, FOPDT processes are considered, including delay and lag dominant ones. The design is based on a new approximation approach, the NRTF method, for the equivalent controller in an IMC loop and on two performance specifications, the gain crossover frequency and the phase margin. As it has been previously demonstrated by the authors, the NRTF method allows for a more accurate approximation of the process dead-time and ensures an improved closed loop response.

The implementation of the final fractional order controller is usually a challenging task, since higher order transfer functions are used to approximate the dynamics of the original controller. In situations where CPU, bandwidth allocation, and energy usage are limited, resources need to be efficiently managed. In this paper, a solution for this is proposed, in terms of an event-based implementation of the FO-IMC controller. Such an approach, has only been recently proposed for fractional order PID-type controllers, but not for other types/structure of fractional order controllers. The originality of the approach consists in a two-step implementation. The equivalent fractional order controller, as obtained according to the proposed FO-IMC approach, is divided into an integer order PI controller and a fractional order filter. Then, the NRTF approach is used to determine a standard discrete-time approximation for the fractional order filter. The remaining PI controller is implemented in an event-based approach. Numerical results are provided for lag and delay dominant FOPDT processes. For comparison purposes, an integer order PI controller, tuned according to the same performance specifications as the FO-IMC, is also implemented as an event-based control strategy. The numerical results show that the proposed event-based implementation for FO-IMC controller is suitable and provides for better reference tracking, disturbance rejection, and robustness, compared to the integer

order event-based PI controller, as well as a smaller computational effort compared to a standard discrete-time implementation, thus being more suitable in various industrial applications where resources need to be drastically limited.

Author Contributions: Conceptualization, C.I.M. and I.R.B.; methodology, C.I.M.; software, I.R.B.; validation, I.R.B.; formal analysis, E.H.D.; investigation, C.I.M.; resources, I.R.B.; data curation, I.R.B.; writing—original draft preparation, C.I.M.; writing—review and editing, C.I.M. and E.H.D.; visualization, I.R.B.; supervision, C.I.M.; project administration, C.I.M.; funding acquisition, E.H.D. All authors have read and agreed to the published version of the manuscript.

Funding: D.E.H. was funded by Hungarian Academy of Science, Janos Bolyai Grant (BO/00313/17) and the ÚNKP-19-4-OE-64 New National Excellence Program of the Ministry for Innovation and Technology. C.I.M. and I.B. have been supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI-UEFISCDI, project number PN-III-P1-1.1-TE-2016-1396, TE 65/2018.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Ionescu, C.M.; Machado, J.A.; De Keyser, R. Modeling of the lung impedance using a fractional-order ladder network with constant phase elements. *IEEE Trans. Biomed. Circuits Syst.* **2011**, *5*, 83–89. [[CrossRef](#)] [[PubMed](#)]
2. Muresan, C.I.; Folea, S.C.; Birs, I.R.; Ionescu, C.M. A Novel Fractional Order Model and Controller for Vibration Suppression in Flexible Smart Beam. *Nonlinear Dyn.* **2018**, *93*, 525–541. [[CrossRef](#)]
3. Muresan, C.I.; Dulf, E.H.; Prodan, O. A Fractional Order Controller for Seismic Mitigation of Structures Equipped with Viscoelastic Mass Dampers. *J. Vib. Control* **2016**, *22*, 1980–1992. [[CrossRef](#)]
4. Ionescu, C.M.; Birs, I.; Copot, D.; Muresan, C.I.; Caponetto, R. Mathematical modeling with experimental validation of viscoelastic properties in non-Newtonian fluids. *Philos. Trans. R. Soc. A* **2020**, *378*, 20190284. [[CrossRef](#)]
5. Birs, I.; Copot, D.; Muresan, C.I.; Nascu, I.; Ionescu, C. Identification for Control of Suspended Objects in Non-Newtonian Fluids. *Fract. Calc. Appl. Anal.* **2019**, *22*, 1378–1394. [[CrossRef](#)]
6. Carvalho, A.R.M.; Pinto, C.M.A.; de Carvalho, J.M. Fractional Model for Type 1 Diabetes. In *Mathematical Modelling and Optimization of Engineering Problems. Nonlinear Systems and Complexity*; Machado, J., Özdemir, N., Baleanu, D., Eds.; Springer: Cham, Switzerland, 2020; Volume 30.
7. Magin, R.L.; Karani, H.; Wang, S.; Liang, Y. Fractional Order Complexity Model of the Diffusion Signal Decay in MRI. *Mathematics* **2019**, *7*, 348. [[CrossRef](#)]
8. Pinto, C.M.A.; Carvalho, A.R.M.; Baleanu, D.; Srivastava, H.M. Efficacy of the Post-Exposure Prophylaxis and of the HIV Latent Reservoir in HIV Infection. *Mathematics* **2019**, *7*, 515. [[CrossRef](#)]
9. Kumar, S.; Ahmadian, A.; Kumar, R.; Kumar, D.; Singh, J.; Baleanu, D.; Salimi, M. An Efficient Numerical Method for Fractional SIR Epidemic Model of Infectious Disease by Using Bernstein Wavelets. *Mathematics* **2020**, *8*, 558. [[CrossRef](#)]
10. Dulf, E.-H.; Vodnar, D.C.; Danku, A.; Muresan, C.-I.; Crisan, O. Fractional-Order Models for Biochemical Processes. *Fractal Fract.* **2020**, *4*, 12. [[CrossRef](#)]
11. Podlubny, I. Fractional-order systems and $PI^\lambda D^\mu$ -controllers. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214. [[CrossRef](#)]
12. Monje, C.A.; Chen, Y.; Vinagre, B.M.; Xue, D.; Feliu-Batlle, V. *Fractional-Order Systems and Controls: Fundamentals and Applications*; Springer: Cham, Switzerland, 2010.
13. Folea, S.; Muresan, C.I.; De Keyser, R.; Ionescu, C. Theoretical Analysis and Experimental Validation of a Simplified Fractional Order Controller for a Magnetic Levitation System. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 756–763. [[CrossRef](#)]
14. Li, X.; Wang, Y.; Li, N.; Han, M.; Tang, Y.; Liu, F. Optimal fractional order PID controller design for automatic voltage regulator system based on reference model using particle swarm optimization. *Int. J. Mach. Learn. Cyber.* **2017**, *8*, 1595–1605. [[CrossRef](#)]
15. El-Khazali, R. Fractional-order $PI^\lambda D^\mu$ controller design. *Comput. Math. Appl.* **2013**, *66*, 639–646.

16. Monje, C.A.; Vinagre, B.M.; Feliu, V.; Chen, Y. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng. Pract.* **2008**, *16*, 798–812. [CrossRef]
17. Muresan, C.I.; Folea, S.; Mois, G.; Dulf, E.H. Development and implementation of an FPGA based fractional order controller for a DC motor. *Mechatronics* **2013**, *23*, 798–804. [CrossRef]
18. Ionescu, C.M.; Dulf, E.H.; Ghita, M.; Muresan, C.I. Robust controller design: Recent emerging concepts for control of mechatronic systems. *J. Frankl. Inst.* **2020**, *357*, 7818–7844. [CrossRef]
19. Das, S.; Pan, I.; Das, S.; Gupta, A. A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices. *Eng. Appl. Artif. Intell.* **2012**, *25*, 430–442. [CrossRef]
20. Castillo-Garcia, F.J.; Feliu-Batlle, V.; Rivas-Perez, R.; Sanchez, L. Time Domain Tuning of a Fractional Order PI α Controller Combined with a Smith Predictor for Automation of Water Distribution in Irrigation Main Channel Pools. *IFAC Proc. Vol.* **2011**, *44*, 15049–15054. [CrossRef]
21. Maamir, F.; Guiatni, M.; El Hachemi, H.M.S.M.; Ali, D. Auto-tuning of fractional-order PI controller using particle swarm optimization for thermal device. In Proceedings of the 4th International Conference on Electrical Engineering, Boumerdes, Algeria, 13–15 December 2015; pp. 1–6. [CrossRef]
22. De Keyser, R.; Muresan, C.I.; Ionescu, C.M. A novel auto-tuning method for fractional order PI/PD controllers. *ISA Trans.* **2016**, *62*, 268–275. [CrossRef]
23. Juchem, J.; Muresan, C.I.; De Keyser, R.; Ionescu, C.M. Robust fractional-order auto-tuning for highly-coupled MIMO systems. *Heliyon* **2019**, *5*, e02154. [CrossRef]
24. Soukkou, A.; Belhour, M.C.; Leulmi, S. Review, Design, Optimization and Stability Analysis of Fractional-Order PID Controller. *Int. J. Intell. Syst. Appl.* **2016**, *8*, 73. [CrossRef]
25. Leng, B.Y.; Qi, Z.D.; Shan, L.; Bian, H.J. Review of Fractional Order Control. 2014. Available online: <https://www.scientific.net/AMR.1049-1050.983> (accessed on 14 August 2020).
26. Shah, P.; Agashe, S. Review of fractional PID controller. *Mechatronics* **2016**, *38*, 29–41. [CrossRef]
27. Dastjerdi, A.A.; Vinagre, B.M.; Chen, Y.Q.; HosseinNia, S.H. Linear fractional order controllers; A survey in the frequency domain. *Annu. Rev. Control* **2019**, *47*, 51–70. [CrossRef]
28. Birs, I.; Muresan, C.I.; Nascu, I.; Ionescu, C. A Survey of Recent Advances in Fractional Order Control for Time Delay Systems. *IEEE Access* **2019**, *7*, 30951–30965. [CrossRef]
29. Copot, D.; Ghita, M.; Ionescu, C.M. Simple Alternatives to PID-Type Control for Processes with Variable Time-Delay. *Processes* **2019**, *7*, 146. [CrossRef]
30. Bettayeb, M.; Mansouri, R.; Al-Saggaf, U.; Mehedi, I.M. Smith Predictor Based Fractional-Order-Filter PID Controllers Design for Long Time Delay Systems. *Asian J. Control* **2016**, *19*, 587–598. [CrossRef]
31. Vu, T.N.L.; Lee, M. Smith predictor based fractional-order PI control for time-delay processes. *Korean J. Chem. Eng.* **2014**, *31*, 1321–1329. [CrossRef]
32. Jesus, I.S.; Machado, J.T. Fractional Control with a Smith Predictor. *J. Comput. Nonlinear Dyn.* **2011**, *6*, 31010–31014. [CrossRef]
33. Safaei, M.; Tavakoli, S. Smith predictor based fractional-order control design for time-delay integer-order systems. *Int. J. Dyn. Control* **2018**, *6*, 179–187. [CrossRef]
34. Maamar, B.; Rachid, M. IMC-PID-fractional-order-filter controllers design for integer order systems. *ISA Trans.* **2014**, *53*, 1620–1628. [CrossRef]
35. Valerio, D.; Sa da Costa, J. Tuning of fractional PID controllers with Ziegler–Nichols-type rules. *Signal Process.* **2006**, *86*, 2771–2784. [CrossRef]
36. Abadi, M.R.R.M.; Jalali, A.A. Fractional order PID controller tuning based on IMC. *Int. J. Inf. Technol. Control Autom. IJITCA* **2012**, *2*, 21–35.
37. Lei, S.; Zhao, Z.; Zhang, J. Design of fractional order smith predictor controller for non-square system. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016. [CrossRef]
38. Muresan, C.I.; Birs, I.; De Keyser, R. An Alternative Design Approach for Fractional Order Internal Model Controllers for Time Delay Systems. *Int. J. Control* **2020**. under review.
39. Muresan, C.I.; Dutta, A.; Dulf, E.H.; Pinar, Z.; Maxim, A.; Ionescu, C.M. Tuning algorithms for fractional order internal model controllers for time delay processes. *Int. J. Control* **2016**, *89*, 579–593. [CrossRef]
40. Arya, P.P.; Chakrabarty, S. IMC based Fractional Order Controller Design for Specific Non-Minimum Phase Systems. *IFAC-PapersOnLine* **2018**, *51*, 847–852. [CrossRef]

41. Jain, S.; Hote, Y.V. Fractional order IMC controller via order reduction and CRONE principle for Load frequency control. In Proceedings of the 2018 8th International Conference on Power and Energy Systems (ICPES), Colombo, Sri Lanka, 21–22 December 2018. [\[CrossRef\]](#)
42. Dulf, E.H. Simplified Fractional Order Controller Design Algorithm. *Mathematics* **2019**, *7*, 1166. [\[CrossRef\]](#)
43. De Keyser, R.; Muresan, C.I.; Ionescu, C.M. An efficient algorithm for low-order discrete-time implementation of fractional order transfer functions. *ISA Trans.* **2018**, *74*, 229–238. [\[CrossRef\]](#)
44. Muresan, C.I.; Ionescu, C.M. Generalization of the FOPDT Model for Identification and Control Purposes. *Processes* **2020**, *8*, 682. [\[CrossRef\]](#)
45. Volanova, R.; Visioli, A. *PID Control in the Third Millennium*; Springer: Cham, Switzerland, 2012.
46. Birs, I.; Nascu, I.; Ionescu, C.M.; Muresan, C.I. Event-based fractional order control. *J. Adv. Res.* **2020**. [\[CrossRef\]](#)
47. Birs, I.; Muresan, C.I.; Ionescu, C.M. An event based implementation of a fractional order controller on a non-Newtonian transiting robot. In Proceedings of the European Control Conference (ECC), Saint Petersburg, Russia, 12–15 May 2020; pp. 1436–1441.
48. Heemels, W.P.; Johansson, K.H.; Tabuada, P. An introduction to event-triggered and self-triggered control. In Proceedings of the IEEE Conference on Decision and Control, Maui, HI, USA, 10–13 December 2012; pp. 3270–3285. [\[CrossRef\]](#)
49. Castilla, M.; Bordons, C.; Visioli, A. Event-based state-space model predictive control of a renewable hydrogen-based microgrid for office power demand profiles. *J. Power Sources* **2020**, *450*. [\[CrossRef\]](#)
50. Liu, Q.; Wang, Z.; He, X.; Zhou, D.H. A survey of event-based strategies on control and estimation. *Syst. Sci. Control Eng.* **2014**, *2*, 90–97. [\[CrossRef\]](#)
51. Petras, I. Tuning and implementation methods for fractional-order controllers. *Fract. Calc. Appl. Anal.* **2012**, *15*, 282–303. [\[CrossRef\]](#)
52. Li, Y.; Sheng, H.; Chen, Y.Q. Analytical impulse response of a fractional second order filter and its impulse response invariant discretization. *Signal Process* **2011**, *91*, 498–507. [\[CrossRef\]](#)
53. Steiglitz, K.; McBride, L.E. A Technique for the Identification of Linear Systems. *IEEE Trans. Autom. Control* **1965**, *10*, 461–464. [\[CrossRef\]](#)



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).